Julius Ott - CS for MRI - Exercise Sheet 1

May 14, 2020

1 Excercise Sheet 1

1.1 Compressive Sensing for MRI

1.1.1 Task 1 : What is the radio-frequency range? How does it relate to MRI?

The radio frequency range is in between 20kHz and 300GHz. As the name implies, these frequencys can be reconstructed, since they are used for radio signal transmission. In relation to MRI, hydrogen nuclei are excited by a pulse in the radio frequency range and omit energy with a resonance frequency, again in the radio frequency range. Therefore, the signal can be reconstructed to an image.

1.1.2 Task 2: What is the nuclear magnetic moment and the gyromagnetic ratio? Describe it for different atoms.

The nuclear magnetic moment is microscopic view of the magnetization of nucleus. It can be described in a sum of the orbit magnetic moment and angular magnetic moment. The two different magnetic moments are well illustrated by the moon and earth. The orbit magnetic moment appears when a nucleus rotates around an axis, like the moon around the world. The angular magnetic moment is the magnetization correlated to the self rotation, or spin, like the moon spins around its own axis. Furthermore, the magnetic moment can be defined as the torque on the nucleus by an external magnetic field.

$$\tau = m \times B$$

, where m is the magnetization of the nucleus and B the external magnetic field.
In addition, the gyromagnetic ratio is defined as the ratio between the magnetic moment and spin moment. $\gamma = \frac{m}{S}$ The gyromagnetic ratio for a single proton is 42.57 $\frac{Hz}{T}$, which is also the ratio for hydrogen with a nuclear spin number of 0.5. In case of water, there are two free protons and whether they spin parallel or antiparallel, which results in a total spin of 1 for parallel and 0 for antiparallel, the gyromagnetic ration is higher or lower.

1.1.3 Task : 3 Do the calculations and check that the function presented as a solution to the Bloch is indeed a solution

The Given Bloch equations: $\frac{dM_x(t)}{dt} = \gamma * B_0 * M_x(t)$
$\frac{dM_y(t)}{dt} = \gamma * B_0 * M_y(t)$
$\frac{dM_z(t)}{dt} = 0$

with the following solutions:
$M_x(t) = M_{0x} * cos(\gamma B_0 t) + M_{0y} sin(\gamma B_0 t)$

$M_x(t) = -M_{0x} * sin(\gamma B_0 t) + M_{0y} cos(\gamma B_0 t)$
$M_x(t) = M_{0z}$

Taking the derivative of the solution functions leads to ...

$\frac{dM_x(t)}{dt} = -M_{0x} * \gamma * B_0 * sin(\gamma B_0 t) + M_{0y} * \gamma * B_0 * cos(\gamma B_0 t) = \gamma B_0 M_y(t)$

$\frac{dM_y(t)}{dt} = -M_{0x} * \gamma * B_0 * cos(\gamma B_0 t) - M_{0y} * \gamma * B_0 * sin(\gamma B_0 t) = -\gamma B_0 M_x(t)$
$\frac{dM_z(t)}{dt} = 0$

Since, the solution solves the equation, the solution is a correct one. For $M_z(t)$ one could choose an arbitrary constant value.

1.1.4 Task: 4 Show that the angle between B and M does not change

Since, M is a rotation around the z-axis with a magnetic moment M_0 and B is the external magnetic field in z-direction, M is always perpendicular to B. Therefore, the angle between them never changes.

1.1.5 Task: 5 What is a rotation matrix?

A rotation matrix rotates a vector or matrix around a certain axis, while preserving the properties of the vector or matrix. For a rotation matrix $R \in R^{n \times n}$, the following properties hold: * det(R) = 1 * $R * R^T = I$ with I = Identity Matrix * $\|(R_\mathrm{i})\| = 1$

[1]:
```
using PyCall
using LinearAlgebra
using Images
#using ImageView
using FFTW
using Wavelets
using Printf
using Base
using PyPlot
using Latexify
using LaTeXStrings
using HDF5
sympy = pyimport("sympy")
pywt = pyimport("pywt");
```

Define the symbolic matrix

[2]:
```
x = sympy.Symbol("x") # symbolic x
y = sympy.cos(x)
z = sympy.sin(x)
co = y.subs(x, pi)      # substitute x with any number for cos(x=any) and␣
 ↪sin(x=any)
si = z.subs(x, pi)
```

```
#convert symbolic substitution to Float to see the real number, otherwise␣
↪PythonObject
R = [convert(Float32, co) convert(Float32, si) 0 ; -convert(Float32, si)␣
↪convert(Float32, co) 0; 0 0 1] ;
latexify(round.(R))|> display
```

$$\begin{bmatrix} -1.0 & 0.0 & 0.0 \\ -0.0 & -1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{1}$$

