

# Matlab Tutorial for Plotting Data and Solving Equations

Quan Wang, ECSE, RPI  
IEA Graduate TA for Section 3  
wangq10@rpi.edu

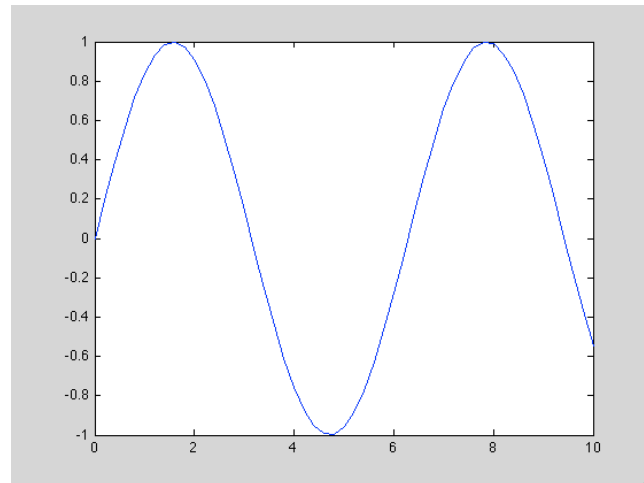
# My Office Hours

- \* Location: JEC 2001
- \* Monday: 4:00 – 5:00
- \* Tuesday: 1:00 – 5:00
- \* Friday: 1:00 – 4:00

# Plot(x,y)

- \* If you have two vectors  $x$  and  $y$ , then `plot(x,y)` will visualize the data  $(x_i, y_i)$  in the 2D  $x$ - $y$  plane
- \*  $x$  and  $y$  must have the same size
- \* E.g.

```
>> x=0:0.2:10;  
>> y=sin(x);  
>> plot(x,y)
```



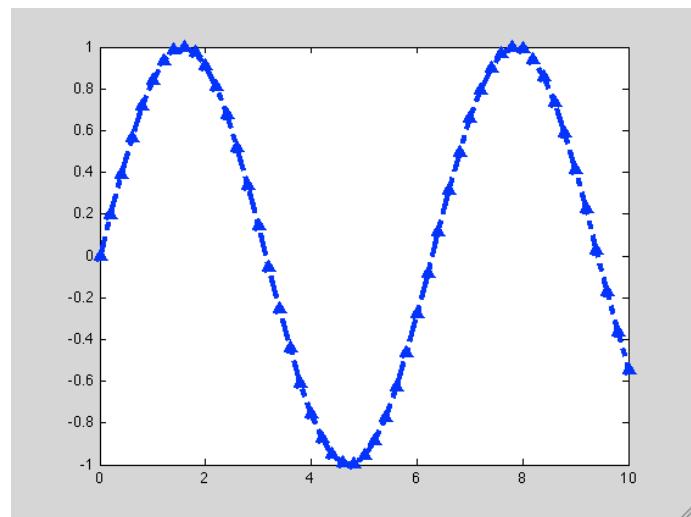
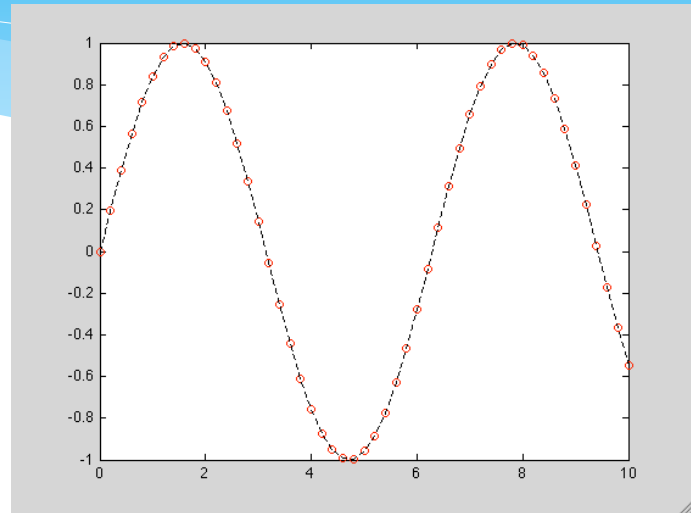
# LineStyle

