实验 3 离散序列的基本运算

1518 班　15352408　张镓伟

一、实验目的

(1)进一步了解离散时间序列时域的基本运算。
(2)通过实验进一步理解卷积定理，了解卷积的过程。
(3)了解 MATLAB 语言进行离散序列运算的常用函数，掌握离散序列运算程序的编写方法。

二、实验涉及的 MATLAB 子函数

1. find

功能：寻找非零元素的索引号。

调用格式：

find((n>＝min(n1))&(n<＝max(n1)))；在符合关系运算条件的范围内寻找非零元素的索引号。

2. fliplr

功能：对矩阵行元素进行左右翻转。

调用格式：

x1＝fliplr(x)；将 x 的行元素进行左右翻转，赋给变量 x1。

3.conv

功能：进行两个序列间的卷积运算。

调用格式：

y＝conv(x，h)；用于求取两个有限长序列 x 和 h 的卷积，y 的长度取 x、h 长度之和减 1。

例如，x(n)和 h(n)的长度分别为 M 和 N，则

y＝conv(x，h)

y 的长度为 N＋M－1。

使用注意事项：conv 默认两个信号的时间序列从 n＝0 开始，因此默认 y 对应的时间序号也从 n＝0 开始。

4.sum

功能：求各元素之和。

调用格式：

Z＝sum(x)；求各元素之和，常用于等宽数组求定积分。

5.hold

功能：控制当前图形是否刷新的双向切换开关。

调用格式：

holdon；使当前轴及图形保持而不被刷新，准备接受此后将绘制的新曲线。

holdoff；使当前轴及图形不再具备不被刷新的性质

6.pause

功能：暂停执行文件。

调用格式：

pause；暂停执行文件，等待用户按任意键继续。

pause(n)；在继续执行之前，暂停 n 秒

三、实验原理

离散序列的时域运算包括信号的相加、相乘，信号的时域变换包括信号的移位、反折、倒相及信号的尺度变换等。

在 MATLAB 中，离散序列的相加、相乘等运算是两个向量之间的运算，因此参加运算的两个序列向量必须具有相同的维数，否则应进行相应的处理。

1.序列移位

将一个离散信号序列进行移位，形成新的序列：

$$x_1(n)=x(n-m)$$

当 m>0 时，原序列 x(n)向右移 m 位，形成的新序列称为 x(n)的延时序列；当 m<0 时，原序列 x(n)向左移 m 位，形成的新序列称为 x(n)的超前序列。

2.序列相加

两个离散序列相加是指两个序列中相同序号 n(或同一时刻)的序列值逐项对应相加，构成一个新的序列：

$$x(n)=x_1(n)+x_2(n)$$

注意若相加的两个序列长度不同或者对应位置不同，则需要对较短序列和缺少位置的序列补 0，保持位置一一对应。

3.序列相乘

两个离散序列相乘是指两个序列中相同序号 n(或同一时刻)的序列值逐项对应相乘，构成一个新的序列：

$$x(n)=x_1(n)\times x_2(n)$$

同样存在着序列维数相同和不同两种情况，处理方法与序列相加相同。

4.序列反折

离散序列反折是指离散序列的两个向量以零时刻的取值为基准点，以纵轴为对称轴反折。在 MATLAB 中提供了 fliplr 函数，可以实现序列的反折。

5.序列倒相

离散序列倒相是求一个与原序列的向量值相反，对应的时间序号向量不变的新的序列。在 MATLAB 中在原序列前添负号“-”可得其序列倒相。

6.序列的尺度变换

对于给定的离散序列 x(n)，序列 x(mn)是 x(n)每隔 m 点取一点形成，相当于时间轴 n 压缩了 m 倍；反之，序列 x(n/m)是 x(n)作 m 倍的插值而形成的，相当于时间轴 n 扩展了 m 倍。

7.直接使用 conv 进行卷积运算

求解两个序列的卷积，很重要的问题在于卷积结果的时宽区间如何确定。在 MATLAB 中，卷积子函数 conv 默认两个信号的时间序列从 n＝0 开始，y 对应的时间序号也从 n＝0 开始。

8.复杂序列的卷积运算

由于 MATLAB 中卷积子函数 conv 默认两个信号的时间序列从 n＝0 开始，因此，如果信号不是从 0 开始，则编程时必须用两个数组确定一个信号，其中，一个数组是信号波形的幅度样值，另一个数组是其对应的时间向量。此时，程序的编写较为复杂，我们可以将其处理过程编写成一个可调用的通用子函数。下面是在 conv 基础上进一步编写的新的卷积子函数 convnew，是一个适用于信号从任意时间开始的通用程序。

```
function［y，ny］＝convnew(x，nx，h，nh)   %建立 convnew 子函数
%x 为一信号幅度样值向量，nx 为 x 对应的时间向量
%h 为另一信号或系统冲激函数的非零样值向量，nh 为 h 对应的时间向量
%y 为卷积积分的非零样值向量，ny 为其对应的时间向量
n1＝nx(1)＋nh(1)；%计算 y 的非零样值的起点位置
n2＝nx(length(x))＋nh(length(h))；%计算 y 的非零样值的宽度
ny＝［n1：n2］；%确定 y 的非零样值时间向量
y＝conv(x，h)；
```

