# Plots and Plotting Software

*IDC603A, 2017-18 , SEMESTER II*

*IIT Kanpur*

# Utility of Plots

- Visualisation
- Description
- Emphasis and analysis

# What to Plot?

- Data
- Mathematical relationships
- Comparable information

# Some Common Practices

- Axis labels and legends
- Choice of scale
- Colour and labels
- Consistency of notation

# Some Common Practices [contd...]

- Legibility
- Look & feel/cluttering
- PPT and/or LaTeX integration
- Coding and hardware issues

# Plotting Software

- **MATLAB**
- **PYTHON LIBRARIES**
- **GNUPlot**
- **MATHEMATICA**

# How to Choose One!

- Application specific
- Quick look or deeper analysis?
- Coding and memory issues
- Liberty with look and feel

# Various Plots and When to Use Them

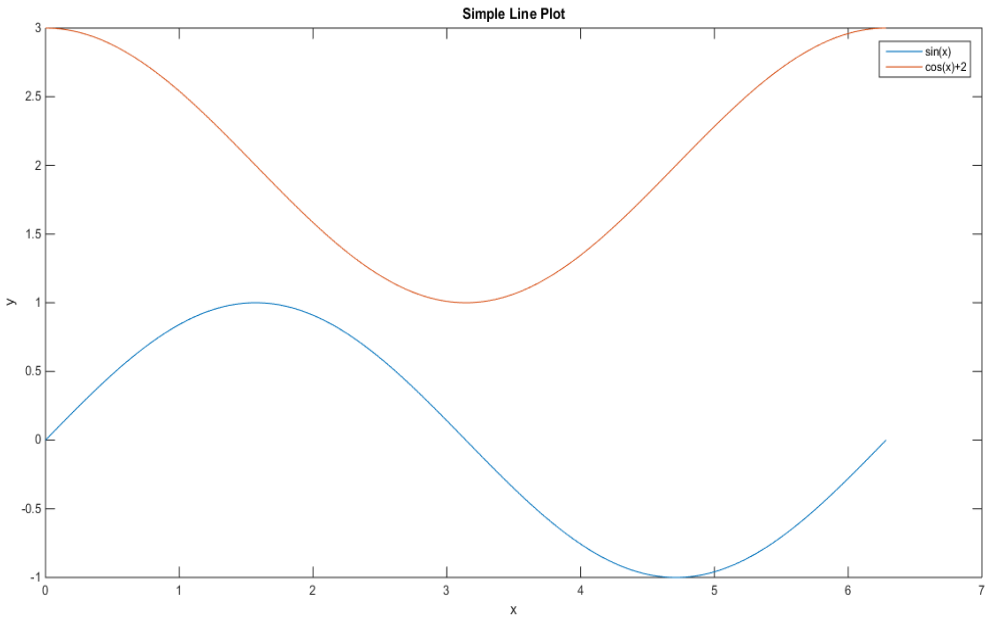
- Line plots
- Scatter plots
- Surface plots
- Contour plots
- Boxplots
- Bar diagram
- Pie charts

# Plotting with *MATLAB*

- Diverse Tools and Functions
- Excellent Documentation
- Computations governed by numerical linear algebra

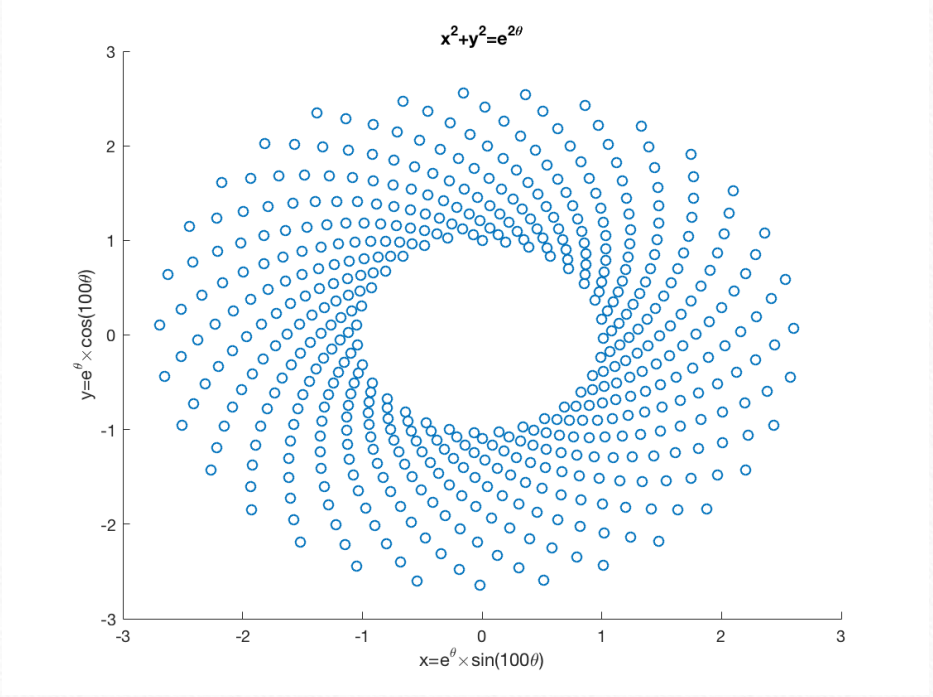
# Line Plots

```
x = linspace(0,2*pi,1000);
y1 = sin(x);
plot(x,y1); hold on
y2 = cos(x)+2; plot(x,y2)
hold off
title('Simple Line Plot');
xlabel('x'); ylabel('y');
legend('sin(x)','cos(x)+2');
```



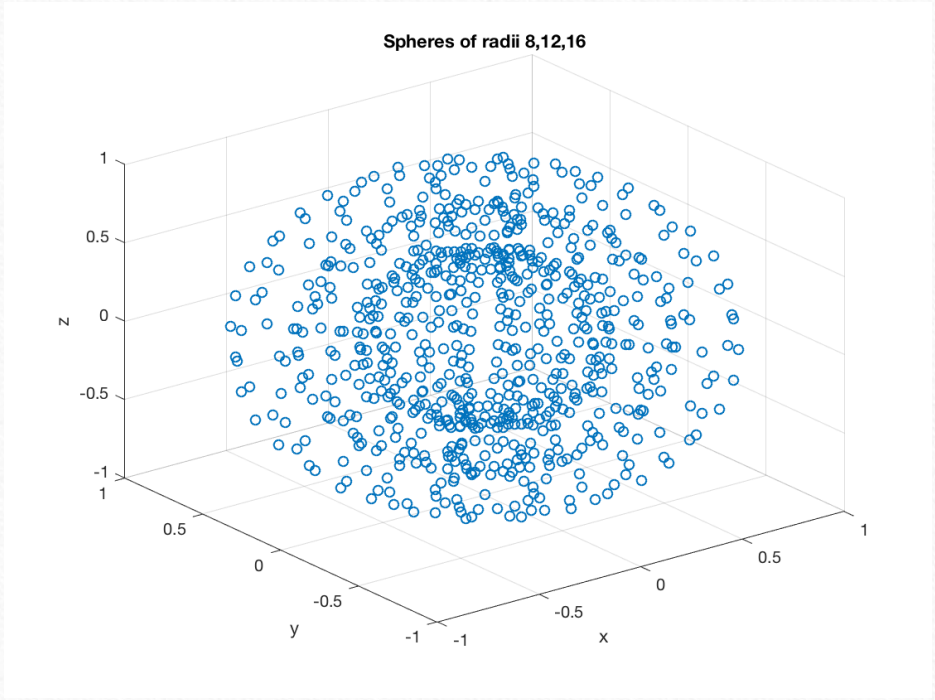
# Plotting of Points: Scatter Plots

```
theta = linspace(0,1,500);
x = exp(theta).*sin(100*theta);
y = exp(theta).*cos(100*theta);
s = scatter(x,y);

title('x^2+y^2=e^{2\theta}')

xlabel('x=e^{\theta}\timessin(100
\theta)')

ylabel('y=e^{\theta}\timescos(100
\theta)')
```



