EE225E/BIOE265 Spring 2016 Miki Lustig
Principles of MRI

Assignment 5

Due Friday Feb 26th, 2016, Self Grading Due Monday Feb 29th, 2016

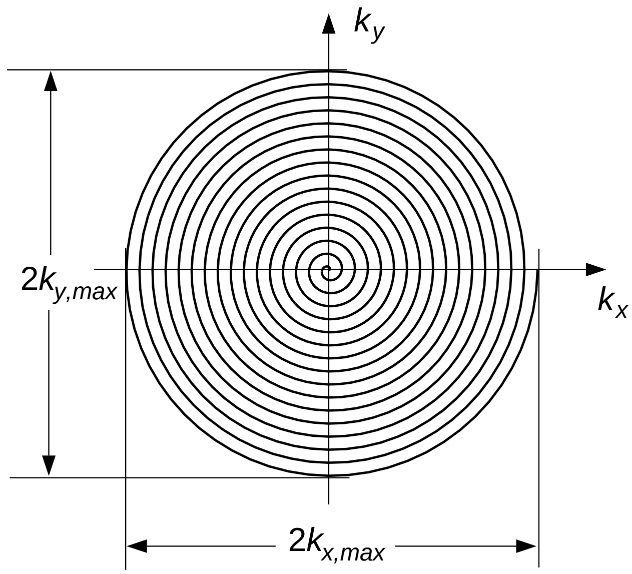
1. Finish reading Nishimura Ch. 5 and start reading Ch.6.

2. For the 16 turn spiral trajectory, plotted below, what is the

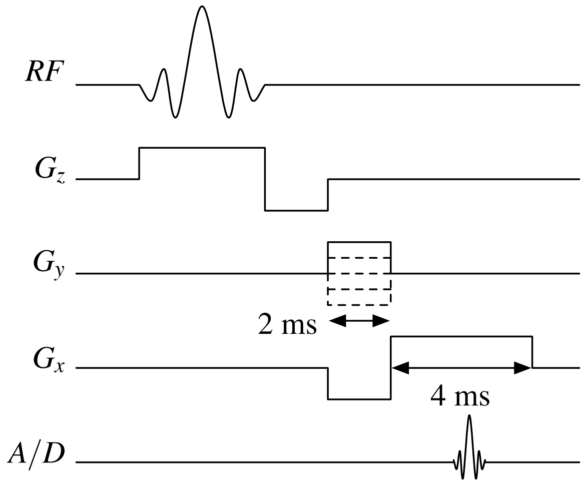
   a) Spatial resolution, and

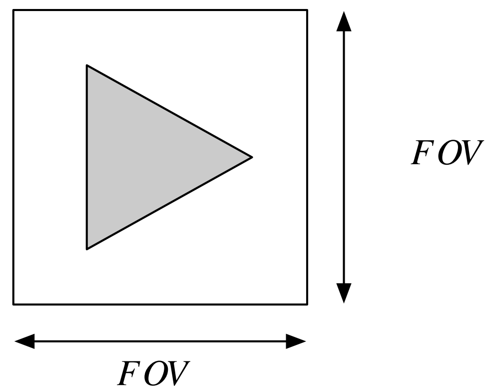
   b) FOV

   given that $k_{x,max} = k_{y,max} = 2.5$ cycles/cm. Assume that the sampling rate along the spiral trajectory is not limiting.



3. Consider the 2DFT pulse sequence, shown below on the left, with the following timing





The amplitude of the readout gradient is 0.94 G/cm, as is the maximum of the phase encode gradient. Also, 256 samples are acquired during the readout, and 256 phase encode steps are used. The initial magnetization is fully relaxed for each acquisition. The RF pulse is a $+90°$ rotation about $+x$.

The object being imaged is shown above on the right. It is a triangle that is larger than half the FOV in each dimension. The reconstructed image is the magnitude of the inverse FT of the sampled data.

(a) What are the resolution and FOV of the pulse sequence?