Check for the matrix properties

[3]:
```
determinant = det(R)
multiplication = R* R' ;
norm1 = norm(R[:,1]) ;
norm2 = norm(R[:,2]);
norm3 = norm(R[:,3]);

L"Determinant\, of\, R" |>display
latexify(determinant) |> display
latexify(L"norms\, of\, columns\, of\, R:\,") |> display
latexify([norm1 norm2 norm3]) |> display
latexify(L"Output, of\, R* R^T:\,") |> display
latexify(round.(multiplication)) |> display
```

$Determinant\, of\, R$

1.0

$norms\, of\, columns\, of\, R:\,$

$$\begin{bmatrix} 1.0 & 1.0 & 1.0 \end{bmatrix} \tag{2}$$

$Output, of\, R* R^T:\,$

$$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{3}$$

As shown, the given matrix is a rotation matrix. The rotation around the z-axis can be shown by multiplying the matrix with the basis vector for the z axis, while the result is again the basis vector for the z-axis. In addition, the entries in the rotation matrix confirm the result, because only x- and y-axis values are multiplied with the angular functions.

[4]:
```
z_ax = [0 0 1]' ; #z basis vector
R_rot = R*z_ax;
latexify(R_rot)
```

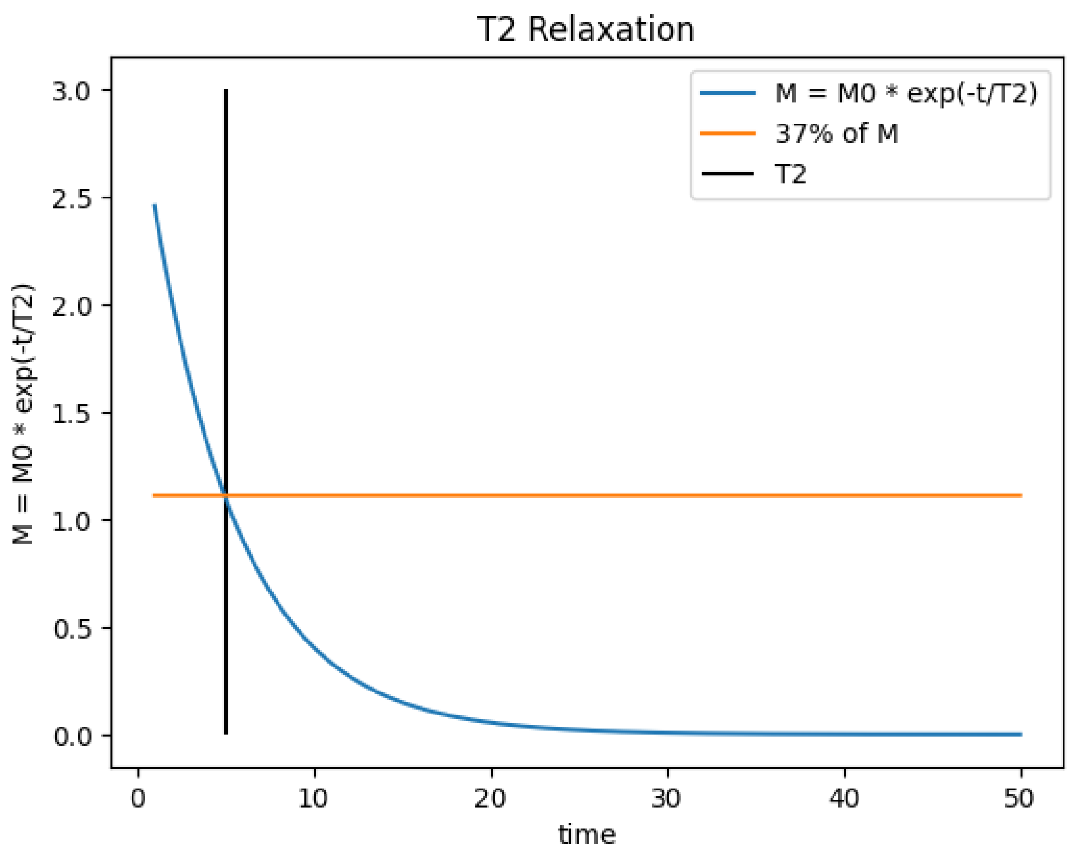
[4]:

$$\begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} \tag{4}$$

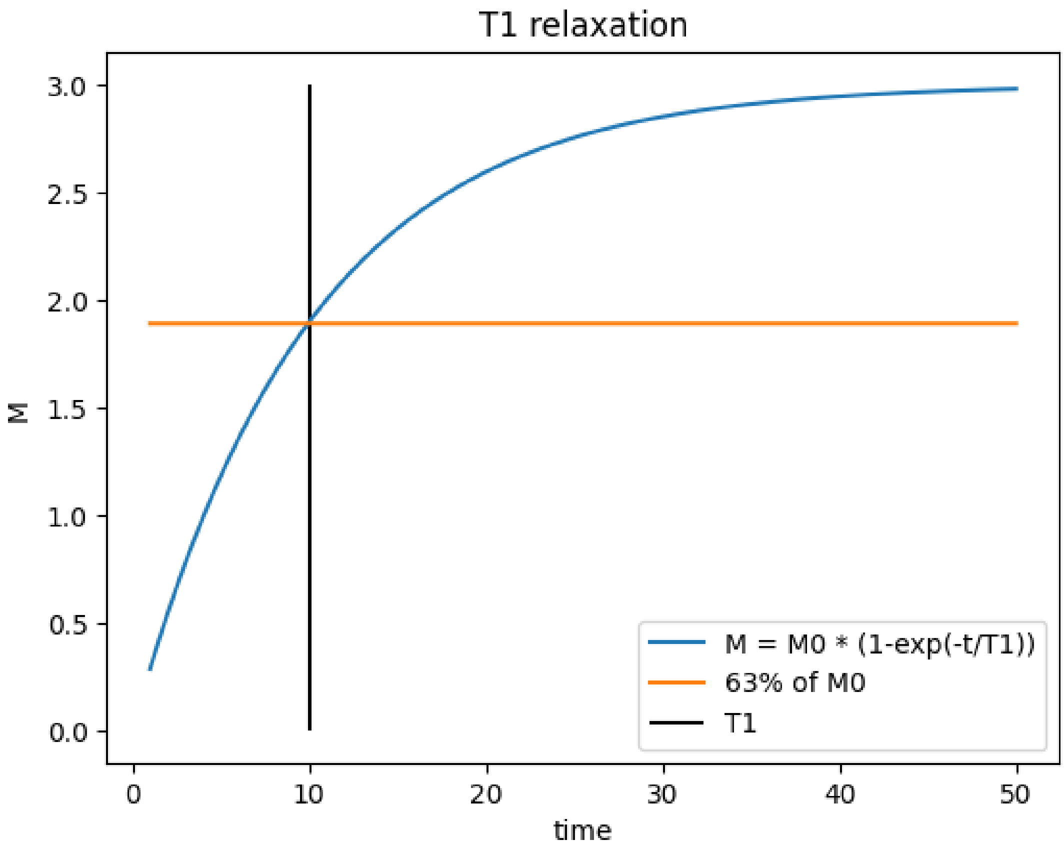
1.1.6 Task 6: What proceeds more rapidly, T1 or T2 ? Why? What do 37% and 63% mean for the relaxation times?

Regulary, T2 proceeds faster than T1, because T1 is the time constant for recovering the magnetization M_z before the pulse and requires the reduction of magenetization $M_x y$ in the transversal direction.
Furthermore, T2 is the time when 37% of the transversal magentization is gone and T1 is the time required to gain 63% of the longitudinal magnetization.
Since, expiring the net magnetization in transversal direction does not excite all spins, and T2 is the minimum time for T1, T2 is mostly much faster.
Both relaxation behaviours a shown in the plots below.

[5]:
```
M0 = 3 ;
ts = 1:0.01:50;  #plot t from 1 to 50 s
T1 = 10 ;
T2 = 5 ;
Ms = [];
Mzs = [];
for t in ts
    Mz = M0*(1- exp(-t/T1));
    Mxy = M0 * exp(-t/T2)
    append!(Mzs, Mz)
    append!(Ms, Mxy)
end
fig = figure();
plot(ts,Ms, label = "M = M0 * exp(-t/T2)")
plot(ts, 0.37*M0*ones(size(ts)), label="37% of M")
vlines(T2, ymin=0, ymax= M0, label="T2")
xlabel("time"); ylabel("M = M0 * exp(-t/T2)")
title("T2 Relaxation")
legend();
savefig("T2 Relaxation")
```