用上述程序可以计算两个离散时间序列的卷积和，求解信号通过一个离散系统的响应。

9.卷积积分的动态过程演示

为了更深入地理解两个序列卷积的原理，下面提供一段演示卷积积分的动态过程的 MATLAB 程序。

动态地演示求解信号序列

f_1＝0.8n (0<n<20)

f_2＝u(n) (0<n<10)

卷积和的过程。

```
clf；                     %图形窗清屏
nf1＝0：20；              %建立 f1 的时间向量
f1＝0.8.^nf1；            %建立 f1 序列
lf1＝length(f1)；         %取 f1 时间向量的长度
nf2＝0：10；              %f2 的时间向量
lf2＝length(nf2)；        %取 f2 时间向量的长度
f2＝ones(1，lf2)；        %建立 f2 序列
lmax＝max(lf2，lf1)；     %求最长的序列
if lf2>lf1 nf2＝0；nf1＝lf2－lf1；%若 f2 比 f1 长，对 f1 补 nf1 个 0
elseif lf2<lf1 nf1＝0；nf2＝lf1－lf2；%若 f1 比 f2 长，对 f2 补 nf2 个 0
else nf2＝0；lf1＝0；    %若 f1 与 f2 同长，不补 0
end
lt＝lmax；        %取长者为补 0 长度基础
                  %先将 f2 补得与 f1 同长，再将两边补最大长度的 0
u＝［zeros(1，lt)，f2，zeros(1，nf2)，zeros(1，lt)］；
t1＝(－lt＋1：2*lt)；
%先将 f1 补得与 f2 同长，再将左边补 2 倍最大长度的 0
f1＝［zeros(1，2*lt)，f1，zeros(1，nf1)］；
hf1＝fliplr(f1)；         %将 f1 作左右反折
N＝length(hf1)；
```

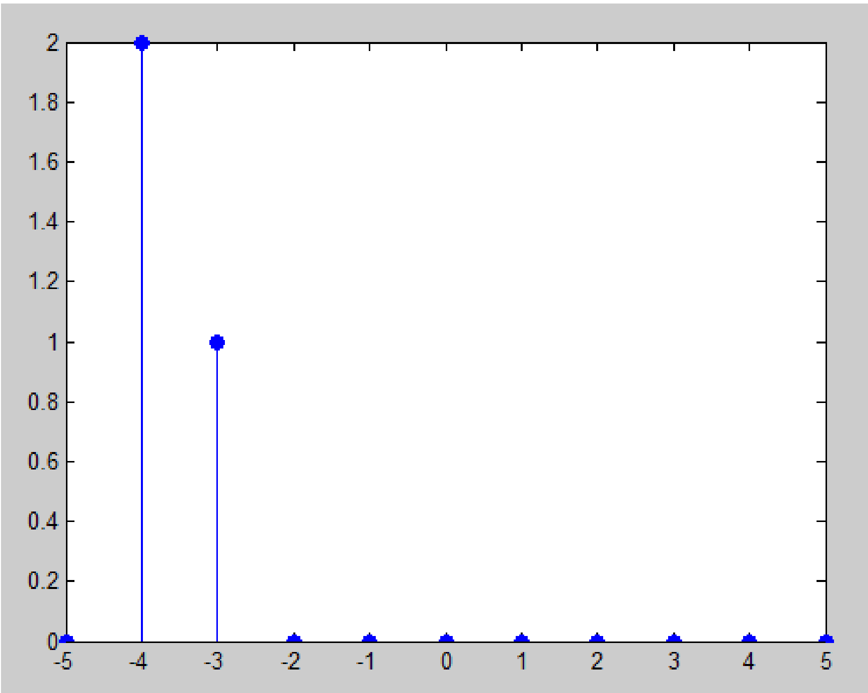
```
y＝zeros(1，3*lt)；         %将 y 存储单元初始化
fork＝0：2*lt               %动态演示绘图
p＝［zeros(1，k)，hf1(1：N－k)］；%使 hf1 向右循环移位
y1＝u.*p；[KG－4]         %使输入和翻转移位的脉冲过渡函数逐项相乘
yk＝sum(y1)；             %相加
y(k＋lt＋1)＝yk；          %将结果放入数组 y
subplot(4，1，1)；stem(t1，u)；
subplot(4，1，2)；stem(t1，p)；
subplot(4，1，3)；stem(t1，y1)；
subplot(4，1，4)；stem(k，yk)；%作图表示每一次卷积的结果
axis(［－20，50，0，5］)；holdon[KG－1]
%在图形窗上保留每一次运行的图形结果
pause(1)；  %停顿 1 秒钟
```

