HL2011: Exercise Set 3
k-space sampling and basic imaging

Andreas Sigfridsson, Peter Nillius

1 Preparations

If you haven't done it already, download the package `seemri.zip` from Bilda and unzip it to a suitable directory. Start Matlab and add the path to seemri using the matlab command `addpath` <*path to seemri*>. In matlab, type `help seemri` and make sure you get the help page. You are now ready to start.

Define some useful constants:

```
gammabar = 42.58e6;
gamma = 2*pi*gammabar;
```

2 k-space sampling

The MRI signal can be interpreted as a signal in k-space, the continuous Fourier transform of the image. By measuring the signal we measure a trajectory in k-space. Because only a finite number of trajectories will be measured and the signal is sampled, the data we acquire will be a sampling of the image's k-space.

In this exercise you will investigate how the sampling of k-space affects the reconstructed image. We will look at the a disc with radius 3. Define the disc as a Matlab function object:

```
u = @(x,y) sqrt(x.^2+y.^2)<=3;
```

You can view the disc by defining a coordinate grid and evaluate the function at each point thereby creating a digital image:

```
x = -5:0.1:5;
y = -5:0.1:5;
[xg, yg] = meshgrid(x, y);
imagesc(x, y, u(xg, yg));
axis image
colormap gray
title('Image space')
```

The continuous Fourier transform (k-space) of this disk can be derived analytically and is equal to

$$\mathcal{F}\{u\} = 3\frac{\mathrm{J}_1\left(3 \cdot 2\pi\sqrt{k_x^2 + k_y^2}\right)}{\sqrt{k_x^2 + k_y^2}} \tag{1}$$

were J_1 is the Bessel function of the first kind. Define a function object for this in Matlab by:

```
Fu = @(kx,ky) 3*besselj(1, 2*pi*(sqrt(kx.^2+ky.^2)+1e-9*(kx==0 & ky==0))*3)...
  ./(sqrt(kx.^2+ky.^2)+1e-9*(kx==0 & ky==0));
```

Note, the construction to handle the limit when $(k_x, k_y) \to (0, 0)$.

The function `mrireconstruct` in the **seemri** package reconstructs an image from a sampling of its k-space. Construct a sampling grid around origin from $-k_{max,x}$ to $k_{max,x} - \Delta k_x$ and $-k_{max,y}$ to $k_{max,y} - \Delta k_y$ with sampling distance Δk_x and Δk_y.[1]. Try to begin with

```
kmaxx = 5;
kmaxy = 5;
dkx = 0.1;
dky = 0.1;
kxa = -kmaxx:dkx:kmaxx-dkx; % Array
kya = -kmaxy:dky:kmaxy-dky; % Array
[kxg, kyg] = meshgrid(kxa,kya); % Grid
```

You can view the k-space by evaluating `Fu` with the sampling grid using:

```
imagesc(kxa, kya, Fu(kxg, kyg));
axis image
colormap gray
title('k-space')
```

Now, use `mrireconstruct` to reconstruct and view the reconstructed image.

```
mrireconstruct(Fu(kxg, kyg), max(kmaxx,kmaxy), 'Plot', true)
title(sprintf('kmaxx = %g, kmaxy = %g, dkx = %g, dky = %g', kmaxx, kmaxy, dkx, dky))
```

The k_{max} argument to `mrireconstruct` is used to compute the coordinates of the object (see `mrireconstruct.m`).

Now, reconstruct and view images using Δk values of 0.05, 0.1 and 0.2 in either or both directions. **Q1: How does the k-space sampling interval Δk affect the reconstructed image?** Specify in terms of image field of view (FOV) and *pixel size*.

Repeat the procedure but, this time, let $\Delta k = 0.1$ and reconstruct and view images using k_{max} values of 2, 4 and 10. **Q2: How does the range of the sampling k_{max} affect the reconstructed images?** Specify in terms of image field of view (FOV) and *pixel size*.

Say that you would like to image a 220 mm wide object with 1 mm large pixels/voxels. **Q3: How do you select k_{max} and Δk?** Don't forget the units!

3 Basic Gradient Echo Imaging

In this exercise you will design a gradient echo imaging sequence with rectilinear (Cartesian) sampling. To keep the simulations fast we are only simulating a 2D plane and therefore don't need to do slice selection.