- \* By default, the data points will be jointed with line segments. If there are plenty of data, the curve will look smooth
- \* We can customize the line by `plot(x,y,LineStyle)`
- \* LineSpec is a string, defined by quotation marks
- \* LineSpec may contain line style, color, width, marker type, marker size
- \* Just use Matlab help and search for LineSpec when you need to plot a specific line

# LineStyle

\* E.g.

```
>> plot(x,y,'or')  
>> hold on  
>> plot(x,y,'--k')  
>> figure  
>> plot(x,y,'-.^b','LineWidth',3)
```



# Hold and Figure

- \* By using **hold on** command, the next figure will directly plot on the previous figure
- \* By using **hold off** command, the next figure will erase the previous figure and then plot
- \* By using **figure** command, the next figure will be plotted in a new window without closing the previous window

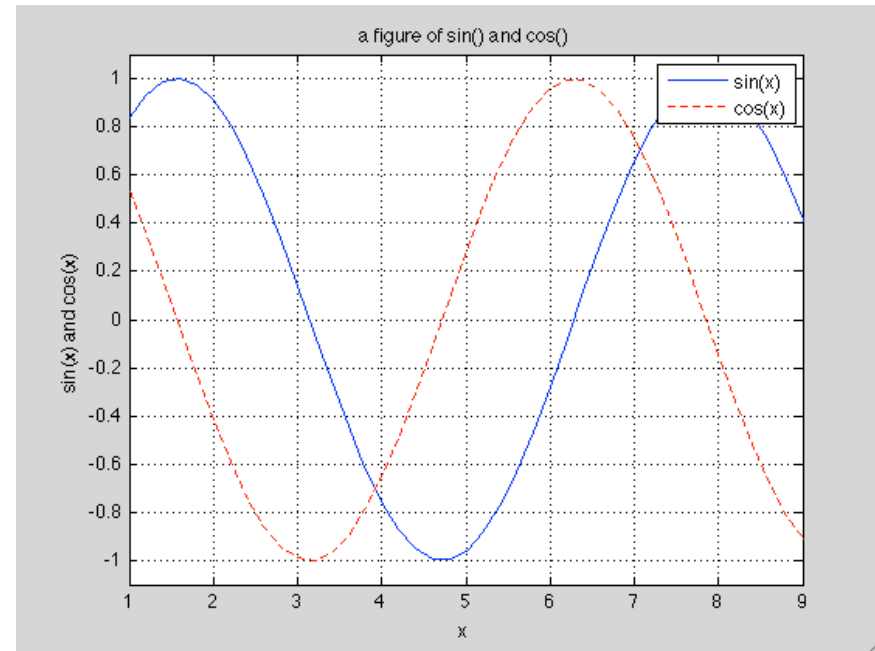
# Title, Label, Legend, Grid, and Axis

- \* `title('text')` will add a title to the figure
- \* `xlabel('text')` will add label to the x axis, similar is `ylabel('text')`
- \* `legend('text1','text2'...)` will display the legend of different lines
- \* **grid on** command will display the grid
- \* **axis** is a series of commands about operations on axis

# Title, Label, Legend, Grid, and Axis

\* E.g.

```
>> x=0:0.2:10;  
>> y=sin(x);  
>> z=cos(x);  
>> plot(x,y,'-b');  
>> hold on  
>> plot(x,z,'--r');  
>> title('a figure of sin() and cos()')  
>> xlabel('x')  
>> ylabel('sin(x) and cos(x)')  
>> grid on  
>> legend('sin(x)', 'cos(x)');  
>> axis([1 9 -1.1 1.1])
```

