BME 595 - Fall 2023
Nicole Haderer (1005721444), Deniz Uzun (1006035005)

MAGNETIC RESONANCE IMAGING LAB #1

PART 1: SURVEY OF MRI SUITE

1. ***Make a crude sketch of the magnet and the magnetic field lines. On your sketch, indicate the relative density of the flux lines.***

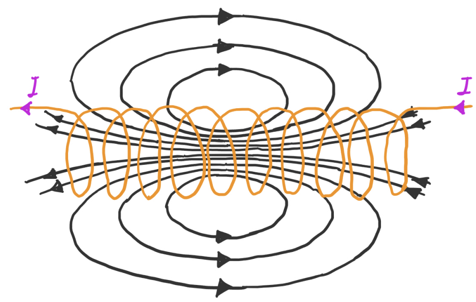

Figure 1. Sketch of a solenoid magnet and the magnetic field lines.

To create a magnetic field (shown in black in Figure 1), a current "I" is run through the solenoid wire (shown in orange in Figure 1). The magnetic field of the solenoid is the superposition of the fields due to the current through each coil.

2. ***Are ferromagnetic objects attracted more strongly near a) the opening to the bore, or b) at the exact center of the bore? Why?***
Ferromagnetic objects are attracted more strongly at the exact center of the bore rather than at the opening. This can be seen in Figure 1, where the magnetic flux lines are the most dense (closest together) at the center of the solenoid coil and become less dense (farther apart) at either ends. The increased density and uniformity in the space between the field lines at the center of the bore indicates a stronger magnetic field, and therefore attraction of ferromagnetic objects.

3. ***How does the magnet stay superconducting?***
The solenoid coils in superconductive MRI magnets are made from alloys that have zero electrical resistance at temperatures around 10 K. First, the coil is connected to a power supply and the current through the coils gradually increases until the desired magnetic field is achieved in a process called siting. After the magnet is sited, it is cooled to superconductive temperatures by filling the cryostat jacket with liquid helium at least up to the 75% level. The electrical terminals to charge the coil windings are located deep within the cryostat. The power supply is then removed from the coils, but the current continues to flow throughout the coil due to no electrical resistance making it superconductive. This overall process of ramping results in a constant magnetic field from the superconductive solenoid magnet.

4. ***What are the main advantages and disadvantages of using surface coils?***
The main advantage of using small surface coils is that they produce a higher signal-to-noise ratio (SNR) than would be possible from a larger more distant coil. The disadvantage of using a surface coil is the non-uniformity of the signal. This is known as surface coil flare, an overall bright surface signal caused by the enhancement of superficial signals and diminishing of deeper signals. This

typically requires applying post-processing correction filters to the images to adjust for low spatial frequency intensity modulations, accentuate image contrast, and reduce noise. The required intensity corrections are generally given by the coil's vendor, but should only be utilized for qualitative rather quantitative analysis.

5. ***In what plane would a flat surface coil NOT receive an MRI signal and why? Which of Maxwell's laws is relevant here? Assume the coil lies flat on the scanner bed, the z-direction is parallel to the magnet bore, the x-direction is horizontal, and y-direction is vertical.***
   An MRI signal is the electric signal that gets induced in the coil when the magnetic flux through coil changes in accordance with Faraday's Law (of which Maxwell's Third Equation is derived). Since the coil lies flat in the x-z plane, a change in magnetic flux perpendicular to this plane (i.e. in the y-direction) is required to induce a current in the coil. As the magnetic field is in the z-direction, moving the coil on the x-y plane will not change the z-component and no MRI signal will be received.

PART 2: IMAGE CONTRAST

1. ***Load the file "mr_lab.mat" using the command "load mr_lab". This file contains two matrices I1 and I2. These represent two objects with different proton densities. I1 has a proton density of 100, and I2 has a proton density of 50. Generate an image containing both objects with the command "I3=I1+I2". Now, add Gaussian noise of magnitude 5 to this image with the command "I3=I3+randn(size(I3)).*5". Display I3. Consider I3 to represent a proton-density-weighted image. What is the SNR of each of the two squares in the image? What is the CNR between the two squares?***

   The code shown in Figure 2 outlines the process used to generate the image of I3 and determine the SNR of the two square objects as well as the CNR between them.

```
load mr_lab

%PART 2 - IMAGE CONTRAST
%Generate an image containing both objects
I3 = I1+I2;
I3 = I3+randn(size(I3)).*5; %Add Gaussian noise of 5
imagesc(I3);

%Q1
y=imcrop(); %crop top left (yellow) box - I1
t=imcrop(); %crop bottom right (teal) box - I2
b=imcrop(); %crop blue area (noise)

%Calculate the means
y_mean = mean2(y); %I1
t_mean = mean2(t); %I2

%Calculate the standard deviation of the noise
b_std = std2(b);

%SNR - Signal to Noise Ratio of each square
y_snr = y_mean/b_std;
t_snr = t_mean/b_std;

%CNR - Contrast to Noise Ratio of each square
cnr = (y_mean - t_mean)/b_std;
```

Figure 2. Screenshot of the MATLAB code for Part 2.1.

