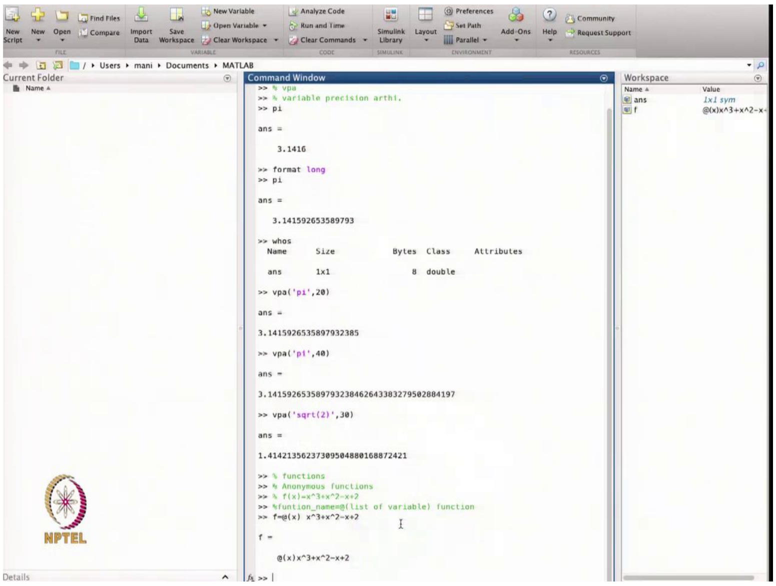
Scientific Computing Using Matlab

Professor. Vivek Aggarwal & Professor. Mani Mehra
Department of Mathematics
Indian Institute of Technology, Delhi

Lecture No. 04
Functions Definition in Matlab

Hello viewers. Welcome back to this course. So, today we are going to do lecture number 4 and in the previous 3 lectures we have discussed some basics about the Matlab and in this lecture also we are going to do some basics. So, in this I will explain how we can define any function in a Matlab and then we can use this function later on in your script files to do other codes. So, let us go to the Matlab windows.
(Refer Slide Time: 0:53)



```
>> % vpa
>> % variable precision arthi.
>> pi

ans =

    3.1416

>> format long
>> pi

ans =

   3.141592653589793

>> whos
  Name      Size            Bytes  Class     Attributes

  ans       1x1                 8  double

>> vpa('pi',20)

ans =

3.1415926535897932385

>> vpa('pi',40)

ans =

3.141592653589793238462643383279502884197

>> vpa('sqrt(2)',30)

ans =

1.41421356237309504880168872421

>> % functions
>> % Anonymous functions
>> % f(x)=x^3+x^2-x+2
>> %funtion_name=@(list of variable) function
>> f=@(x) x^3+x^2-x+2

f =

    @(x)x^3+x^2-x+2

>> 
```

So, this is my Matlab window and so, there is no variable there in the workspace. So, today we will discuss that, so today I will define you the command that is vpa. So, this is another command that is the, so this command gives you that variable precision arithmetic. So, what it saves, because like suppose I want to define what is pi.

So, pi gives you the value 3.161416 or if I write format long and then if I define the pi, so it gives you the 16 digit value and if you see this one whos so it shows you that it is a 8 byte and double. So, it means it is giving you everything. Whatever we are doing in the Matlab it is giving you double precision. The number is given to you in double precision means correct up to 16 digits. But suppose I want to increase the precision, so I will write down the command that vpa.

So, vpa command gives you that I will define then that π and suppose I want to define this with the 20 digits accuracy, so I will write this one. So, this is the value of the pi and it is coming with the 20 digits or maybe I want to. Suppose I want to increase the accuracy digits, so I want 40 digits so that gives you the value of the pi up to 40 digits. Similarly, I can define the value of maybe I want to define square root square root of 2 with 30 digit accuracy.

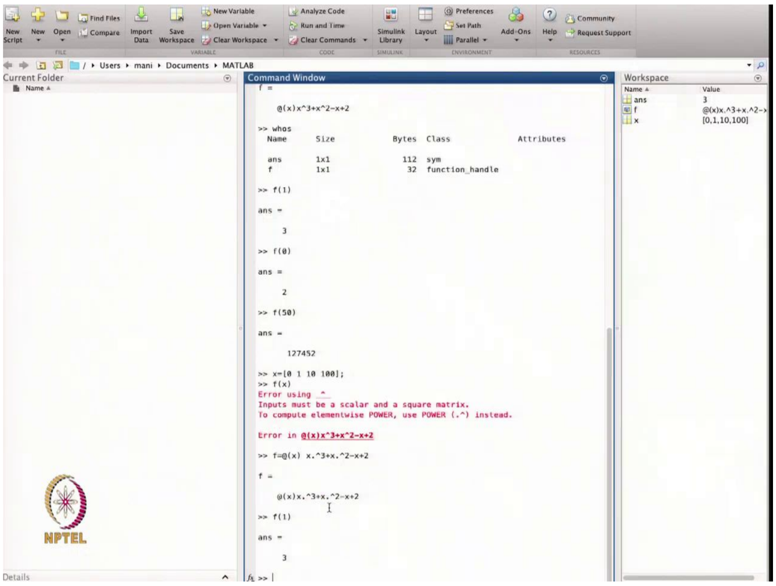
So, that is the value of the $\sqrt{1.41}$ and then this is the accurate up to the 30 digits accuracy. So, this type of command vpa is used when we want to define or we want to use the values of any fractions or a function like π or $\sqrt{\pi}$ and for a large number of accuracy or maybe an accuracy up to 40 digit or 30 digits. So, we use this vpa command. Now we start with the functions. So, in this case we will define the anonymous function. So, let us do the functions. How to define the functions?