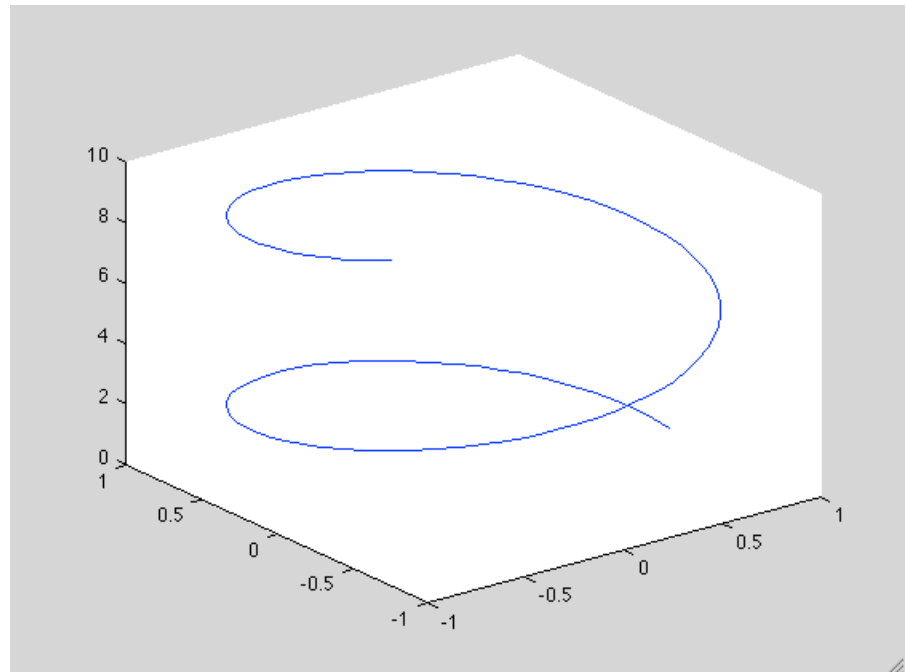




# Plot3(x,y,z)

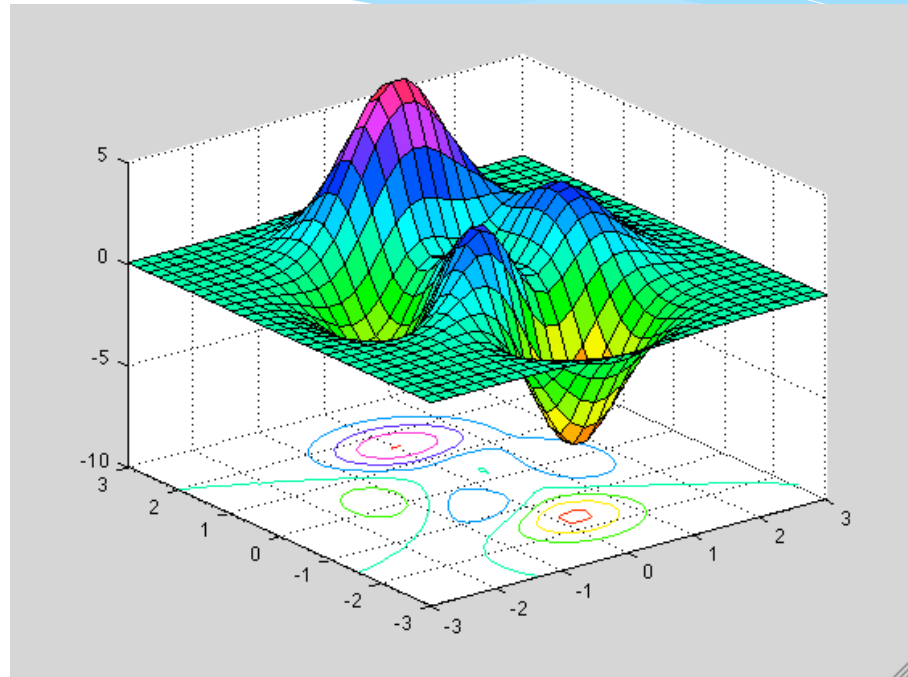
- \* `plot3(x,y,z)` visualize a line in 3D space
- \* E.g.

```
>> x=0:0.1:10;  
>> plot3(cos(x),sin(x),x);
```



# Other Functions for 2D or 3D Visualization

- \* `surf()`
- \* `surface()`
- \* `polar()`
- \* `imagesc()`
- \* `fplot()`
- \* `subplot()`
- \* Explore by yourself!



