



MATLAB

Chapter 2

程式流程控制



```
for 變數 = 向量  
    運算式;  
end
```

```
>> x = zeros(1,6);  
>> for a = 1:6  
>>     x(a) = 3*a;  
>> end  
>> disp(x)
```

```
3 6 9 12 15 18
```

% 預先配置矩陣將使得迴圈中
變數矩陣執行速度大增

程式流程控制_2



```
while 條件式  
    運算式;  
end
```

```
>> x = zeros(1,6);           % 結果同for loop  
>> a = 1;  
>> while a <= 6  
>>     x(a) = a;  
>>     a = a+1;  
>> end  
disp(x)
```

if – else - end



if 條件式

 運算式;

else

 運算式;

end

% 同C++之用法

Convolution



- $\mathbf{C} = \text{conv}(\mathbf{A}, \mathbf{B})$
- $\text{length}(\mathbf{C}) = \text{length}(\mathbf{A}) + \text{length}(\mathbf{B}) - 1$
- 向量 \mathbf{C} 中的第 n 個元素

$$C[n] = \sum_{k=-\infty}^{\infty} A[k]B[n-k]$$

Filter



- $y = \text{filter}(b,a,x)$
- 用來處理 FIR 或 IIR 系統
- 其中 b, a, x, y 為向量，其關係為

$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k]$$

- 因為 IIR 系統的 impulse response 為無限多項，所以經過此 IIR 系統的輸出會為無限多項，所以在此 filter 這指令的輸出向量的長度取和輸入向量的長度一樣
- 因此，在 $a=1$ 時為 FIR 系統，如 $y1 = \text{filter}(b,1,x)$ ，其結果和 $y2 = \text{conv}(b,x)$ 相似，但其結果 $y1$ 向量的長度會和向量 x 的長度一樣

Ex 1

$$x[n] = \delta[n] + 2\delta[n-1];$$

$$h[n] = \delta[n] + \delta[n-1] + 2\delta[n-2];$$

則 $y[n] = x[n] * h[n] = ?$

Sol:

$$X = [1 \ 2];$$

$$h = [1 \ 1 \ 2];$$

$$y = \text{conv}(x, h)$$

y =

1 3 4 4 % length(y)=4

所以 $y[n] = \delta[n] + 3\delta[n-1] + 4\delta[n-2] + 4\delta[n-3]$

Ex2



```
X=[1 2];
```

```
h=[1 1 2];
```

```
y=filter(h,1,X)
```

```
y=
```

```
1 3
```

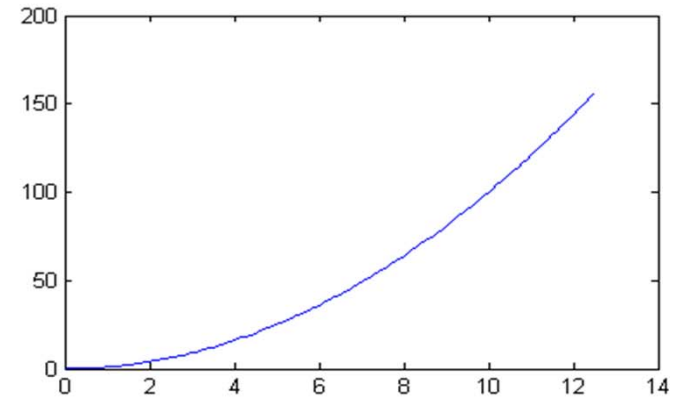
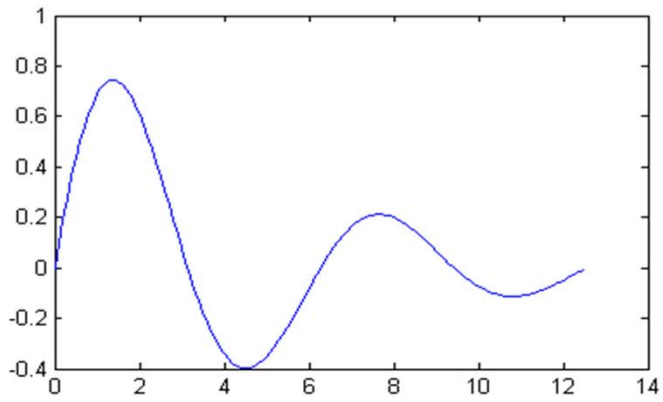
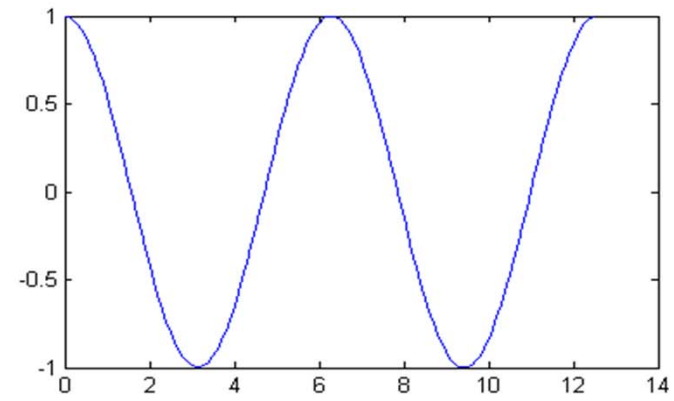
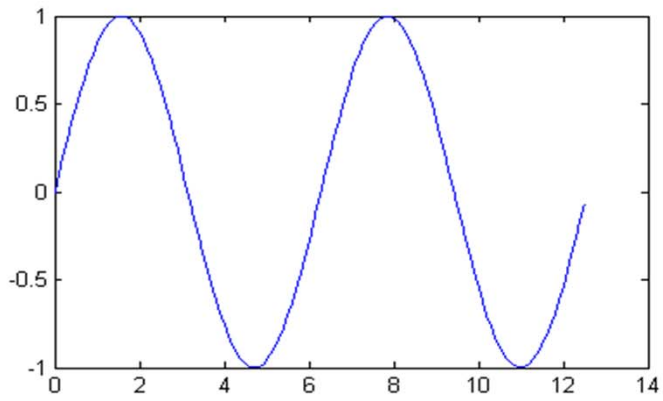
```
% length(y)=2
```


subplot



```
>> x = 0:0.1:4*pi;  
>> subplot(2,2,1); plot(x, sin(x));  
>> subplot(2,2,2); plot(x, cos(x));  
>> subplot(2,2,3); plot(x, sin(x) .* exp(-x/5));  
>> subplot(2,2,4); plot(x, x.^2);
```

subplot 結果



作業



- 當 $x[n] = \delta[n + 2] + 3\delta[n] - 4\delta[n - 2]$

$$h[n] = \delta[n + 1] + \delta[n - 1] + \delta[n - 2]$$

$y[n]$ 為 $x[n]$ 與 $h[n]$ 做 convolution 後的結果，試以手算的方式以及 MATLAB 的 `conv` 及 `filter` 兩指令分別求答案，並比較用手算出的是否相符