So, in anonymous functions, so anonymous functions means suppose I have a function like this one. My function I define a function f(x) is equal to a small I want to calculate $x^3 + x^2 - x + 2$. So, this function I want to use in my Matlab program. So, how to define these functions in a Matlab command window or in a later on in the script file.

So, let us define this one. So, this is what will define. So, that is that definition this function is defined, so it gives you the definition. So, first is the function name. So, this is the name. So, that is equal to and I will use the word at the rate and then I will find out the list of the variables and then the function. So, I will write the function name and then I will put the sign at the rate and then the list of variables and then I will define the function.

So for example I define the function whatever I just defined. I will write f is equal to at the rate and now suppose this function is a one dimensional just function in a 1D so I will define the function of only one variable and then I will define $x^3 + x^2 - x + 2$. So that is my function.

(Refer Slide Time: 5:39)



```
f =

    @(x)x^3+x^2-x+2

>> whos
  Name      Size            Bytes  Class              Attributes

  ans       1x1               112  sym
  f         1x1                32  function_handle

>> f(1)

ans =

     3

>> f(0)

ans =

     2

>> f(50)

ans =

      127452

>> x=[0 1 10 100];
>> f(x)
Error using  ^
Inputs must be a scalar and a square matrix.
To compute elementwise POWER, use POWER (.^) instead.

Error in @(x)x^3+x^2-x+2

>> f=@(x) x.^3+x.^2-x+2

f =

    @(x)x.^3+x.^2-x+2

>> f(1)

ans =

     3

fx >> |
```

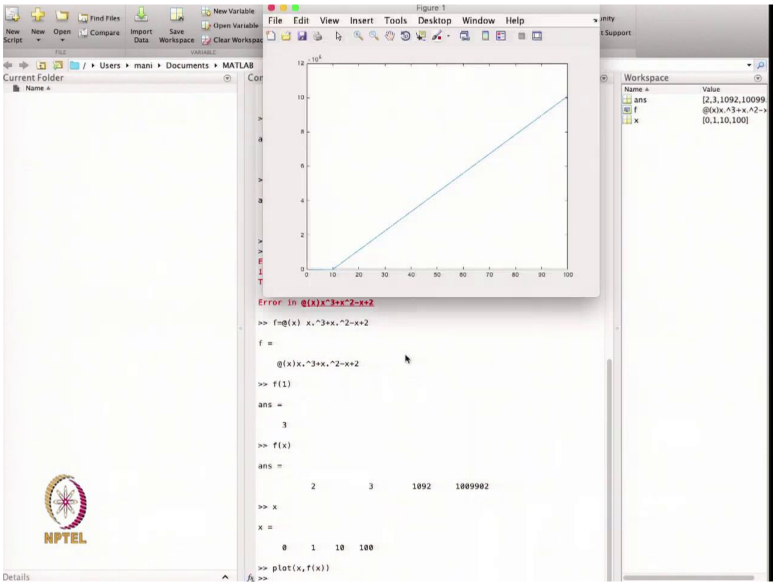
Name ▲	Value
ans	3
f	@(x)x.^3+x.^2-x
x	[0,1,10,100]

Now you can see that it is coming to the workspace and its value is f is a function with a function handle. So, function handle means it gives you the name of the function. Now, suppose I want to find the value of the function f1. So f1 if I put there it gives you a value of f1(3) or maybe f1(0), This gives the value of 0.

Maybe I can define what is the value of the function f at maybe a large value 50? So, that gives you the value of f(50). Suppose I want to calculate the value of the function for not at one value but suppose I take a vector of value. So, I define [0 1 10 100]. So, this is the one vector I define.

So, that is my vector and now I want to find the value of f(x). So, now it gives you the error and error using the sign this. It means that in this case we have defined the function. That function is a scalar but the value I am giving in the vector form because it is a 1 dimensional vector. So, in this case what I will do, I will redefine the function so that this function can be compatible with the vector values also.

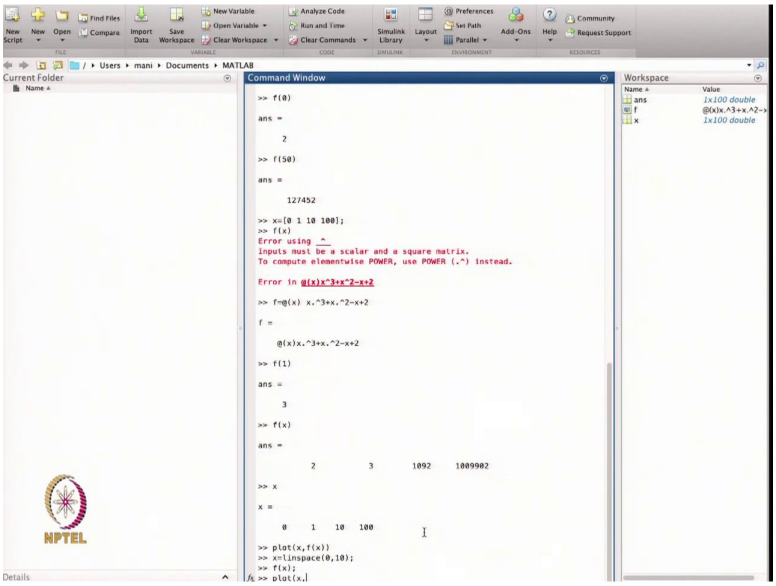
So, in this case I will define this one f in the same way but only I will put the dot sign as we have done in the case of defining the vectors. So, now this is my function and now everything will be the same, the value, the function will be the same.
(Refer Slide Time: 7:22)



```
Error in @(x)x^3+x^2-x+2

>> f=@(x) x.^3+x.^2-x+2

f =

    @(x)x.^3+x.^2-x+2

>> f(1)

ans =

     3

>> f(x)

ans =

           2           3        1092     1009902

>> x

x =

     0     1    10   100

>> plot(x,f(x))
>>
```