After adding the signal intensities of the I1 and I2 matrices to create I3, Gaussian noise of 5 was applied and the image describing I3 was generated using the "imagesc()" function. This image is shown in Figure 3.

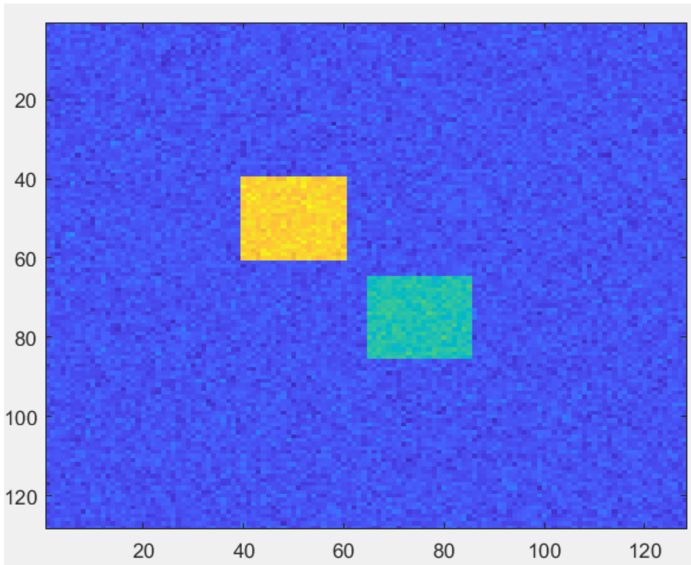

Figure 3. Display of I3 with Gaussian noise of magnitude 5. The yellow coloured square on the upper left is the result of I1 and the teal coloured square on lower right is the result of I2.

The "imcrop()" function was then used to extract the signal intensities producing each of the squares in the image; these were saved to matrix variables "y" (yellow square) and "t" (teal square). The mean of the signal intensities for both of these squares was determined using the "mean2()" function. This was done for use in SNR calculations. To evaluate the noise, "imcrop()" was used again on a section of the image background and was saved as matrix "b". For both SNR and CNR calculations, the standard deviation of the background's signal intensity is required. Therefore, the standard deviation of "b" was determined using the "std2()" function. The mean signal intensities of the squares and the standard deviation of the noise determined from MATLAB are listed below.

- Mean of I1, y_mean = 100.2 (yellow square in Figure 3)
- Mean of I2, t_mean = 49.9 (teal square in Figure 3)
- Standard Deviation of I3's Gaussian noise, b_std = 5.01 (Note: This is consistent with the known Gaussian noise applied before generating the image.)

By substituting these values into the equations shown in Figure 2 and below, the SNR for each square and CNR between the two squares were calculated.

$$\text{SNR} = \frac{\textit{mean signal intensity of object}}{\textit{standard deviation of noise}}$$

$$\text{CNR} = \frac{\textit{mean signal intensity of object \#1} - \textit{mean signal intensity of object \#2}}{\textit{standard deviation of noise}}$$

<u>Results Summary:</u>

- **SNR of I1 = 20.0 (yellow square in Figure 3)**
- **SNR of I2 = 10.0 (teal square in Figure 3)**
- **CNR between I1 & I2 squares = 10.1**

2. ***Next, we will generate T2-weighted images using a TE of 20 ms. For I1, we will use a T2 value of 10 ms. Generate a new T2-weighted image (call it I4) by applying the appropriate weighting function to I1. For I2, we will use a T2 value of 50 ms. Generate a new T2-weighted image (call it I5) in the same manner. Generate an image containing both objects with the command I6=I4+I5, and add Gaussian noise of magnitude 5 in a similar manner to step #1. What is the SNR of each of the two squares in the image? What is the CNR between the two squares? How does the contrast between the two objects compare to the image in step #1?***

The code shown in Figure 4 outlines the process used to generate a T2-weighted version of I3 called I4 and determine the SNR of the two square objects as well as the CNR between them.

```
%Q2
%Generate T2-weighted Images at 10ms and 50ms, for TE = 20ms
I4 = I1 * exp(-20/10); %T2 = 10ms
I5 = I2 * exp(-20/50); %T2 = 50ms
imagesc(I4);
imagesc(I5);

%Image containing both
I6 = I4 + I5;
I6 = I6+randn(size(I3)).*5;
imagesc(I6);

y2=imcrop(); %crop top left (teal) box - I4
t2=imcrop(); %crop bottom right (yellow) box - I5
b2=imcrop(); %crop noise

%Calculate the means
y2_mean = mean2(y2); %I4
t2_mean = mean2(t2); %I5

%Calculate the standard deviation of the noise
b2_std = std2(b2);

%SNR
y2_snr = y2_mean/b2_std;
t2_snr = t2_mean/b2_std;

%CNR
cnr2 = (t2_mean - y2_mean)/b2_std;
```

Figure 4. Screenshot of the MATLAB code for Part 2.2.