# Symbols and Solving Equations

- \* `x=sym('x')` will define a symbol `x`
- \* `solve(eq)` will solve for the zeros of an equation with symbols
- \* E.g.

```
>> x=sym('x')
```

```
x =
```

```
x
```

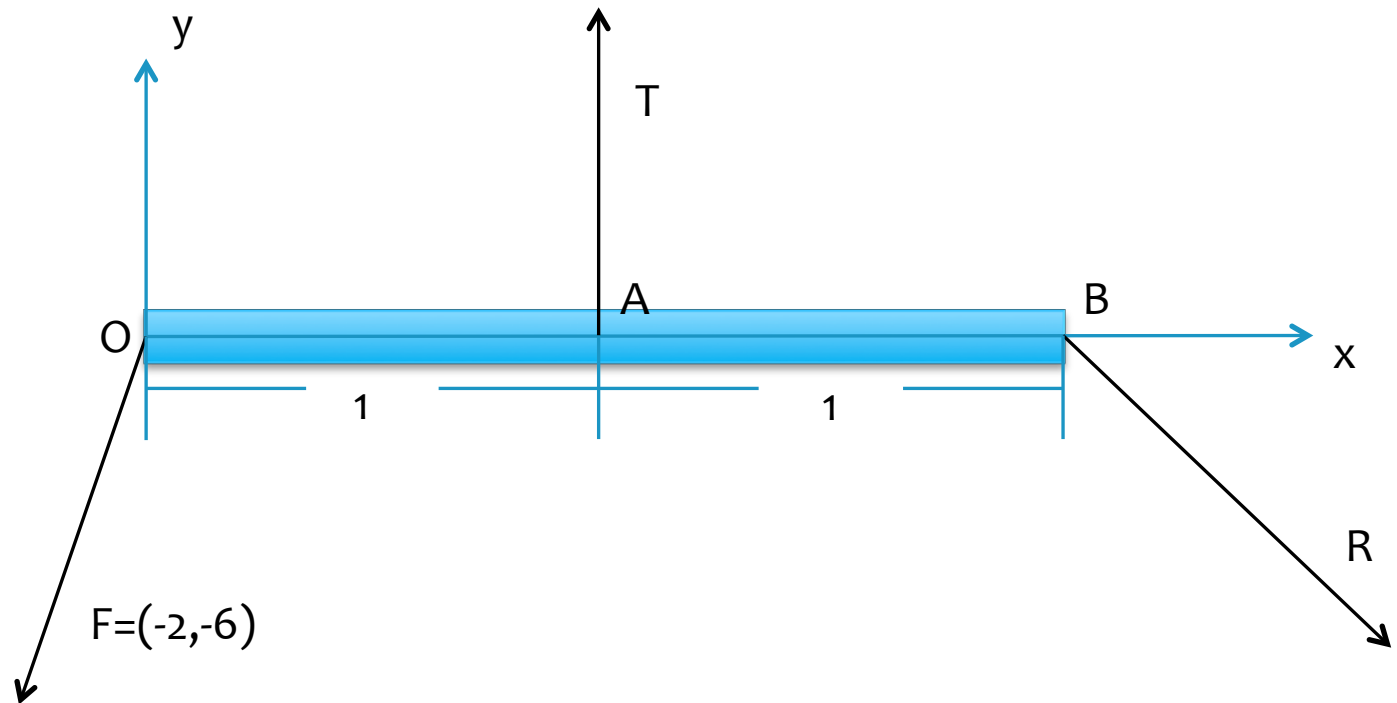
```
>> solve(x-5)
```

```
ans =
```

```
5
```

# Symbols and Solving Equations

\* E.g. For this equilibrium system, what is T and R?