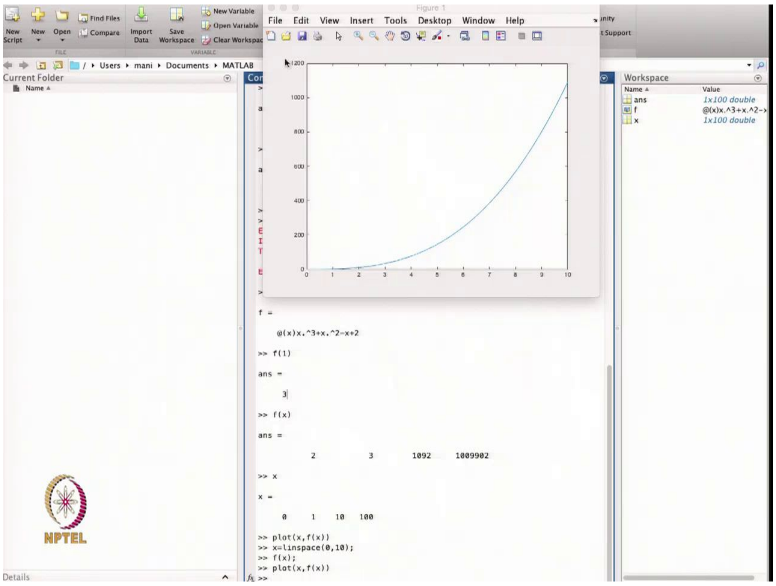
But now I can pass the value of x here. So, now I will define the value and this is the value of the function. So, my value of x was 0, 1, 10 and 100 and that is the value of function at 2, 3, 1092 and this is. So, this is the value of the function given at various values of this one.

Then I can plot the value of the function x and then f(x). So, I can plot the function that this is the value of the function. So, in this case I can define the value like I define a function x = linspace(0,10). So, this is the function I am defining.
(Refer Slide Time: 8:23)



```
>> f(0)

ans =

     2

>> f(50)

ans =

      127452

>> x=[0 1 10 100];
>> f(x)
Error using  ^
Inputs must be a scalar and a square matrix.
To compute elementwise POWER, use POWER (.^) instead.

Error in @(x)x^3+x^2-x+2

>> f=@(x) x.^3+x.^2-x+2

f =

    @(x)x.^3+x.^2-x+2

>> f(1)

ans =

     3

>> f(x)

ans =

           2           3        1092     1009902

>> x

x =

     0     1    10   100

>> plot(x,f(x))
>> x=linspace(0,10);
>> f(x);
>> plot(x,
```

Name	Value
ans	1x100 double
f	@(x)x.^3+x.^2-x
x	1x100 double



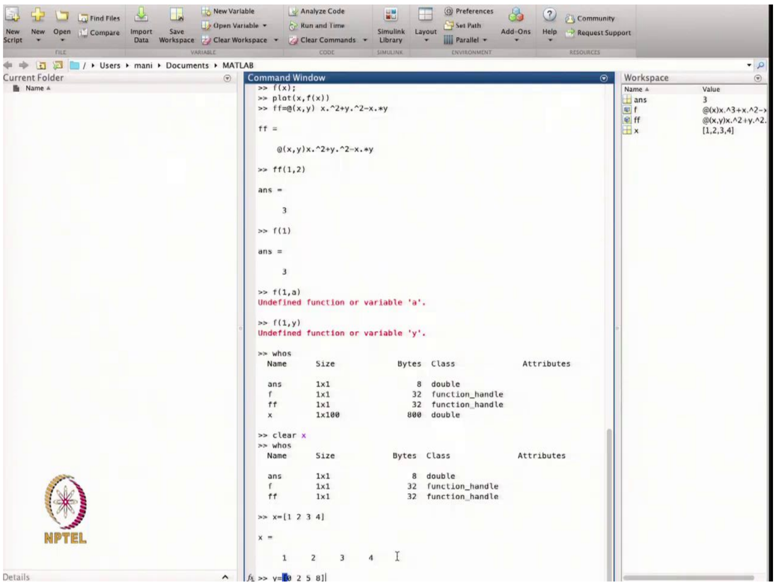
```
f =

    @(x)x.^3+x.^2-x+2

>> f(1)

ans =

     3

>> f(x)

ans =

           2           3        1092     1009902

>> x

x =

     0     1    10   100

>> plot(x,f(x))
>> x=linspace(0,10);
>> f(x);
>> plot(x,f(x))
>>
```

Name	Value
ans	1x100 double
f	@(x)x.^3+x.^2-x
x	1x100 double

Then I define f(x) and then I define the plot. So, plot x f(x). So, that is the value of the. So, in this case what it is giving you? That this is the x I am taking from 0 to 10 and this is linspace I am using.

So, it is giving you the default 100 values of x from 0 to 10 and based on this one I will find the value of f(x) and this is the plot I am doing. So, that is the plot of the function this one for the x range from 0 to 10. So, this is the way we can define the function. So, this function is a we have defined in 1 dimensional. The same way I can define the function. So, let us find out another function.