四、实验任务

(1)用 MATLAB 实现下列信号序列：

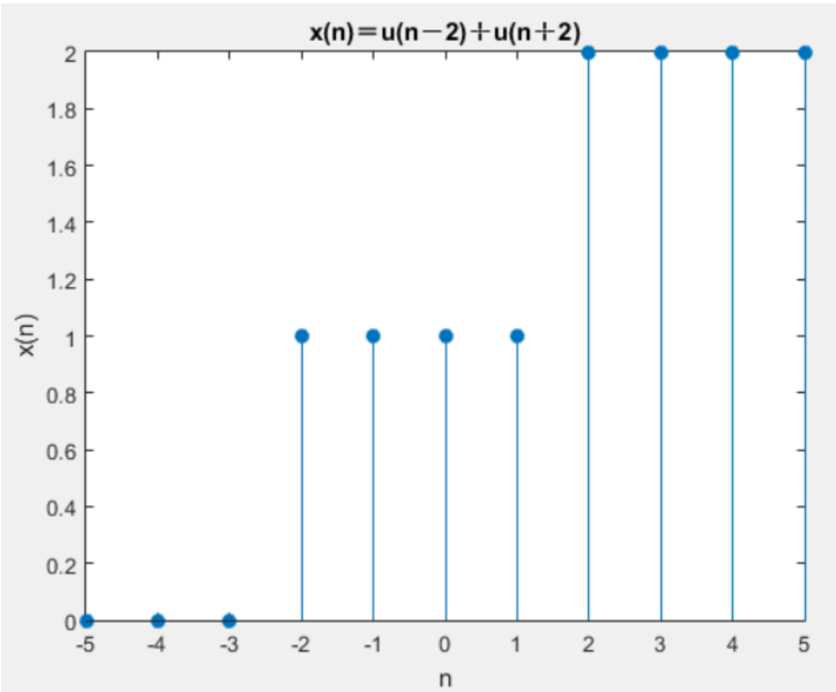
①x(n)＝δ(n＋3)＋2δ(n－4)，(－5<n<5)

```
clc
clear all;
n1=-5;n2=5;
n=n1:n2;        %确定序列区间
k1=-3;k2=-4;
x1=[n==k1];   %生成序列1
x2=2*[n==k2];%生成序列2
x=x1+x2;        %序列相加
stem(n,x,'fill');%绘制相加后的序列的图像
```



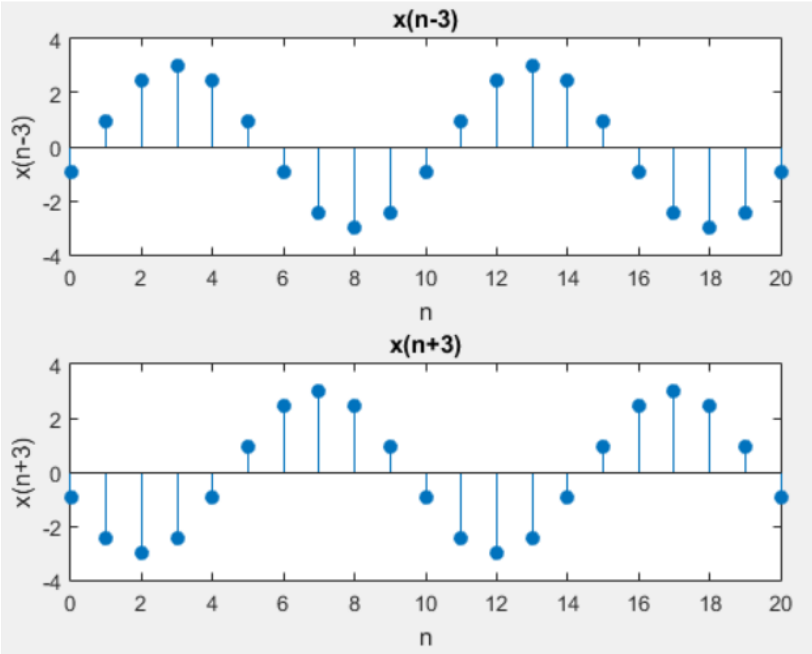
②x(n)＝u(n－2)＋u(n＋2)，(－5<n<5)

```
clc
clear all;
n1=-5;n2=5;
k1=2;k2=-2;
n=n1:n2;      %确定序列区间
x1=[n>=k1]; %生成序列u[n-2]
x2=[n>=k2]; %生成序列u[n+2]
x=x1+x2;      %两序列相加
stem(n,x,'fill');
title('x(n)=u(n-2)+u(n+2)');
xlabel('n');
ylabel('x(n)');
```



