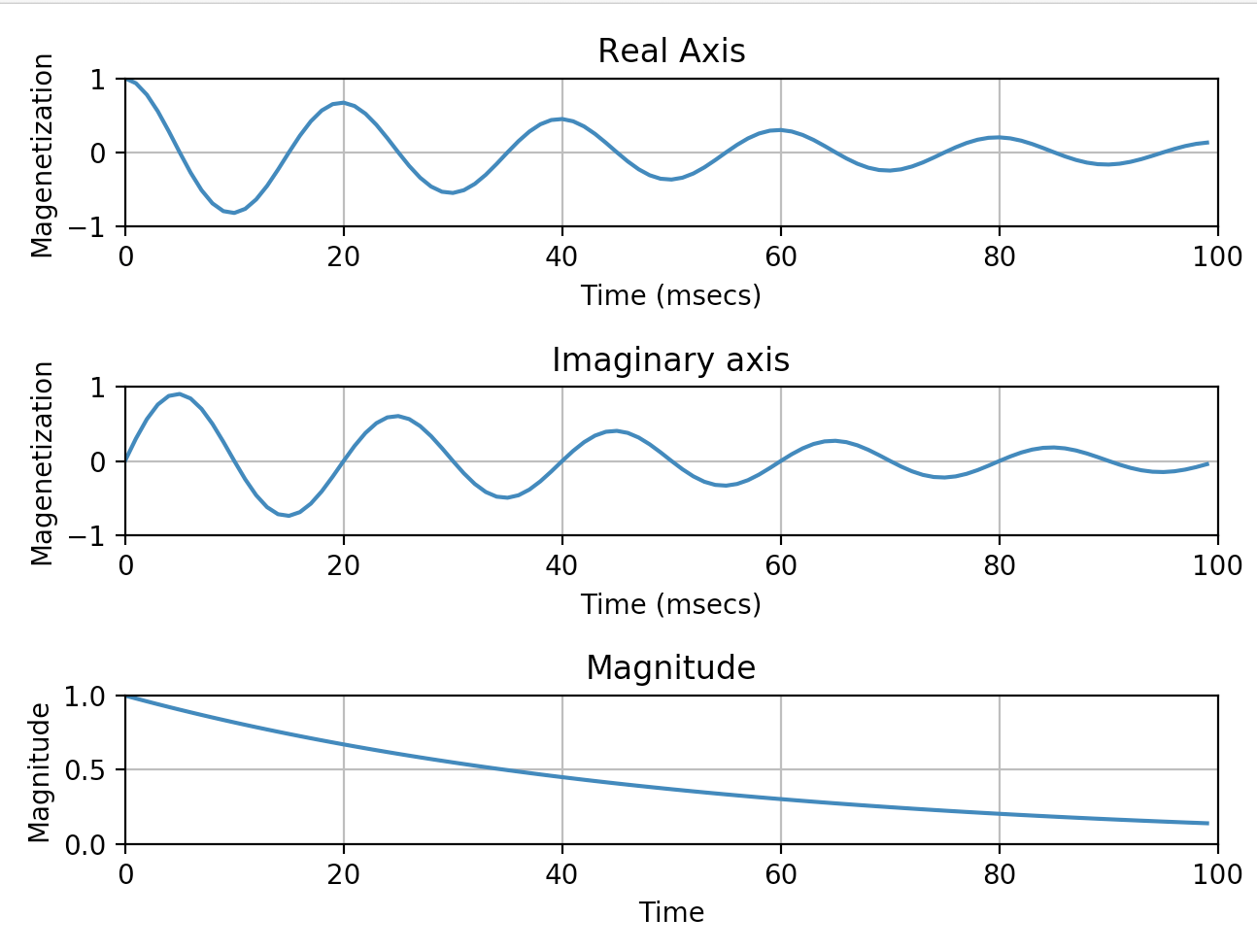
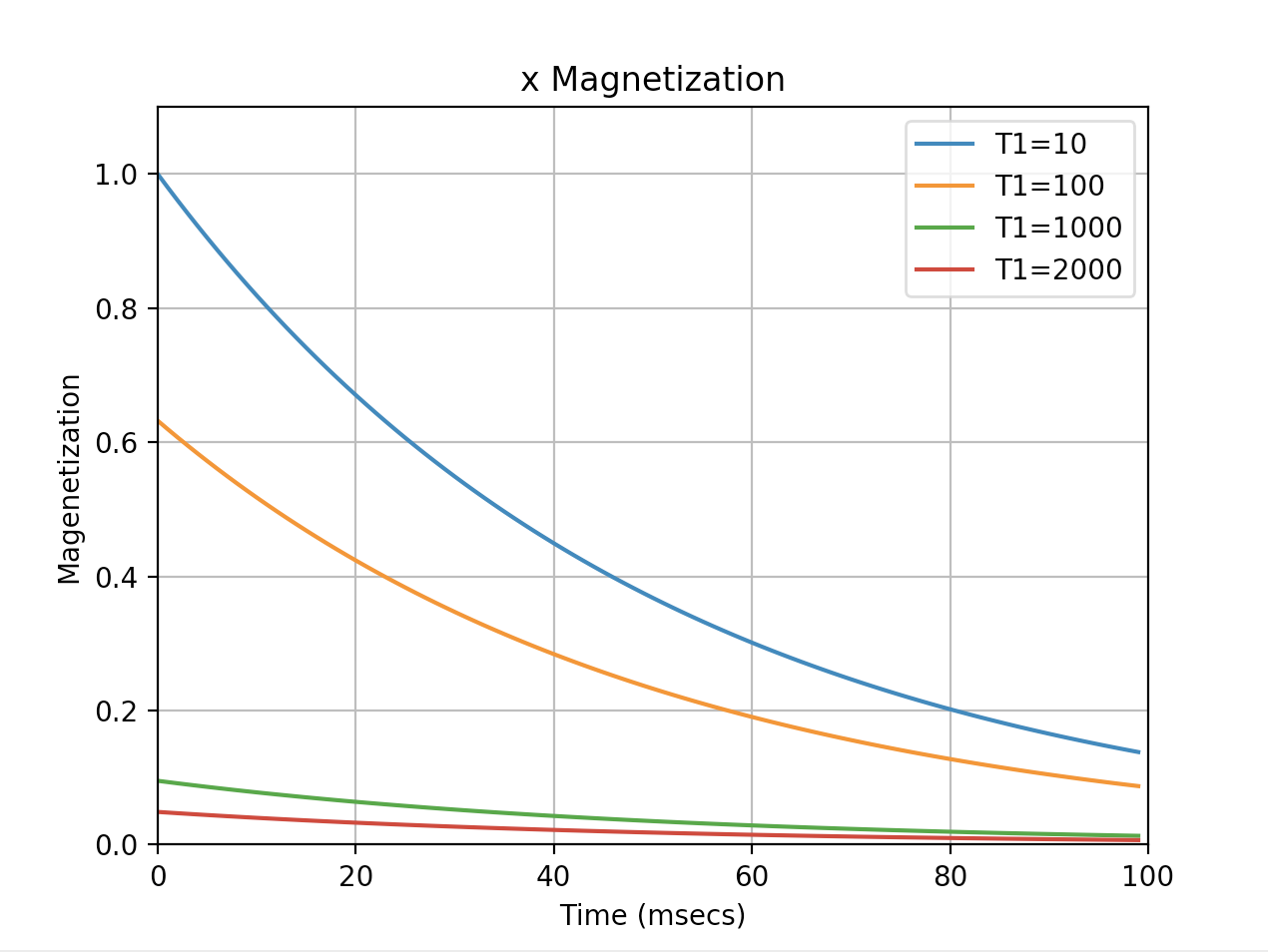
Python/Matlab Program Explanations

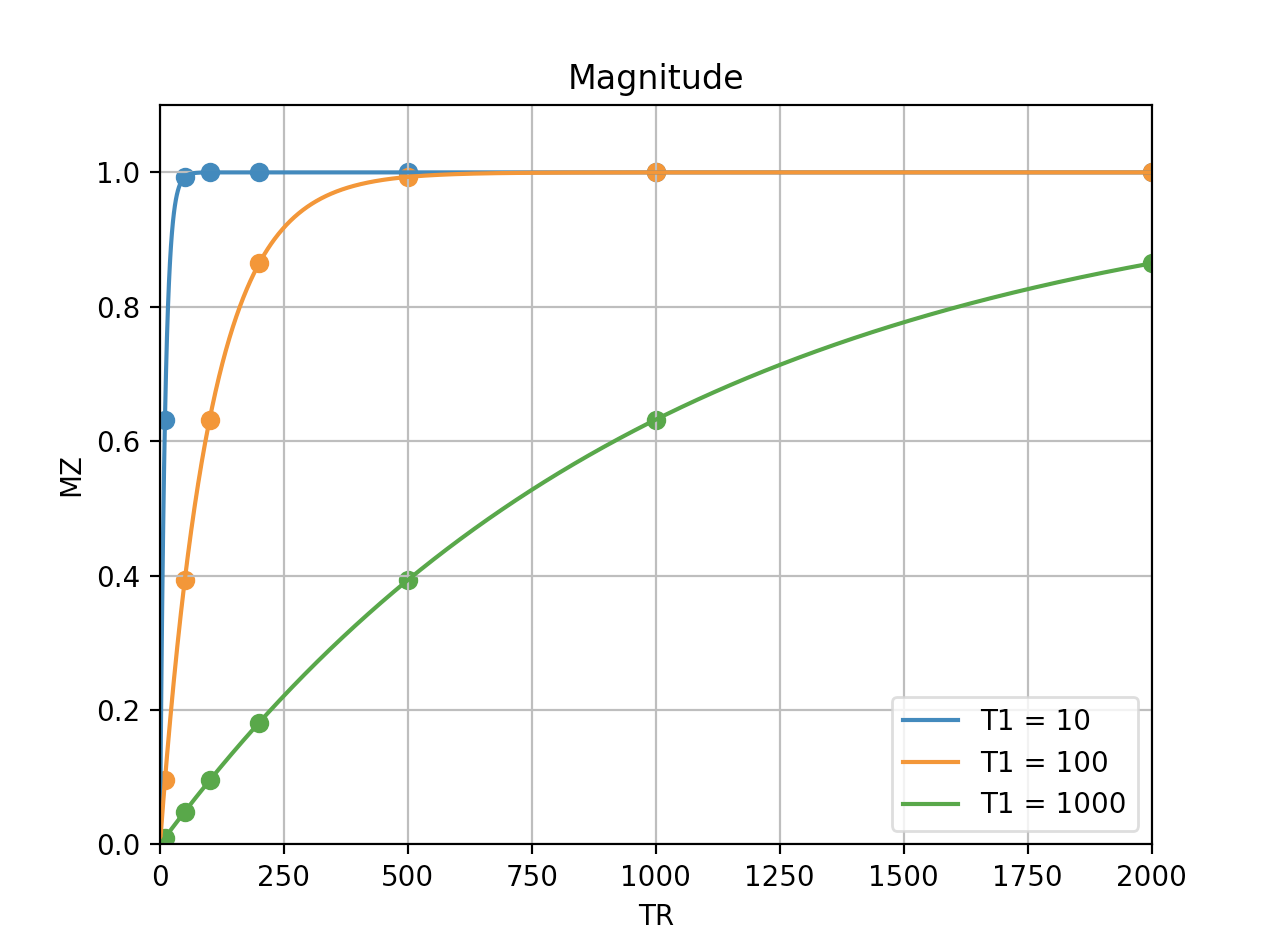
1. T2PulseAcquire plots the magnetization in the xy plane after the 90-degree pulse sequence between t=0 and t=100 msecs. Run the program and enter the desired frequency in Hz (0, 50, and 100 are suggested) and T2 (1, 10, 50, and 100 are suggested) in msecs by typing the number as an integer into the prompt and clicking