(b) Sketch the image that would be produced if we just used the even numbered phase encodes.

(c) What does the image look like if we doubled the x gradient, and used the original phase encode gradient?

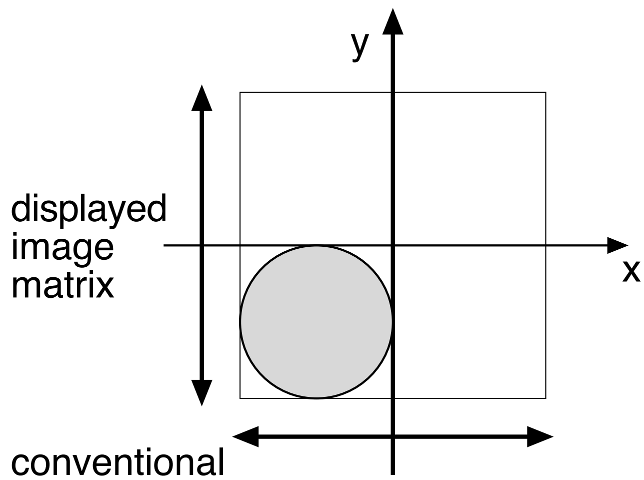
(d) What does the image look like if we doubled the maximum y gradient, and used the original readout gradient?

(e) Assume that the sign of the RF is alternated every other phase encode, so that it produces a $+90°$ rotation about the $+x$ axis on the even phase encodes, and a $-90°$ rotation on the odd phase encodes. We reconstruct as usual with an inverse FT. What does the image look like?

(f) Now consider the case where we are using the original acquisition gradients, but are imaging sodium, which has a $\gamma/2\pi$ of 1.126 kHz/G. What is the resolution and FOV? What does the image now look like?

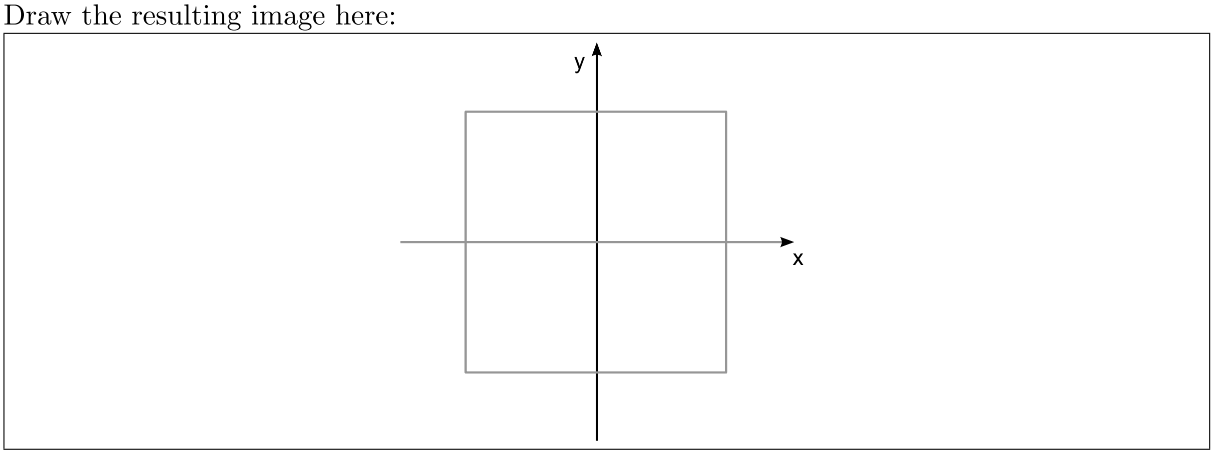
4. Artifacts in 2DFT (From Midterm I 2012)

Consider a conventional 2DFT sequence that produces the image shown below (within the square).