To apply the T2-weighting to I1 and I2, the following function was used: $M = M_0$*exp(-TE/T2), where M_0 is the original matrix of signal intensities and M is the T2-weighted version. Therefore, to generate I4 and I5, the I1 and I2 matrices were scaled by a factor of exp(-20 ms/10 ms) and exp(-20 ms/50 ms) respectively. These matrices were added together to produce I6 and a Gaussian noise of 5 is applied. Using the “imagesc()” function, this produced the image shown in Figure 5.

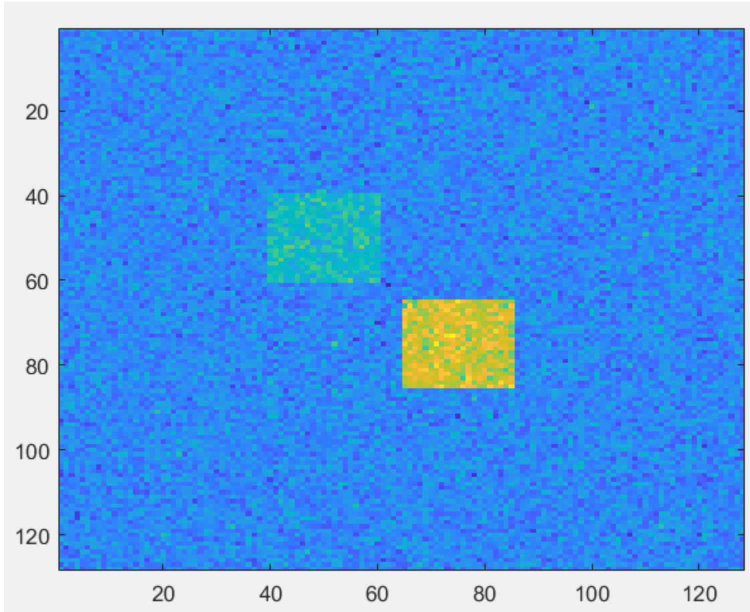

Figure 5. Display of I6 with Gaussian noise of magnitude 5. The teal coloured square on the upper left is the result of I4 and the yellow colored square on lower right is the result of I5.

Contrast and brightness are predominantly determined by the T2 properties of the object. As a result of T2-weighting the colours of the objects in Figure 5 are reversed and the contrast between the two objects are reduced in comparison to Figure 2.

The same procedure to determine the mean signal intensities of the squares and the standard deviation of the noise in Part 2.1 was repeated using the "imcrop()", "mean2()", and "std2()" functions. The same SNR and CNR equations were also utilized. The intermediate values and final results are listed below:

- Mean of I4, y2_mean = 13.3 (teal square in Figure 5)
- Mean of I5, t2_mean = 33.5 (yellow square in Figure 5)
- Standard Deviation of I6's Gaussian Noise, b_std = 5.1 (Note: This is consistent with the known Gaussian noise applied before generating the image.)

<u>Results Summary:</u>

- **SNR of I4 = 2.6 (teal square in Figure 5)**
- **SNR of I5 = 6.6 (yellow square in Figure 5)**
- **CNR between I4 & I5 = 4**

3. ***Let the proton densities of two objects be defined as PA and PB. Let the T2 values of the two objects be defined as T2_A and T2_B. Derive a general expression for the T2-weighted contrast. Now, derive an expression that indicates the echo time necessary to produce zero contrast between the two objects. Using the proton density and T2 values described in steps #1 and #2, calculate the TE that would produce zero contrast in this case. Generate an image in the same manner as step #2 to confirm your result.***

The expressions representing the T2-weighted versions of PA and PB can be generated using the same equation in Part 2.2: $M = M_0$*exp(-TE/T2). Substituting for PA and PB and their respective T2 values ($T2_A$ and $T2_B$) produces the following expressions for their T2-weighted versions, PA_{T2} and PB_{T2}.

