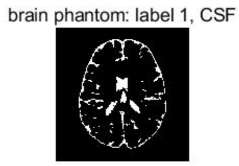
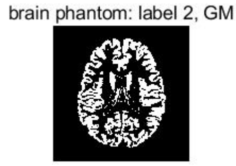
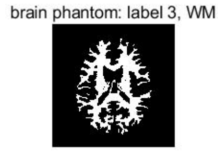
1. Inspect the digital brain phantom







T1 CSF = 2569ms

T2 CSF = 329ms

SD CSF = 1

T1 GM = 833ms

T2 GM = 83ms

SD GM = 0.86

T1 WM = 500ms

T2 WM = 70ms

SD WM = 0.77

2.Simulate MR image contrast from pulse sequences

2.1.Spin Echo Proton density weighted (PDw):

The table below is the definition of various contrast weighting.

TABLE 14.4

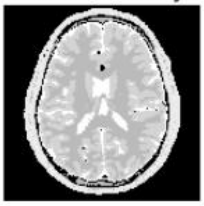
Combinations of TE and TR Values Used to Generate Various Contrast Weightings in Spin Echo Imaging

	Short TE ($\leq$ 20 ms)	Long TE ($\geq$ 80 ms)
Short TR (<700 ms)	T_1-weighted	Not commonly used
Long TR (>2000 ms)	Proton density–weighted	T_2-weighted

The results are as follow:

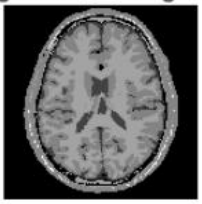
```
TR = 12500
TE = 5
```

Signal of Proton density weighted



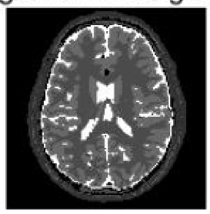
```
TR = 500
TE = 10
```

Signal of T1 weighted



```
TR = 5000
TE = 100
```

Signal of T2 weighted



2.2 FLAIR sequence:

TI =log(2)*T1_csf. This is T2-weighted-FLAIR. The definition of parameters and the image are as follow.

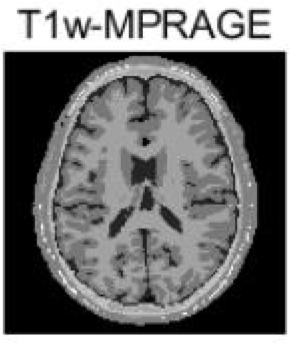
```
TR = 7000
TE = 100
TI = 1.7807e+03
```

T2w-FLAIR

2.3 MPRAGE

The detailed initialization refer to the code below.
Use the signal function of MPRAGE to calculate:

```
S = ph.sd.*(sin(alpha/180*pi*B1)).*RP.*Q;
```



3.1 Add noise to the simulated data

1). calculate the variance of noise according to the definition of SNR

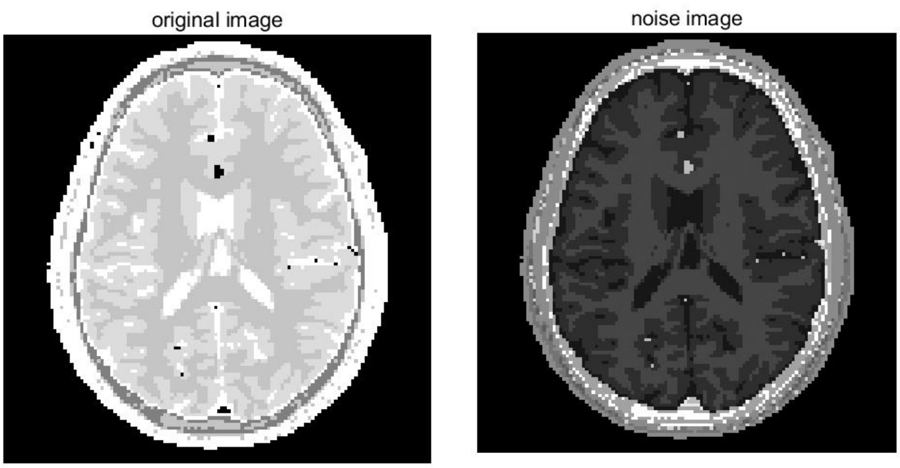
```
kspace = fft2(ph.label)
SNRdB = 50;
var_noise = norm(kspace,2)/10^(SNRdB/10)
```

2). create nois_ kspace and add to kspace

```
noise_real = normrnd(0,sqrt(var_noise),size(kspace,1),size(kspace,2))*mean(kspace(:))
noise_imag = normrnd(0,sqrt(var_noise),size(kspace,1),size(kspace,2))*mean(kspace(:))
kspace_noise = kspace + noise_real + noise_imag*1i;
```