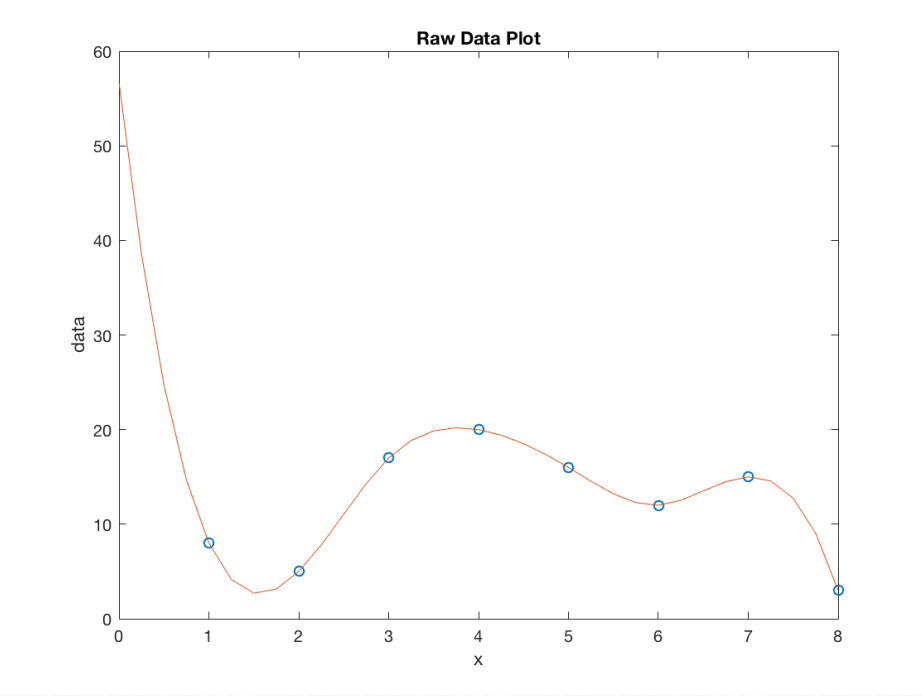
# 3-D Scatter Plots

```
[X,Y,Z] = sphere(16);
x = [0.5*X(:); 0.75*X(:); X(:)];
y = [0.5*Y(:); 0.75*Y(:); Y(:)];
z = [0.5*Z(:); 0.75*Z(:); Z(:)];
scatter3(x,y,z)

title('Spheres of radii 8,12,16')

xlabel('x')

ylabel('y')

zlabel('z')
```



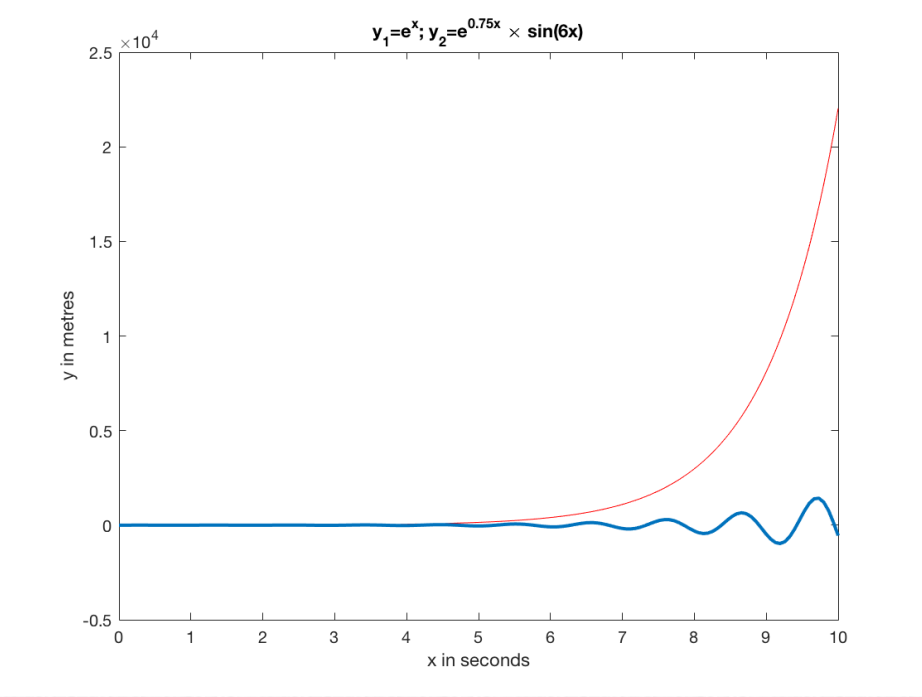
# Spline Plots

```
x=[1,2,3,4,5,6,7,8];
y=[8,5,17,20,16,12,15,3];
xx=0:0.25:8;
yy=spline(x,y,xx);
plot(x,y,'o',xx,yy)
title('Raw Data Plot')
xlabel('x')
ylabel('data')
```



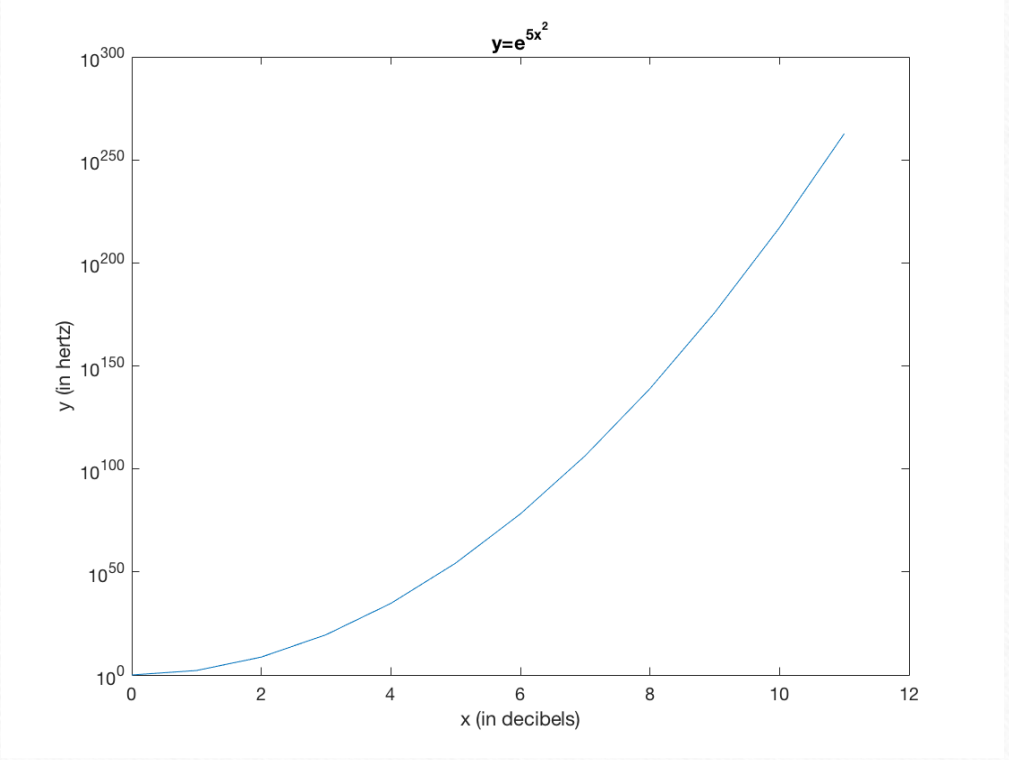
# Plotting of Curves

```
t=linspace(0,10,150);
y1=exp(t);
y2=exp(0.75*t).*sin(6*t);
plot(t,y1,'r');
hold on
plot(t,y2,'LineWidth',2)
hold off
title('Time response')
xlabel('t (in seconds)')
ylabel('y (in metres)')
legend('y_1=e^{t}','y_2=e^{0.75t}
\times sin(6t)')
```