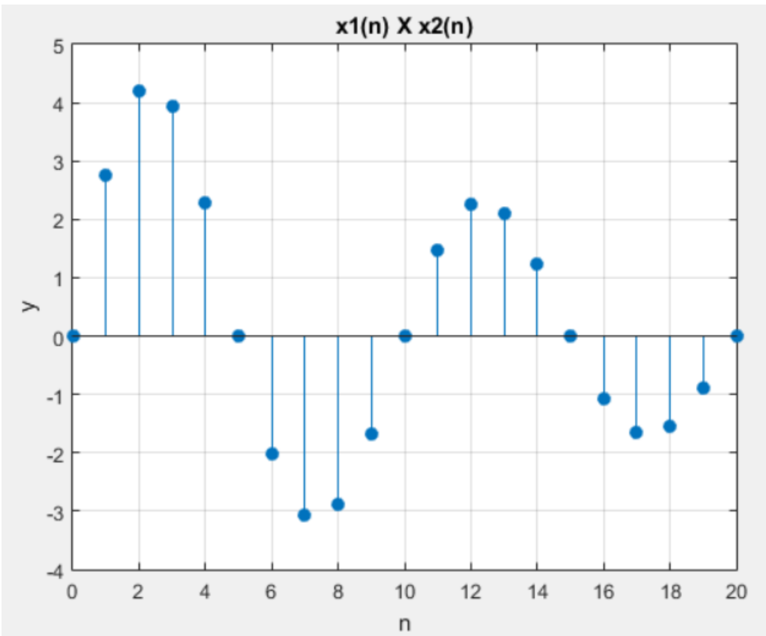
③已知 x(n)＝3cos(2πn/10)，试显示 x(n－3)和 x(n＋3)在 0<n<20 区间的波形。

```
clc
clear all;
n1=0;n2=20;
k1=2;k2=-2;
n=n1:n2;      %确定序列区间
x1=3*cos(2*pi*(n-3)/10);%x(n-3)
x2=3*cos(2*pi*(n+3)/10);%x(n+3)
subplot(2,1,1);
stem(n,x1,'fill');
title('x(n-3)');xlabel('n');ylabel('x(n-3)');
subplot(2,1,2);
stem(n,x2,'fill');
title('x(n+3)');xlabel('n');ylabel('x(n+3)');
```



④已知 $x1=e^{-n/16}$，$x_2(n)=5\sin(2\pi n/10)$，试显示 $x_1(n)\times x_2(n)$在 0<n<24 区间的波形。

```
clc
clear all;
n1=0;n2=24;
k1=2;k2=-2;
n=n1:n2;      %确定序列区间
x1=exp(-1*n/16);
x2=5*sin(2*pi*n/10);
x=x1.*x2 %x1和x2的乘积
stem(x,'fill');grid on;
title('x1(n) X x2(n)');xlabel('n');ylabel('y');
```