[6]:
```
fig = figure();
plot(ts, Mzs,label="M = M0 * (1-exp(-t/T1))")
plot(ts, 0.63*M0*ones(size(ts)), label="63% of M0")
vlines(T1, ymin=0, ymax =M0, label="T1")
xlabel("time"); ylabel("M ")
title("T1 relaxation")
legend(loc="best");
savefig("T1 Relaxation")
```



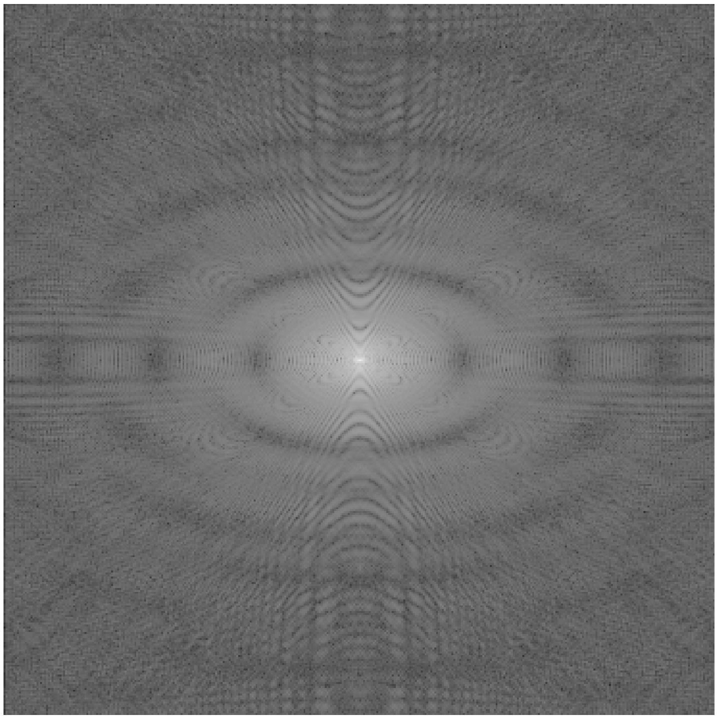
1.1.7 Task 7: Describe the order of the magnitude of T1 and T2 for different biological tissues.

While observing the T1 and T2 values for different tissues for $B_0 = 1.5T$, there a three main observations. First, T2 is mostly 5-10 times longer than T1. Second, liquids like water have very long T1 and T2 times. Last, dense or solid tissues like bones have a short T2 value. These observations can be described by the molecular motion. While free molecules lead to more spin and therefore to a longer T2, a dense tissue with less free molecules will result in a shorter T2 and a larger T1.

1.1.8 Task 8: Hello World in Julia

[7]:
```
# image
N = 500   # high N results in high resolution
I = shepp_logan(N, highContrast = true)
imshow(I, cmap="gray");
axis("off");
savefig("Shepp_Logan")
```

[8]:
```
#Fourier tranbsform of shepp logan
fourier_shepp = FFTW.fft(I) ;
fshift = FFTW.fftshift(fourier_shepp)
magnitude =  log.(abs.(fshift))
@show size(magnitude)
imshow(magnitude, cmap="gray")
axis("off")
@show size(I);
savefig("Shepp Logan Fourier")
```

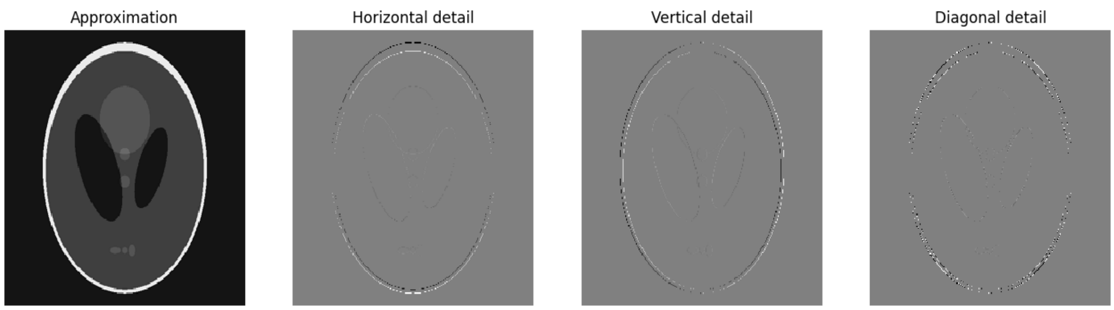


```
size(magnitude) = (500, 500)
size(I) = (500, 500)
```