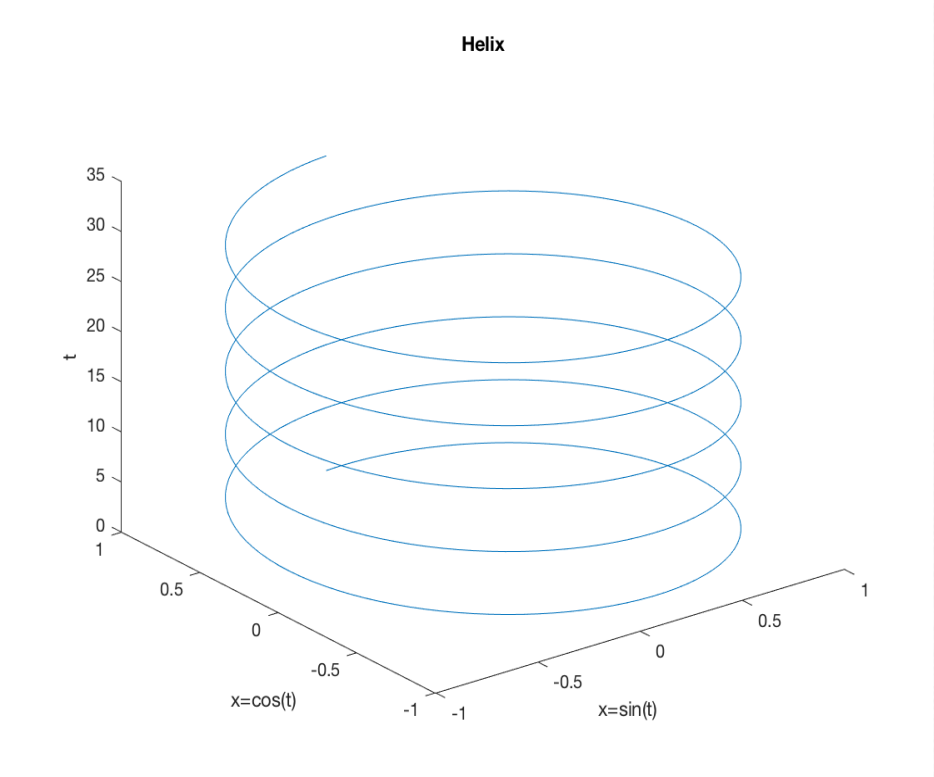
# Semi-log Plots

```
x=0:1000;
y=exp(5*x.^2);
semilogy(x,y)
title('y=e^{5x^2}')
xlabel('x (in decibels)')
ylabel('y (in hertz)')
```



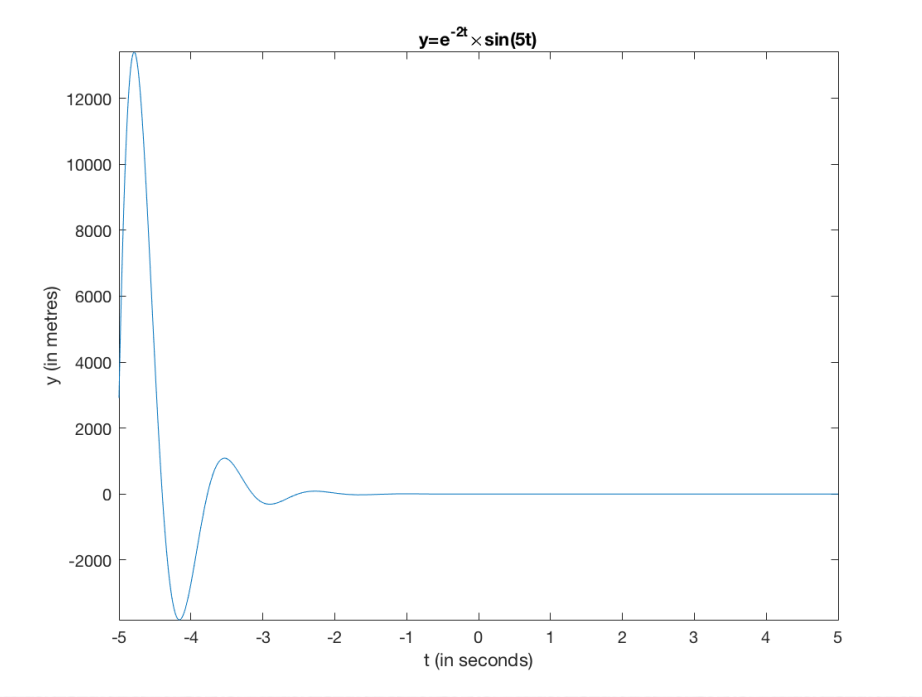
# Plotting of 3-D Curves

```
t = 0:pi/50:10*pi;
x = sin(t);
y = cos(t);
plot3(x,y,t)
title('Helix')
xlabel('x=sin(t)')
ylabel('x=cos(t)')
zlabel('t')
```



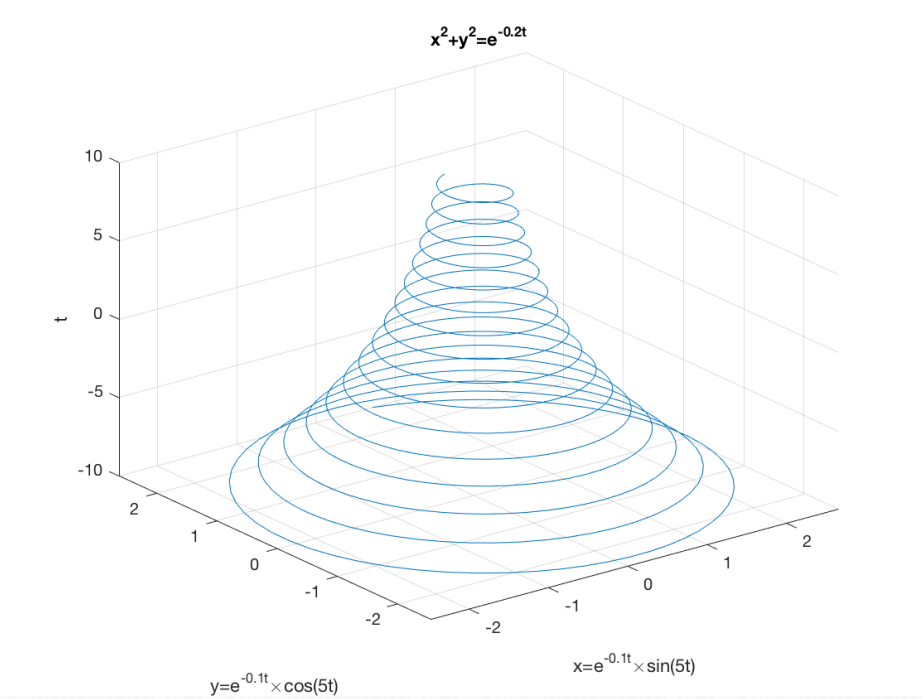
# Plotting a Function in 2-D

```
fplot(@(t) exp(-
2*t)*sin(5*t))

title('y=e^{-
2t}\timessin(5t)')

xlabel('t (in
seconds)')

ylabel('y (in metres)')
```



