Homework 10
Due: Thursday, April 29, 2004, at 4pm

Reading OWN Chapter 10: 10.1, 10.2, 10.3, 10.5, 10.6.

Practice Problems *(Suggestions.)* OWN 10.1, 10.2, 10.3, 10.4

Problem 1 *(z-Transform)* 20 Points

(a) OWN 10.21 (a)

(b) OWN 10.21 (g)

(c) OWN 10.22 (b)

(d) OWN 10.22 (d)

Problem 2 *(Inverse z-Transform)* 30 Points

(a) (5 Pts) OWN 10.23, the first $X(z)$. No need to do the Taylor series method.

(b) (5 Pts) OWN 10.23, the second $X(z)$. No need to do the Taylor series method.

(c) (10 Pts) The signal $x[n]$ has $ROC(x[n]) = \{z | z \neq 0\}$ and z-Transform

$$X(z) \quad = \quad \sin(3z^{-1}). \tag{1}$$

Determine the signal $x[n]$.

(d) (10 Pts) OWN 10.26 (a), (b), (c).

Problem 3 *(Advanced properties of the z-Transform)* 20 Points

OWN 10.44 (a), (b), (c). In each case, also determine the region of convergence (in terms of the region of convergence R of the original signal $x[n]$).

Remark. These properties are at the foundations of wavelets.

Problem 4 *(Two-dimensional Fourier Transform)* 30 Points

In this homework problem, we explore some of the essential properties of the two-dimensional Fourier transform, defined simply as

$$X(\omega_1, \omega_2) \quad = \quad \int\int x(t_1, t_2) e^{-j\omega_1 t_1} e^{-j\omega_2 t_2} dt_1 dt_2 \tag{2}$$

(a) Consider the signal

$$x(t_1, t_2) \quad = \quad \cos(t_1/3)\cos(t_2/5). \tag{3}$$

Use matlab to get a sense for what the signal $x(t_1, t_2)$ looks like:
t1 = [-20:0.1:20]; % If your computer is slow, you may want to reduce the resolution.

```
t2 = [-20:0.1:20];
mesh(t1, t2, cos(t1/3)'*cos(t2/5));
```

Turn in a printout of this three-dimensional plot of $x(t_1, t_2)$.

Then, calculate (by hand) $X(\omega_1, \omega_2)$ and sketch (by hand) a two-dimensional figure with the two axes corresponding to ω_1 and ω_2 and appropriate symbols representing the value of $X(\omega_1, \omega_2)$. *Hint:* First determine $X(\omega_1, \omega_2)$. Looking at the formula, it should become clear at once how to sketch it... ;-)

(b) Repeat *(a)* for the signal

$$x(t_1, t_2) = \mathrm{sinc}(t_1/3)\mathrm{sinc}(t_2/5). \quad (4)$$

Again, sketch (by hand) $X(\omega_1, \omega_2)$. In the two-dimensional plane with axes ω_1 and ω_2, where are the low frequencies?

(c) As we have seen in class, one way to do image compression is to take the (two-dimensional) Fourier transform of the image, and then only retain the low frequencies.

To explore this idea, download the jpeg file brain.jpg from the class web page. In matlab, try the following:

```
I = imread('brain.jpg');
colormap('gray');
image(I);
```

Write the following m-file, called compress.m:

```
function Ic = compress(I, k);
if sum(size(I)==[256 256]) == 2,
if k > 128, k = 128; end;
if k < 1, k = 1; end;
Ifourier = fftshift(fft2(I));
Ifouriercompressed = zeros(size(I));
Ifouriercompressed(128-k+2:128+k, 128-k+2:128+k) = Ifourier(128-k+2:128+k, 128-k+2:128+k);
Ic = ifft2(ifftshift(Ifouriercompressed));
Ic = real(Ic);
else
Ic = 0; 'Code works only for image size 256 by 256 pixels!'
end;
```

Try the program for different values of k, and understand what the program is doing. Turn in the following:

- Plots for three representative values of k (ie high, medium, and low compression). Describe (in words) what the compression algorithm does to the image. *Hint:* It's pretty cool to check out, one after the other, $k = 1, 2, 3, 4, 5,$ etc....
- In the two-dimensional Fourier plane (corresponding to the two-dimensional Fourier transform of the image), explain what the program does in order to compress the image. Why will the compressed image yield a smaller file?

(d) (Extra Credit 5 Points) Extend the above matlab code to the case where the image size is not 256 by 256 pixels.