### 信号与系统实验报告

名 称： 数字信号卷积和的实现及应用

学 院： 计算机科学与工程学院

专 业： 计算机科学与技术专业

学 号： 09021235

姓 名： 江紫弦

日期： 2023年 3 月 21 日

评分：

1. 实验目的
2. 掌握Matlab中相关函数的使用，程序代码编制与调试的流程。
3. 熟悉卷积和的运算规则及其意义，加深对离散时间信号分析的理解。
4. 实验任务
5. 完成实验内容全部题目，分析解决调试代码过程中出现的问题。
6. 认真完成本次实验小结，思考卷积和的应用。
7. 主要设备、软件平台
8. 硬件：计算机
9. 软件：Matlab
10. 实验内容
11. 输出杨辉三角。
    * + 1. 函数编写
        2. 控制台输出
        3. 循环语句、条件语句
        4. 程序运行、调试
12. 编程实现信号，间的卷积和运算函数，并绘制出下列信号卷积和波形。
    * + 1. ，
        2. ，
        3. ，
13. 设计游戏时，若对小怪使用一次技能的效果是“小怪会在接下来5秒内持续掉血，每秒掉血量分别为[5 4 3 2 1]”；如果间隔1秒连续发动3次技能，请绘制出每次攻击后小怪的累计掉血量情况。
14. 探究拓展
15. 给定一个如下所示的二维矩阵，实现其自身的卷积运算。
16. **实验结果  
    1、杨辉三角  
    （1）思路** 杨辉三角的特点：第n+1行的第i个元素等于第n行的第i-1个数和第i个数之和，即triangle(i,j) = triangle(i-1,j-1) + triangle(i-1,j)。我们知道杨辉三角的每一行的最左边和最右边元素均为1，则杨辉三角的第一行和第二行均为1，所以我们可以用循环逐行求出指定行数的杨辉三角。 **（2）代码**  
    % 定义杨辉三角的行数

n = 10;

% 初始化杨辉三角矩阵

triangle = zeros(n);

% 计算杨辉三角

for i = 1:n

triangle(i,1:i) = 1;%先将三角范围全初始化为1（也可以只初始化左边和后边两列为1）

for j = 2:i-1

triangle(i,j) = triangle(i-1,j-1) + triangle(i-1,j);

end

end

% 输出杨辉三角

disp('杨辉三角：');

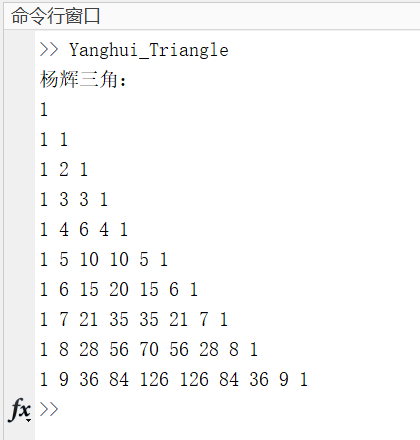
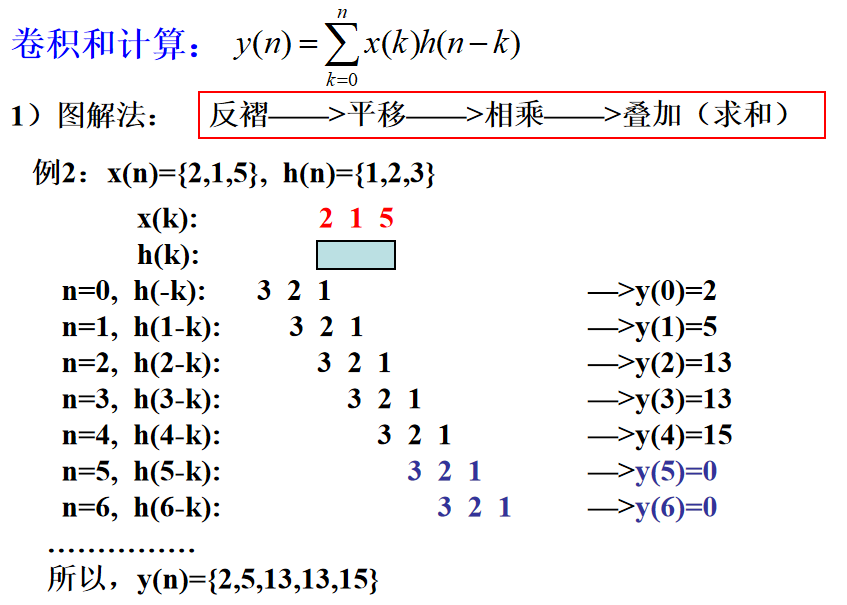
for i = 1:n