[9]:
```
# Wavelet transform of Shepp Logan
titles = ["Approximation", "Horizontal detail",
          "Vertical detail", "Diagonal detail"]
coeffs2 = pywt.dwt2(I, "bior1.3")
LL, (LH, HL, HH) = coeffs2

fig = figure(figsize=(16,4))
fig.add_subplot(141)
imshow(LL, interpolation="nearest", cmap="gray", aspect="auto")
title(titles[1])
axis("off")
fig.add_subplot(142)
imshow(LH, interpolation="nearest", cmap="gray", aspect="auto")
title(titles[2])
axis("off")
fig.add_subplot(143)
imshow(HL, interpolation="nearest", cmap="gray", aspect="auto")
title(titles[3])
axis("off")
fig.add_subplot(144)
```

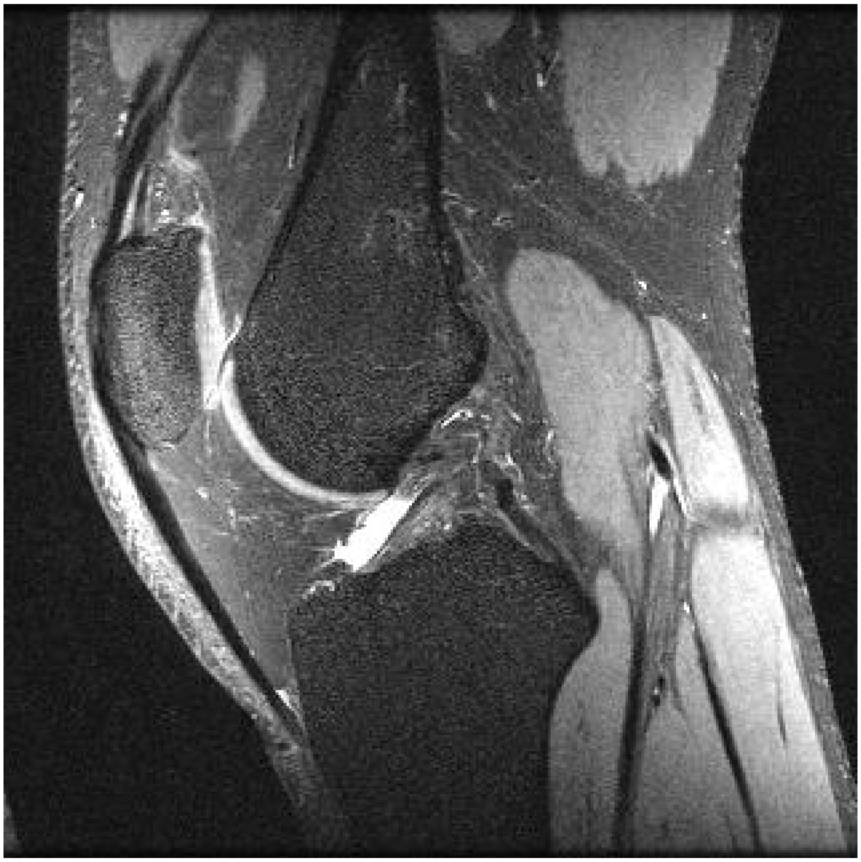
```
imshow(HH, interpolation="nearest", cmap="gray", aspect="auto")
title(titles[4])
axis("off");
savefig("Shepp Logan Wavelet")
```



1.1.9 Task 9: Now get some data from http://mridata.org/, load it in Julia and do the same procedure as in the exercise above.

