MRI Practical – MRI/Matlab Focus
Non-Ionising Functional & Tissue Imaging

Dr. Neal Bangerter, January 2021

Part 1: Visualizing and Manipulating 2D Signals as Images

This exercise aims to give you some familiarity with visualizing and manipulating 2D signals as images. I assume a basic knowledge of Matlab. If you need additional help with Matlab, I have posted a beginning tutorial on Blackboard.

For these problems, I assume some of the background on simple 2D signals I assume some background from the supplementary material from the January 19th lecture. If you haven't had time to review some of that material, you should review at least the supplement on the 2D Fourier transform and its properties now.

First, let's learn how to create and display some basic signals. Start Matlab and enter the following code:

```
[x, y] = meshgrid(-4:0.05:4);     ← type “help meshgrid” if you haven’t seen this before
s1 = exp(-pi*(x.^2 + y.^2));
imshow(s1, []); ← the [] scales the image so the largest value is white and the smallest is black
```

(a) The 2D signal s1 belongs to an important class of signals. What is it?

Now let's create and display a 2D sinc signal:

```
s2 = sin(-pi*x).*sin(-pi*y)./(pi^2*x.*y);
imshow(s2, []);
```

(b) Does this image look as you would expect? What is wrong with it? What is causing the problem?

Try the following:

```
[x, y] = meshgrid(-4:0.05:4 + 10*eps);    ← type “help eps” (a useful thing to know)
s2 = sin(-pi*x).*sin(-pi*y)./(pi^2*x.*y);
imshow(s2, []);
```

(c) Did that fix the problem you identified in (b)? Why?

(d) BONUS (for those interested in the limitations of finite precision arithmetic): Why did I use `10*eps` instead of `eps`?

Now let's take a look at the exponential signal:

```
kx_0 = 0.25;
ky_0 = 0.5;
s3 = exp(j*2*pi*(kx_0*x + ky_0*y));
figure;
subplot(1,2,1);         ← A useful thing to know for putting multiple graphs in a single figure
```

```
imshow(real(s3), []);      ← s3 is complex, so let's look at both the real part…
subplot(1,2,2);
imshow(imag(s3), []);      ← …and the imaginary part.
```

(e) Try several different values for `kx_0` and `ky_0`. What do these parameters represent?

How about creating rect(x,y)? Try the following code:

```
s4 = double(abs(x)<0.5 & abs(y)<0.5);
figure;
imshow(s4,[]);
```

Remember that `x` and `y` are 2D matrices returned by the `meshgrid` function. The expression `abs(x)<0.5` returns a 2D matrix whose values are either 1 or 0 depending on whether the corresponding element in `x` has an absolute value less than 0.5. The `&` performs a logical AND operation. Puzzle through this expression and figure out why it works if it isn't clear. Why do I cast it to a double afterwards?

NOTE: For next week you will use the 2D signals s1, s3, and s4 that you created in this problem. It might be useful to figure out how to save your variable space in Matlab so it can be loaded at a future time.