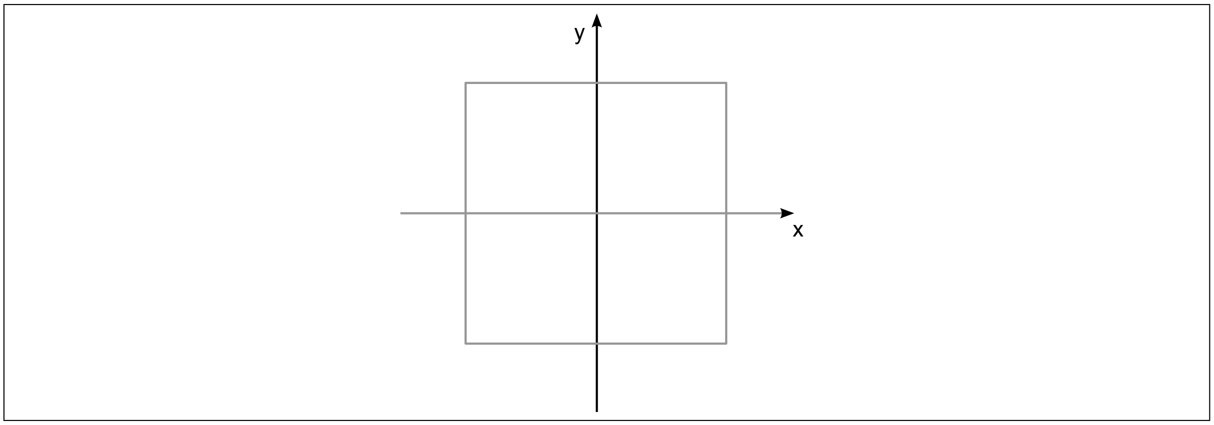
a) If the scan is repeated but with all gradient amplitudes scaled by a factor of **1.25** (this is the only change), sketch the resultant image. Note that the dumb (inflexible) reconstruction computer blindly takes the inverse FFT of the raw data and displays the image matrix.

Draw the resulting image here:



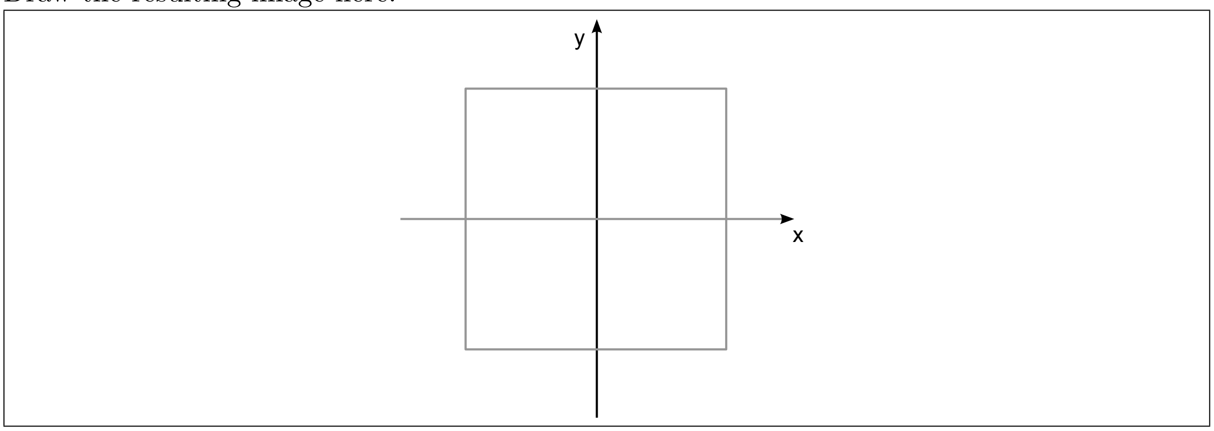
b) Now, the scan is repeated with the phase encode gradients turned off (this is the only change), sketch the resultant image. Again assume a dumb reconstruction computer.