for j = 1:i

fprintf('%d ',triangle(i,j));

end

fprintf('\n');

end  
  
**(3)输出（打印一个十行的杨辉三角）**  
  
**2.卷积和运算  
（1）思路  
①公式法**卷积公式为;由于题目中要求对有限长离散信号进行卷积运算，在matlab中可对n,k进行双重循环，对数组进行访问，按公式累积求和，即可得到卷积结果。 **②图解法**图解法原理如下图；在matlab运用数组进行数据存储，易对h\_n数组进行反褶、平移，再将x\_n数组右移h\_len-1个，即可完成x(k)与h(-k)元素的下标对齐。之后利用循环，进行相乘叠加，每次循环后对h\_n数组右移一位，直至两个数组的下标完全错开，这之后对应n处的y\_n值恒等于0。由于在运算过程中反复用到右移操作，单独设计一个右移函数MoveRight。 **  
（2）代码  
1)右移函数**  
 function [x\_MoveRight,Newleft] = MoveRight(x,left,m)

%将数组x右移m位，前面补0(有效位数的下标+m)

%left是x最左边一个有效数字的下标

%Newleft是x\_MoveRight最左边一个有效数字的下标

x\_len=length(x);

x\_MoveRight(1,1+m:x\_len+m)=x(1,1:x\_len);

for i=1:m

x\_MoveRight(i)=0;

end

Newleft=left+m;

end

**2)公式法实现卷积运算**  
 function [y\_n] = convolution1(x\_n,h\_n)

%用公式法实现卷积

x\_len=length(x\_n);

h\_len=length(h\_n);

for n=1:x\_len+h\_len

y\_n(n)=0;

for k=max(1,n-h\_len):min(x\_len,n-1)

y\_n(n)=y\_n(n)+x\_n(k)\*h\_n(n-k);

end

end

y\_n(1)=[];

end

**3)图解法实现卷积运算**  
 function [y\_n] = convolution(x\_n,h\_n)

%编程实现信号h(n),x(n)间的卷积和运算函数y(n) 图解法

%取长度

x\_len=length(x\_n);

h\_len=length(h\_n);

%偏移量

offset=h\_len-1;

% 翻转h(n)

h\_n=fliplr(h\_n);

%将x\_n右移offset个

[x\_n,l]=MoveRight(x\_n,1,offset);

%每次循环h\_n右移一位,初始化h\_n有效数字的最左一位下标

left=1;

%用循环相乘后叠加

for i=1:x\_len+h\_len

if x\_len+offset<left

break;%之后的y\_n(i)==0

else

y\_n(i)=0;%初始化y\_n(i)的值

for j=1:min(h\_len+left-l,x\_len)%j为当前列数关于x\_n数组有效数字最左边一个值的下标的偏移量

y\_n(i)=y\_n(i)+x\_n(j+offset)\*h\_n(j+offset);%相乘后叠加

end

[h\_n,left]=MoveRight(h\_n,left,1);%每次循环后h\_n右移一位

end

end  
  
**4）分别调用公式法和图解法公式运算卷积函数完成题中要求**  
 clc

clear

%% 编程实现信号h(n),x(n)间的卷积和运算函数y(n)

% 用图解法实现卷积

x\_n=[1 2 3 4 5 6 7 8 9 10];

h\_n=[1 1];

y\_n1=convolution(x\_n,h\_n)

x\_n=[1 2 3 9 4 5 6 0 7 8];

h\_n=[-1 2 -1];

y\_n2=convolution(x\_n,h\_n)

x\_n=[1 2 3 4 5 6 7 8 9 10];

h\_n=[1 2 3 4 5 6 7 8 9 10];

y\_n3=convolution(x\_n,h\_n)

%% 用公式法实现卷积

x\_n=[1 2 3 4 5 6 7 8 9 10];

h\_n=[1 1];

y\_n1=convolution1(x\_n,h\_n)

x\_n=[1 2 3 9 4 5 6 0 7 8];

h\_n=[-1 2 -1];

y\_n2=convolution1(x\_n,h\_n)

x\_n=[1 2 3 4 5 6 7 8 9 10];

h\_n=[1 2 3 4 5 6 7 8 9 10];

y\_n3=convolution1(x\_n,h\_n)