# Symbols and Solving Equations

```
Ty=sym('Ty');  
T=[0,Ty,0];  
F=[-2,-6,0];  
Rx=sym('Rx');  
Ry=sym('Ry');  
R=[Rx,Ry,0];  
result=solve(T+F+R,cross([1,0,0],T)+cross([2,0,0],R));  
result.Ty  
result.Rx  
result.Ry
```

Output:

Warning: 6 equations  
in 3 variables.

ans =

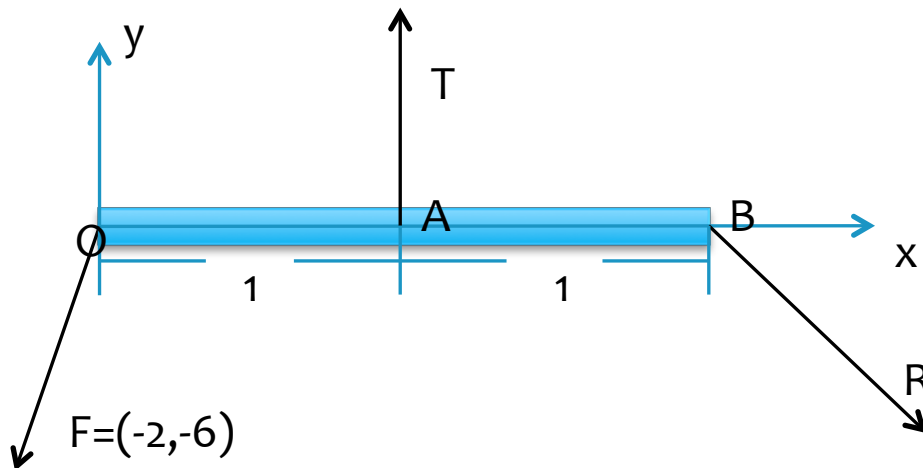
12

ans =

2

ans =

-6



# Symbols and Solving Equations

\* Verify our results:

```
T2=[0,result.Ty,0];  
R2=[result.Rx,result.Ry,0];  
T2+F+R2  
cross([1,0,0],T2)+cross([2,0,0],R2)
```

```
ans =
```

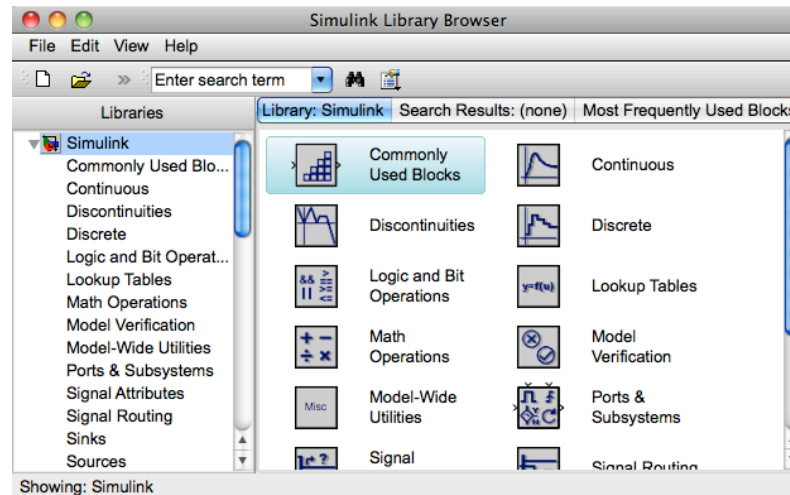
```
[ 0, 0, 0]
```

```
ans =
```

```
[ 0, 0, 0]
```