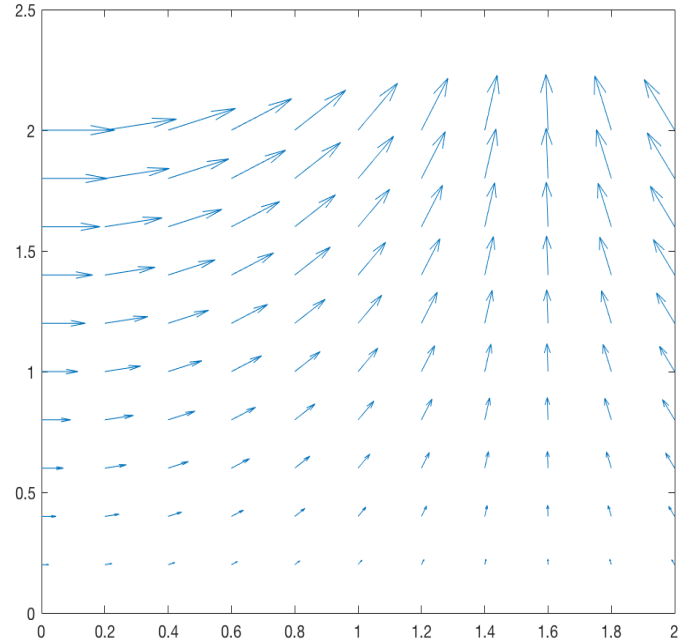
# Plotting of a Function in 3-D

```
xt = @(t) exp(-t/10).*sin(5*t);
yt = @(t) exp(-t/10).*cos(5*t);
zt = @(t) t;
fplot3(xt,yt,zt,[-10 10])

title('x^2+y^2=e^{-0.2t}')

xlabel('x=e^{-
0.1t}\timessin(5t)')

ylabel('y=e^{-
0.1t}\timescos(5t)')

zlabel('t')
```



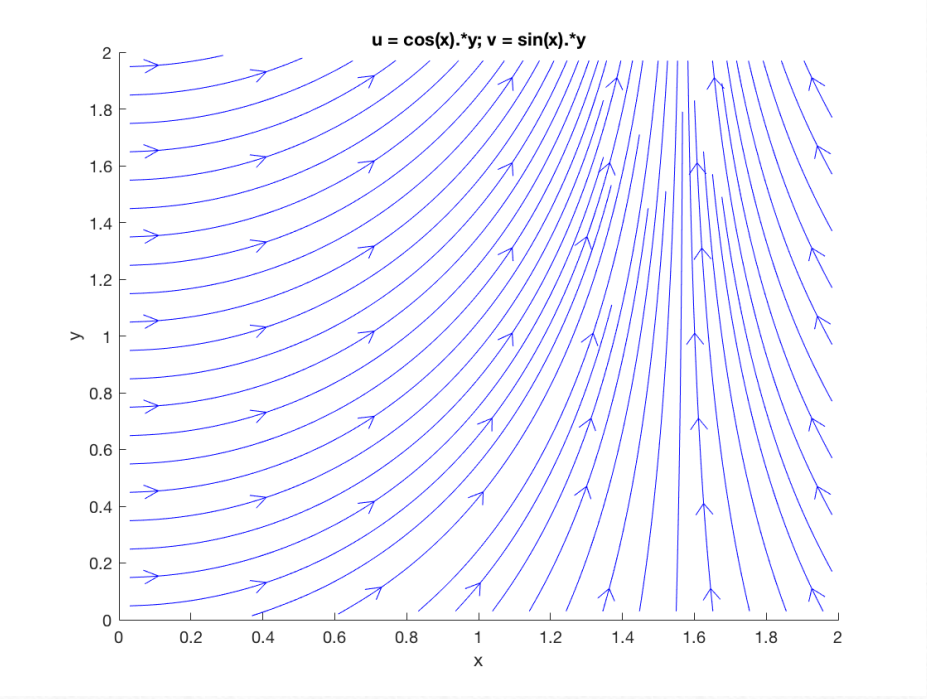
# Vector Fields

```
[x,y]=meshgrid(0:0.2:2,
0:0.2:2);
u = cos(x).*y;
v = sin(x).*y;
quiver(x,y,u,v)
```



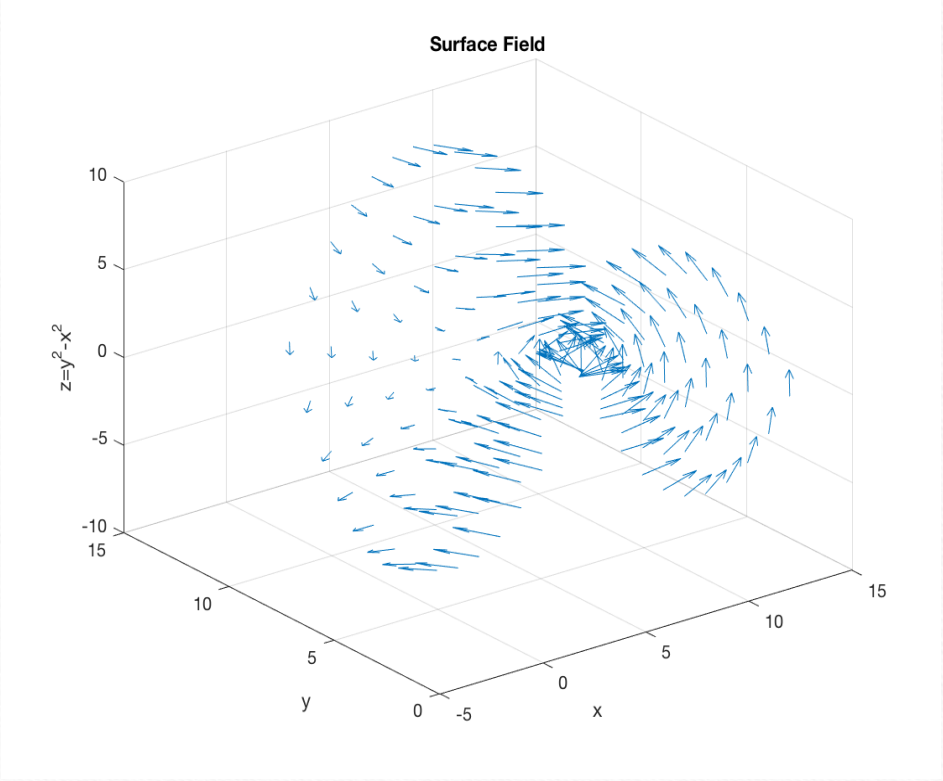
# Plotting of Streamlines on vector fields

```
[x,y]=meshgrid(0:0.2:2,0:0.2:2);
u = cos(x).*y;
v = sin(x).*y;
streamslice(x,y,u,v)
title('u = cos(x).*y; v =
sin(x).*y')
xlabel('x')
ylabel('y')
```