%% 绘制

n=1:length(y\_n1)+1;

figure('name','第一小题运算结果');

stem(y\_n1,'filled');

n=1:length(y\_n2)+1;

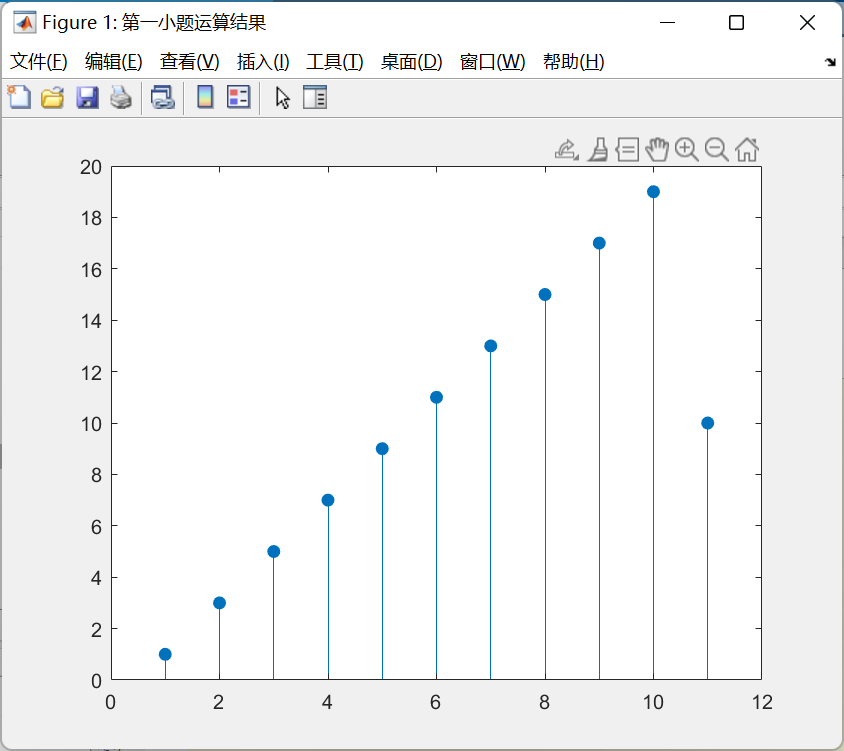
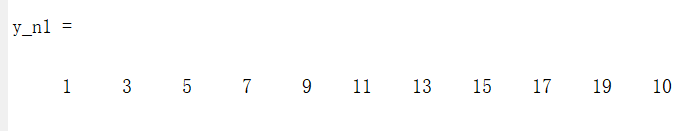
