# HL2027 3D reconstruction and analysis in medicine

# Project 1

# Radial sampling in MRI

## Introduction:

As mentioned in class, one of the main issues with MRI is its time-consuming acquisition. As we know from the lecture, cartesian sampling complies with Nyquist's theorem. If we acquire less data than that, we are exposed to artifacts or noise. Still, adding extra assumptions, it is possible to get reasonably good images with sub-Nyquist sampling. Radial sampling is one of these techniques in which "spikes" (lines at different angles) are acquired in the k-space. The goal of the project is to test different parameters and reconstruction methods and assess their quality.

Sigpy is a nice toolbox for performing MRI reconstruction. We will use it as a basis for the project:

https://github.com/mikgroup/sigpy

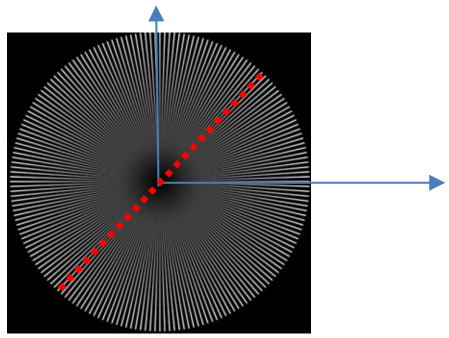
The data consists of a T1w brain image of size 256x256x90 (t1.nii.gz). For tasks 1-6, please use one slice, for example the central one, i.e., slice 45. For task 7, you can use the complete image. You can use ITKsnap to visualize the 3D volume. SimpleITK can read NIfTI files. The code looks like:

```
import SimpleITK as sitk
import numpy as np
sitk_t1 = sitk.ReadImage('t1.nii.gz')
t1 = sitk.GetArrayFromImage(sitk_t1)
```

## Tasks:

1. **Simulation of radial sampling**: **(15%)**
   Use **`sigpy.mri.radial`***(coord_shape, img_shape)* to generate the coordinates in the k-space. coord_shape is a vector with [number of spikes, number of acquisitions per spike, dimension]. Img_shape is the dimension of the image. In the project, we are interested in testing different values of "number of spikes" so you can call the method as:

   coord = **sigpy.mri.radial**([spikes,256,2],[256,256])



One “spike” is depicted in red in the figure. 256 samples of the k-space are acquired per spike. The spikes are equally distributed in the circle. As mentioned, the image size is 256x256 in plane and 90 slices.

We will use these coordinates to simulate the sampling of the k-space. For this, we will use a high-resolution image that can be used as ground truth. Take the Fourier transform of that image and use the coordinates to select the values that will be used for reconstruction. The original image can be used for comparing the quality of the reconstruction (see below). Estimate the acceleration factor with respect to the number of spikes compared to cartesian sampling. Test acceleration factors up to 10x.

2. **Quality assessment: (10%)**

   In order to assess the quality of the reconstructions, one can use qualitative and quantitative tools. For qualitative analysis, in addition to the reconstructed images, one can show difference maps, i.e., Original-Reconstructed. Regarding quantitative analysis, one can use distance measurement between images. Scikit-image has a set of functions for this. For example, the code:

   psnr = **peak_signal_noise_ratio**(img_original,img_reconstructed)

   will compute the peak signal to noise ratio (PSNR) between the two images, which is a measurement of difference between images. Compute at least PSNR and SSIM in the experiments below. **Optional**: you can explore skimage for more measurements.

3. **Reconstruction with the non-uniform Fourier transform (NUFT): (30%)**
   The first approach for reconstruction is to use non-uniform Fourier transform. Use the functions from sigpy for this. You can get inspiration from the code of the tutorial:

https://github.com/mikgroup/sigpy-mri-tutorial/blob/master/01-gridding-reconstruction.ipynb

   The data of the tutorial can be found in canvas as well: ‘projection_ksp.npy’ and ‘projection_coord.npy’

   Try the method with a different number of spikes. Read the papers suggested in the documentation and explain in the report the method and why it is necessary to perform density compensation. Compare the results with the standard inverse Fourier transform (it’s also implemented in sigpy).

4. **Other reconstruction methods: (20%)**
   **`sigpy.mri.app.L1WaveletRecon`** is designed to perform compressed sensing reconstruction. Try that method with radial sampling for different number of spikes. Do the same with **`sigpy.mri.app.TotalVariationRecon.`** Play around with the parameters of these methods and discuss their influence on the reconstruction in the report. Notice that these methods can also use different coils, but we are working with only one. Set the appropriate sensitivity map to disregard that.

5. **CT reconstruction methods for radial MRI: (15%)**
   In class, we briefly mentioned that radial sampling is related to CT reconstruction. That means that all CT reconstruction methods can be applied for reconstructing MR images.

Describe with equations to model the problem as a CT reconstruction one, that is, how to set up radial sampling in terms of a sinogram.

6. **Bonus (optional): (+10%)**
   Run filtered back projection (FBP) to reconstruct the images. **Hint**: scikit-image implements FBP and other CT reconstruction methods.

7. **3D reconstruction: (10%)**
   Apply the best performing method for getting an acceleration of 5x and 10x in a 3D volume. **Hint:** apply reconstruction to every slice in the 3D volume independently.

The project work can be done in groups of **2 people**. Each group has to hand in a report of **maximum 4 pages** of text (figures and refs do not count for this limit) describing

1. underlying **problems**
2. basic **theories**
3. solution **strategies** used
4. findings from the **experiments**
5. final **conclusions**.

Make sure to **include content for each** of the given **tasks**. The highlighted parts may help to put emphasis on certain aspects.

Each group will do a **demo and present** its **results** to the rest of the class on the **25$^{th}$ of October**. Each group will have **7 min** for this. The report and code must be submitted no later than the **23$^{rd}$ of October**. The grades of the mini-exams of the modules *Reconstruction* and *Restoration* are part of the final assessment for this project:

| | |
|---|---|
| **Report**: | 45 points |
| **Demo & presentation**: | 10 points |
| **Code**: | 25 points |
| **Mini exams 1 and 2**: | 20 points |

Good luck!!!