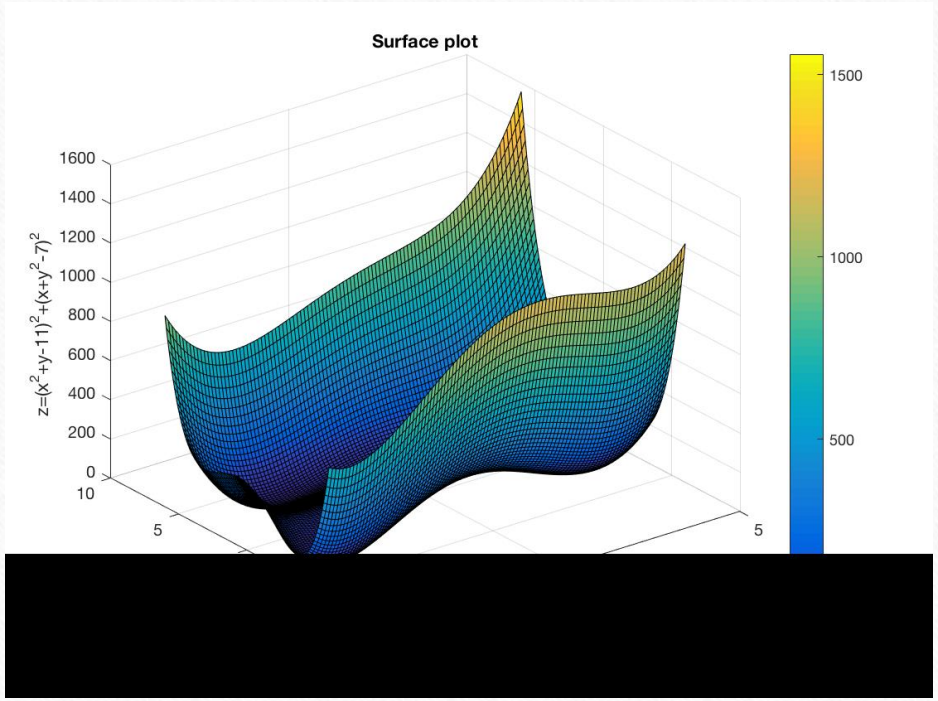
# 3-D Vector Fields

```
x = -3:0.5:3;
y = -3:0.5:3;
[X,Y] = meshgrid(x, y);
Z = Y.^2 - X.^2;
[U,V,W] = surfnorm(Z);
quiver3(Z,U,V,W)

title('Surface Field')

xlabel('x')

ylabel('y')

zlabel('z=y^2-x^2')
```



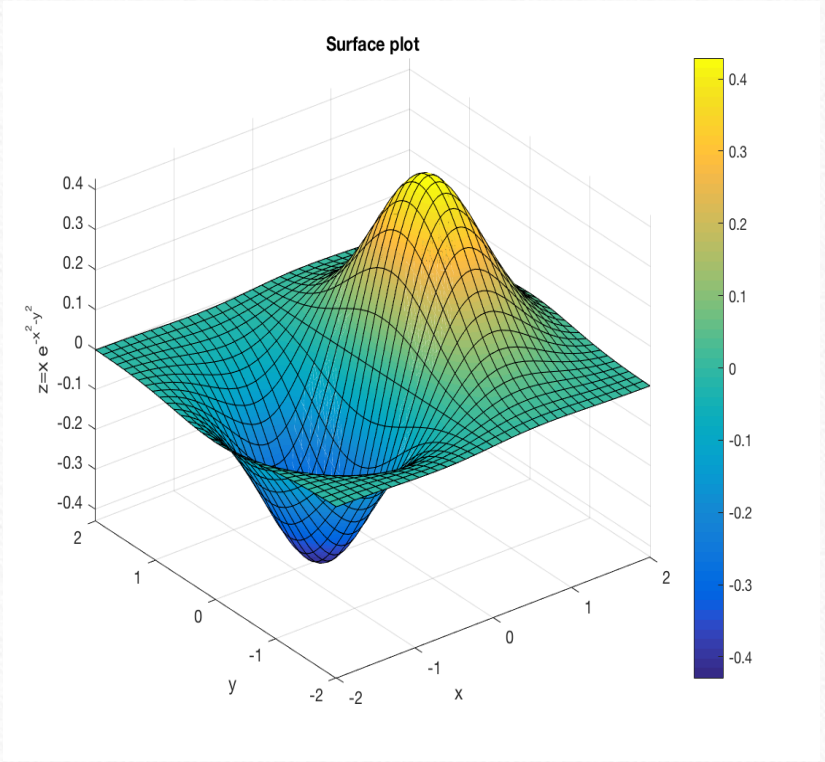
# 3-D Surface Plots

```
[x,y]=meshgrid(-5:0.1:5,-6:0.1:6);
z=(x.^2+y-11).^2+(x+y.^2-7).^2;
surf(x,y,z)
title('Surface plot')
xlabel('x')
ylabel('y')
zlabel('z=(x^2+y-11)^2+(x+y^2-7)^2 ')
colorbar
```



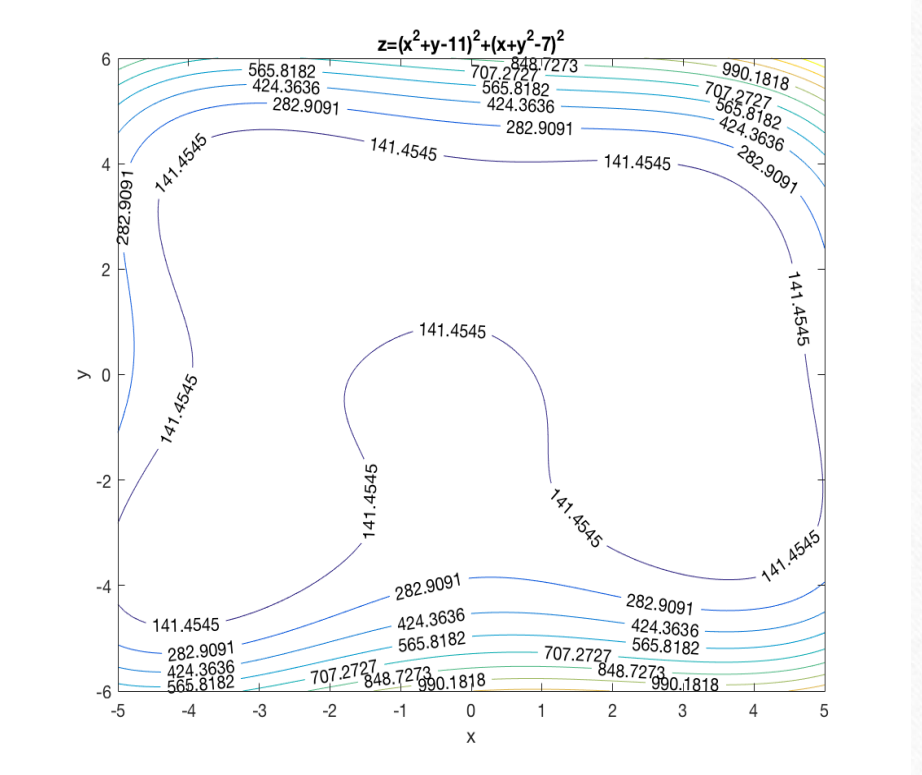
# 3-D Surface Plots (by function)

```
fsurf(@(x,y) x*exp(-x^2-y^2),[-2 2 -2 2])

title('Surface plot')
xlabel('x')
ylabel('y')
zlabel('z=x e^{-x^2-y^2}')
colorbar
```