Draw the resulting image here:

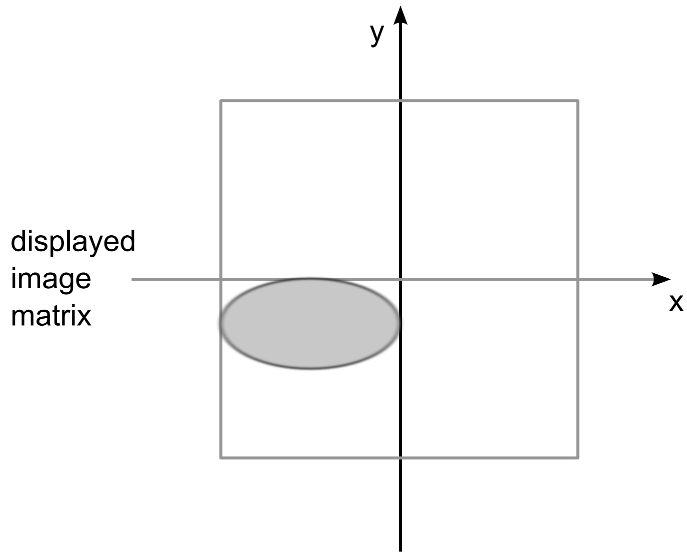


c) Now, the scan is repeated, but the imaging plane is rotated counterclockwise by 45°, sketch the resultant image. Again assume a dumb reconstruction computer. Note, that only the imaging plane is rotated, not the sample!.

Draw the resulting image here:



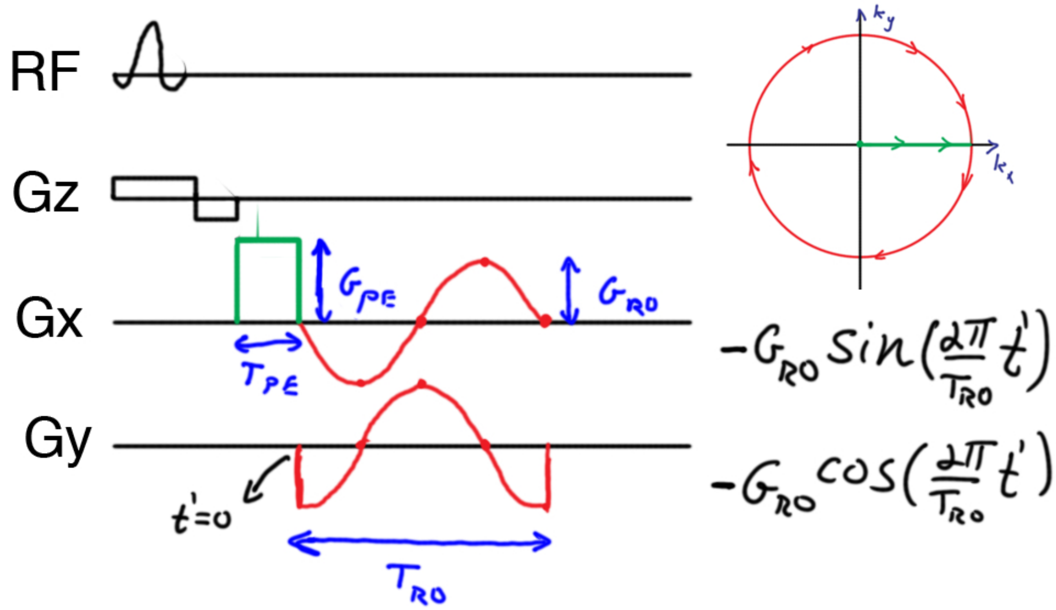
d) Another 2DFT sequence is applied and the image shown below gets displayed. Explain what might have changed in the pulse sequence as compared to the conventional sequence.



Explain briefly:

5. The Rings Trajectory (from midterm I 2012) [1]

Consider the following pulse sequence :



Each repetition is supposed to trace a single angular ring trajectory in k-space. The trajectory is designed by determining the parameters for the outmost ring and then scaling the gradients to trace the inter rings.