enter. The first graph uses the given information to plot the following function on the real axis: . The graph below it is the same but plots on the imaginary axis using . The t in both equations are divided by 1000 because the t is in msecs but omega is in rad/sec. The third graph on the bottom plots the magnitude by taking the sum of the squares of both of the previous functions. The equation for that is . A frequency of 50 Hz and T2 of 50 msecs results in:



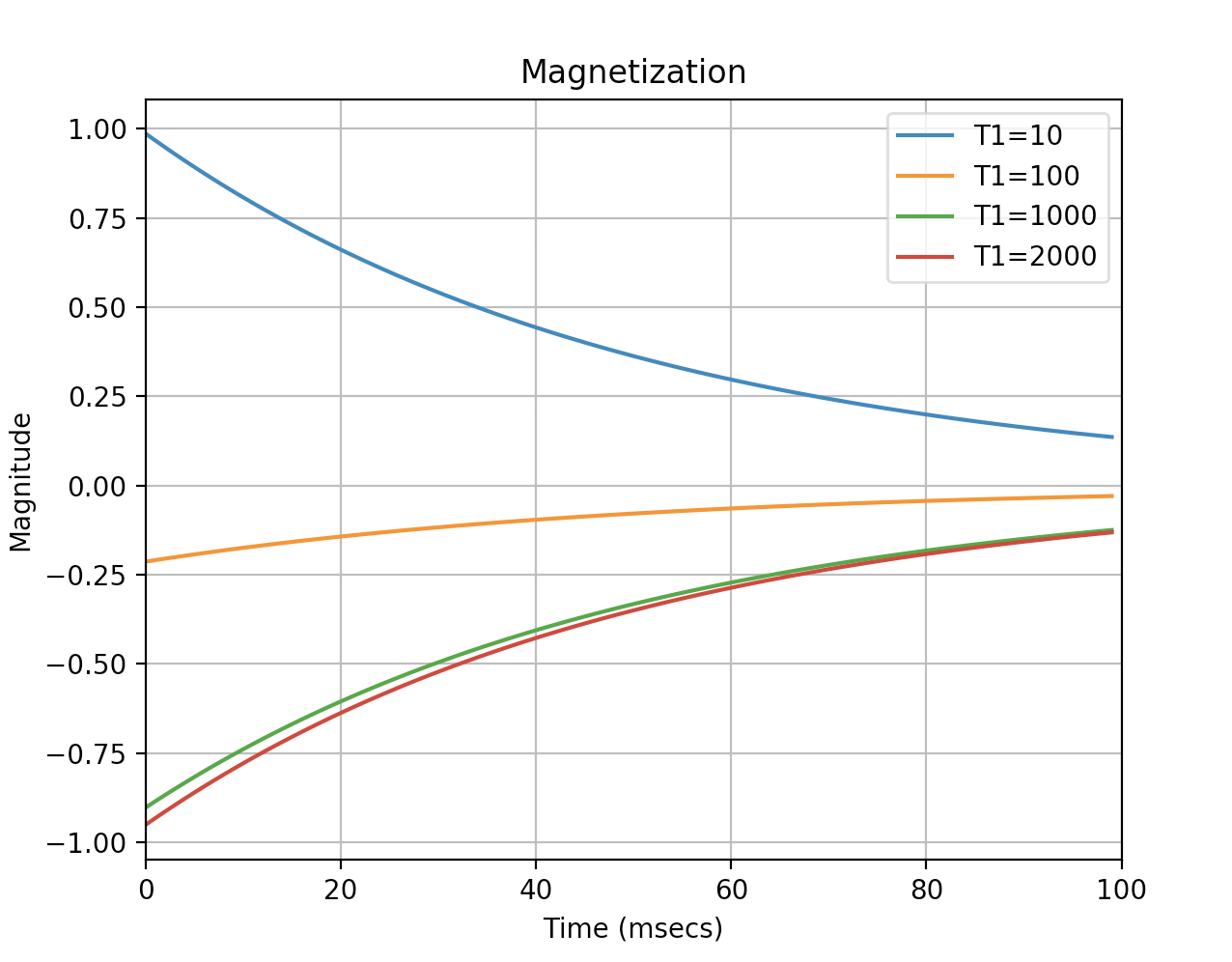
1. T1SaturationRecovery plots the x magnetization of saturation recovery after a pulse between t=0 and t=100 msecs. Run the program and enter the desired TR value in msecs by typing in the number as an integer and clicking enter. Using the equation, , with T2 constant at 50 msecs, the graph shows the response when T1 = 10, 100, 1000, and 2000 msecs. A TR of 50 should look like this:



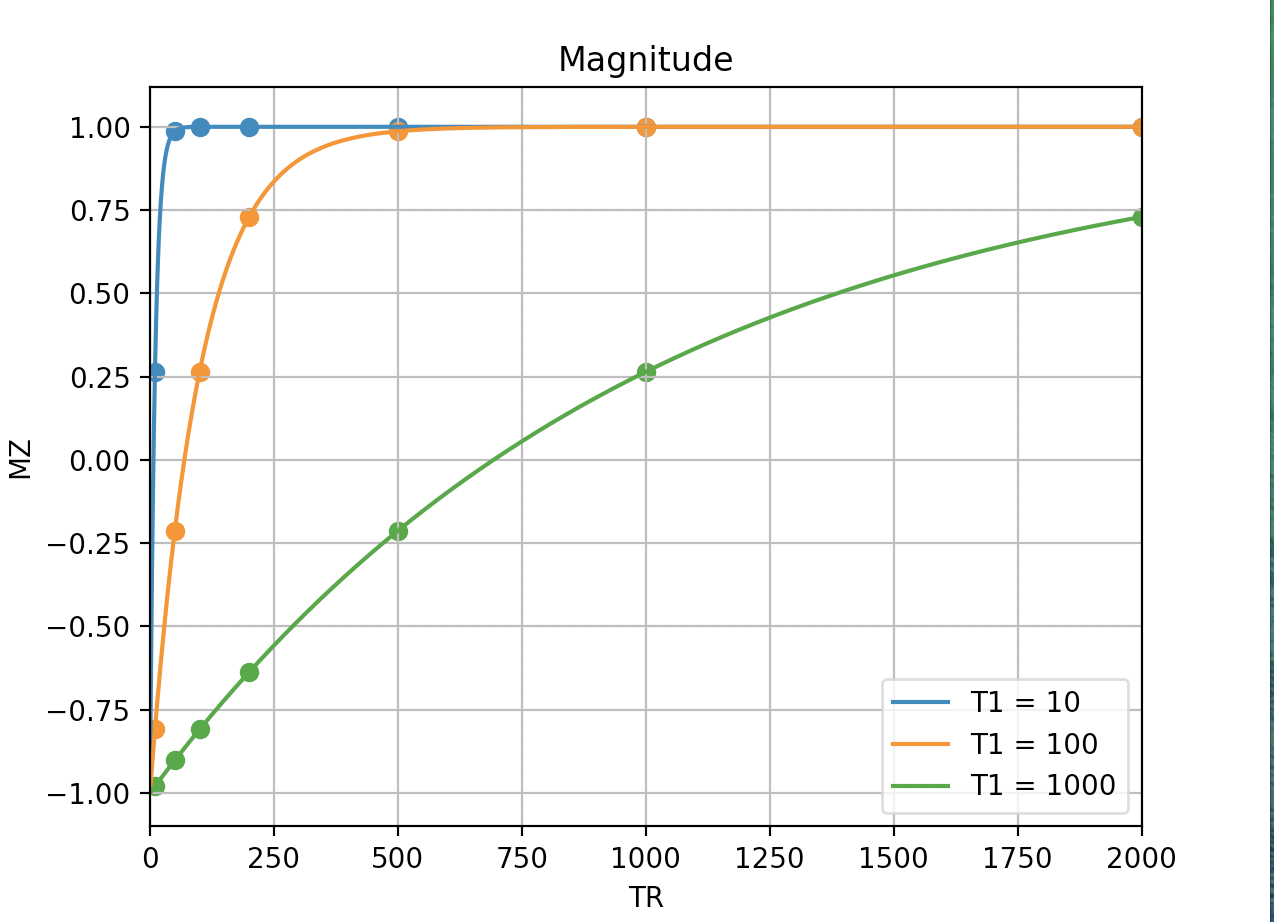
