EN-74 ECE: Introduction to Image Processing
Tufts University
Fall 2007
Problem Set 5
Due October 11, 2007

READING: McAndrew Chapter 5

1. **Convolve by hand the signals** ***x[n] = {$\underline{1}$,2,3,4}*** **and** ***h[n] = {-1, $\underline{2}$, 1}***
From Lecture 5, slide 8 we have

$$y[n] = \sum_{k=-1}^{1} h[k]x[n-k] = h[-1]x[n+1] + h[0]x[n] + h[1]x[n-1]$$

From slide 14 of the same set of notes, we have that the first non-zero point in *y[n]* will occur at *n=-1* and that there will be a total of 6 points (since the length of *y* is one less than the sum of the lengths of *x* and *h*). Hence, the nonzero portion of *y[n]* will run from *n=-1* to *n=4*. Now since we take all values of *x[n]* equal to zero except those for *n=0,1,2,3* we just plug and chug in the above equation:

$$\begin{aligned}
y[-1] &= -1\times x[0] + 2\times x[-1] + 1\times x[-2] = -1\times 1 + 2\times 0 + 1\times 0 = -1 \\
y[0] &= -1\times x[1] + 2\times x[0] + 1\times x[-1] = -1\times 2 + 2\times 1 + 1\times 0 = 0 \\
y[1] &= -1\times x[2] + 2\times x[1] + 1\times x[0] = -1\times 3 + 2\times 2 + 1\times 1 = 2 \\
y[2] &= -1\times x[3] + 2\times x[2] + 1\times x[1] = -1\times 4 + 2\times 3 + 1\times 2 = 4 \\
y[3] &= -1\times x[4] + 2\times x[3] + 1\times x[2] = -1\times 0 + 2\times 4 + 1\times 3 = 11 \\
y[4] &= -1\times x[5] + 2\times x[4] + 1\times x[3] = -1\times 0 + 2\times 0 + 1\times 4 = 4
\end{aligned}$$

2. **Convolve by hand the signals**

$$x[n] = \begin{cases} n & n = 0, 1, 2, 3, \ldots \\ 0 & \text{else} \end{cases} \quad \text{and} \quad h[n] = \{\underline{-1}, 1\}$$

Explain how/why we can think of *h* as a discrete form of a differentiator?
Following the same reasoning as above we have

$$y[n] = \sum_{k=-1}^{1} h[k]x[n-k] = h[-1]x[n+1] + h[0]x[n] = x[n] - x[n-1]$$

Now, for the *x* given above we have that $x[n] - x[n-1] = n - (n-1) = 1$ so long as $n \geq 1$. Otherwise the output of the system is just 0. This is like a differentiator in that we have turned a signal that is a discrete line with "slope" equal to one into a constant, in this case one, which is just the slope. Similarly, if we had *x[n]* just a constant, say 3, then *x[n]-x[n-1] =0* which is just like what happens when we take the derivative of a constant, we get zero.

3. **Figure out how to use the `conv` function in Matlab to convolve two 1D signals. Note that `conv` does not keep track of the index set of the output signal. How can you do this if you know the index sets of `x` and `h`? Verify the correctness of the first problem**.

First, the Matlab code for the `conv` function is

```
>> x=[1 2 3 4]';
>> h=[-1 2 1]';
>> conv(x,h)
ans =
    -1
     0
     2
     4
    11
     4
```

which is the same as we got when doing this by hand. To keep track of the indices, *n*, you must do all the book keeping yourself. The following Matlab function would do the trick

```
function [y,yn] = fullconv(x,xn,h,hn)

% Function to perform discrete convolution of the signals
% x and h and to keep track of the indices over which
% the output is defined.

% Inputs
%   x = one of the two functions to convolve
%   xn = vector of indices over which x[n] is nonzero
%   h = second of the two functions to convolve
%   hn = vector of indices over which h[n] is nonzero

% Outputs
%   y = numerical values of x convolved with h
%   yn = vector of indices over which y[n] is nonzero

% The convolution is done using the conv function
y = conv(x,h);

% The leftmost index of y is equal to the leftmost index of
% x plus the leftmost index of y (slight generalization of
% Lecture 5, slide 14).
y_left = xn(1)+hn(1);

% The length of y is the length of x plus the length of h minus 1.
% Hence the index set starts at y_left and runs
% to y_left+length(x)+length(y)-2. Note the -2 instead of -1.  If
% we used -1, the we would end up with an index set which is one
% too long.  Try it.
yn = (y_left:(y_left+length(x)+length(h)-1-1))';
```