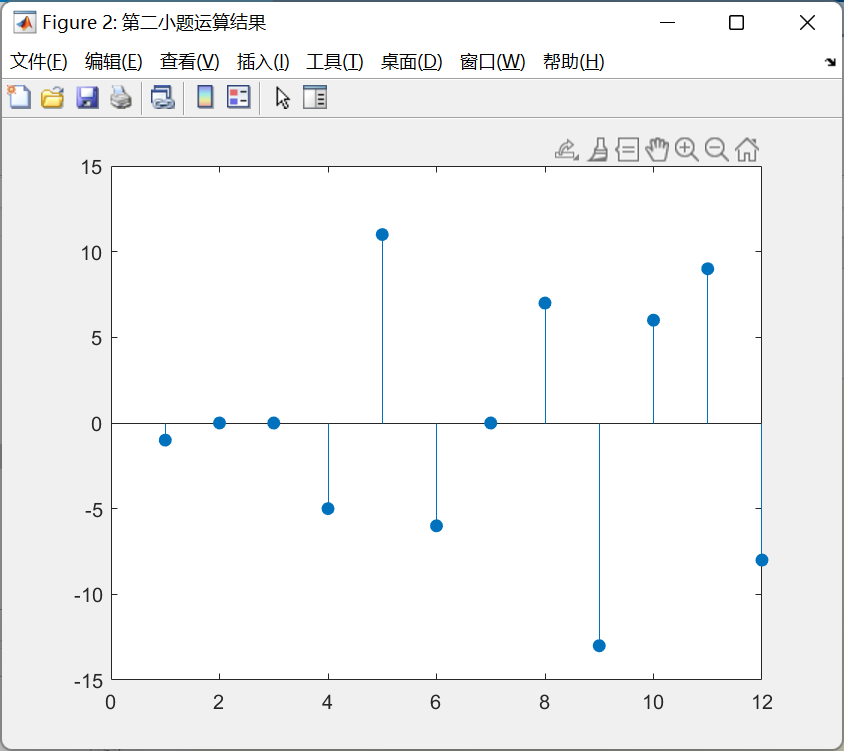
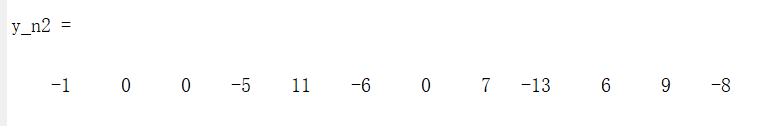
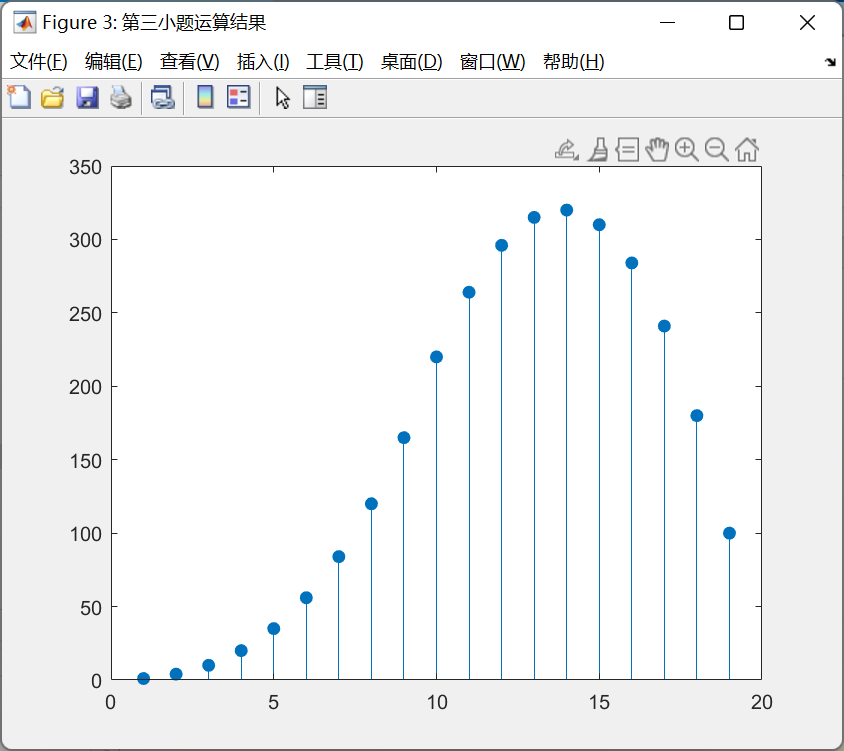
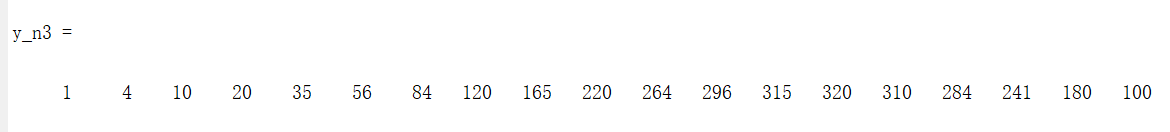
figure('name','第二小题运算结果');