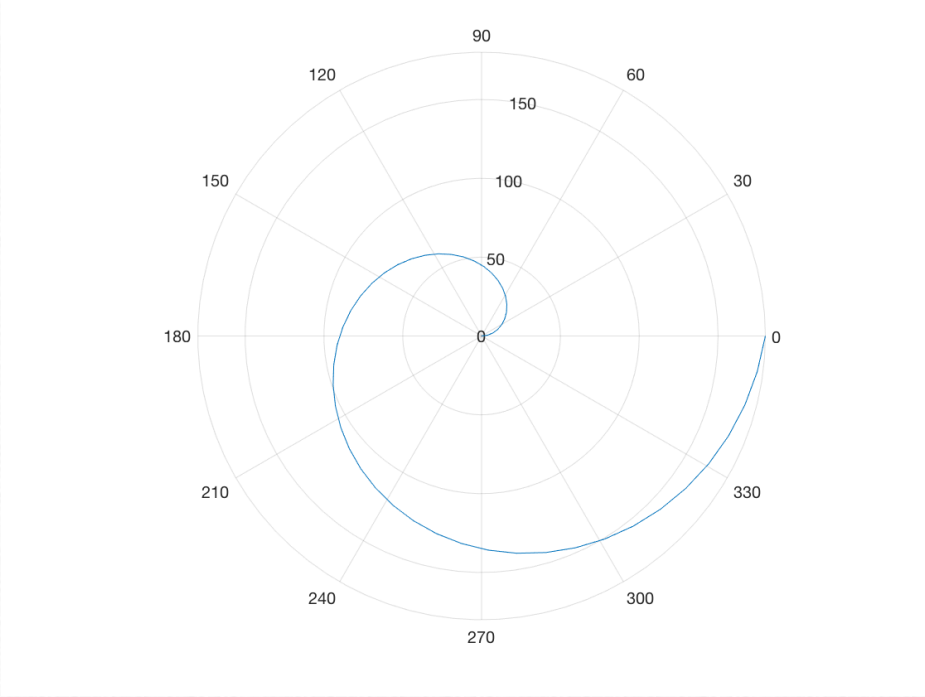
# 2-D Contour Plots

```
[x,y]=meshgrid(-5:0.1:5,-6:0.1:6);

z=(x.^2+y-11).^2+(x+y.^2-7).^2;

[c,f]=contour(x,y,z,10);

title('z=(x^2+y-11)^2+(x+y^2-7)^2')

xlabel('x')

ylabel('y')

clabel(c,f)
```



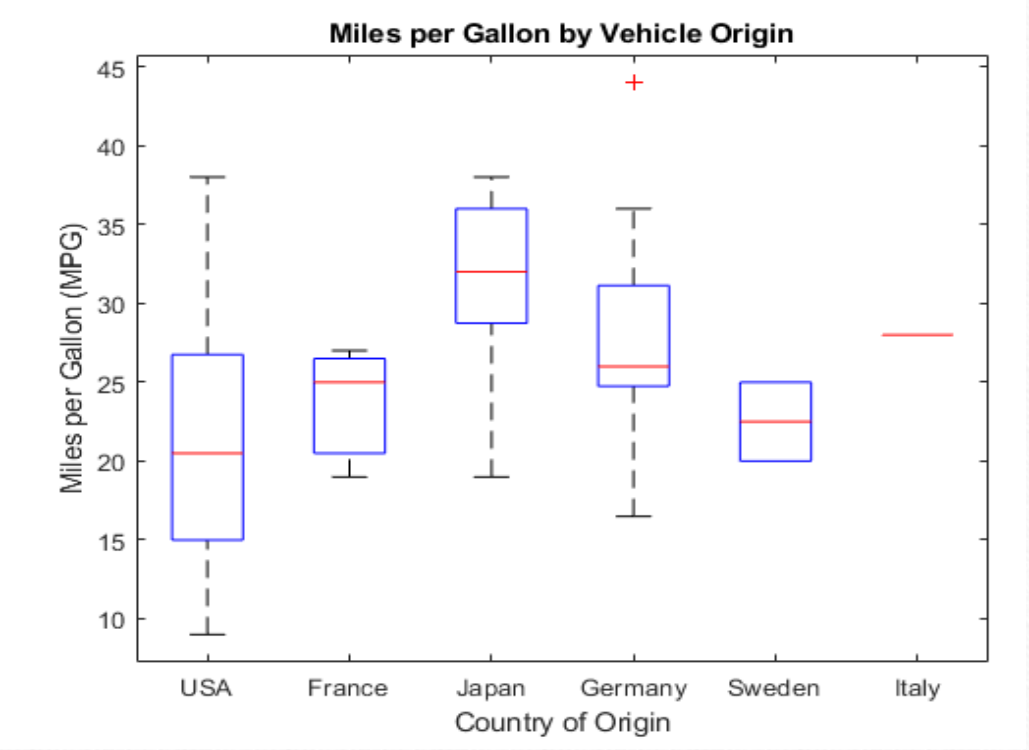
# Plotting in Polar Coordinates

```
theta=linspace(0,360,50);

rho=0005*theta/10;

theta_radians=deg2rad(the
ta);

polarplot(theta_radians,
rho)
```



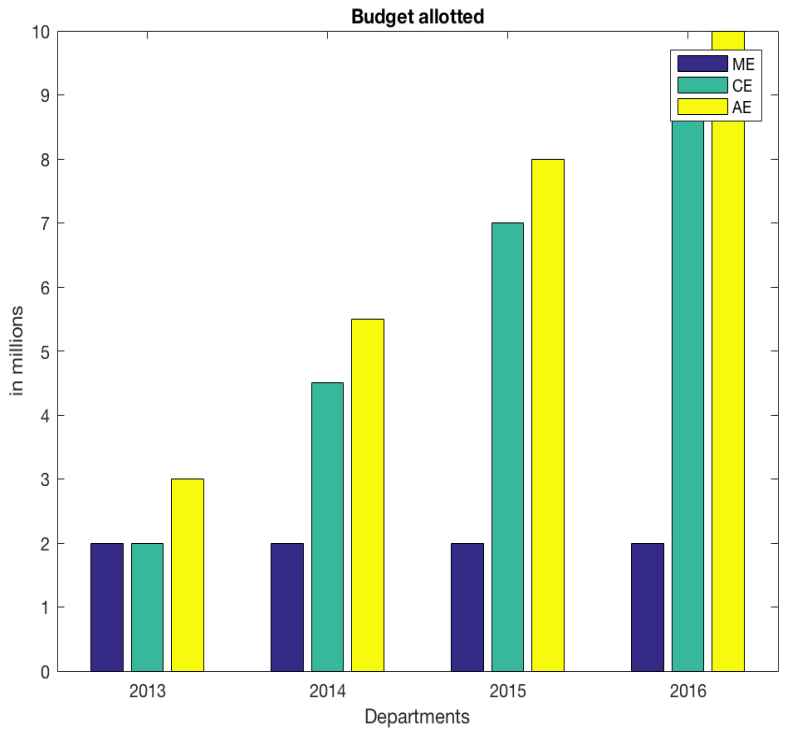
# Boxplots

```
load carsmall

boxplot(MPG,Origin)

title('Miles per Gallon by
Vehicle Origin')

xlabel('Country of Origin')

ylabel('Miles per Gallon (MPG)')
```



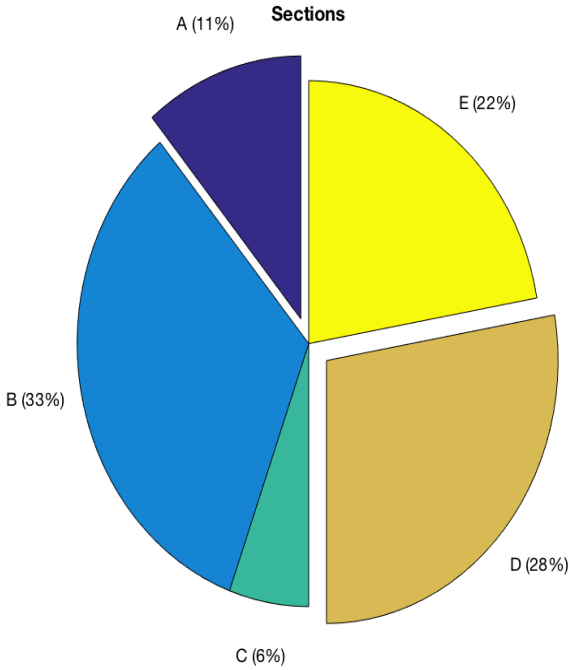
# Bar Diagrams

```
x=[2013:2016];

y=[2 2 3; 2 4.5 5.5; 2 7 8;
2 9 10];

bar(x,y)

title('Budget allotted')

xlabel('Departments')

ylabel('in millions')

legend('ME','CE','AE')
```