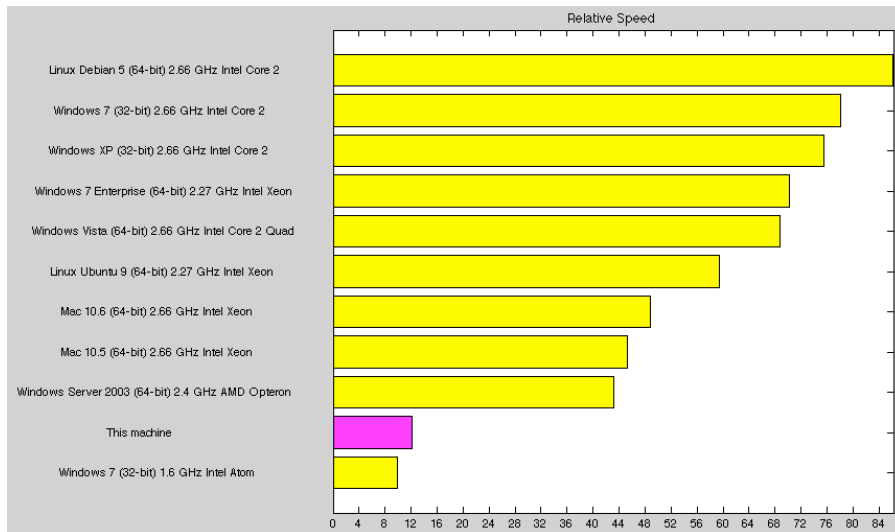
# Fun Topics: Simulink

- \* Type **simulink** then press enter in the command window, you are in a new world
- \* Explore by yourself



# Fun Topics: Bench

- \* Type **bench** then press enter in the command window, the performance of your computer will be assessed