(Refer Slide Time: 9:12)



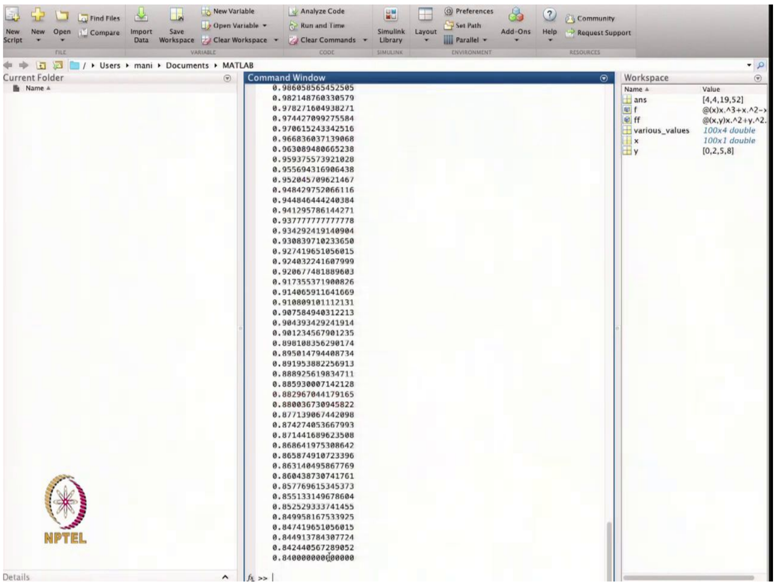
```
>> f(x);
>> plot(x,f(x))
>> ff=@(x,y) x.^2+y.^2-x.*y

ff =

    @(x,y)x.^2+y.^2-x.*y

>> ff(1,2)

ans =

     3

>> f(1)

ans =

     3

>> f(1,a)
Undefined function or variable 'a'.

>> f(1,y)
Undefined function or variable 'y'.

>> whos
  Name      Size            Bytes  Class              Attributes

  ans       1x1                 8  double
  f         1x1                32  function_handle
  ff        1x1                32  function_handle
  x         1x100             800  double

>> clear x
>> whos
  Name      Size            Bytes  Class              Attributes

  ans       1x1                 8  double
  f         1x1                32  function_handle
  ff        1x1                32  function_handle

>> x=[1 2 3 4]

x =

     1     2     3     4

>> y=[0 2 5 8]
```

Name ▲	Value
ans	3
f	@(x)x.^3+x.^2-x
ff	@(x,y)x.^2+y.^2.
x	[1,2,3,4]

So, let say I take ff, the other function I define. So, this function I define at the rate maybe I can define for 2 variables now, x and y and this is I defining $x^{2}.+y^{2}-x.\cdot y$. So, that is another function I am defining. Now I can define the function that is compatible for both scalar and vector. So, that is why I put the dot here. So, this is a 2 dimensional function I have. Now I can calculate the value of 1 and 2 for this function. Putting the value of the function here.

So, that gives you the value 1 or maybe I can define the function, this x I can put 1 and y I can put. So, just it gives the value 1. So, y I will put as a. So, that gives you the value that undefined function on a variable a. So, in this case what I can do, I can take the value of x and here I can put maybe y only. So, this is f(x).

So, I have not defined the value of the function y. So, y maybe I can define, y is equal to, I can define linspace or maybe I can define clear clear x. So, we have cleared the x. Now I can define that x=[1, 2, 3, 4], so this is the vector I am defining and similarly I define the values maybe x=[0, 2, 5, 8]. So, that is the other value I have defined y.
(Refer Slide Time: 11:23)



```
0.986058565452505
0.982148760330579
0.978271604938271
0.974427099275584
0.970615243342516
0.966836037139068
0.963089480665238
0.959375573921028
0.955694316906438
0.952045709621467
0.948429752066116
0.944846444240384
0.941295786144271
0.937777777777778
0.934292419140904
0.930839710233650
0.927419651056015
0.924032241607999
0.920677481889603
0.917355371900826
0.914065911641669
0.910809101112131
0.907584940312213
0.904393429241914
0.901234567901235
0.898108356290174
0.895014794408734
0.891953882256913
0.888925619834711
0.885930007142128
0.882967044179165
0.880036730945822
0.877139067442098
0.874274053667993
0.871441689623508
0.868641975308642
0.865874910723396
0.863140495867769
0.860438730741761
0.857769615345373
0.855133149678604
0.852529333741455
0.849958167533925
0.847419651056015
0.844913784307724
0.842440567289052
0.840000000000000
>>
```

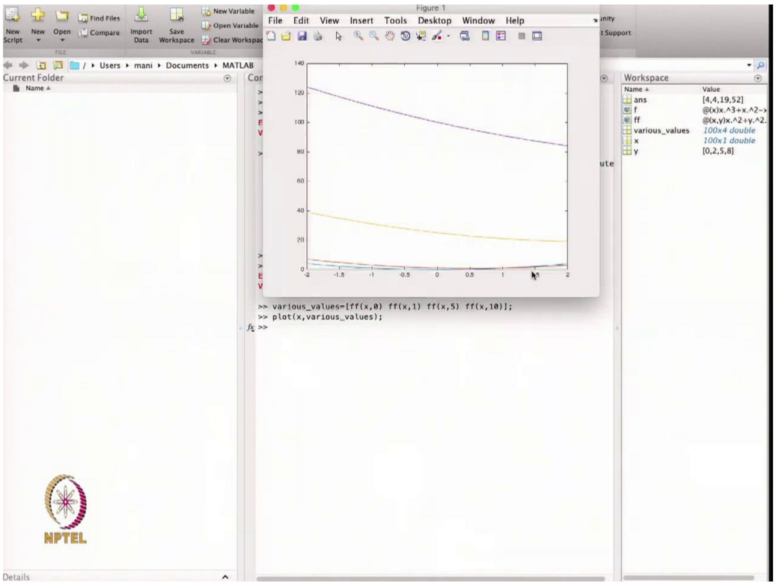
Name	Value
ans	[4,4,19,52]
f	@(x)x.^3+x.^2-x
ff	@(x,y)x.^2+y.^2.
various_values	100x4 double
x	100x1 double
y	[0,2,5,8]

Now I can pass this one x and y. So, that is the value. So, this value gives you for this value of x and this value of y. Similarly I can define that suppose I define it for 2, so that gives the value.

In this case my x=2 fixed and this y I can find. So, in this way I can define the function value of the function for various this one. Now I define x =linspace(-2,2) suppose I define this one and then my x is equal to x. So, I just make the row vector the column vector. Then I define a function with various values.