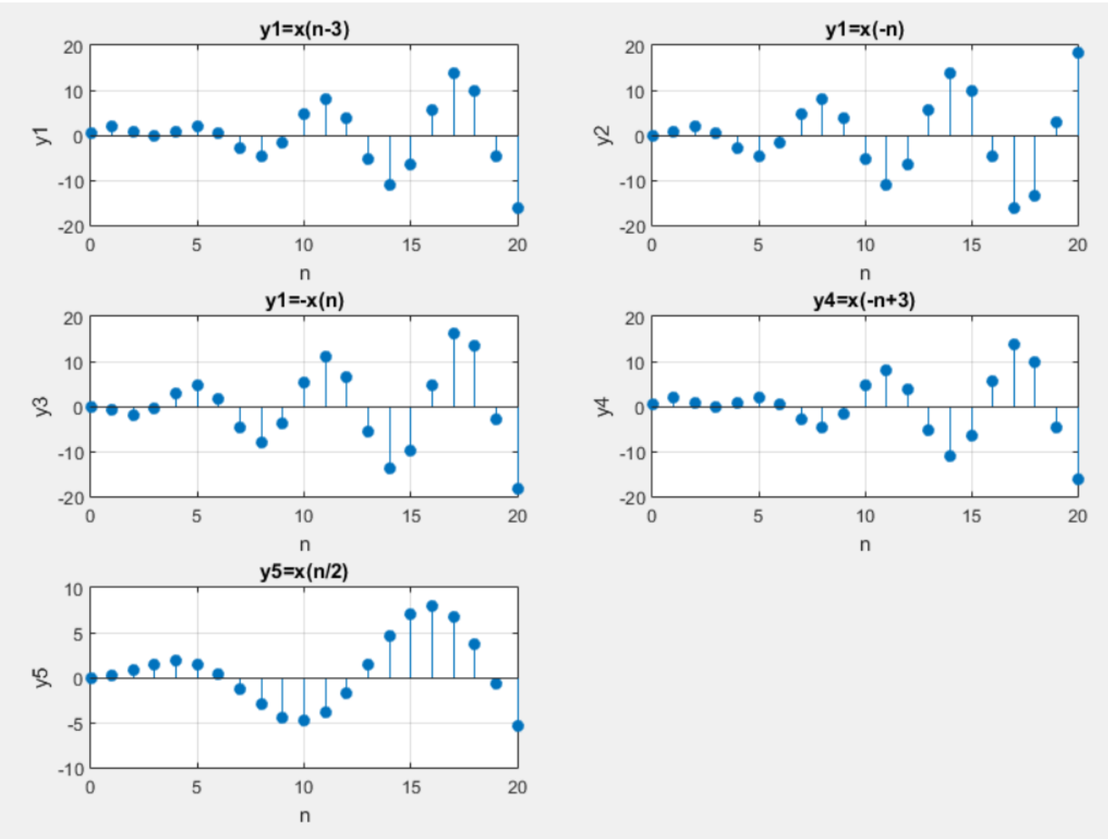


(2) 已知信号 x(n)＝n sin(n)，试显示在 0<n<20 区间的下列波形：

y1(n)＝x(n－3)，y2(n)＝x(－n)，y3(n)＝－x(n)，

y4(n)＝x(－n＋3)，y5(n)＝x(n/2)

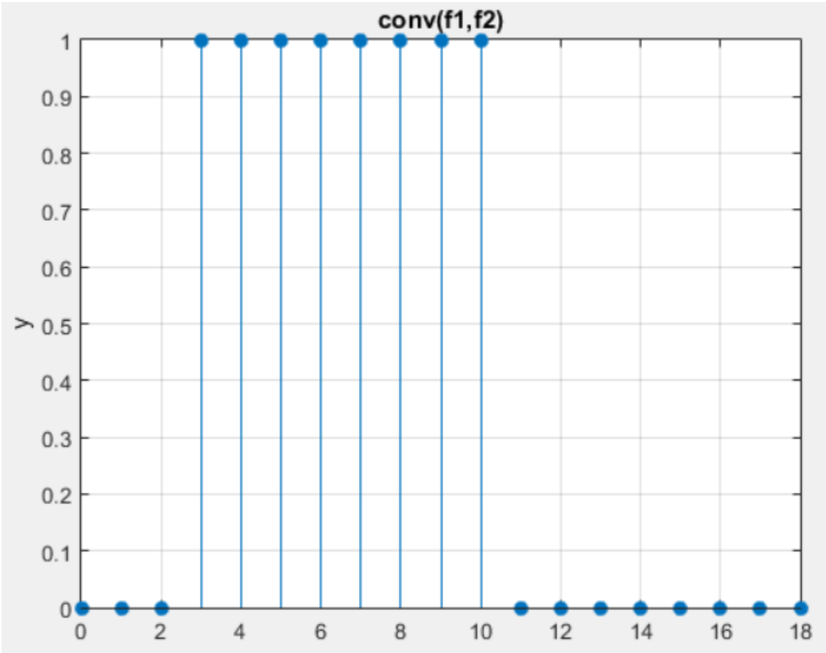
```
clc
clear all;
n1=0;n2=20;
n=n1:n2;      %确定序列区间
y1=(n-3).*sin(n-3);%x(n-3)
y2=(-n).*sin(-n);  %x(-n)
y3=-1*n.*sin(n);   %-x(n)
y4=(3-n).*sin(3-n);%x(-n+3)
y5=(n/2).*sin(n/2);%x(n/2)
subplot(3,2,1);
stem(n,y1,'fill');grid on;
title('y1=x(n-3)');xlabel('n');ylabel('y1');
subplot(3,2,2);
stem(n,y2,'fill');grid on;
title('y1=x(-n)');xlabel('n');ylabel('y2');
subplot(3,2,3);
stem(n,y3,'fill');grid on;
title('y1=-x(n)');xlabel('n');ylabel('y3');
subplot(3,2,4);
stem(n,y4,'fill');grid on;
title('y4=x(-n+3)');xlabel('n');ylabel('y4');
subplot(3,2,5);
stem(n,y5,'fill');grid on;
title('y5=x(n/2)');xlabel('n');ylabel('y5');
```