# Pie Diagrams

```
p=[1 3 0.5 2.5 2];

e=[1 0 0 1 0];

l={'A (11%)';'B (33%)';'C
(6%)';'D (28%)';'E
(22%)'};

pie(p,e,l)

title('Sections')
```



# MATLAB Figure Properties

# Best Way to Learn is to Code!

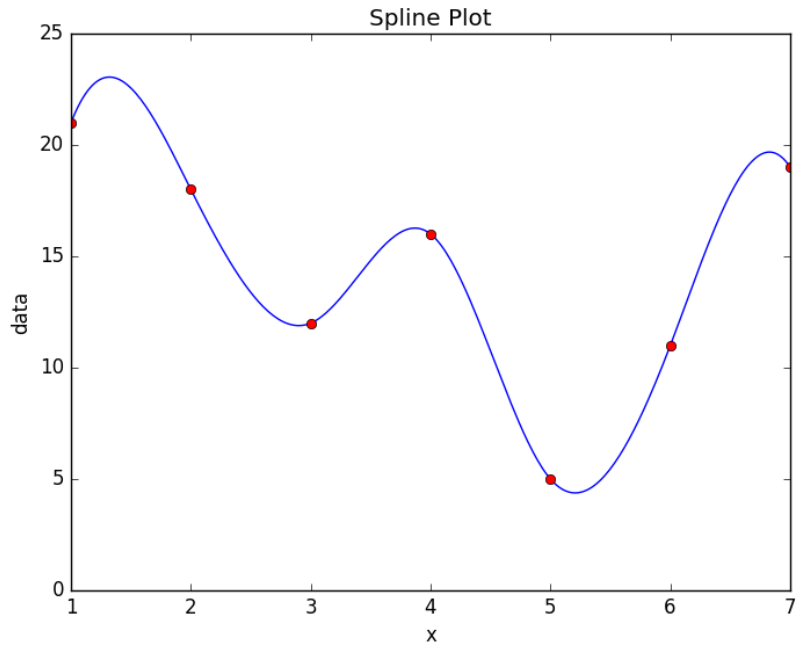
- Mathworks Documentation
- MATLAB Forums
- Plenty of Textbooks

# Plotting in Python software

- Customized Library
- Styling Options
- Platform Independence
- Better visibility
- Better real time output
- Troubleshooting

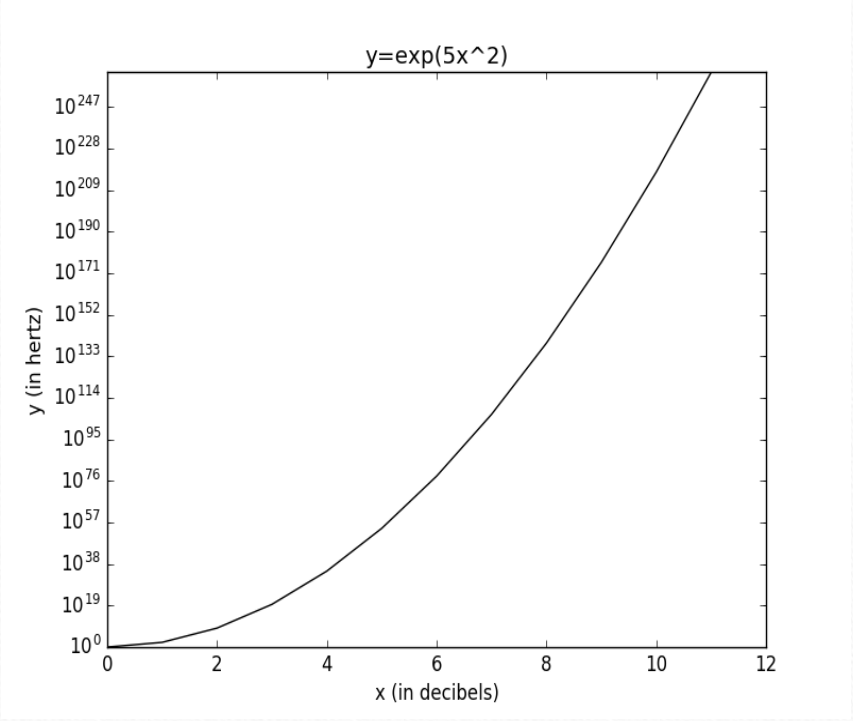
# Spline Plot

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.interpolate import spline
T = np.array([1, 2, 3, 4, 5, 6, 7])
power = np.array([21, 18, 12, 16, 5, 11,
19])
xnew = np.linspace(T.min(),T.max(),300)
power_smooth = spline(T,power,xnew)
plt.plot(xnew,power_smooth)
plt.plot(T,power,'ro')
plt.title("Spline Plot")
plt.xlabel("x")
plt.ylabel("data")
plt.show()
```



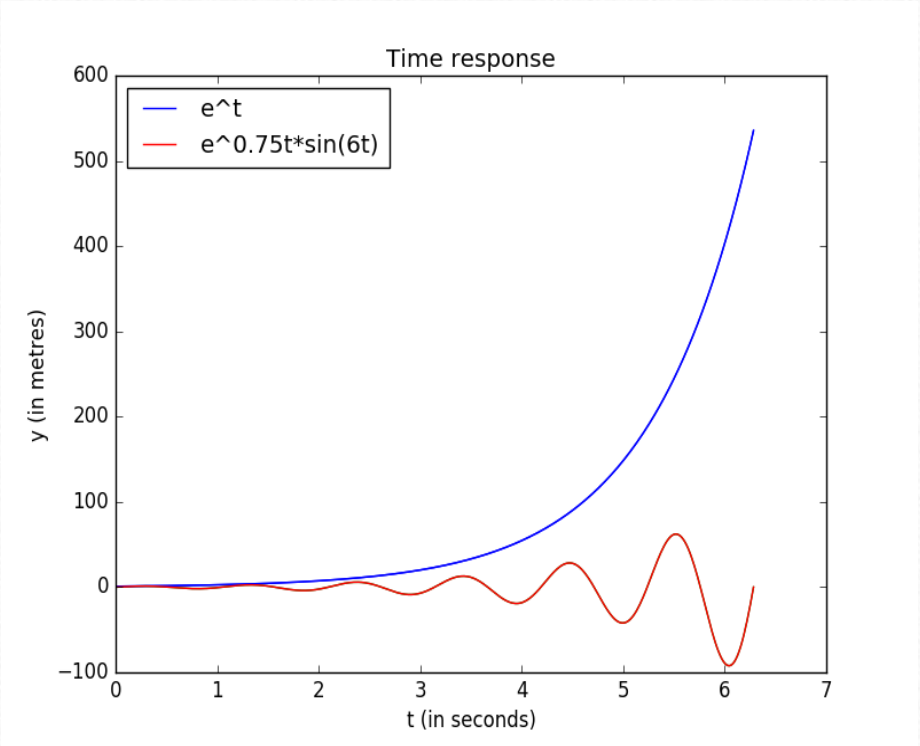
# Semi-log Plot