- PA_{T2} = PA*exp(-TE/$T2_A$)
- PB_{T2} = PB*exp(-TE/$T2_B$)

Since contrast is the difference in signal intensities, T2-weighted contrast is the difference between the T2-weighted signal intensities, PA_{T2} and PB_{T2}:

Contrast = Signal 1 - Signal 2

T2-Weighted Contrast = T2-Weighted Signal 1 - T2-Weighted Signal 2
∴ **T2-Weighted Contrast = PA*exp(-TE/$T2_A$) - PB*exp(-TE/$T2_B$)**

To determine the TE necessary to produce zero contrast between two objects, the T2-Weighted Contrast term above is set to 0.

T2-Weighted Contrast = 0 = PA*exp(-TE/$T2_A$) - PB*exp(-TE/$T2_B$)

The equation is manipulated as follows to isolate for TE:

PA*exp(-TE/$T2_A$) = PB*exp(-TE/$T2_B$)
PA/PB = exp(-TE*(1/$T2_B$ - 1/$T2_A$))
ln(PA/PB) = -TE*(1/$T2_B$ - 1/$T2_A$)

$$\therefore TE = \frac{ln(\frac{PA}{PB})}{\frac{1}{T2_A} - \frac{1}{T2_B}}$$

Substituting the mean signal intensity for I1, 100.2, as PA, the mean signal intensity for I2, 49.9, as PB, 10 ms as $T2_A$, and 50 ms as $T2_B$, the TE required to produce zero T2-weighted contrast between the squares is calculated:

$$TE = \frac{ln(\frac{mean\ of\ I1}{mean\ of\ I2})}{\frac{1}{10\ ms} - \frac{1}{50\ ms}}$$
$$= 12.5\ ln(\frac{100.2}{49.9})$$
$$= 8.7 \text{ ms}$$

∴ A **TE of 8.7 ms** is required for zero T2-weighted contrast between the squares.

To confirm this result, a TE of 8.7 ms was used in the procedure outlined in Part 2.2 instead of the previous 20 ms; the code for this application is shown in Figure 6. The image generated from this is shown in Figure 7.

```
%Q3
PA = 100; %proton density of I1 & I4 (top left box)
PB = 50; %proton denstiy of I2 & I5 (bottom right box)

T2_A = 10; %T2 value of I1 & I4
T2_B = 50; %T2 value of I2 & I5

%To produce zero contrast - TE = 8.6643ms
I7 = I1 * exp(-8.6643/10);
I8 = I2 * exp(-8.6643/50);

I9 = I7 + I8;
I9 = I9+randn(size(I3)).*5;
imagesc(I9);
```

Figure 6. Screenshot of the MATLAB code for Part 2.3.

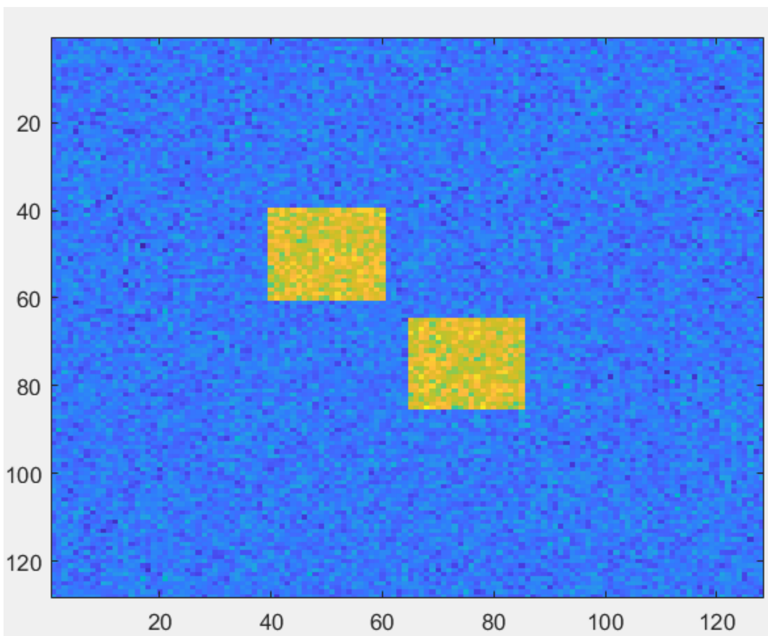

Figure 7. Display of I9 with Gaussian noise of magnitude 5. The square on the upper left is the result of I7 and the one on lower right is the result of the I8 matrix.

As seen above in Figure 7, there is no visible contrast between the squares when the T2-weighting uses an echo time of 8.7 ms in comparison with Figure 5 when the echo time was 20 ms. Both squares now have similar overall signal intensities.