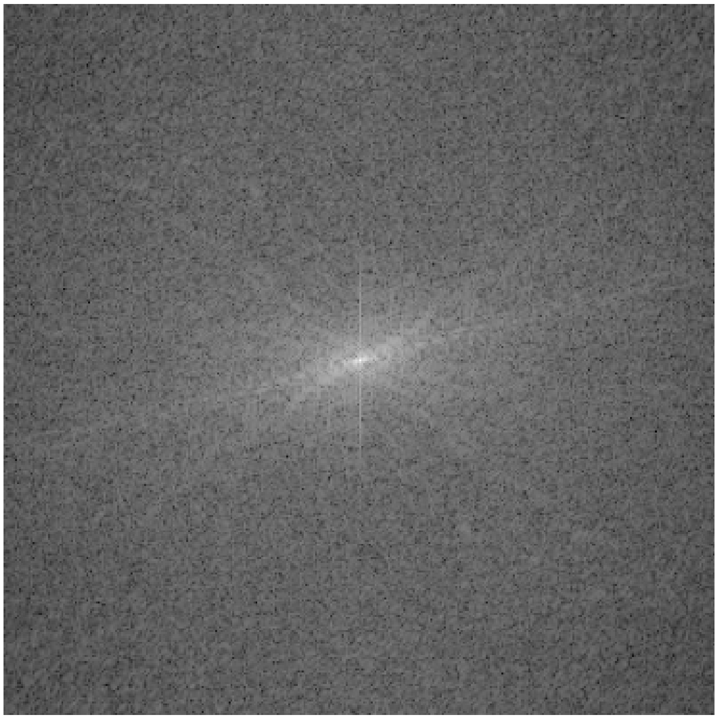
[10]:
```
knee = Images.load("MRI_knee.jpg")
savefig("Knee MRI")
knee
```

[10]:



[11]:
```
knee = convert(Array{Float64}, knee);
```

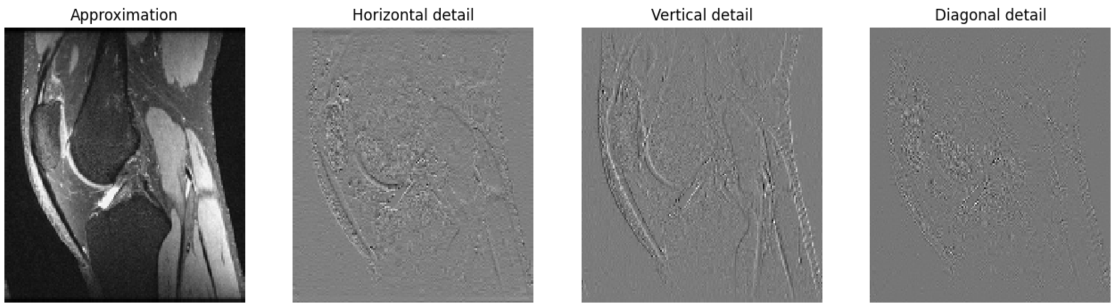
[12]:
```
#Fourier transform of Knee MRI
fourier_knee = FFTW.fft(knee) ;
fshift = FFTW.fftshift(fourier_knee)
magnitude =  log.(abs.(fshift))
imshow(magnitude, cmap="gray")
axis("off");
savefig("Fourier Knee")
```



[13]:
```
# Wavelet transform of Knee MRI
titles = ["Approximation", "Horizontal detail",
          "Vertical detail", "Diagonal detail"]
coeffs2 = pywt.dwt2(knee, "bior1.3")
LL, (LH, HL, HH) = coeffs2

fig = figure(figsize=(16,4))
fig.add_subplot(141)
imshow(LL, interpolation="nearest", cmap="gray", aspect="auto")
title(titles[1])
axis("off")
fig.add_subplot(142)
imshow(LH, interpolation="nearest", cmap="gray", aspect="auto")
title(titles[2])
axis("off")
fig.add_subplot(143)
imshow(HL, interpolation="nearest", cmap="gray", aspect="auto")
title(titles[3])
axis("off")
fig.add_subplot(144)
imshow(HH, interpolation="nearest", cmap="gray", aspect="auto")
title(titles[4])
axis("off");
```

```
savefig("Knee Wavelet")
```



1.1.10 Task 9: Load an angiogram and a brain image in Julia, apply the Wavelets transform and the DCT transform to both and plot the histogram of coefficients. Compare, for the two different images, which transform performs better for a sparse representation. Discuss the results

[14]:
```
angiogram = Images.load("angiogram.jpg")
angiogram = Gray.(angiogram)
@show size(angiogram)
angiogram
```