We would like to use it to scan a circular object with a FOV of 25.6cm at a spatial resolution of 1mm. For this question assume that the sampling interval is $\Delta T = 4\mu s$ and the maximum gradient amplitude is limited to $|G_x|, |G_y| < 4$ G/cm.

Recall $\int \sin(at)dt = \frac{-1}{a}\cos(at)$, and $\int \cos(at)dt = \frac{1}{a}sin(at)$.

a) What are W_k, the extent of the trajectory in k-space, and Δk, the minimum required spacing between samples in k-space?

$W_k =$	$\Delta k =$

b) For the outmost ring, what are T_{PE}, G_{PE}, G_{RO}, and T_{RO} that result in the fastest scan that does not violate the FOV and gradient amplitude constraints?

$T_{PE} =$	$G_{PE} =$	$G_{RO} =$	$T_{RO} =$

c) How many rings, N, are required to cover k-space?

$N =$

[1] This is a tribute to Holden Wu, who's PhD thesis was on the ring trajectory

d) You scan a point object located at $x = 10$cm. Unfortunately, due to systematic errors, the start and end of the A/D window is delayed by 10 samples with respect to the gradient waveforms. Assuming a "dumb" reconstruction computer that is not aware of the delay, draw is the image that is going to be reconstructed? Comment on the sensitivity of rings to delays.

Draw the image here. Label and put numbers everywhere you can.

6. **Matlab assignment**

   The assignment will be concerned with reconstructing 2DFT MR data. Two raw data files are available on the web site. These are

   - **se_t1_sag_data.mat** Sagittal (midline) T1 weighted spin-echo image of the head of a normal volunteer, acquired at 0.5T. The data set has 256 phase encodes, and 256 samples. CSF is dark due it its very long T1, white matter is brighter than gray matter. Fat is very bright due to it's short T1.
   - **phantom.mat** This is an axial gradient-echo image of a phantom (test object) acquired at 1.5T.

   These data files are in the class website.

   Centered 2DFT's Generally in medical image reconstruction, the origin for both the k-space data and the reconstructed image is the center of the image or data array. The usual convention for the fft is that the origin is at the beginning of the array, or the corner of a 2D array. To do a centered fft, you want to do an fftshift both before and after the fft. In homework 2 we wrote the functions fftc and ifftc, which you should use here.

   Displaying Images For displaying images the best option is "imshow" which is part of the image processing toolbox. If you have a complex image "im", you display it with

   ```
   >> imshow(abs(im),[]).
   ```

   This automatically scales the image from the image minimum to maximum. If you want to explicitly specify these, use

   ```
   >> imshow(abs(im),[win_min win_max]).
   ```

   This is quite common. Often the brightest part of the image is of little interest (a vessel, or fat, for instance). You will often want to specify a narrower window to accentuate the part of the image that is of interest.

   The built in "image" function in Matlab does a remarkably poor job of displaying images. If you don't have the image processing toolbox, this is what you will need to use. Matlab assumes you want a pseudocolor image, which is almost never true. To set the colormap to grayscale,

   ```
   >> colormap('gray');
   ```

   Also, you want the image to be displayed with a square aspect ratio, and you probably don't care about the axes,

   ```
   >> axis('square');
   >> axis('off');
   ```

   The default colormap has 64 gray levels. The image(im) function uses the value of im at each pixel as an index into the colormap. Hence, you want im to be real valued, and scaled from 1 to 64.

   ```
   >> imx = max(max(abs(im)));
   >> image(abs(im)*64/imx);
   ```

   You will want to incorporate all of these steps into an m-file, since you will do this often.

Orientation Conventions In displaying medical images there are conventions about the orientations the images are presented. The axes are described in "subject" space. These are right-left (R/L) for subject right and left, anterior-posterior (A/P) for subject front and back, and superior-inferior (S/I) for subject up and down. For axial slices, the image is oriented as it would appear if you were looking at the subject from the feet. The subject's left is on the right side of the image, and their front is up. For a sagittal slice (down the middle) you are viewing the subject from their left side, so up is up, the subject's front is to the left, and back is to the right. For coronal slices you are viewing the subject from the front, and up is up, and left is right. This all clear, right? It is important, though. You want to make sure they take out the diseased kidney, and not the healthy one.