PART 3: SNR TRADEOFFS

1. The calculated SNR is computed as follows:

$$SNR = mean\ intensity\ in\ ROI\ \div\ standard\ deviation\ in\ background$$

The SNR changes in proportion to the voxel volume and imaging time as follows:

$$SNR \propto voxel\ volume\ \times imaging\ time^{0.5}$$

Complete the following table

Image	Imaging parameter	Mean intensity in ROI	Standard deviation background	Computed SNR	Expected relative SNR	Voxel volume change	Imaging time change
1	Reference	780	20	39	1	1	1
2	$NSA \times 2$	800	12.7	63	$\sqrt{2} = 1.41$	1	2
3	$SL \div 2$	769	38	20	0.5	0.5	1
4	$FOV \times 2,$ $SL \div 2$	760	8.4	90	2	2	1
5	$N_{phase} = 128$	777	15	52	$\frac{3}{2}\sqrt{\frac{2}{3}}$ = 1.22	$\frac{192}{128} = \frac{3}{2}$	$\frac{128}{192} = \frac{2}{3}$

NOTE: for Images 1 to 4, $N_{phases} = 192$.

NSA stands for number of signal averages. SL stands for slice thickness. FOV stands for field-of-view (i.e. width of the entire image in the x-direction and y-direction). N_{phase} refers to the number of phase encode lines (i.e. number of voxels in the y-direction); the greater N_{phase}, the smaller the voxel thickness in the y-direction for the same FOV.

2. ***If we now take the computed SNRs and divide them by the SNR of the reference, we can compare the experimental relative SNRs to the expected relative SNRs. Complete the following table.***

Image	Experimental relative SNR	Expected relative SNR
1	1	1
2	1.62	1.41
3	0.51	0.5
4	2.31	2
5	1.33	1.22

Why do you think there are differences between the two columns? What factors may account for these differences?

In general, the experimental relative SNRs were slightly larger than the expected relative SNRs indicating there is a greater mean signal intensity of ROI, lower standard deviation of background noise, or both experimentally. One factor that may account for the disparity between the experimental and expected SNRs is the assumption that the noise is distributed equally across all frequencies in a Gaussian amplitude distribution. Experimentally, measurements of the ROI's signal intensities are performed on magnitude-reconstructed images. During this reconstruction, the pixels corresponding to noise are no longer Gaussian and may take other distributions form such as Rician or Raleigh. As such, correction factors would need to be applied to the experimental SNR's for a more accurate representation.

PART 4: QUANTITATIVE T1 AND T2 MEASUREMENT

1. ***The Matlab file 'T1andT2.mat' contains data acquired from a MRI contrast agent-doped solution for measuring T1 and T2 relaxation times. Load the data using the command "load T1andT2". T2 measurements were acquired from a single-slice multi-echo spin-echo sequence. Plot signal intensity acquired at each echo time (TE) by typing: plot(TE, T2 Signal, '.'); Knowing the relationship T2 Signal=ke$^{-TE/T2}$, use cftool to determine the line of best fit. Plot original data points and fitted lines. Estimate T2 and express in ms.***

   The code shown in Figure 8 outlines the process used to plot the T2 signal data against the echo time (TE) data and to determine the T2 value from the line of best fit.

```
load T1andT2.mat

%PART 4 - QUANTITATIVE T1 AND T2 MEASUREMENT
%Q1
%Plot signal intensity acquired at each echo time (TE)
plot(TE, T2signal, '.');

%T2signal=ke^(-TE/T2)
%Plot original datapoints and fitted line
%cftool;

%Estimate T2 and express in ms
T2 = 1/0.005136;
```

Figure 8. Screenshot of the MATLAB code for Part 4.1.

The T2 signal vector was plotted against the TE vector in MATLAB to produce Figure 9 below.

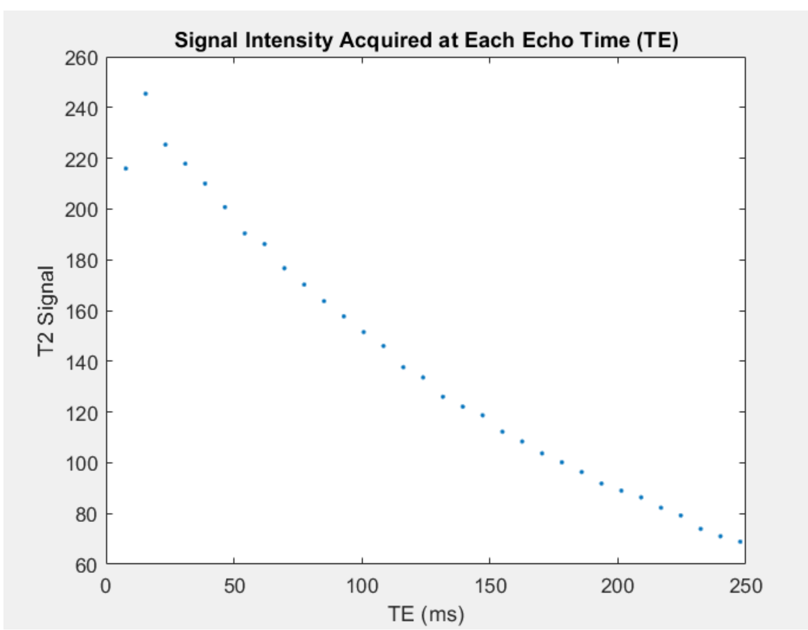

Figure 9. T2 signal intensity acquired at each echo time (TE).

