

Plotting in Matlab

The goal of this document is to assist you in plotting functions using Matlab.

The following instructions can be manually inputted into the command window of Matlab. In plotting functions, computers evaluates the function on a finite number of points. Languages such as Mathematica will automatically produce the finite points to plot, however Matlab requires the user to produce the array themselves. To do this we can define an empty vector "x" that will collect the plot points.

```
>> x=[];
```

Next we can employ a for-loop that will create the evenly spaced points The for-loop acts by performing a calculation

```
>> for j=0:1000
x=[x j/100];
end;
```

a thousand times over. Each time the calculation is performed, the index "j" (begining at zero) increases by one until it reaches 1000. The calculation performed (second line of the above figure) appends one one-hundredth of the index "j" to the vector "x". Note the presence of the semi-colon at the end of the calculation, it acts to suppress the output of the for-loop, had it not been there, each iteration of the for-loop would have have displayed the vector "x" in the command window. Alternatively, there are built in functions in Matlab that will generate an array of evenly spaced values.

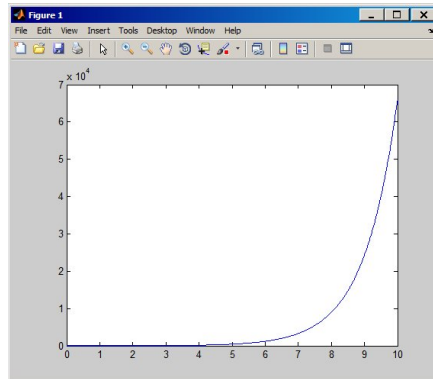
```
>> x=linspace(0,10,1000)
```

The linspace function above creates 1000 evenly spaced points between 0 and 10. Now we'll take the "x" vector and evaluate it on a simple exponential function. The "y" is a vector with values corresponding to each entry of the "x" vector.

```
>> y=3*exp(x);
```

Now all there is to do is plot our two vectors together.

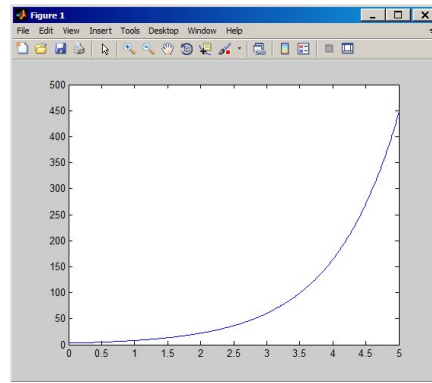
```
>> plot(x,y)
```



Note that Matlab has automatically connected each point.

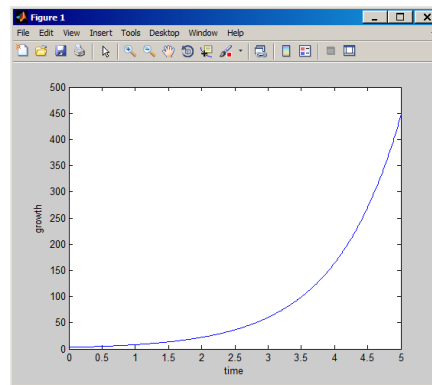
From this point it's just a matter of appearance. We can change the range of the axes

```
>> axis([0 5 0 500])
```



Next we can add labels to the axes

```
>> xlabel('time');  
>> ylabel('growth');
```

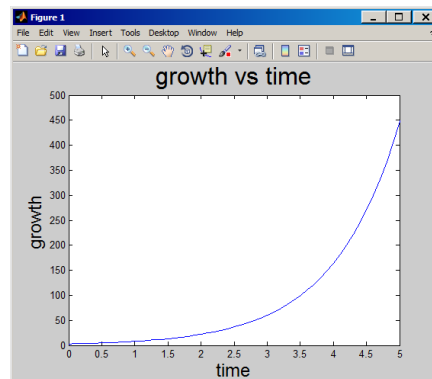


Alternatively we could have included a third argument in the label function to change the font size

```
>> xlabel('time','FontSize',18)
```

The last thing you can include is a title for the plot

```
>> title('growth vs time','FontSize',24)
```



This is the basics of plotting with Matlab. More than what was shown in this document can be done with Matlab and an excellent resource is the MathWorks website.