(3)编写 MATLAB 程序，描绘下列信号序列的卷积波形：

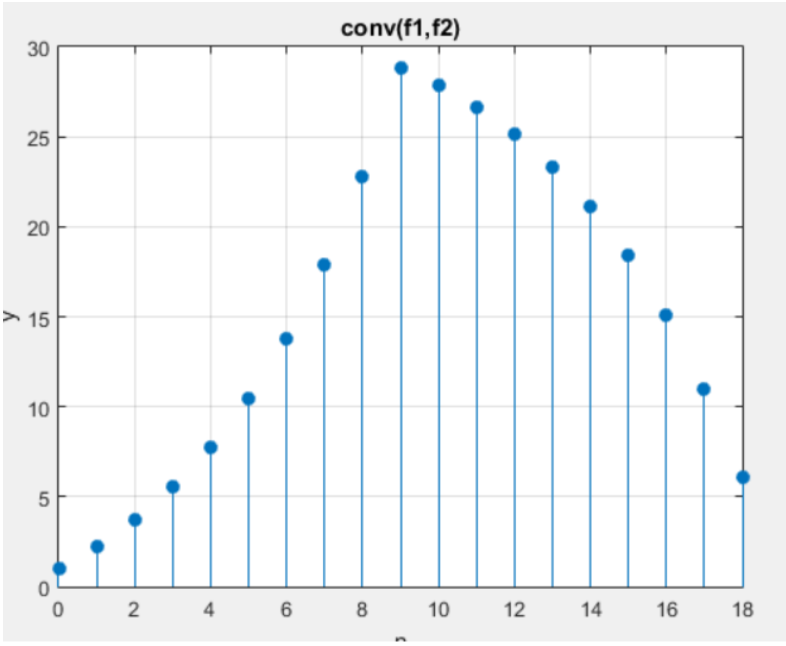
①$f1(n) = \delta(n-1)$，$f2(n) = u(n-2)$，$(0 \le n < 10)$

```
clc
clear all;
n1=0;n2=10-1;
n=n1:n2;      %确定原始序列区间
f1=zeros(1,length(n));
f2=zeros(1,length(n));
for i=n1+1-n1:n2+1-n1 %生成f1(n) = δ (n-1)  f2(n) =u(n-2)
    if i-1+n1==1
        f1(i)=1;
    end
    if i-1+n1>=2
        f2(i)=1;%u(n-2)
    end
end
y=conv(f1,f2)    %f1与f2的卷积,conv的结果默认从0开始
stem([n1+n1:n2+n2],y,'fill');grid on;
title('conv(f1,f2)');xlabel('n');ylabel('y');
```



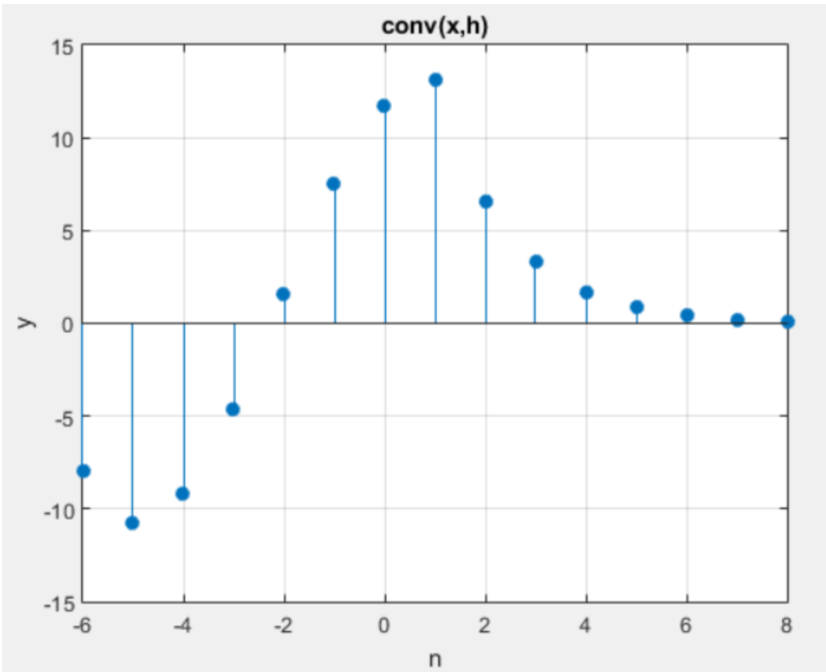
②$f1(n) = u(n)$，$f2(n) = e^{0.2n}u(n)$，$(0 \le n < 10)$

```
clc
clear all;
n1=0;n2=10-1;
n=n1:n2;      %确定原始序列区间
f1=zeros(1,length(n));
for i=n1+1-n1:n2+1-n1 %生成f1(n) =u(n)
    if i-1+n1>=0
        f1(i)=1;
    end
end
f2=exp(0.2*n).*f1       %生成f2(n) = (e^(0.2n))u(n)
y=conv(f1,f2)    %f1与f2的卷积,conv的结果默认从1开始
stem([n1+n1:n2+n2],y,'fill');grid on;
title('conv(f1,f2)');xlabel('n');ylabel('y');
```