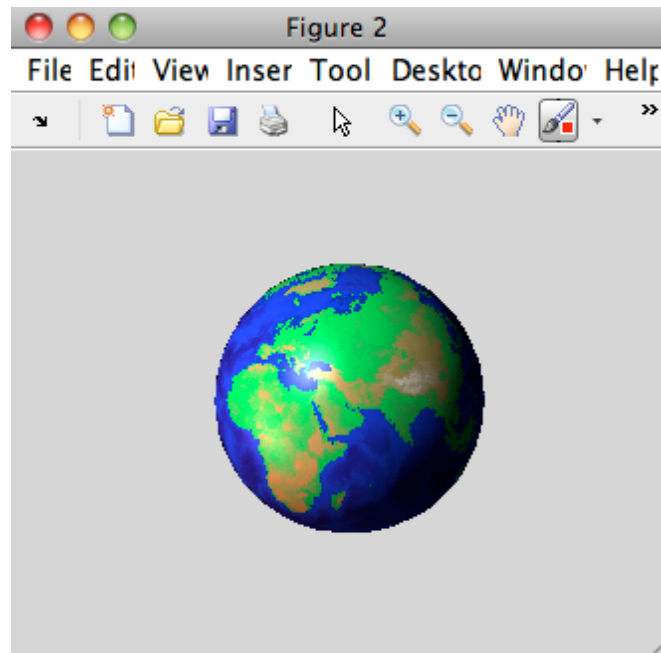


Computer Type	LU	FFT	ODE	Sparse	2-D	3-D
Linux Debian 5 (64-bit) 2.66 GHz Intel Core 2	0.0665	0.0955	0.1619	0.2600	0.2581	0.2109
Windows 7 (32-bit) 2.66 GHz Intel Core 2	0.0759	0.1188	0.1138	0.2425	0.3933	0.3341
Windows XP (32-bit) 2.66 GHz Intel Core 2	0.0760	0.1127	0.1109	0.2392	0.3036	0.5459
Windows 7 Enterprise (64-bit) 2.27 GHz Intel Xeon	0.0405	0.0570	0.1871	0.2114	0.4217	0.7185
Windows Vista (64-bit) 2.66 GHz Intel Core 2 Quad	0.0735	0.1026	0.1964	0.2728	0.3955	0.3876
Linux Ubuntu 9 (64-bit) 2.27 GHz Intel Xeon	0.0480	0.0464	0.1639	0.2439	0.6367	0.9800
Mac 10.6 (64-bit) 2.66 GHz Intel Xeon	0.0735	0.1566	0.1789	0.3330	0.6779	0.8531
Mac 10.5 (64-bit) 2.66 GHz Intel Xeon	0.0801	0.1623	0.1783	0.5197	0.6881	0.7907
Windows Server 2003 (64-bit) 2.4 GHz AMD Opteron	0.1039	0.1322	0.2701	0.7535	0.5609	0.5016
This machine	0.2513	0.1458	0.1858	6.8818	0.6999	0.8306
Windows 7 (32-bit) 1.6 GHz Intel Atom	1.8847	0.7619	0.6514	1.5816	2.3777	1.9173



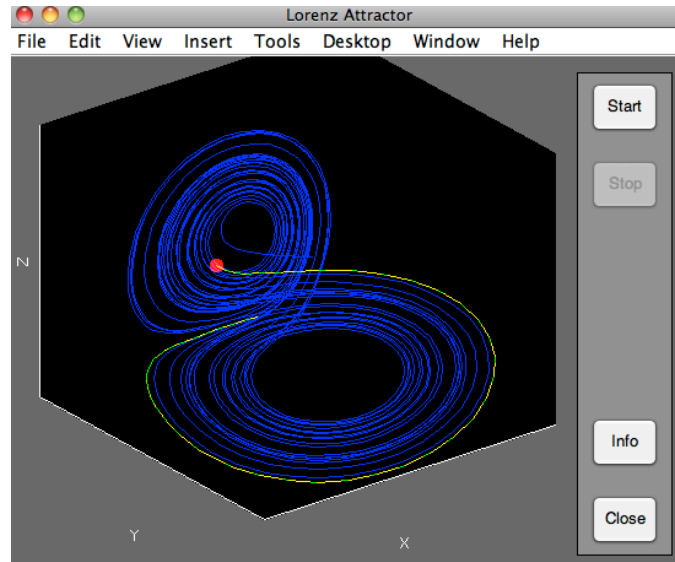
# Fun Topics: Earthmap

- \* Type **earthmap** then press enter in the command window, you will see the map of the earth



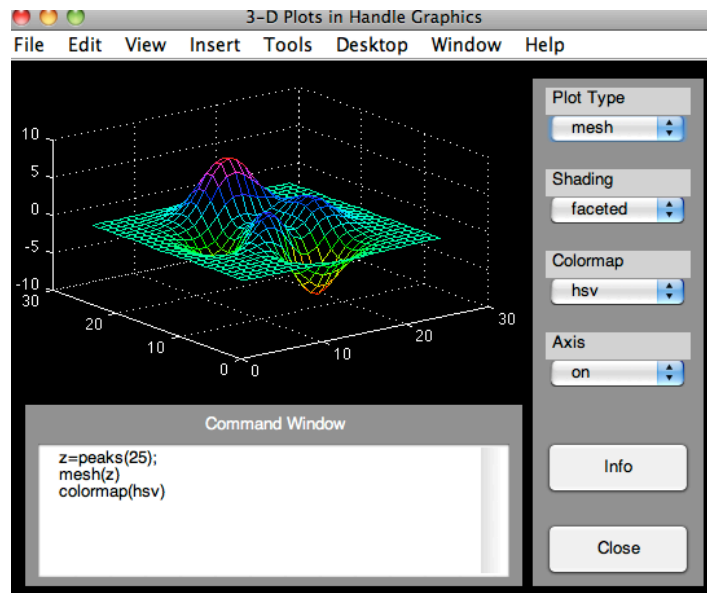
# Fun Topics: Lorenz

- \* Type **lorenz** then press enter in the command window, you will see an application for the animation of the orbit around the Lorenz chaotic attractor