Using MATLAB's "cftool", a one-term exponential line of best fit was determined as shown in Figure 10. This type of line, f(x) = a*exp(b*x), was selected as it takes the form of the known relationship between the T2 signal and echo time: T2 Signal = $ke^{-TE/T2}$.

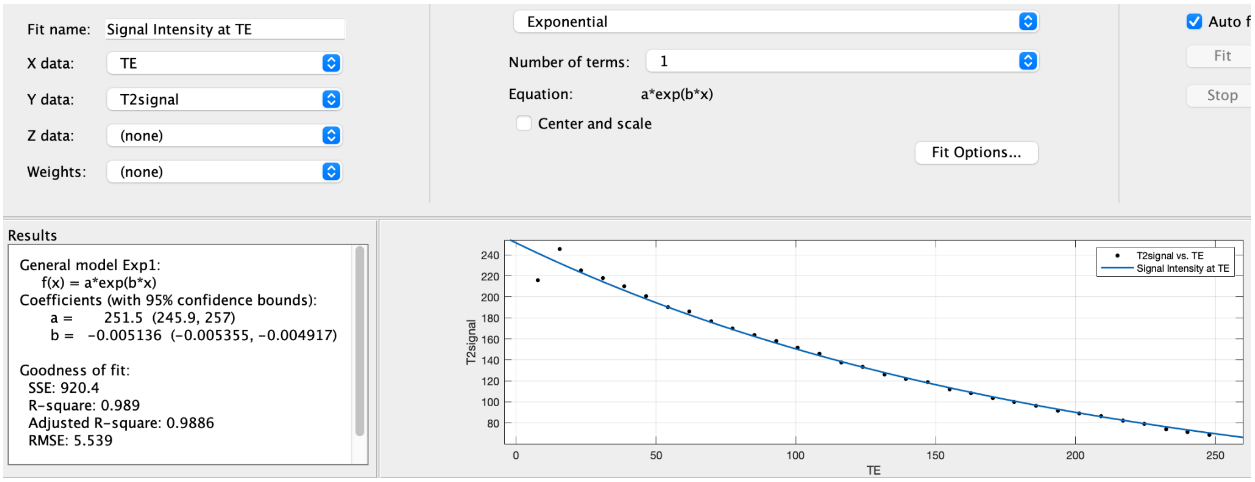

Figure 10. Using cftool to determine the best fit based on T2 Signal = $ke^{-TE/T2}$.

From Figure 10, the equation of the fitted line was determined to be f(x) = $251.5*e^{-0.005136x}$. This equation has strong correlation with the data points as it has an R^2 value of 0.989 (the closer the R^2 value is to 1, the better the quality of fit). The T2 value can be estimated using the expression for T2-weighted signals in the following way:

f(x) = T2 Signal, x = TE ⇒ T2 Signal = $251.5*e^{-0.005136*TE}$

For the above equation to take this form: T2 Signal = $ke^{-TE/T2}$

k = 251.5

-1/T2 = -0.005136 ms^{-1} (units are ms^{-1} since TE is in units of ms)

∴ T2 can be estimated to be 1/(0.005136 ms^{-1}) = **194.7 ms**

2. ***T1 measurements were acquired from a single-slice inversion-recovery spin-echo sequence. Plot signal intensity acquired at each inversion time (TI) by typing: plot(TI, T1signal, '.'); Notice that the signal intensity for the "early" points should be negative but are displayed as positive. Change the sign (invert) for these early points. Plot the original data points and the corrected data points. Knowing the relationship T1 Signal=k(1 - 2$e^{-TI/T1}$ + $e^{-TR/T1}$) and TR=3000ms, determine the line of best fit. Plot corrected data points and fitted line. Estimate T1 and express in ms.***

The code shown in Figure 11 outlines the process used to correct and plot the T1 signal data against the inversion time (TI) data and determine a line of best fit.

```
%Q2
%Invert first 2 points
T1signal(1) = -T1signal(1);
T1signal(2) = -T1signal(2);

T1_original = abs(T1signal);

%Plot original datapoint
plot(TI, T1_original, '.');

%Plot corrected datapoint
plot(TI, T1signal, '.');

cftool;
```

Figure 11. Screenshot of the MATLAB code for Part 4.2.

The original T1 signal vector was plotted against the TI vector in MATLAB to produce Figure 12 below.