1. T1Satpt2 plots the saturation recovery magnetization absolute magnitude right after the 90-degree pulse (when t=0). After running the program, it graphs for T1 = 10, 100, and 1000 msecs. Furthermore, it highlights the points where TR = 10, 50, 100, 200, 500, 1000, 2000 msecs. The resulting graph is:

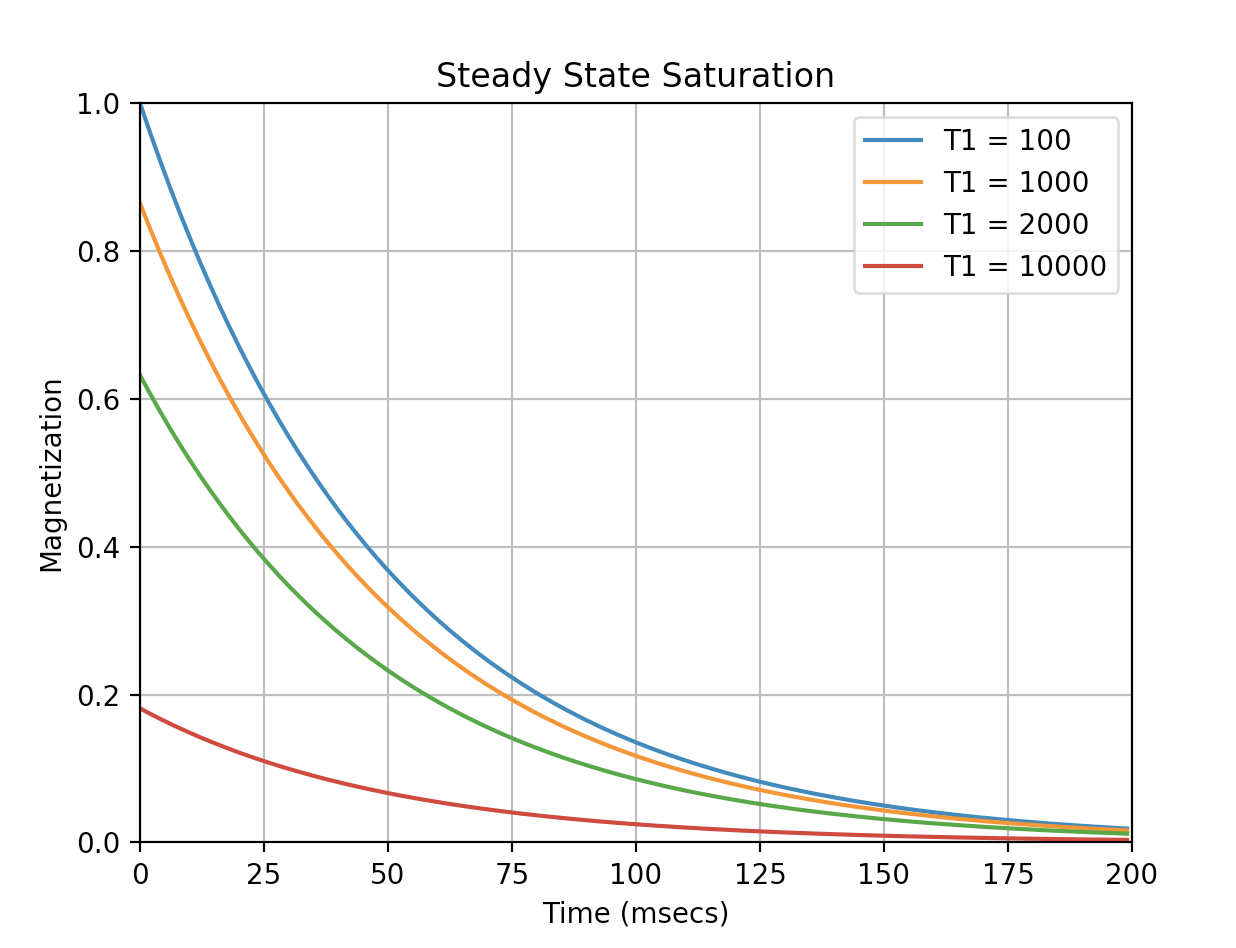


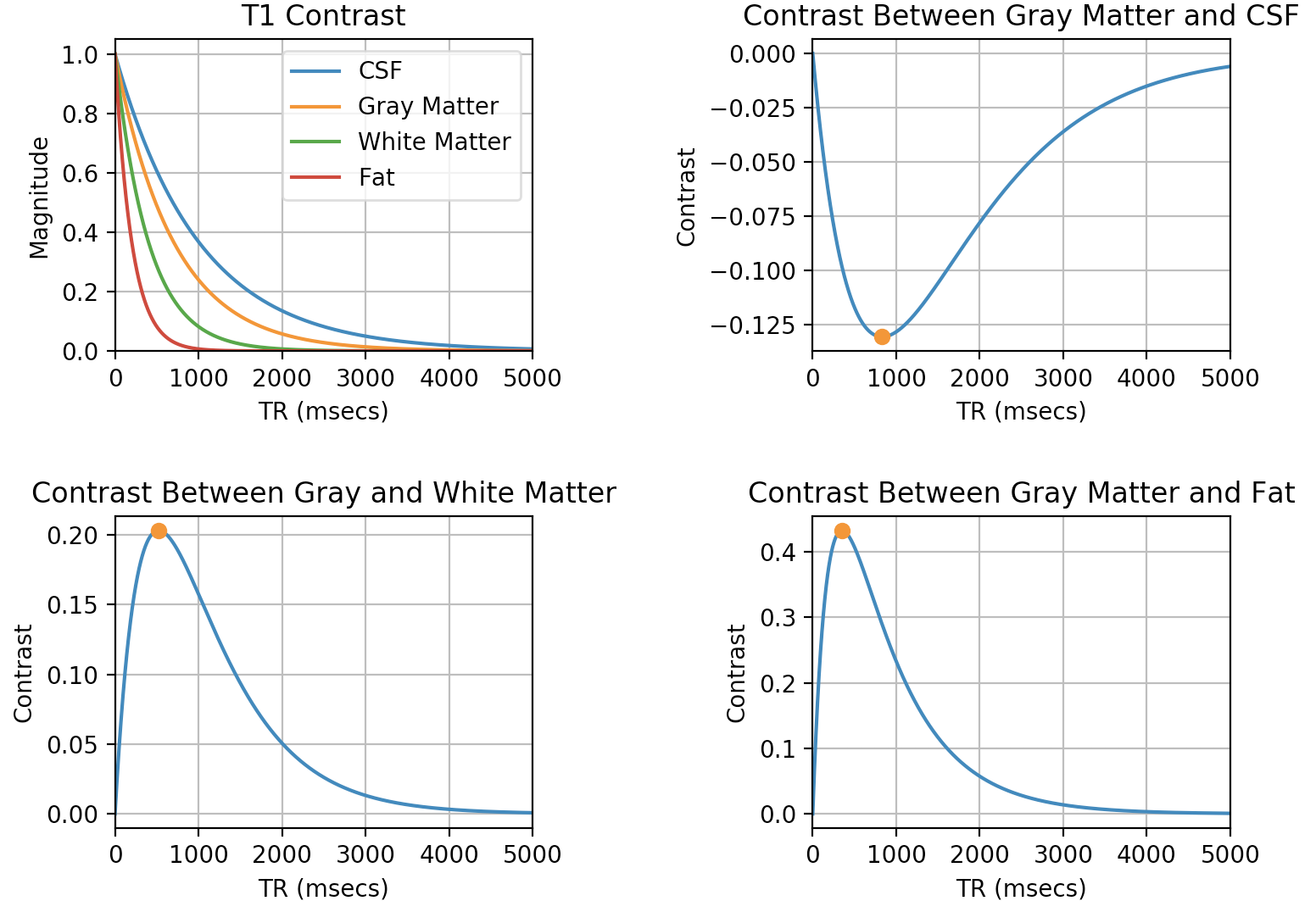
1. T1InversionRecovery plots the magnetization of inversion recovery between t=0 and t=100 msecs. Run the program and enter the