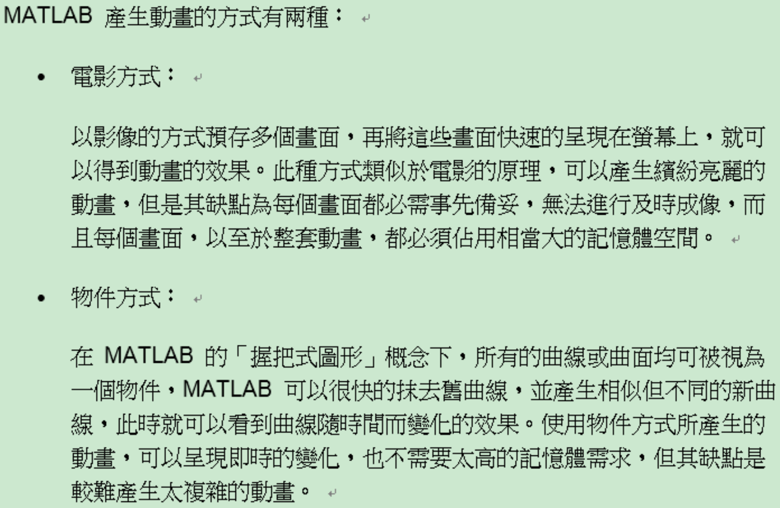
# Animation

Instructor: Pei-Chiang Shao (邵培強)

Department of Mathematics, National Central University
Jhongli District, Taoyuan City 32001, Taiwan (R.O.C.)

E-mail: shaopj823@gmail.com

# Matlab animation



MATLAB 產生動畫的方式有兩種： ↵

- 電影方式： ↵

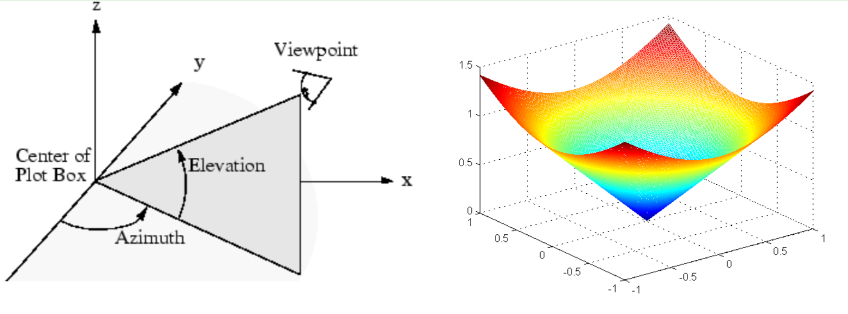
  以影像的方式預存多個畫面，再將這些畫面快速的呈現在螢幕上，就可以得到動畫的效果。此種方式類似於電影的原理，可以產生繽紛亮麗的動畫，但是其缺點為每個畫面都必需事先備妥，無法進行及時成像，而且每個畫面，以至於整套動畫，都必須佔用相當大的記憶體空間。 ↵

- 物件方式： ↵

  在 MATLAB 的「握把式圖形」概念下，所有的曲線或曲面均可被視為一個物件，MATLAB 可以很快的抹去舊曲線，並產生相似但不同的新曲線，此時就可以看到曲線隨時間而變化的效果。使用物件方式所產生的動畫，可以呈現即時的變化，也不需要太高的記憶體需求，但其缺點是較難產生太複雜的動畫。 ↵

# Movie-based animation

Produce an animation by changing the view of the graph of $f(x,y) = \sqrt{x^2+y^2}$ in cartesian coordinate.



# Movie-based animation

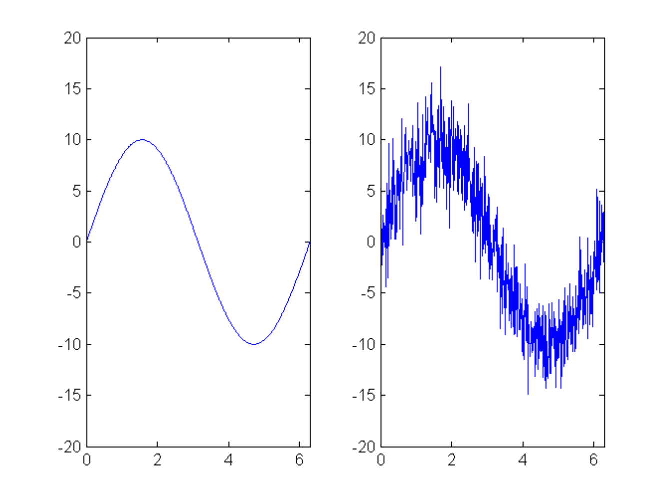
Produce an animation by changing the view of the graph of $f(x, y) = \sqrt{x^2 + y^2}$ in cartesian coordinate.

```matlab
% matlab code
x = -1 : 0.01 : 1;
y = -1 : 0.01 : 1;
[X,Y] = meshgrid(x,y);
Z = sqrt(X.^2 + Y.^2);
mesh(X,Y,Z)

n = 50;
for i = 1 : n
    view(i*360/n, 30);
    pause(0.1);
end
for i = 1 : n
    view(360, 30+i);
    pause(0.1);
end
```

# Movie-based animation

Produce an animation by changing the noise level of the Gaussian noise which contaminates the signal $y = 10\sin(x)$ on $[0, 2\pi]$.