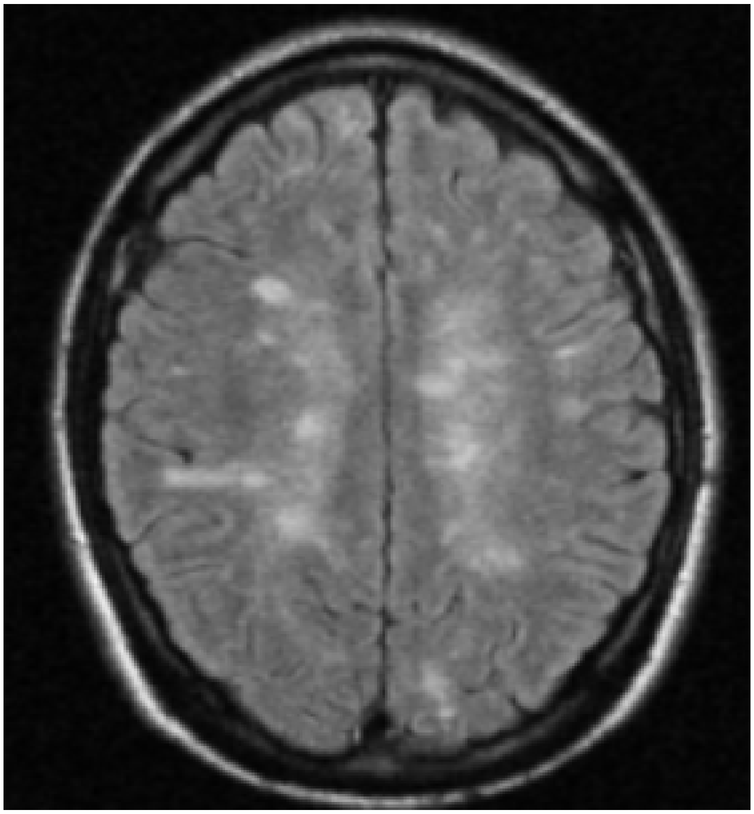
```
size(angiogram) = (302, 280)
```

[14]:

[15]:
```julia
brain = Images.load("MRI_brain.jpg");
brain = imresize(brain, (302,280));
brain = Gray.(brain)
@show size(brain)
brain
```

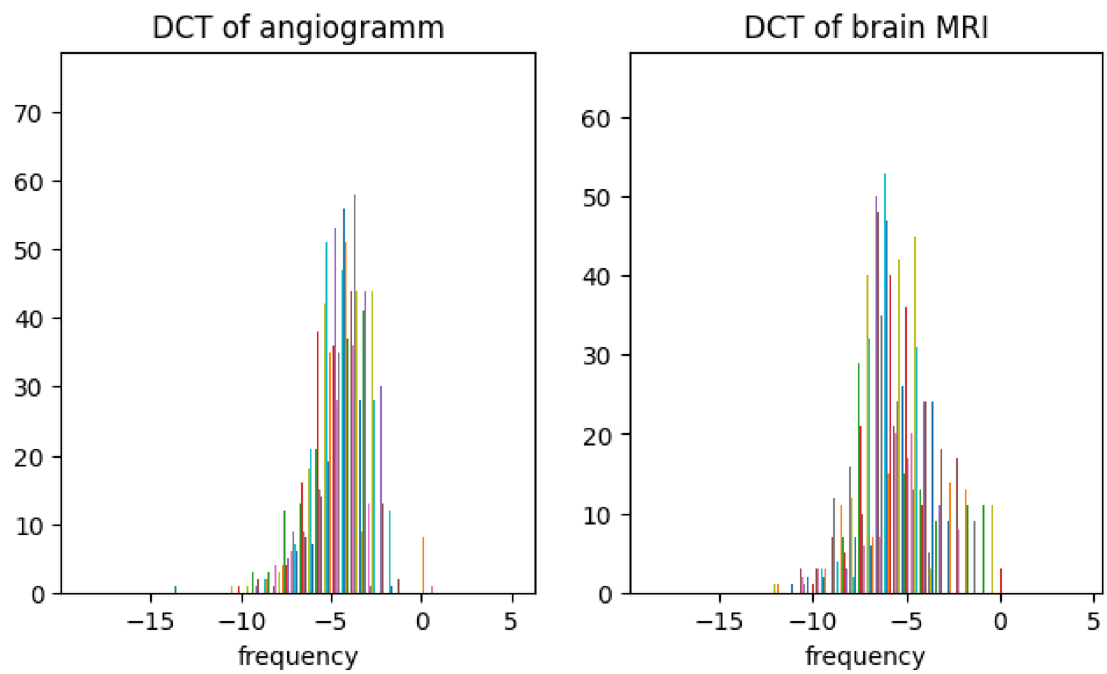
```
size(brain) = (302, 280)
```

[15]:



[16]:
```julia
angiogram = convert(Array{Float64}, angiogram);
brain = convert(Array{Float64}, brain);
#Fourier Transform angiogram
fourier_angio = FFTW.dct(angiogram) ;
fshift_angio = FFTW.fftshift(fourier_angio)
magnitude_angio =  log.(abs.(fshift_angio));
#Fourier Transform brain
fourier_brain = FFTW.dct(brain) ;
fshift_brain = FFTW.fftshift(fourier_brain)
magnitude_brain =  log.(abs.(fshift_brain));
```