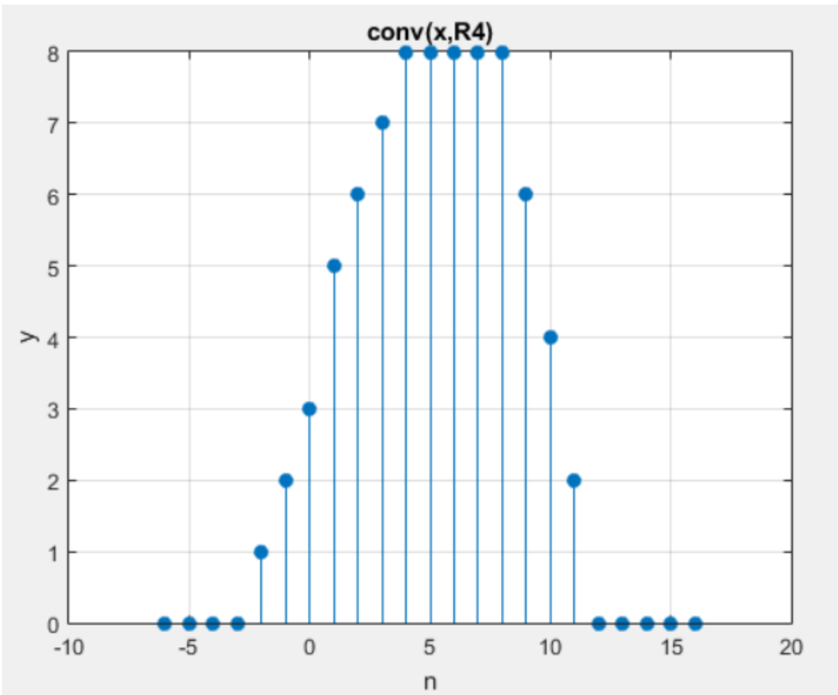
③x(n)＝sin(n/2)， h(n)＝$(0.5)^n$，（－3≤n≤4)

```
clc
clear all;
n1=-3;n2=4;
n=n1:n2;
x=sin(n/2);
h=0.5.^n;
y=conv(x,h);    %x与h的卷积
stem([n1+n1:n2+n2],y,'fill');grid on;
title('conv(x,h)');xlabel('n');ylabel('y');
```



④x(n)＝δ(n＋2)＋δ(n－1)，h(n)＝R_4(n)，(－3≤n≤8)

```
clc
clear all;
n1=-3;n2=8;
n=n1:n2;
x=zeros(1,length(n));
R4=zeros(1,length(n));
for i=n1+1-n1:n2+1-n1
    if i-1+n1>=1      %生成x(n) = δ(n+2) + δ(n-1)
        x(i)=x(i)+1;
    end
    if i-1+n1>=-2
        x(i)=x(i)+1;
    end
    if i-1+n1>=0    %生成R4(n)
        R4(i)=R4(i)+1;
    end
    if i-1+n1>=4
        R4(i)=R4(i)-1;
    end
end
x
R4
y=conv(x,R4);    %x与h的卷积
stem([n1+n1:n2+n2],y,'fill');grid on;
title('conv(x,R4)');xlabel('n');ylabel('y');
```



(4)已知信号

$$x(n)=\begin{cases}2n+5, & -4\le n\le -1\\ 6, & 0\le n\le 4\\ 0, & \text{其它}\end{cases}$$