# Movie-based animation

Produce an animation by changing the noise level of the Gaussian noise which contaminates the signal $y = 10\sin(x)$ on $[0, 2\pi]$.

```matlab
% matlab code
x = 0 : 0.01 : 2*pi;
y = 10*sin(x);

n = 50;
for i = 1 : n
    A = 5*rand;
    z = randn(1, length(y));
    plot(x, y+A*z);
    axis([0, 2*pi, -20, 20])
    title(['Noise level = ', num2str(A)])
    pause(0.1);
end
```

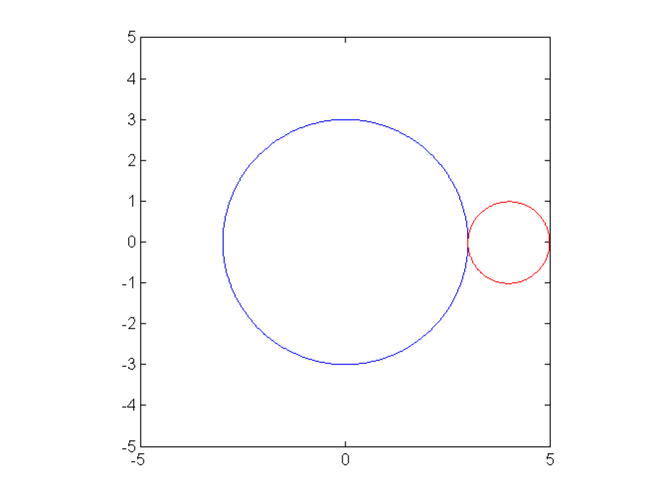
# Movie-based animation

Produce an animation by first diminishing the amplitude of $y = 100\sin(x)$ with factor 0.9 and then oscillating it with factor -1.1 (Use stem(x,y)).

```
% matlab code
A = 100; x = 0 : 0.1 : 2*pi; y = A*sin(x); n = 60;
for i = 1 : n
    if i <= n/2
        y = y*0.9;
        stem(x,y);
        title(['diminishing...', num2str(i)])
    else
        y = y*(-1.1);
        stem(x,y);
        title(['oscillating...', num2str(i)])
    end
    axis([0, 2*pi, -A, A]);
    pause(0.1);
end
```

# Movie-based animation

Produce an animation by rolling the small red circle along the big blue one.



# Movie-based animation

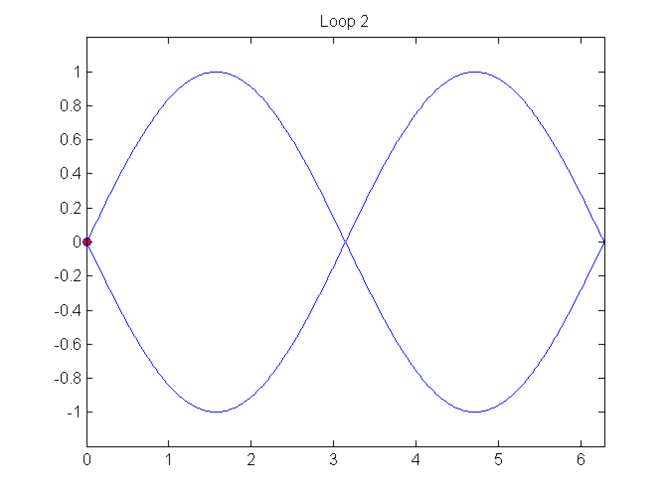
Produce an animation by rolling the small red circle along the big blue one.

```matlab
% matlab code
theta1 = 0 : 0.01 : 2*pi; theta2 = 0 : 0.05 : 6*pi;
r1 = 3; x1 = r1*cos(theta1); y1 = r1*sin(theta1);
r2 = 1; x2 = r2*cos(theta1); y2 = r2*sin(theta1);
R = r1 + 2*r2;

n = length(theta2) ;
for i = 1 : n
    center = (r1+r2)*[cos(theta2(i)), sin(theta2(i))];
    plot(x1, y1);
    hold on;
    plot(x2+center(1), y2+center(2), 'r');
    axis([-R, R, -R, R]); axis square;
    hold off;
    pause(0.01);
end
```

# Movie-based animation

Produce an animation by rolling the magenta ball along the $\infty$-shape curve.



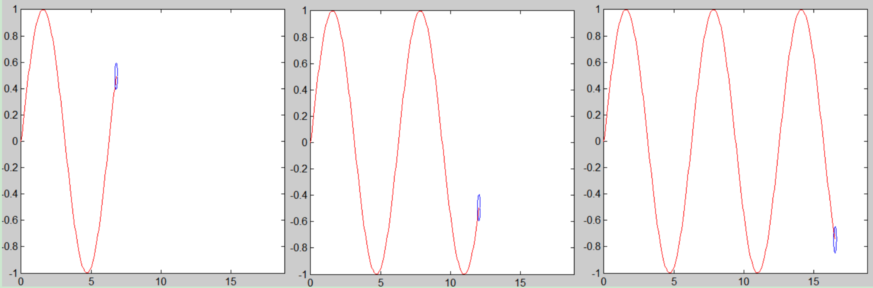
# Movie-based animation

Produce an animation by rolling the magenta ball along the ∞-shape curve.

```matlab
% matlab code
x = 0 : pi/100 : 2*pi; y = sin(x);
xx = [x, fliplr(x)]; yy = [y, fliplr(-y)];
n = 1;
while (n<=3)
    for i = 1 : length(xx)
        hold off
        plot(xx, yy);
        axis([0, 2*pi, -1.2, 1.2])
        hold on
        title(['Loop ', num2str(n)])
        plot(xx(i),yy(i),'o','MarkerFaceColor','r');
        pause(0.01)
    end
    n = n + 1;
end
```

# Quiz

Produce an animation by embedding a small disk into the front size of the wave $y = \sin(x)$ and moving them forward together.


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15