3). inverse fft

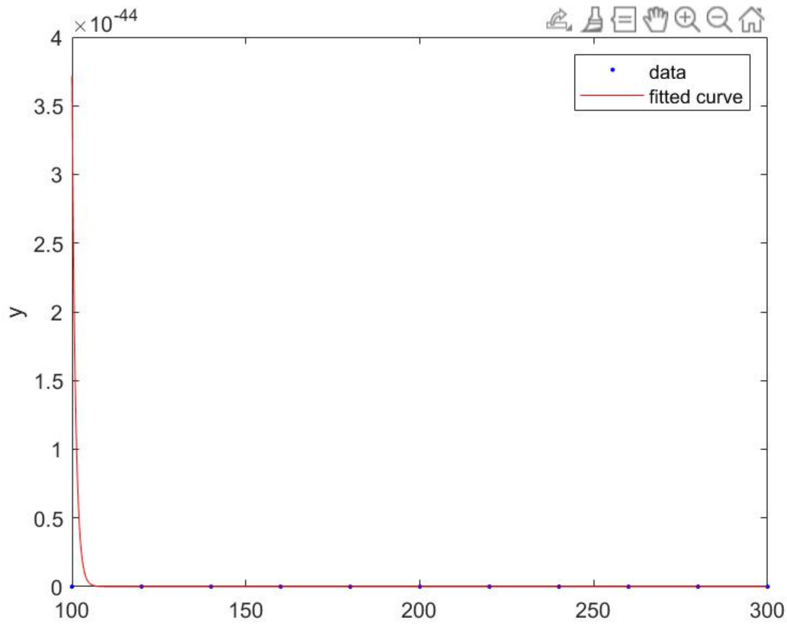
```
noise_img = ifft2(kspace_noise)
```



3.2 T2 mapping for CSF, GM and WM

3.2.1 TE is 11. The more TE, the more precise the fitting function is.

The fitting result is as followed. T2 is the x value, mapping to 63% of the maximal y value.



Appendix: code

```
%%2.1 Simulate MR image contrast from pulse sequences
%M0 = ph.sd
%Proton density weighted (PDw), longTR, shortTE
TR = 12500
TE = 5
S = ph.sd.*(1-2*exp(-(TR-TE/2)./ph.t1) + exp(-TR./ph.t1)).*exp(-TE./ph.t2);
figure
imshow(S,[]); title('Signal of Proton density weighted');

%T1 weighted (T1w), shortTR, shortTE
TR = 500
TE = 10
S = ph.sd.*(1-2*exp(-(TR-TE/2)./ph.t1) + exp(-TR./ph.t1)).*exp(-TE./ph.t2);
figure
imshow(S,[]); title('Signal of T1 weighted');

%T2 weighted (T2w), longTR, longTE
TR = 5000
TE = 100
S = ph.sd.*(1-2*exp(-(TR-TE/2)./ph.t1) + exp(-TR./ph.t1)).*exp(-TE./ph.t2);
figure
imshow(S,[]); title('Signal of T2 weighted');
```

```
%%2.2 FLAIR sequence--T2-weighted-FLAIR
% matlab: .*pixel-wise multiply   * matrix multiply
% python: * pixel-wise multiply   .dot() matrix multiply
TR = 7000
TE = 100
TI = log(2)*T1_csf
S = ph.sd.*(1-2*exp(-TI./ph.t1)+exp(-TR./ph.t1)).*exp(-TE./ph.t2);

figure
imshow(S,[]); title('T2w-FLAIR');
```

```
%%2.3 MPRAGE--T1-weighted-MPRAGE
% syms E1 E2 E3 E4 T1_star a

TR = 2420; TE = 3.68; TI = 960;
ES = 10.2; %  time difference between excitation pulses
alpha = 9;
n1 = 80; % number of echo
N2D = 224; N3D = 160;
tao1 = TI - ES*n1;
tao2 = TR - tao1 -ES*N3D;
B1 = 1;
RP = 1; % receive sensitivity profile of  RF coil.
T1_star = 1./(1./ph.t1 - 1/ES*log(cos(alpha/180*pi*B1)));
E1 = exp(-tao1./ph.t1);
E2 = exp(-tao2./ph.t1);
E3 = exp(-(N3D*ES)./T1_star);
E4 = exp(-(n1*ES)./T1_star);
q1 = E4.*(1-2*E1+E1.*E2);
q2 = T1_star./ph.t1.*(1+E1.*E2.*E3-E1.*E2.*E4-E4);
q3 = (1+E1.*E2.*E3);
Q = (q1+q2)./q3;
S = ph.sd.*(sin(alpha/180*pi*B1)).*RP.*Q;

figure
imshow(S,[]); title('T1w-MPRAGE');
```

```
%%3.1 Add noise to the simulated data
kspace = fft2(ph.label)
SNRdB = 50;
var_noise = norm(kspace,2)/10^(SNRdB/10)

noise_real = normrnd(0,sqrt(var_noise),size(kspace,1),size(kspace,2))*mean(kspace(:))
noise_imag = normrnd(0,sqrt(var_noise),size(kspace,1),size(kspace,2))*mean(kspace(:))
kspace_noise = kspace + noise_real + noise_imag*1i;
noise_img = ifft2(kspace_noise)

figure
imshow(ph.sd,[]); title('original image');
imshow(real(noise_img),[]); title('noise image');
```

```
%%3.2 T2 Mapping
T1_img = zeros(size(ph.label,1),size(ph.label,1))
% S = zeros(10,size(ph.label,1),size(ph.label,1))
TE = linspace(100,300,11);
TR = 5000;

for i = 1:length(TE)
    Signal(i,:,:) = ph.sd.*(1-2*exp(-(TR-TE(i)/2)./ph.t1) + exp(-TR./ph.t1)).*exp(-TE(i)./ph.t2);
end

for i =1:size(ph.sd,1)
    for j =1:size(ph.sd,2)
        for img = 1:length(TE)
           y(img) = Signal(img,i,j);
        end
        x = TE;
        f{i,j} = fit(x(:), y(:), ft, 'StartPoint', [1 1]);
    end
end
```