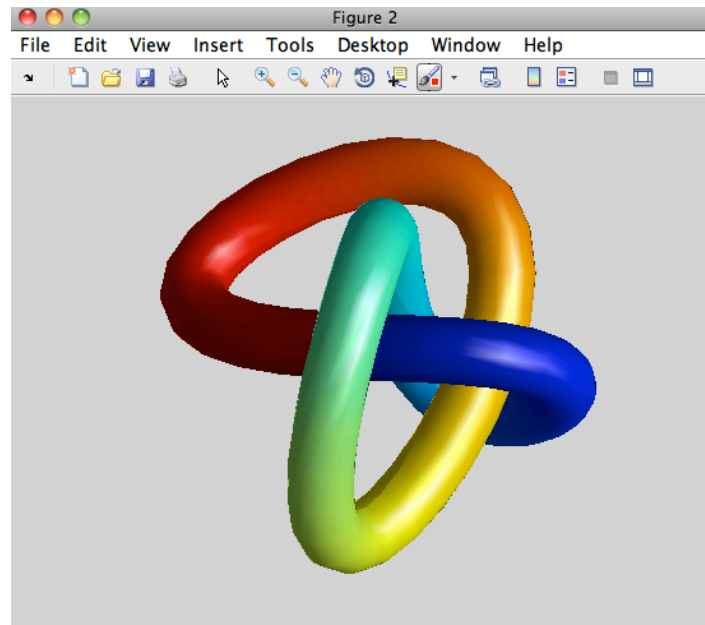
# Fun Topics: Graf3d

- \* Type `graf3d` then press enter in the command window, you will see a demo for the graphic visualization functions of Matlab



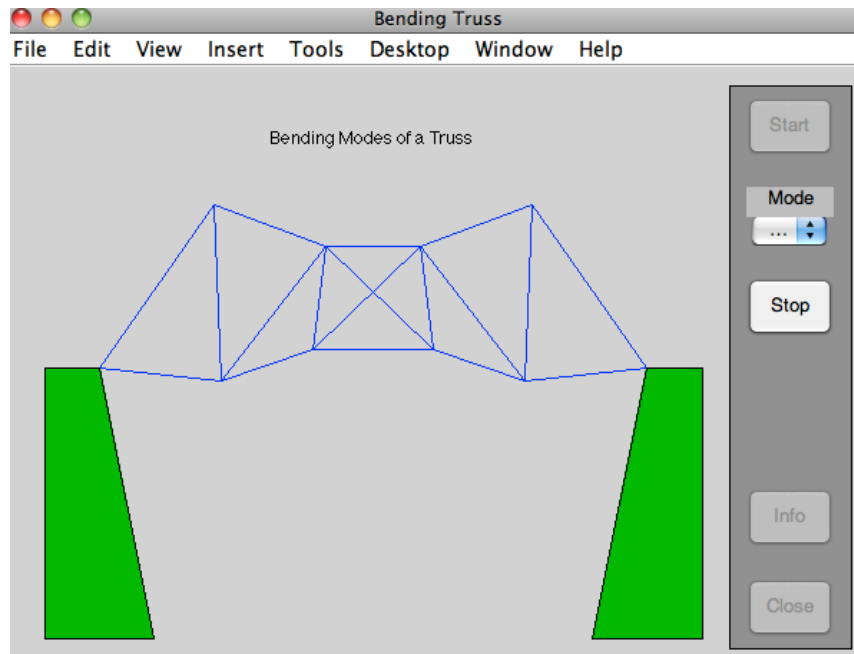
# Fun Topics: Knot

- \* Type **knot** then press enter in the command window, you will see a 3D knot



# Fun Topics: Truss

- \* Type **truss** then press enter in the command window, you will see a demo for the bending modes of a truss





# Thank you!

This is the last Matlab session, but this is just the beginning for you to use Matlab as an engineer