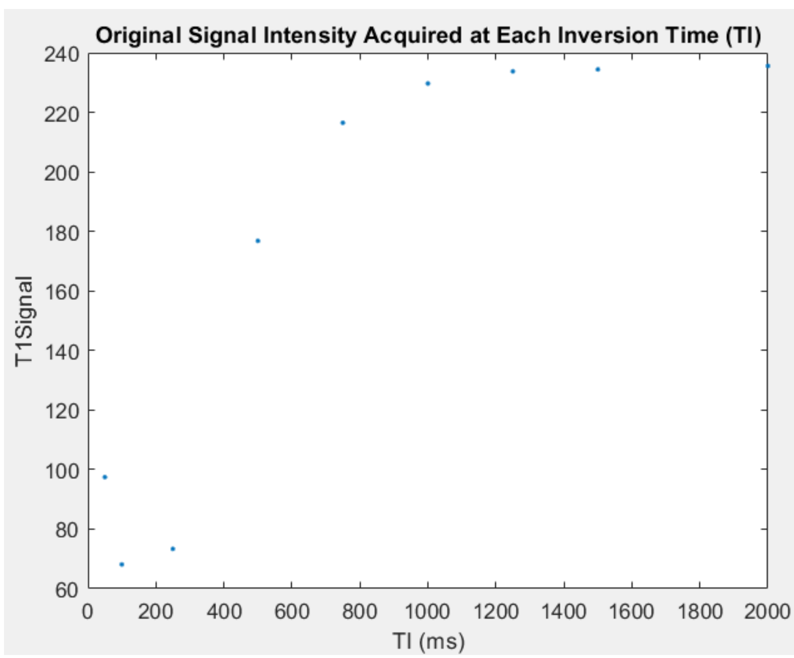

Figure 12. T1 Signal intensity acquired at each inversion time (TI) with original data points.

As shown in Figure 12, the first two data points are displayed as positive T1 signals but should be negative in keeping with the data trend. These data points were corrected by a factor of -1 as shown in Figure 11 and the data was replotted as displayed in Figure 13.

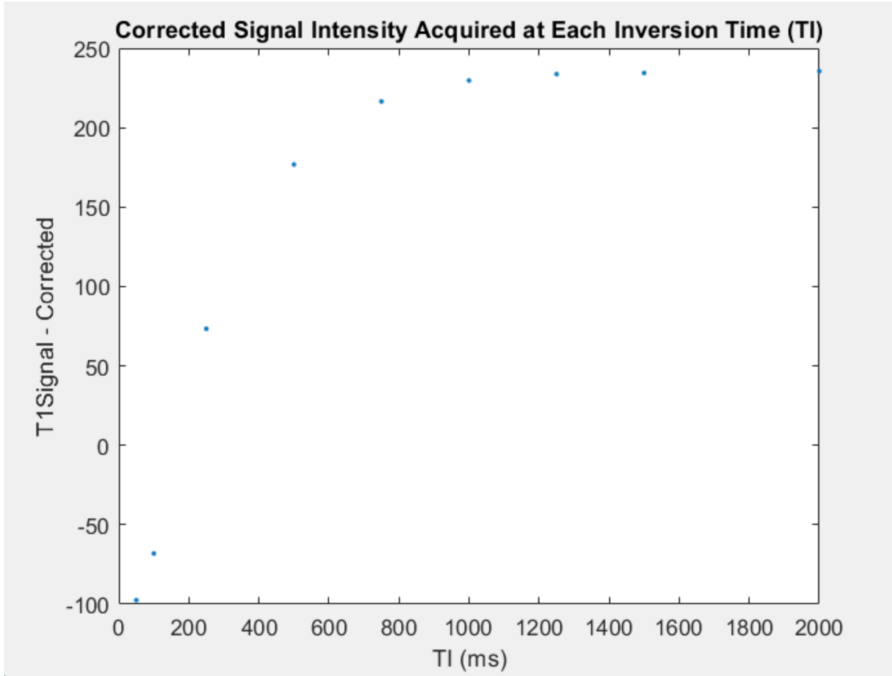

Figure 13. T1 Signal intensity acquired at each inversion time (TI) with corrected data points.

Using MATLAB's "cftool", a custom line of best fit was determined as shown in Figure 14. This custom equation, f(x) = a*(1-2*exp(-x*b)+exp(-3000*b)), is based on the known relationship between T1 signal and TI: T1 Signal = $k(1 - 2e^{-TI/T1} + e^{-TR/T1})$, where the repetition time (TR) is 3000 ms for this data set.