desired TR value in msecs by typing in the number as an integer and clicking enter. The program then graphs the equation, for T1= 10, 100, 1000, and 2000 msecs with T2 constant at 50 msecs. An entered TR value of 50 results in the graph:



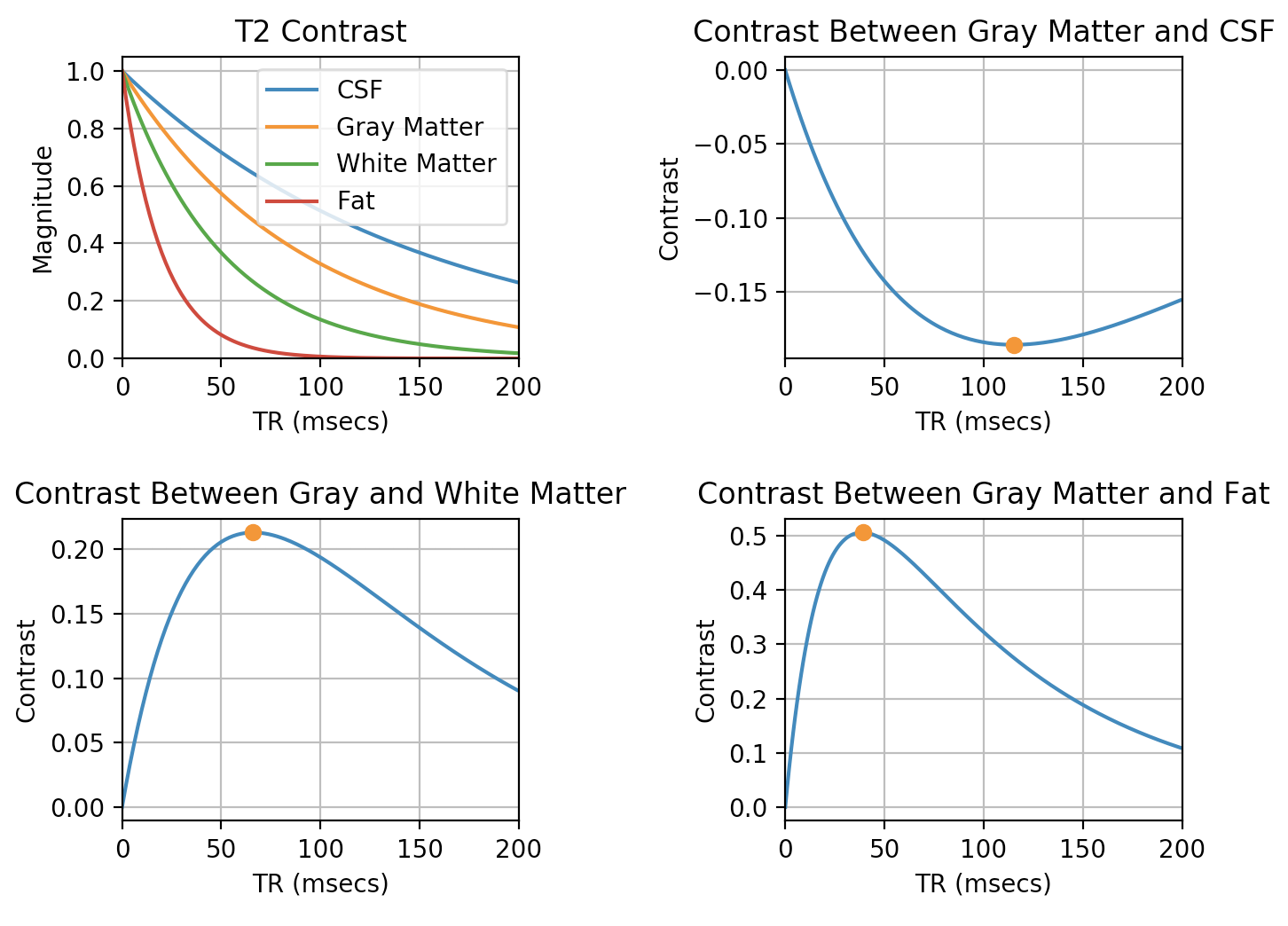
1. T1Invpt2 plots the inversion recovery magnetization absolute magnitude right after the 90-degree pulse (when t=0). The program graphs for T1 = 10, 100, and 1000 msecs. It also shows the specific points