stem(y\_n2,'filled');

n=1:length(y\_n3)+1;

figure('name','第三小题运算结果');

stem(y\_n3,'filled');  
  
**（3）输出  
①第一小题**

  
**②第二小题**  
  
**③第三小题**  
  
**3、打怪游戏  
（1）思路**  
将问题转化为x(n)=[1 1 1],h(n)=[5 4 3 2 1]两个信号的卷积运算  
**（2）代码**  
clc

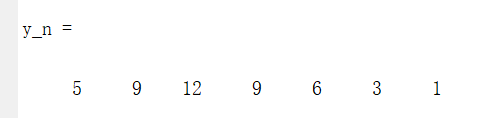
clear

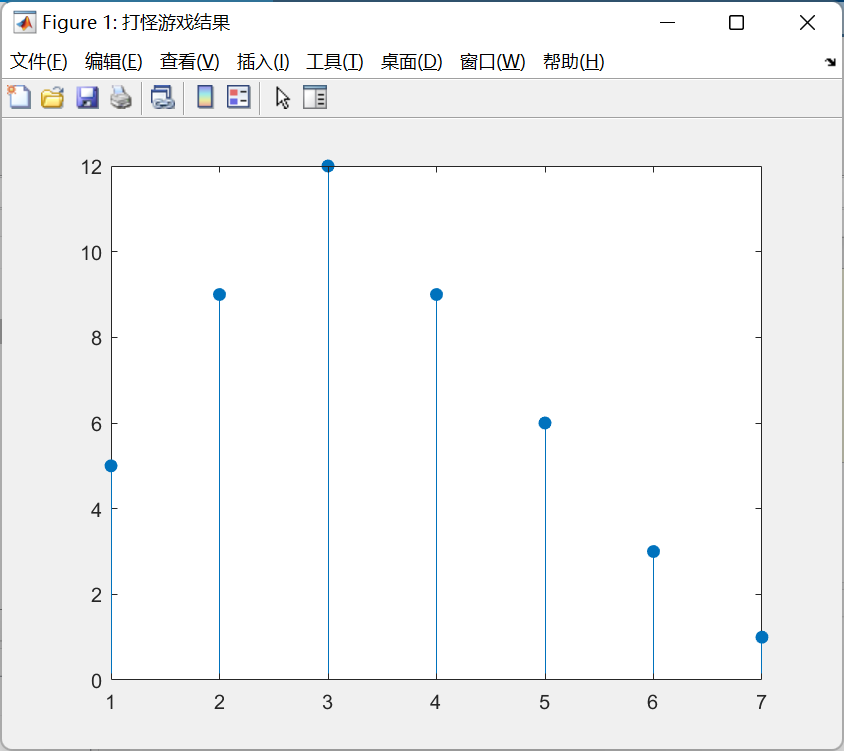
%将问题转化为[1 1 1]和[5 4 3 2 1]两个信号的卷积  
x\_n=[1 1 1];

h\_n=[5 4 3 2 1];

y\_n=convolution(x\_n,h\_n)

**(3)输出**





每次攻击后的掉血情况为[5 9 12 9 6 3 1]。

**4、矩阵卷积  
（1）思路**  
设待运算的两矩阵为A，B，其大小分别为m\*m,n\*n（假设n为奇数）。矩阵卷积运算步骤如下：  
a.让卷积核B翻转180度  
b.对矩阵A补零，即对矩阵上下左右都补（n-1）/2行（列）0，得到一个(m+n-1)\* (m+n-1)大小的矩阵Y；

c.滑动旋转后的B，使B的中心位于矩阵Y的每一个元素，并求卷积和（即将旋转后的B在Y上进行华东，然后对应位置相乘后相加的结果就是当前矩阵Y元素对应位置上的卷积结果）

d.在计算好的矩阵Y中心取m\*m大小的矩阵，即为矩阵A，B卷积后的结果

**（2）代码  
1）实现矩阵卷积的函数**  
function [Y] = MatrixConvolution(A,B)

%矩阵A和B卷积 前提：卷积核B的行列数为奇数

%先将B翻转180度

B=flip(B);

B=flip(B,2);

m=size(A);

m=m(2);

n=size(B);

n=n(2);

Y=zeros(m+n-1,m+n-1);

Y1=zeros(m+n-1,m+n-1);

Y((n-1)/2+1:(n-1)/2+m,(n-1)/2+1:(n-1)/2+m)=A;

for i=1:m

for j=1:m

re=0;

for k1=1:n

for k2=1:n

re=re+B(k1,k2)\*Y(i+k1-1,j+k2-1);

end

end

Y1((n-1)/2+i,(n-1)/2+j)=re;

end

end

Y=Y1((n-1)/2+1:(n-1)/2+m,(n-1)/2+1:(n-1)/2+m);

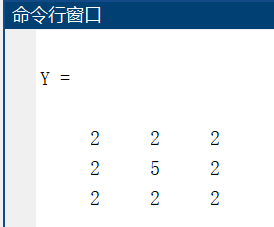
end  
  
**2)调用矩阵卷积函数完成题目要求运算**  
clc

clear

%% 实现矩阵二维卷积

A=[0 1 0;1 1 1;0 1 0];

Y=MatrixConvolution(A,A)

**（3）输出  
**

**七、实验心得**

在写卷积运算代码时，我的第一反应是实现更为复杂的图解法。图解法需对数组的下标严谨计算，同时需要防止数组越界问题，让我对课堂上介绍的图解法有了更深的印象和理解。其次，我还用公式法实现了卷积运算，公式法的实现较为简单，但同样需要对下标进行严谨计算。两种思路下实现对离散信号的卷积运算，让我巩固了上课所学知识。同时拓宽了视野，让我学习了二维矩阵卷积的计算方法并成功实现。