So, various values I can define as, so I define the value of the function for x y 0 then maybe x at 1, x at 5 not 5, y is equal to 5 maybe and then ff x and 10. So, let us take only this value. So, this is the value of the function. So, I have taken it now. I have calculated the value of the function for x, whatever the x I have just defined the linspace.

(Refer Slide Time: 13:03)

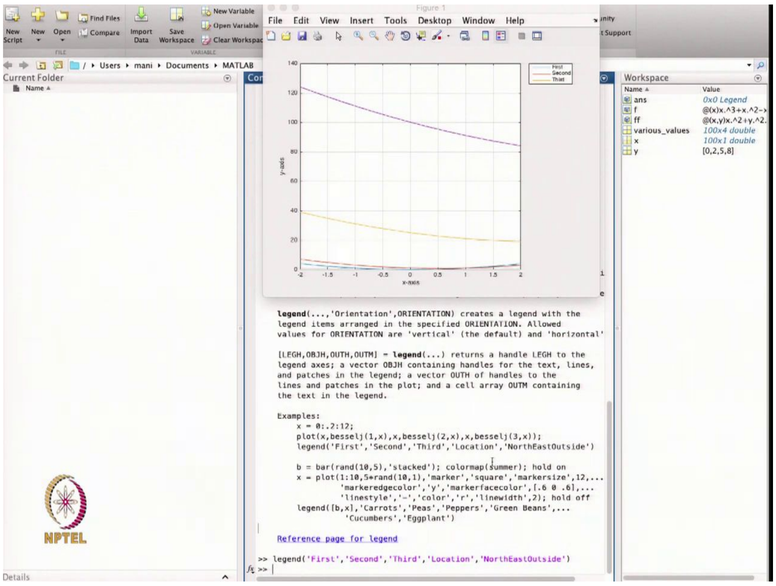


```
>> various_values=[ff(x,0) ff(x,1) ff(x,5) ff(x,10)];
>> plot(x,various_values);
>>
```

Name	Value
ans	[4,4,19,52]
f	@(x)x.^3+x.^2-x
ff	@(x,y)x.^2+y.^2.
various_values	100x4 double
x	100x1 double
y	[0,2,5,8]

Clc I can do it. So, this is the value of the function I have defined and this is the x I have defined. This is my x and then I define the value of the function. Now, I can plot all values together. So, in this case I will do x and then various values. So, this one various values I can define. So, using plot because my x is, so I just want to check the size of this one. So, this is the size. Now various values, this is one row and the columns and this is one row, 100 columns. Now, what I do is x=x'.

Now, I can define various values also. And now maybe, this one is valid because earlier actually it was showing you that only these are the column vectors and that was the row vector. But now x is the column vector and with respect to x I am finding plotting all the functions which the first column is the first function, the second column is the second function. So, all the 4 values functions I can plot in just a single way. So, that is the plot, x various values of x.
(Refer Slide Time: 14:48)



```
legend(...,'Orientation',ORIENTATION) creates a legend with the
legend items arranged in the specified ORIENTATION. Allowed
values for ORIENTATION are 'vertical' (the default) and 'horizontal'

[LEGH,OBJH,OUTH,OUTM] = legend(...) returns a handle LEGH to the
legend axes; a vector OBJH containing handles for the text, lines,
and patches in the legend; a vector OUTH of handles to the
lines and patches in the plot; and a cell array OUTM containing
the text in the legend.

Examples:
    x = 0:.2:12;
    plot(x,besselj(1,x),x,besselj(2,x),x,besselj(3,x));
    legend('First','Second','Third','Location','NorthEastOutside')

    b = bar(rand(10,5),'stacked'); colormap(summer); hold on
    x = plot(1:10,5*rand(10,1),'marker','square','markersize',12,...
             'markeredgecolor','y','markerfacecolor',[.6 0 .6],...
             'linestyle','-','color','r','linewidth',2); hold off
    legend([b,x],'Carrots','Peas','Peppers','Green Beans',...
              'Cucumbers','Eggplant')

Reference page for legend

>> legend('First','Second','Third','Location','NorthEastOutside')
>> 
```

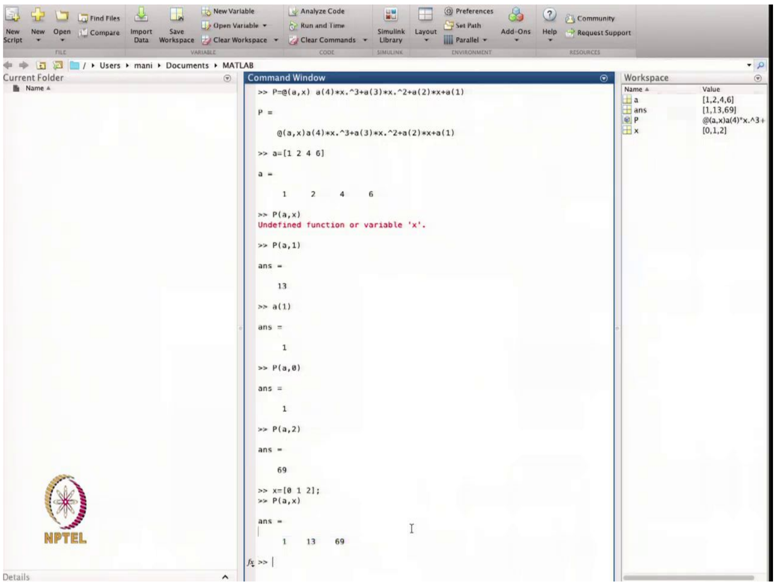
Workspace

Name ▲	Value
ans	0x0 Legend
f	@(x)x.^3+x.^2-x
ff	@(x,y)x.^2+y.^2.
various_values	100x4 double
x	100x1 double
y	[0,2,5,8]

Maybe I can find the put grid there. So, this will give you the grid and then in the I can find the x labels. So, x label is that, x is and similarly I can define the y axis. This gives you the y axis and that is the plot or maybe I can put the legends. So, in the legend, that is the way we can define the legend.