Let $B_0 = 1.5$ T and create a $90^\circ_{x'}$ RF pulse, with $t_p = 1$ ms, for resonance. Let the envelope function be rectangular since this makes the simulation faster.

```
B0 = 1.5;
tp = 1e-3;
B1 =
f_rf =
rf = RectPulse(B1, f_rf, 0, tp);
```

[1]This asymmetric sampling from $-k_{max}$ to $k_{max} - \Delta k$ makes it easy to map the data to the discrete Fourier transform (DFT) and is how `mrireconstruct` works.

Create a small volume to image using the command `disc`. For now make it small when you are about to debug your pulse sequence. Choose a radius 3 mm and resolution 1 mm by:

```
iv = disc(3,1);
```

Let the echo time $T_E = 10$ ms and repetition time $T_R = 2$ s, i.e.

```
TE = 10e-3;
TR = 2;
```

Design the sequence such that the signal is measured over the time interval $T_E - \tau \leq t \leq T_E + \tau$. Let $\tau = 4$ ms, i.e.

```
tau = 4e-3;
```

Now choose your sampling grid, by computing the required k_{max} and Δk to get a field-of-view of at least 8 mm and a pixel size of 1 mm. The lines below recomputes `dkx` and `dky` to ensure that we sample the origin.

```
kmaxx =
kmaxy =
dkx =
dky =
dkx = kmaxx/ceil(kmaxx/dkx);
dky = kmaxy/ceil(kmaxy/dky);
kxa = -kmaxx:dkx:kmaxx-dkx; % Array
kya = -kmaxy:dky:kmaxy-dky; % Array
[kxg, kyg] = meshgrid(kxa,kya); % Grid
```

Q4: How many excitation-measurement repetitions N will we need to achieve this k-space sampling?

Let the phase encoding be in the x-direction. The values in `kxa` are the k_x-coordinates of the k-space trajectories. Each trajectory goes from $k_y = -k_{max}$ to $k_y = k_{max} - \Delta k_y$. **Q5: Sketch the measurement trajectories and sampling points in k-space.**

Let the phase encoding duration be τ. **Q6: What should the phase encoding gradient amplitudes be?** There will be one for each trajectory. Compute them and store them in an array, `Gpexs`, (`Gpexs(n)` should contain the gradient amplitude for the nth trajectory, where $n = 1, ..., N$).

```
Gpexs =
```

You can now create an object that represents the x-gradient, with all its repetitions, by providing the array as you would a single amplitude value to `Gradient`, by doing the following:

```
gx = Gradient([tp tp+tau], {Gpexs 0});
```

Now compute and create an object for the gradient in the frequency encoding direction (y). Let the frequency encoding be in the time interval $T_E - \tau \leq t \leq T_E + \tau$. **Q7: What are the required amplitudes (before (prewinder) and during signal acquisition) for the frequency encoding gradient?** Hint: compute the gradient strength by the k-space traversal of $2k_{max}$ over the duration of 2τ.

```
Gfey1 =
Gfey2 =
gy = Gradient([tp tp+tau TE-tau TE+tau], [Gfey1 0 Gfey2 0]);
```

Q8: The signal is sampled at time intervals corresponding to the sampling points in k-space. What is the required sampling interval Δt? Hint: compute the sampling interval using the readout gradient strength and Δk.

Create an analog-to-digital converter object that samples the signal during the frequency encoding by the commands

```
dt =
adc = ADC(TE-tau, TE+tau, dt);
```

You are now ready to try your imaging sequence. The repetition time T_R and the number of repetitions is provided last.

```
[S, ts] = seemri(iv, B0, rf, gx, gy, adc, TR, length(Gpexs), 'PlotKSpace', true);
```

The last arguments turn on plotting of the k-space interpretation of the signal. It can be good to check that the sequence diagram looks correct. You can use the option `..., 'Pause', 0)` to pause between each repetition.

The nth column of the matrix `S` now contains the signal for repetition n. If everything worked, and we have traversed k-space in order, `S` should contain a rectilinear sampling of k-space.

Try reconstructing the image from `S` using:

```
mrireconstruct(S, max(kmaxx,kmaxy), 'Plot', true);
```

The image should look similar to `imagesc(reshape(iv.Mz0, 7, 7))`.

When things work you can increase the resolution to `disc` and speed up the simulation by turning off the plotting.

Q9: Present the pulse sequence, the nominal image and the reconstructed image in your report.

4 Report

Submit an individual report and source code (as separate PDF and .m files) on Canvas by April 25 23:59, or before 13:15 to count as attendance for the session.