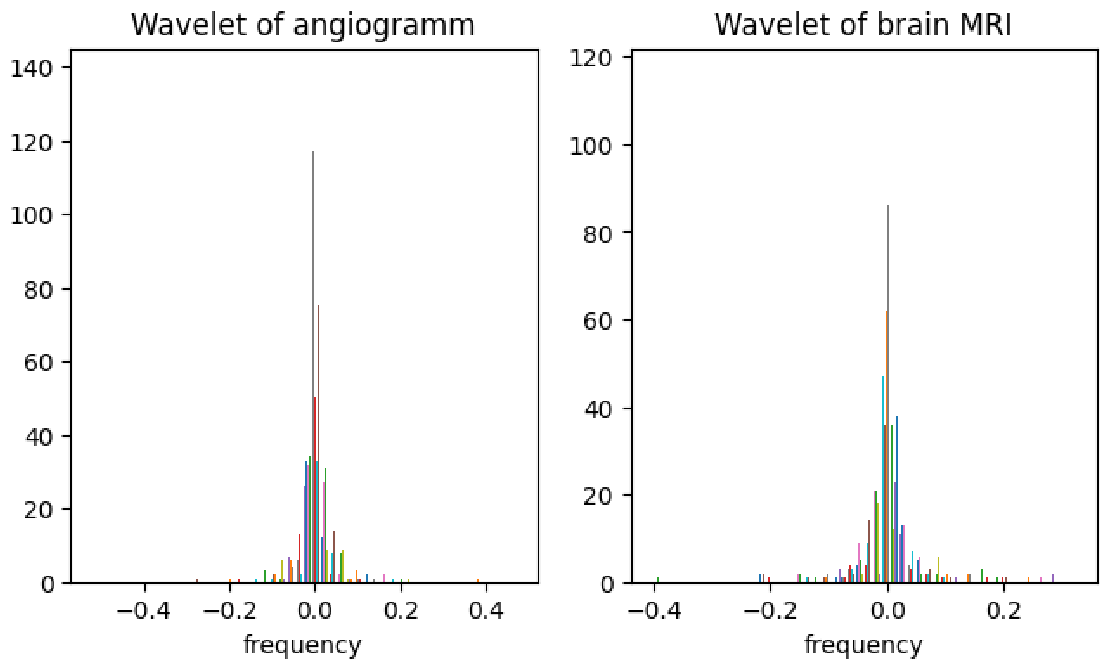
[17]:
```
fig = figure(figsize=(16,4))
fig.add_subplot(141)
hist(magnitude_angio, bins = 50)
title("DCT of angiogramm")
xlabel("frequency")
fig.add_subplot(142)
hist(magnitude_brain, bins = 50)
title("DCT of brain MRI");
xlabel("frequency");
savefig("DCT")
```



[18]:
```
#wavelet transformation angiogram
coeffs2_angi = pywt.dwt2(angiogram, "bior1.3")
LL_ang, (LH_ang, HL_ang, HH_ang) = coeffs2_angi ;
#wavelet transformation brain MRI
coeffs2_brain = pywt.dwt2(brain, "bior1.3")
LL_brain, (LH_brain, HL_brain, HH_brain) = coeffs2_brain ;
```

[19]:
```
fig = figure(figsize=(16,4))
fig.add_subplot(141)
hist(LH_ang+HL_ang+HH_brain, bins = 50)
title("Wavelet of angiogramm");
xlabel("frequency");
fig.add_subplot(142)
hist(LH_brain+HL_brain+HH_brain, bins = 50)
```

```
title("Wavelet of brain MRI");
xlabel("frequency")
savefig("Wavelet")
```



For a sparse representation, it is essential how much information is stored in a limited amount coefficiants. Intuitively, the angiogramm is better suited for a sparse representation instead of the brain mri because the region of interest is concentrated on the small vessels.
As shown in the histograms for the furier coefficiants, the coefficiants of the angiogram are more concentrated while the coefficiants of the brain mri show a higher variance.
In comparison to the fourier coefficiants, the wavelet coefficiants of both images are much more dense. Since, all histograms follow approximately a gaussian distribution, the one for the wavelet transformation have much smaller variance than the histograms for the fourier transformation.
Therefore, the wavelet transformation is better suited for sparse representation.