```
import matplotlib.pyplot as plt
import numpy as np
plt.gca().set_yscale('log')
x = np.arange(0, 50, 1)
y = np.exp(5*(x**2))
plt.plot(x, y, 'k-')
plt.title("y=exp(5x^2)")
plt.xlabel("x (in decibels)")
plt.ylabel("y (in hertz)")
plt.show()
```

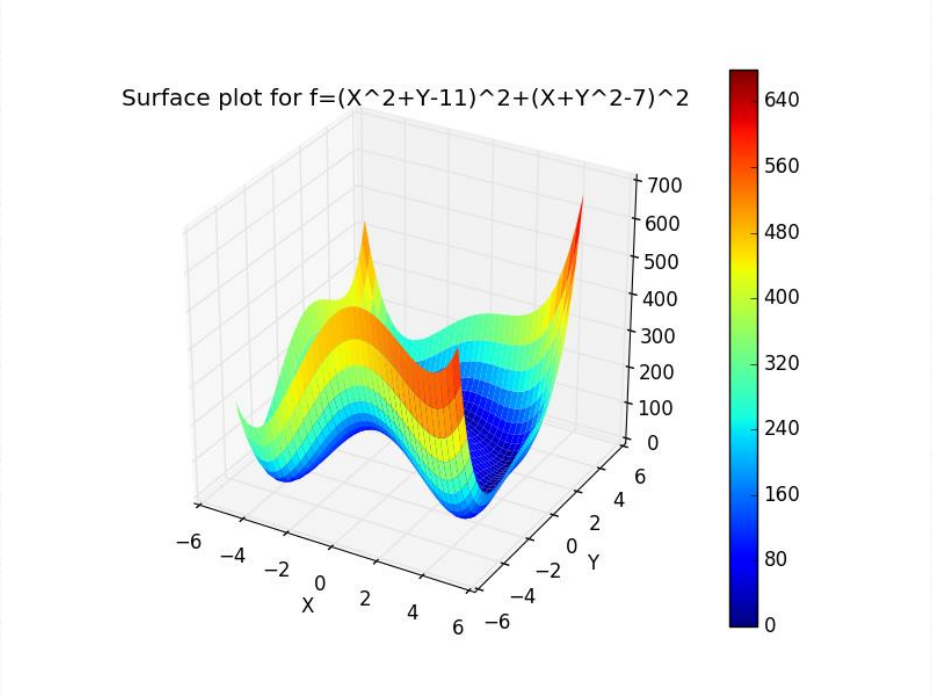


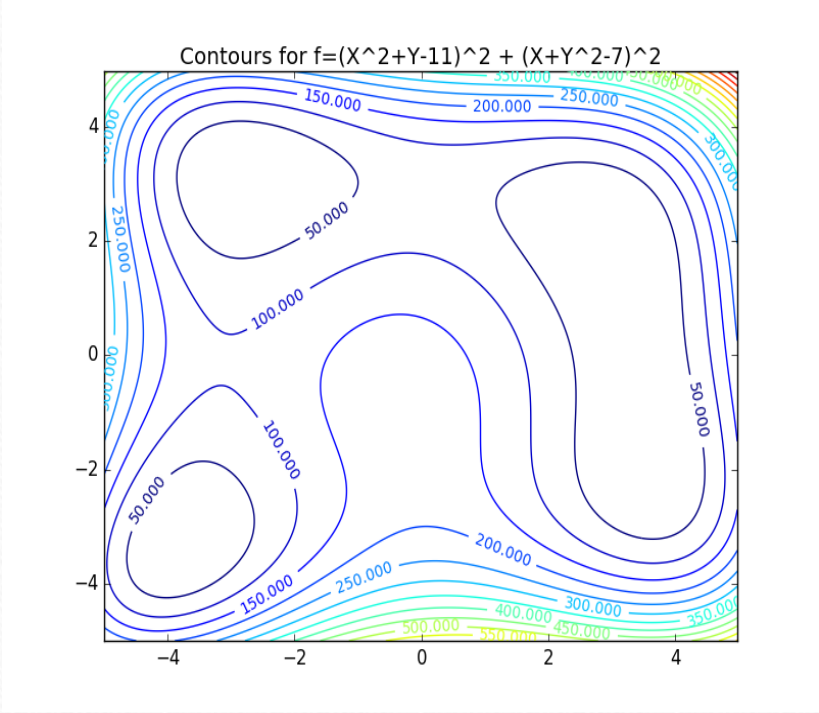
# Plotting of Curves

```
from numpy import *
import math
import matplotlib.pyplot as plt
import pylab
t = linspace(0,2*math.pi,400)
y1 = exp(t)
y2 = exp(0.75*t)*sin(6*t)
plt.plot(t, y1, t, y2)
plt.plot(t, y1, '-b', label='e^t')
plt.plot(t, y2, '-r', label='e^0.75t*sin(6t)')
pylab.legend(loc='upper left')
plt.title("Time response")
plt.xlabel("t (in seconds)")
plt.ylabel("y (in metres)")
plt.show()
```



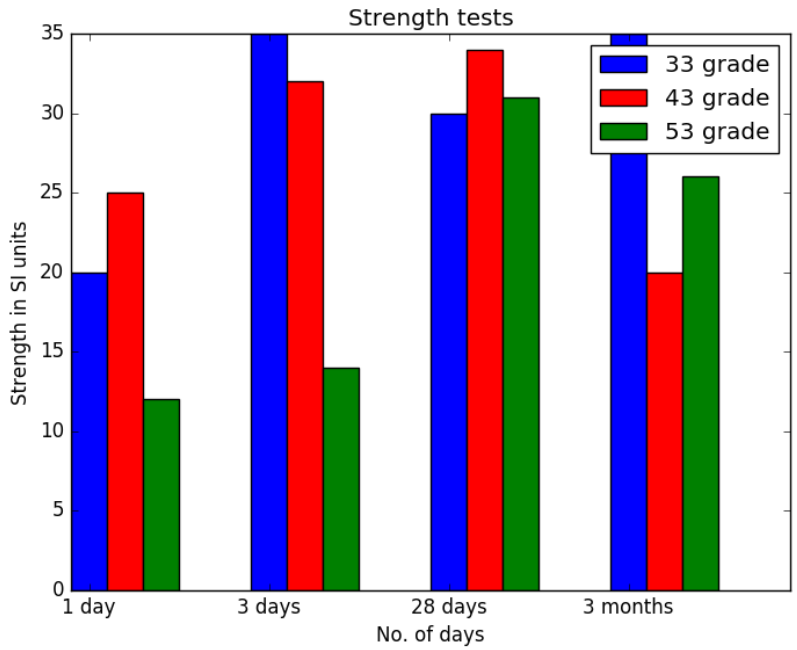
# Surface Plot and Contours



# Bar Diagrams

```
import numpy as np; import matplotlib.pyplot as plt

n_groups = 4; grade_33 = (20, 35, 30, 35)

grade_43 = (25, 32, 34, 20); grade_53 = (12, 14, 31, 26)

fig, ax = plt.subplots(); index = np.arange(n_groups)

bar_width = 0.20

rects1 = plt.bar(index, grade_33, bar_width,
color='b',label='33 grade')

rects2 = plt.bar(index + bar_width, grade_43, bar_width,
color='r', label='43 grade')

rects3 = plt.bar(index + 2*bar_width, grade_53,
bar_width, color='g', label='53 grade')

plt.xlabel('No. of days'); plt.ylabel('Strength in SI
units')

plt.title('Strength tests')

plt.xticks(index + bar_width / 2, ('1 day', '3 days', '28
days', '3 months'))

plt.legend(); plt.show()
```



# Some Special Plotting Software

- Mathematica
- MayaVi
- SigmaPlot
- ROBOANALYSER
- Ploticus

# Keynotes

- Choice of software
- Customization
- Clarity
- Explanatory nature