where TR = 10, 50, 100, 200, 500, 1000, 2000 msecs. The result of running the program is:

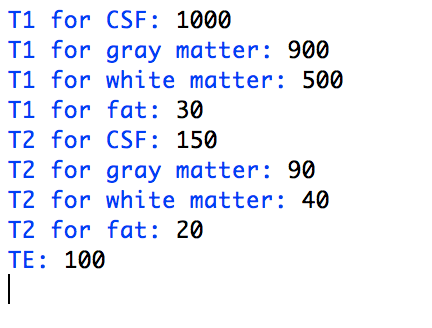
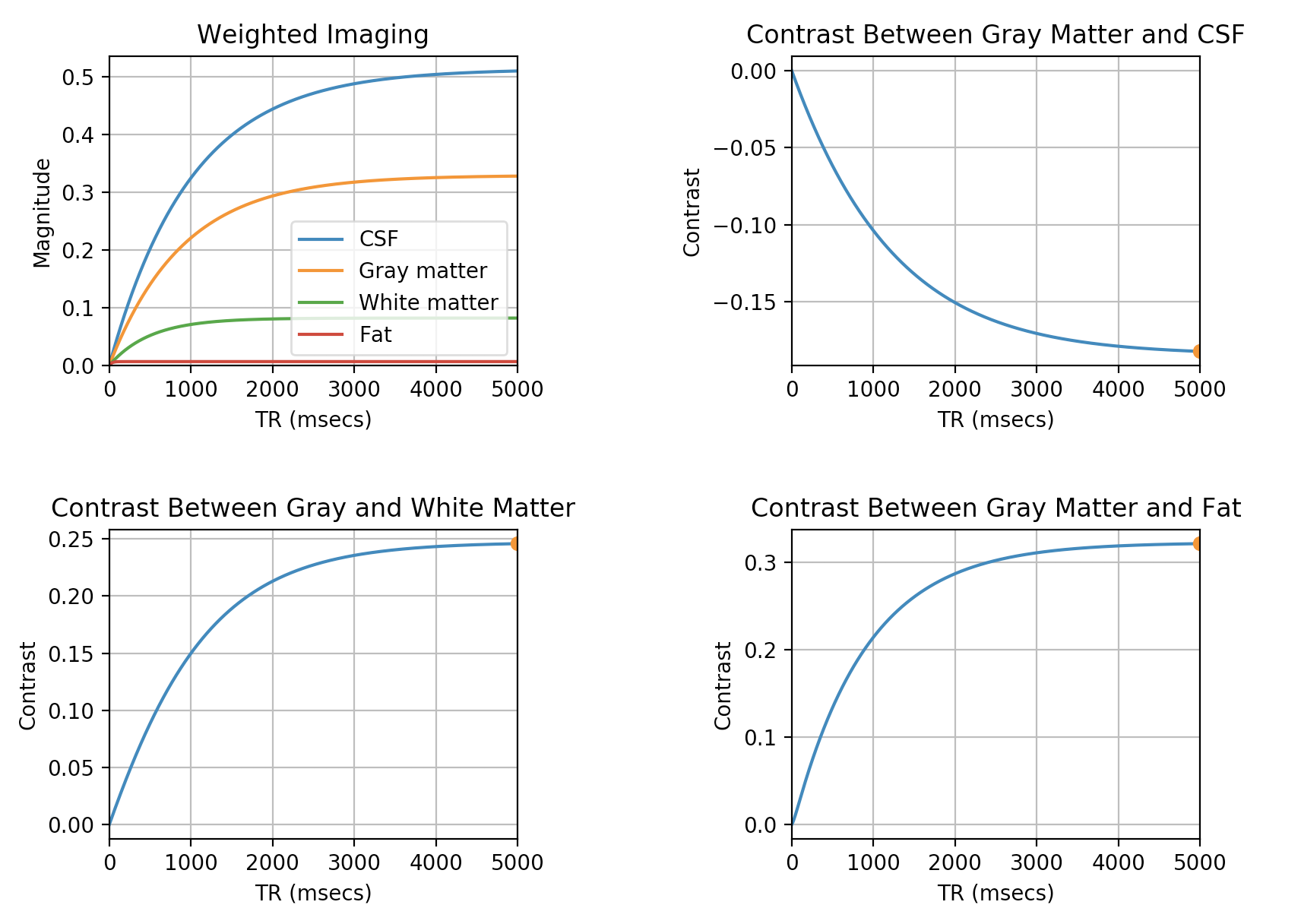
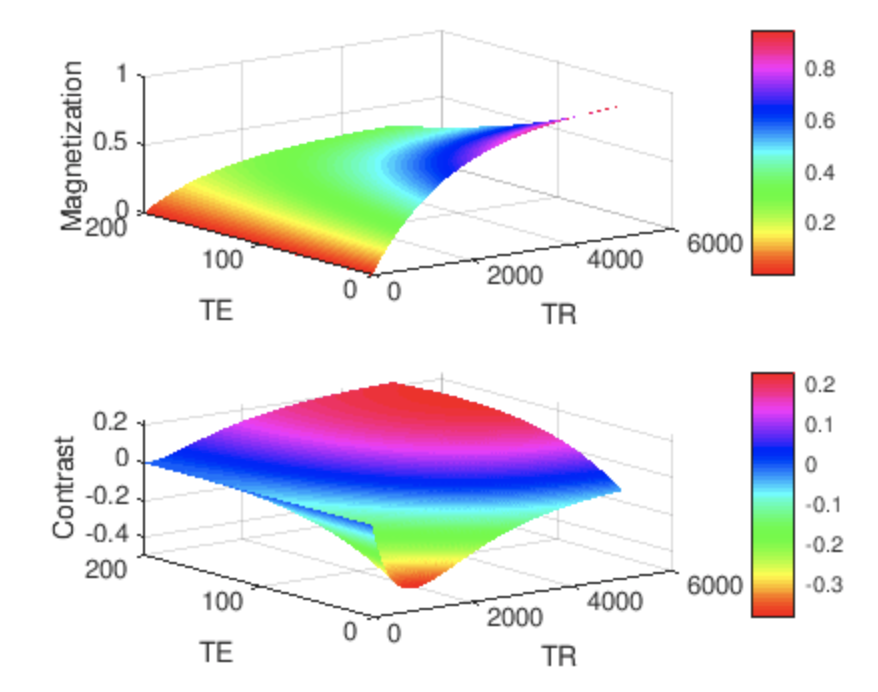


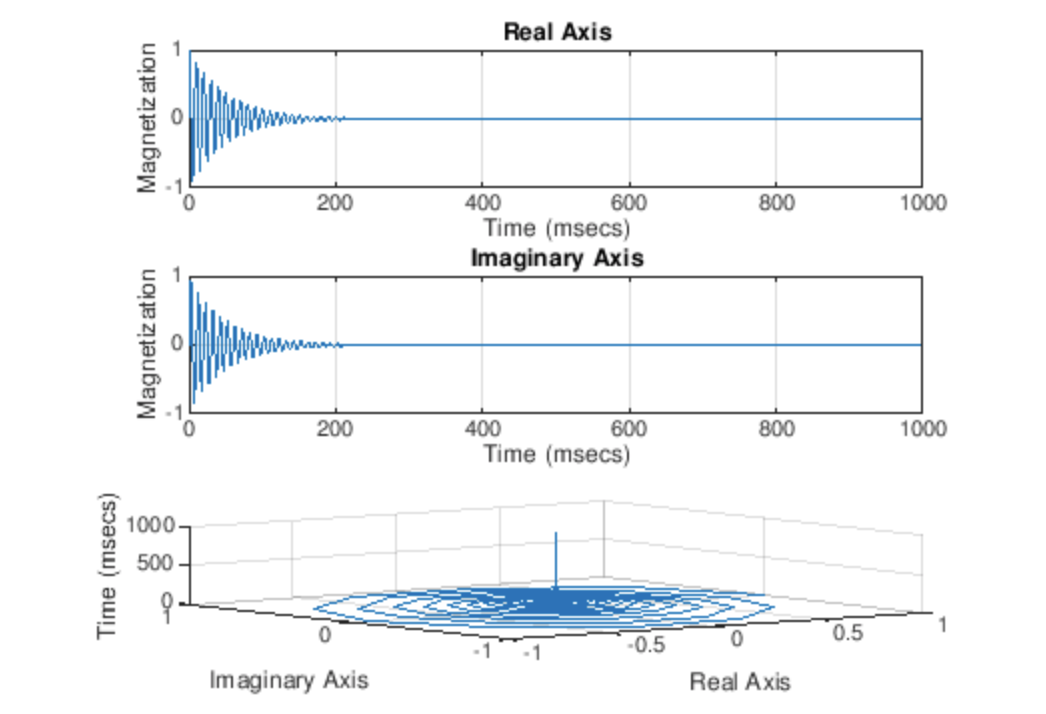
1. Steady State Saturation plots the magnetization of steady state saturation between t=0 and t=200 msecs. Run the program and enter the desired TR (100, 1000, 2000, and 10000 msecs are suggested) value as an integer. The