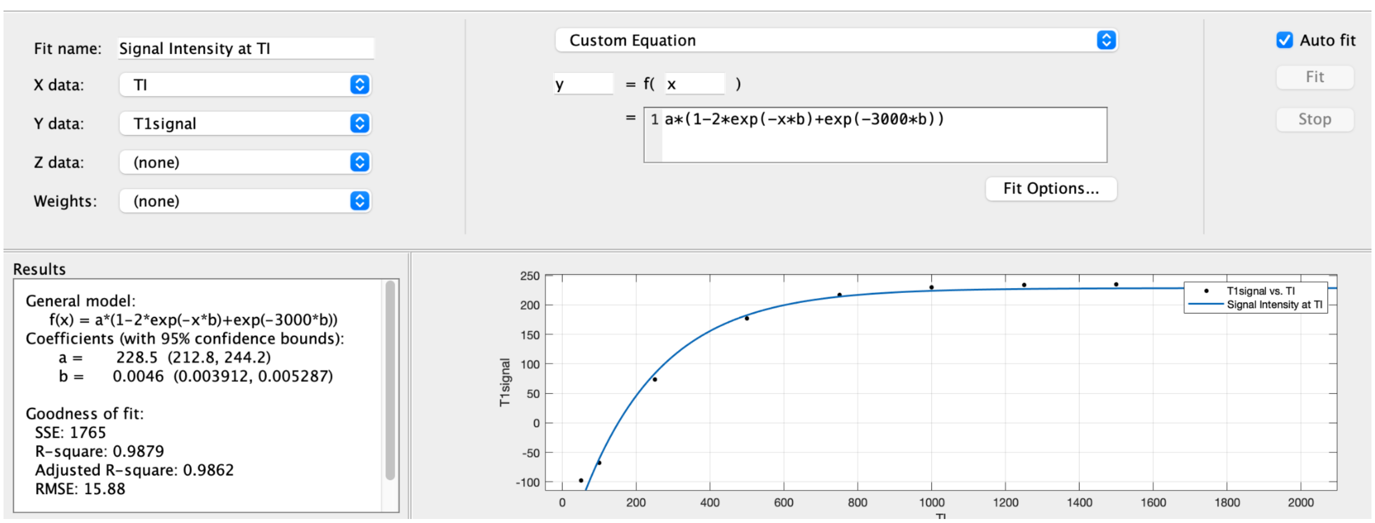

Figure 15. Using cftool to determine the best fit based on T1 Signal = $k(1 - 2e^{-TI/T1} + e^{-TR/T1})$ for the corrected data points.

Please note the coefficient bounds required adjustment from the default ranges of (-Inf, Inf) to produce approximate values within the range of the dataset. As such, the bounds for coefficient "a" were changed to [0, 400] and those for "b" were changed to [0, Inf). This is shown in Figure 16.

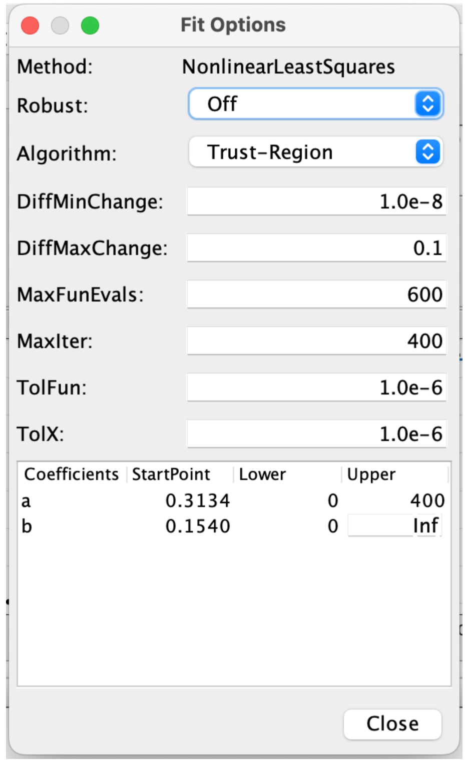

Figure 16. Fit options adjustments for the curve fitting in Figure 15.

From Figure 15, the equation of the fitted line was determined to be f(x) = 228.5*(1-2*exp(-x*0.0046)+exp(-3000*0.0046)). This equation has a strong correlation with the data points as it has an R^2 value of 0.988 (the closer the R^2 value is to 1, the better the quality of fit). The T1 value can be estimated using the expression for T1-weighted signals in the following way:

f(x) = T1 Signal, x = TI ⇒ T1 Signal = 228.5*(1-2*exp(-TI*0.0046)+exp(-3000*0.0046))

For the above equation to take this form: T1 Signal = $k(1 - 2e^{-TI/T1} + e^{-TR/T1})$

k = 228.5

TR = 3000 ms

1/T1 = 0.0046 ms^{-1} (units are ms^{-1} since TR is in units of ms)

∴ T1 can be estimated to be 1/(0.0046 ms^{-1}) = **217.4 ms**