So, legend gives you the value of the function. So, legend is string 1, string 2, string 3 and this gives you that which function is represented by what. So, this is the command I can and then I can define the legend first, second, third. So that gives you the legend. So, this is the first.

This is the second. So, this is the first function, second function, third function and I give you that the location, it is a north east outside. So, that is the north east outside. So this is the north east and we put this one and the outside. So, similarly this one I can define and anywhere wherever you want to like. So, this way we can define the function. Now maybe I can define another function.
(Refer Slide Time: 16:42)



```
>> P=@(a,x) a(4)*x.^3+a(3)*x.^2+a(2)*x+a(1)

P =

    @(a,x)a(4)*x.^3+a(3)*x.^2+a(2)*x+a(1)

>> a=[1 2 4 6]

a =

     1     2     4     6

>> P(a,x)
Undefined function or variable 'x'.

>> P(a,1)

ans =

    13

>> a(1)

ans =

     1

>> P(a,0)

ans =

     1

>> P(a,2)

ans =

    69

>> x=[0 1 2];
>> P(a,x)

ans =

     1    13    69

fx >> |
```

Name ▲	Value
a	[1,2,4,6]
ans	[1,13,69]
P	@(a,x)a(4)*x.^3+
x	[0,1,2]

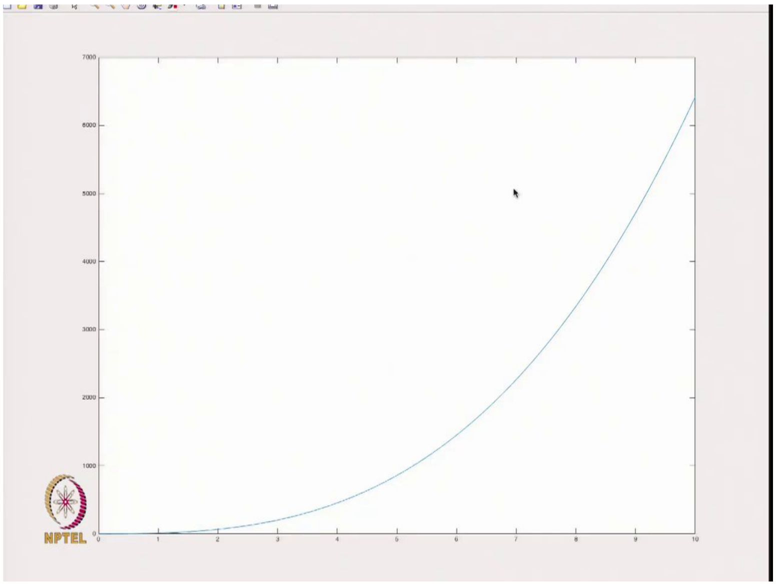
So, I just close this one and I do the clc and I will clear all the variables. Now maybe I can define another function that I just, z is equal to at the rate or I can define the polynomials. So, capital P is equal to at the rate some a and x where a is the some parameter I am defining and then I define a polynomial, a $4x^3$+ $3x^{.2}$ + 2.x.

So, this is just the x, I am defining and then plus a_1. So, this is the polynomial I have defined where I have defined that a is the parameter and this is the coefficient I am defining a_4, a_3 , a_2. So, this is a general cubic polynomial. Now I define what is my a. So, I just defined here that a is, suppose I give a = 1 then 2 then 4 then 6 and I want to this one.

So, this is my a. Now I want to find what is my P. So, that is the value. So, I will just define what gives the value of this x here. So, suppose I give the value maybe 1 here, so that gives the value 13. So, in this case my a will be, so this a 1 will be a 1. So, this is the vector I have given because here I have to give the value of the vector a which has the 4 components a4, a3, a2 and a1.

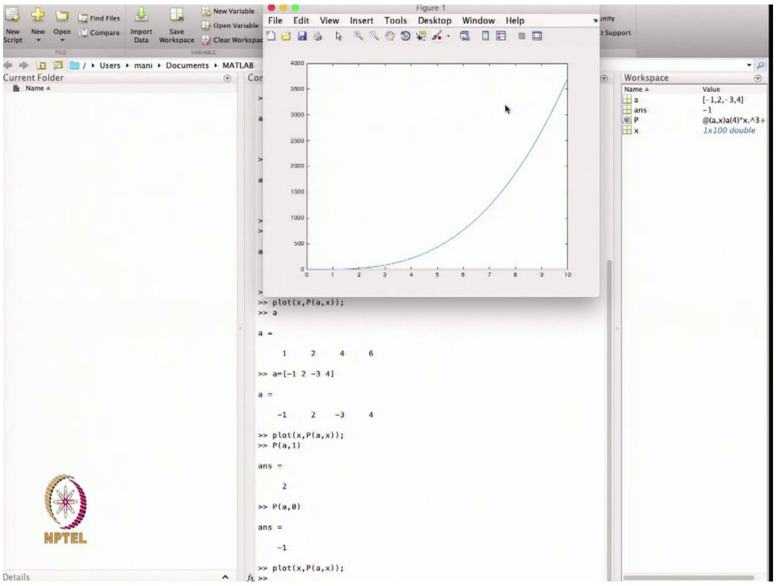
So, this is the a 1, a 2, a 3 and a 4. So, that gives you the value of the function here or maybe I can define what is my P for a 0. So, you put the value of 0, it gives the a 1 only. So, that is my a 1. So, in this case we can define this function polynomial in this way or maybe I can define some other values or I can define if x is equal to 0, 1, 2.