There are several ways to flip images in matlab. You can flip an image about the diagonal with the transpose

```
>> imt = im.';
```

Note the " . ". The default transpose (i.e., just " ' ") also takes the conjugate, and that can cause problems when we care about the phase. You can flip images left-right and/or up-down by indexing the image in the reverse order

```
>> im_tb_flip = im(256:-1:1,:);
>> im_lr_flip = im(:,256:-1:1);
```

The ":" is a placeholder meaning that dimension is in the usual order. You can also rotate an image with the image processing toolbox function "imrotate" which rotates an image in the counter-clockwise direction

```
>> im_rot = imrotate(im,ang);
```

where "ang" is specified in degrees.

Once you have the image displayed, you can save it in eps format with

```
>> print -deps your_filename.eps
```

This produces an eps file in the current directory. Another option, that I prefer, is to save in pdf format,

```
>> print -dpdf your_filename.pdf
```

This can be imported into most word processing programs, such as LaTeX.

Questions

1. For the first data set, `se_t1_sag_data.mat`, load the raw data into matlab. This will give you a 256 by 256 complex array `d`, which contains the raw data.
   a) Make an image of the log of the magnitude of the raw data,
      ```
      >> imshow(log(abs(d)),[])
      ```
      Is the largest signal at the origin? Find where the largest signal is (the `max` function with two output arguments is useful here).
   b) Reconstruct the MR image with your `ifft2c` m-file.
      ```
      >> im = ifft2c(d);
      ```
      and show the real part of the reconstructed image

```
>> imshow(real(im)),[])
```

If you are using image, you will need to account for the fact that the real part is both positive and negative, Imshow does this for you. Explain the appearance of the real part based on the location of the maximum signal you found in (a).

c) Show magnitude of the reconstructed image,

```
>> imshow(abs(im)),[])
```

This is what is usually presented. Which direction is the phase encode direction? Which direction is the largest signal shifted in k-space (readout or phase encode?).

You can combine these plots on a single page using `subplot`. to save paper.

2. The second data set is `phantom.mat`. First clear matlab with `clear` to eliminate your previous data. Then load the phantom data set. The k-space data is again in the matlab variable `d`, and is 256 readout samples by 256 phase encodes.

a) Make an image of the log of the magnitude of the raw data. What sort of features would you expect to see in the image (e.g. round objects, square objects, periodic structures).

b) Reconstruct the image using `ifft2c`, and show the magnitude of the result.

c) The result of (b) doesn't look like you would expect. This is due to way the acquisition was performed. What sequence of RF pulses would produce this?

d) Correct the k-space data to compensate for the RF excitation sequence in (c), and perform the reconstruction again. Show the magnitude of the result. What direction is the phase-encode direction?

Again, you can combine plots using `subplot`. to save paper. Use publish to submit.

7. Matlab Exercise: 2DFT Pulse sequence design.
In this assignment we will write functions to design a 2DFT pulse sequence, and then simulate the design on a Bloch simulator.

The first step is to design a readout gradient. The readout gradient is composed of a prewinder, and a readout part. In the readout, we are interested in having a portion of the gradient that will scan the desired k-space length (gradient area) in which the gradient waveform is constant. This will give us a steady linear scan in k-space. For the prewinder, we are only interested in generating a gradient area that is half the area of the readout part. This should be as fast as possible to minimize the scan time. In addition, the ramps for the readout part should also be as fast as possible. The gradient waveform should always terminate with G=0.

a. Write a function genReadoutGradient.m that designs a readout gradient given the sequence parameters and the system constraints.

```
>> [gro,rowin] = genReadoutGradient(Nf, FOVr, bwpp, Gmax, Smax, dt);
```

The inputs to the function are :
Nf is the number of frequency encodes.
FOVr (in cm) is the desired field-of-view.
bwpp (in Hz/pixel) is the desired bandwidth per pixel
Gmax (in Gauss/cm) is the maximum gradient.
Smax (in Gauss/cm/s) is the maximum slew-rate.
dt (in s) is the duration for each sample.