①描绘 x(n)序列的波形；

②试用延迟的单位脉冲序列及其加权和表示 x(n)序列；

$x(n)= -4\delta(n+4)-1\delta(n+3)+\delta(n+2)+3\delta(n+1)+6\delta(n)$

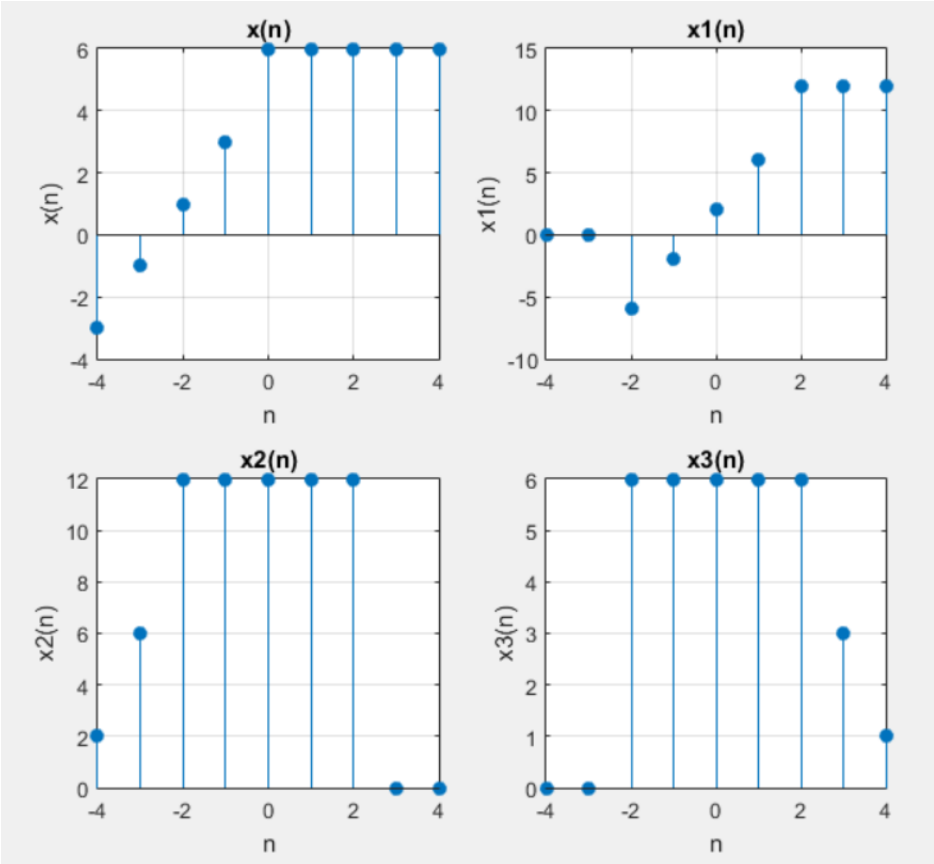
$+6\delta(n-1)+6\delta(n-2)+6\delta(n-3)+6\delta(n-4)$

③试描绘以下序列的波形：

$x_1(n)=2x(n-2)$，$x_2(n)=2x(n+2)$，$x_3(n)=x(2-n)$

```
clc
clear all;
n1=-4;n2=4;
n=n1:n2;
x=zeros(1,length(n));
R4=zeros(1,length(n));
for i=n1+1-n1:n2+1-n1 %生成x(n)
    if i-1+n1>=-4 &&  i-1+n1<=-1
        x(i)=2*(i-1+n1)+5;
    else
        x(i)=6;
    end
end
x1=2*x;x2=2*x;x3=x;
for i=n1+1-n1:n2+1-n1 %生成x1(n),x2(n),x3(n)
    if i-1+n1<n1+2 x1(i)=0;
    else x1(i)=2*x(i-2);
    end
    if i-1+n1>n2-2 x2(i)=0;
    else x2(i)=2*x(i+2);
    end
    if 2-(i-1+n1)<-4 || 2-(i-1+n1)>4 x3(i)=0;
    else x3(i)=x(2-(i-1+n1)+1-n1);
    end
end

subplot(2,2,1)
stem(n,x,'fill');grid on;
title('x(n)');xlabel('n');ylabel('x(n)');
subplot(2,2,2)
stem(n,x1,'fill');grid on;
title('x1(n)');xlabel('n');ylabel('x1(n)');
subplot(2,2,3)
stem(n,x2,'fill');grid on;
title('x2(n)');xlabel('n');ylabel('x2(n)');
subplot(2,2,4)
stem(n,x3,'fill');grid on;
title('x3(n)');xlabel('n');ylabel('x3(n)');
```



(5)思考题：

① 当进行离散序列的相加、相乘运算时，如果参加运算的两个序列向量维数不同，应进行怎样的处理？请简述：调用子函数 convnew 进行卷积积分处理前要做哪些准备，与使用 conv 有何不同。

答：当进行离散序列的相加、相乘运算时，如果参加运算的两个序列向量维数不同，则需对长度较短的序列补零，同时保持位置一一对应。若长度相同，但是位置不一一对应，则两个序列都应进行补 0 扩展直到位置一一对应。

调用子函数 convnew 进行卷积积分处理前要做的准备:

每个信号用两个数组确定，一个数组是信号波形的幅度样值，另一个数组是其对应的时间向量。将两个波形共 4 个数组作为参数转入 convnew 可得卷积后的波形的幅度样值和时间向量。

MATLAB 中卷积子函数 conv 默认两个信号的时间序列从 n＝0 开始，但 convnew 中两个信号的时间序列可以分别从任意时刻开始。

② 当进行离散序列的相乘运算时，例 3-5 题程序中有 x＝y1.*y2，请问此处进行的相乘运算是矩阵乘还是数组乘，为什么要这样使用? MATLAB 中提供的 conv 卷积子函数，使用中需满足什么条件？如果条件不满足，应如何处理？

答：例 3-5 中的 x＝y1.*y2 进行的是数组乘运算，因为此题目的是 y1 和 y2 对应位置的数相乘。

MATLAB 中提供的 conv 卷积子函数，使用是需确保两个信号的时间序列从 0 开始。若不符合这个条件，可以自己在 conv 的基础上编写一个卷积函数，如前面提到的 convnew。