So, this is a vector I am defining and now I can pass the value of to this capital P, a and x. So, that gives you the value of the polynomial for value of a and then this x is the vector. So, it gives you the value of the polynomial at 0, 1 and 2. So, the 3 values together we can find out. Then I can maybe plot this function.
(Refer Slide Time: 20:12)



So, for plotting maybe I need more value of x. So, I will define the x is equal to linspace and I will put the value from 0 to maybe 10. So, this is the value I am defining linspace and then I will define the plot.

So, in this case I will define x and then P for a_x. So, that is the plot. So, this is the polynomial we just defined. So, it gives you the value of the polynomial from 0 to 10 for those values of coefficient. So, my coefficients are the same or maybe I can change the coefficient now and I just put the coefficient -1, then 2, then -3, then 4.
(Refer Slide Time: 21:13)



```
>> plot(x,P(a,x));
>> a

a =

     1     2     4     6

>> a=[-1 2 -3 4]

a =

    -1     2    -3     4

>> plot(x,P(a,x));
>> P(a,1)

ans =

     2

>> P(a,0)

ans =

    -1

>> plot(x,P(a,x));
fx >>
```

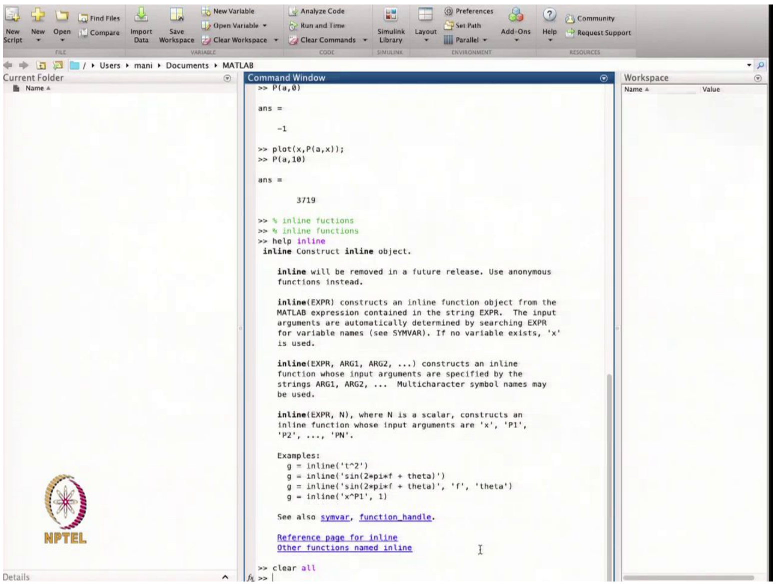
Workspace

Name ▲	Value
a	[-1,2,-3,4]
ans	-1
P	@(a,x)a(4)*x.^3+
x	1x100 double

So, let us take this point. So, in this case my polynomial will change and this is the polynomial now we are getting. So, this is the value of a and so let us see what is the value of capital P at a 1 2 or maybe I will put 0 and the minus 1. So, this is the plot we are getting from here. That is the polynomial we have.

So, this polynomial in this case is starting from 0 and going up to this the value 3500 may be more than that. That I can define from, so P at a 10 is, so that is the value. It is going up to 3719. So this is the way we can define the plot and now the same way the function we have defined that the anonymous function we have defined, the same way I can define the function that is called inline functions. So, what is the inline function?

(Refer Slide Time: 22:42)

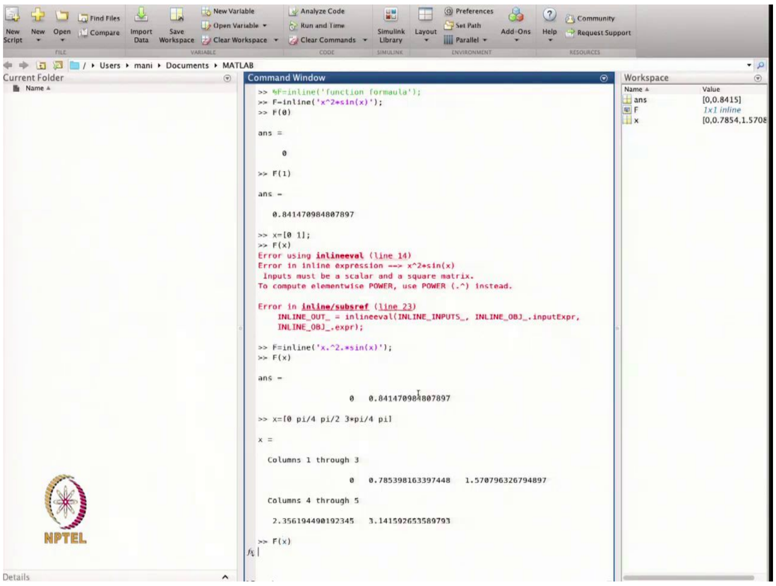


```
>> P(a,0)

ans =

    -1

>> plot(x,P(a,x));
>> P(a,10)

ans =

        3719

>> % inline fuctions
>> % inline functions
>> help inline
 inline Construct inline object.

    inline will be removed in a future release. Use anonymous
    functions instead.

    inline(EXPR) constructs an inline function object from the
    MATLAB expression contained in the string EXPR.  The input
    arguments are automatically determined by searching EXPR
    for variable names (see SYMVAR). If no variable exists, 'x'
    is used.

    inline(EXPR, ARG1, ARG2, ...) constructs an inline
    function whose input arguments are specified by the
    strings ARG1, ARG2, ...  Multicharacter symbol names may
    be used.

    inline(EXPR, N), where N is a scalar, constructs an
    inline function whose input arguments are 'x', 'P1',
    'P2', ..., 'PN'.

    Examples:
      g = inline('t^2')
      g = inline('sin(2*pi*f + theta)')
      g = inline('sin(2*pi*f + theta)', 'f', 'theta')
      g = inline('x^P1', 1)

    See also symvar, function_handle.

    Reference page for inline
    Other functions named inline

>> clear all
>>
```

So, the same way I want to define the function. So, in this case I will just give the help inline. So, this gives the inline construct, inline objects. So, inline functions means while you are doing some calculation or you are writing some code and in between you want to define functions. So, that is called the inline function.