The outputs of the function are:
gro - an array containing the gradient waveform.
rowin - an array containing the indexes in gro that correspond to the readout portion of the gradient. This will be used to crop the interesting part of k-space for reconstruction.

bwpp is something we have not discussed before. It basically defines the gradient amplitude we are going to use during the flat portion of the readout gradient. In essence, $bwpp = \frac{\gamma}{2\pi} G \frac{FOV}{Nf}$. Now, the A/D has a sampling bandwidth of 1/dt. So, the effective number of digital readout samples may be higher than our desired Nf frequency encodes. This is OK, since after we get all the samples, we will filter them to a bandwidth of Nf*bwpp and subsample to get Nf samples.

(Hint: You should first design a trapezoid that meets the criteria and then use the minimum-time-gradient function you wrote in previous homework to deign the prewinder. Remember to compensate for the ramp of the readout in the prewinder!!!!)

b. To design a phase encode gradient, we only need to design the gradient for the largest phase-encode and then scale it accordingly for the others.
Write a function genPEGradient.m that designs the gradient phase encode gradient for the largest phase encode and a phase encode table to scale it.

```
>> [grpe, petable] = genPEGradient(Np, FOVp, Gmax, Smax, dt);
```

The inputs to the function are :
Np is the number of phase encodes.
FOVp (in cm) is the desired phase-encode field-of-view.
Gmax (in Gauss/cm) is the maximum gradient.

Smax (in Gauss/cm/s) is the maximum slew-rate.
dt (in s) is the duration for each sample.

The outputs of the function are:
grpe - an array containing the gradient waveform.
petable - Npe x 1 array containing the phase encode table to scale the phase-encode gradient for each phase-encode. The array entries should be bounded between [-1 : 1]

Now that we have a way to generate the waveforms of a 2DFT sequence, we will simulate such a sequence for a distribution of spins. Download the file hw5_img.mat from the class website. This file contains the arrays dp [7715x2], mx [7715x1], my[7715x1], and mz[7715x1],. These array represents the positions of 7715 spins in space and their magnetization. We will now image them!

c. Design a 2DFT sequence with readout/phase-encode FOV of 14/7 cm, Nf/Np of 64/32 (giving a resolution of $\approx$ 2.2mm. Use a bandwidth per pixel of about 1.8624 Khz/cm. Use $dt = 4\mu s$, Gmax=4G/cm and Smax=15000G/cm/s. Design a hard-pulse RF with 90 degree excitation to use with the gradient sequence. Plot k-space by integrating the gradient waveforms. Make sure it makes sense!

d. Simulation: Simulate the sequence acquisition using the Bloch simulator one phase encode at a time (The simulation takes about 1-2sec per phase encode). Use T1=T2=100. The output of the simulator needs to be integrated across all the spins to get the signal. The code for the simulation part should look like:

```
>> ....
>>  g = [gro, gpe*phtable(n)];
>>  [mx,my,mz] = bloch(rf,g,4e-6,100,100,0,dp,2,mx,my,mz);
>> mxy = sum(mx,2) + sqrt(-1)*sum(my,2);
>> ...
```

You will find that the number of readout samples is bigger than Nr because we sampled at 250Khz ($dt = 4\mu s$). Scanners also sample at that rate and then apply a digital filter to get the desired number of readout points. There are two options, to do the filtering and subsampling of k-space or cropping the image. For simplicity, we will take the 2nd approach.

take a 2D centered IFFT of the resulting k-space data. Crop the image to the desired FOV. You should be able to read something. If you can't, ... then something is wrong. Submit a plot of the gradient waveforms and the image.

Enjoy!

e. Reduce the FOV by a factor of 1.5 in the phase encode and repeat the scan. What do you get? submit the image.