Principles of Magnetic Resonance Imaging.
Exercise session 3

Exercise 1: Discrete Fourier transform. Algebraic approach

Write the (one dimensional) DFT as a matrix-vector product, that is, construct the matrix **D** such that, given the signal (f_n) with $n = 0 \dots N-1$:

$$\hat{\mathbf{f}} = \mathbf{D}\mathbf{f}$$

where $\mathbf{f}$ and $\hat{\mathbf{f}}$ are the column vectors representing (f_n) and $(\hat{f}_k)$, respectively.
Implement **D** in Matlab for $N = 256$. Define a test signal (for example, `f=rand(N,1)`) and check your code against the matlab function `fft(f)/N`. Hint: you could check the norm (function `norm`) of the error `D*f-fft(f)/N` or plot it.
How do the columns and the rows of **D** look like? Plot some of them (Hint: m-th row: `D(m,:)`, n-th column: `D(:,n)`)
Write the inverse DFT as a matrix-vector product (denote the matrix by **E**). Do NOT implement it in Matlab. Compare the elements of **D** and **E**. What do you notice? What is the euclidean norm of the columns and rows of the two matrices? Modify slightly the definition of DFT and inverse DFT in such a way that the column of the two matrices have norm 1.
Derive the relation between **D** and **E**. Now you can easily derive `E` from `D` in Matlab. Do it and check if everything works, that is: $\mathbf{EDf} = \mathbf{DEf} = \mathbf{f}$.

Exercise 2: FFT vs Matrix-vector multiplication

Calculate the DFT by **a)** FFT and **b)** by the matrix-vector (MV) multiplication as derived in the previous exercise. Try for several values of N: $N = 128, 256, 512, 1024, 2048, 4096$. Plot graphs of the computing time of FFT versus the matrix-vector multiplication for the different values of N. Use the matlab commands `tic` and `toc`. What do you observe?
HINT: to have an accurate computing time measurement, perform the calculation a large number of times and take the average value. For example, save the following code into a matlab script and run it:

```
N = 128;
f = rand(N,1);
tic;
for j = 1:100000
  fft(f);
end
total_time = toc;
average_time = total_time/100000;
```

Exercise 3: A spiral k-space trajectory

Suppose we want to scan an object with a certain field of view, L, and spatial resolution, Δ_s. The number of voxel along each spatial direction is N. We choose the spiral k-space trajectory from Fig. 1, which can be expressed in the following form:

$$(k_x(t), k_y(t)) = \alpha t(\cos \omega t, \sin \omega t)$$

where $t \in [0, t_{\max}]$ denotes the time variable and $\omega \in \mathbb{R}^+$, $\alpha \in \mathbb{R}^+$ are the parameters of the trajectory. Obviously, the trajectory starts at $(0, 0)$. The samples along the trajectory are taken at time steps equal to Δ_t.
Try to answer the following questions:

1. What is the effect of α and ω on the trajectory?
2. Derive the values of α and ω in terms of L, Δ_t and Δ_s.
3. Derive a formula for the corresponding gradients, $(G_x(t), G_y(t))$.
4. In practice, the gradient's amplitude is bounded by the fixed value $G_{\max}$, thus $|G_x(t)| \leq G_{\max}$ and $|G_y(t)| \leq G_{\max}$ for all $t \in [0, t_{\max}]$. Modify the expressions for ω and α to fulfill these constraints.
5. Write a Matlab code to compute and plot the gradients G_x and G_y. Take $L = 1$, $N = 64$, $\Delta_t = 0.001$, $\gamma = 0.01$ and $G_{\max} = 4 \cdot 10^5$.
6. Change the values of $G_{\max}$, what do you observe?

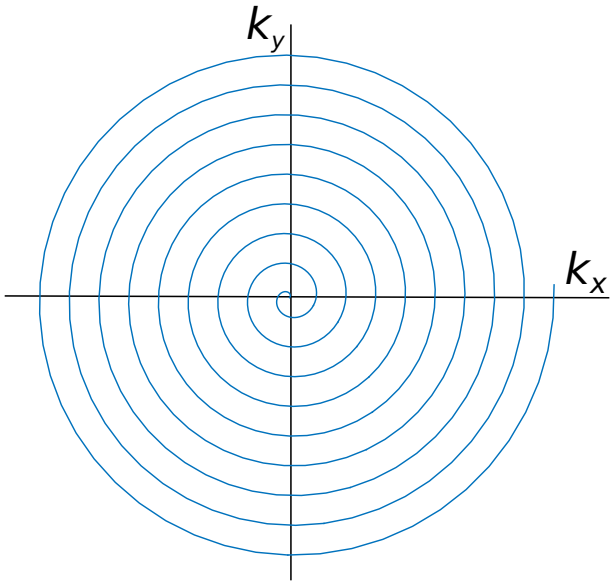

Figure 1: Spiral k-space trajectory.