So, this is the expression of a function or maybe I can define that g is the handler, the name of the function and then I can define inline t square or inline this function we have defined. So, similarly I can define the same function. So, let us define another function. So, before that I just clear all. Now clc.

(Refer Slide Time: 23:30)



```
>> %F=inline('function formaula');
>> F=inline('x^2*sin(x)');
>> F(0)

ans =

     0

>> F(1)

ans =

   0.841470984807897

>> x=[0 1];
>> F(x)
Error using inlineeval (line 14)
Error in inline expression ==> x^2*sin(x)
 Inputs must be a scalar and a square matrix.
To compute elementwise POWER, use POWER (.^) instead.

Error in inline/subsref (line 23)
    INLINE_OUT_ = inlineeval(INLINE_INPUTS_, INLINE_OBJ_.inputExpr,
    INLINE_OBJ_.expr);

>> F=inline('x.^2.*sin(x)');
>> F(x)

ans =

                   0   0.841470984807897

>> x=[0 pi/4 pi/2 3*pi/4 pi]

x =

  Columns 1 through 3

                   0   0.785398163397448   1.570796326794897

  Columns 4 through 5

   2.356194490192345   3.141592653589793

>> F(x)
```

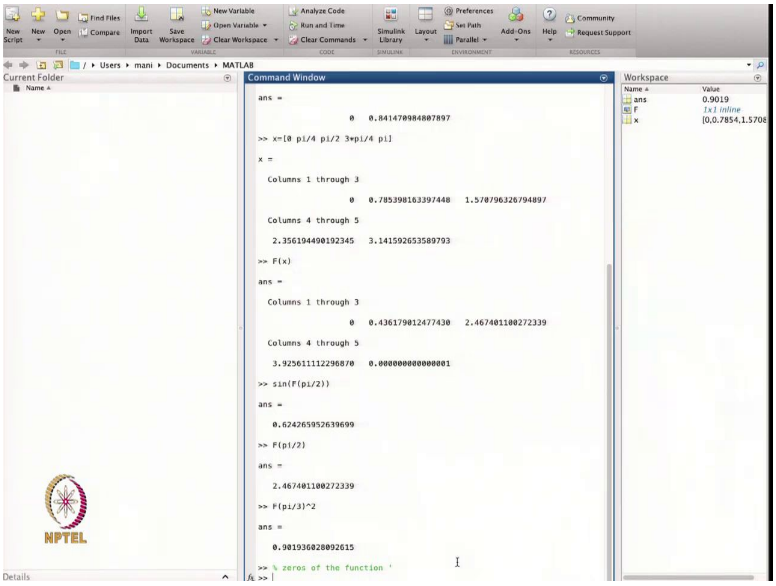
Name	Value
ans	[0,0.8415]
F	1x1 inline
x	[0,0.7854,1.5708

So, in this case you just want to define capital F is equal to let us define inline and then I so that is the way I can define and here I will give you the function formula function formula and that is the way we can define the function. So, before that I just put the percentage sign. So, that is the command. Now so suppose I define f is equal to inline. Maybe I defined x square into multiplied by sin x.

So, this is a function I just defined. So, that is my capital F. Now I can find what is the value of capital F at 0 or what is the value of the capital F(1). So, all the values we can put the same way. We have defined the anonymous function the same way we have defined the inline function or maybe I can define the because now suppose I want to give the value of the function.

I just define x = 0 and 1. So, this is the vector I have defined and now I want to give this value. So, it will not accept the value. Because this function is a scalar function. So, the same way I have to redefine the function by the vector form.

So, putting here dot before the exponential sign and the multiplication sign. Now, I can define my function and that gives you the value of the function at x = 0 and x = 2. So, in this case maybe I can define this function for 0 maybe π/4 , π/2 , π / 3, π/4 and π. So, this is the function. The value of the x I am defining and then I can define the value.
(Refer Slide Time: 25:57)



```
ans =

                 0   0.841470984807897

>> x=[0 pi/4 pi/2 3*pi/4 pi]

x =

  Columns 1 through 3

                 0   0.785398163397448   1.570796326794897

  Columns 4 through 5

   2.356194490192345   3.141592653589793

>> F(x)

ans =

  Columns 1 through 3

                 0   0.436179012477430   2.467401100272339

  Columns 4 through 5

   3.925611112296870   0.000000000000001

>> sin(F(pi/2))

ans =

   0.624265952639699

>> F(pi/2)

ans =

   2.467401100272339

>> F(pi/3)^2

ans =

   0.901936028092615

>> % zeros of the function '
>> |
```

Name ▲	Value
ans	0.9019
F	1x1 inline
x	[0,0.7854,1.5708

So, that is the value of the function here. Now maybe I can define what will be the sign of the function. Capital F(π/2), so this one I can use. So, first it will give you the value of the f. So, this is π/ 2 and then I have defined the sign of this one or maybe this is the way we can define another function or another value I can define.

Maybe I can find the value of f, some value at π/3 and then I can take the square of this one. So, that gives you the value of the square order of this one or maybe I can define the zeros. So, I can define the zeros of the functions. So, this one will do in the, so zeros of the function and other commands will continue in the next lecture.

So, in this lecture we just talked about the anonymous function and the inline function and then how we can use this function in between to find out to do the more computation using these functions. So, in the next lecture we will continue with the basics of the Matlab. So, thanks very much for viewing this one.