program will then graph for T1 values of 100, 1000, 2000, and 10000. A TR value of 2000 looks like this:
2. T1Contrast shows four graphs to understand T1 contrast saturation recovery. Run the program and enter the T1 for CSF, gray matter, white matter, and fat in msecs by typing them in as integers and pressing the enter key. The program then calculates the signal magnitude at the start of acquisition using the equation and plots them on a graph of TR in msecs versus magnitude. The three graphs next to the first one are the contrasts relative to gray matter, which are found by subtracting the magnitudes from the magnitude of gray matter and looking at the difference. On the contrast graphs, the maximum or minimum point is plotted to highlight the time where the difference is the greatest. When the T1 values are 1000, 700, 400, and 200 for CSF, gray matter, white matter, and fat respectively, the graph looks like:



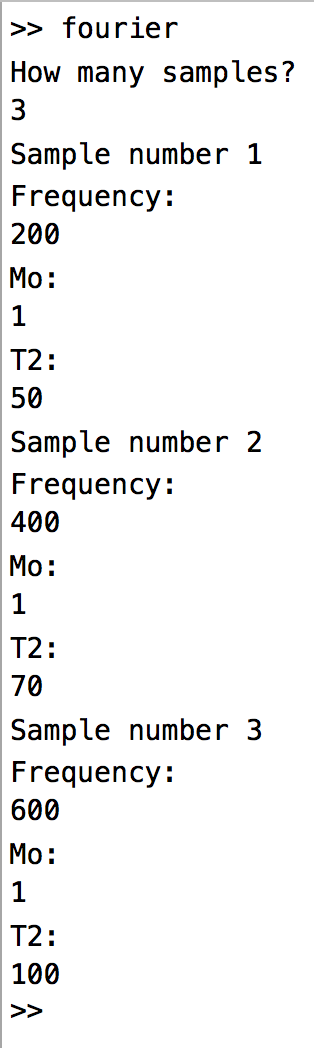
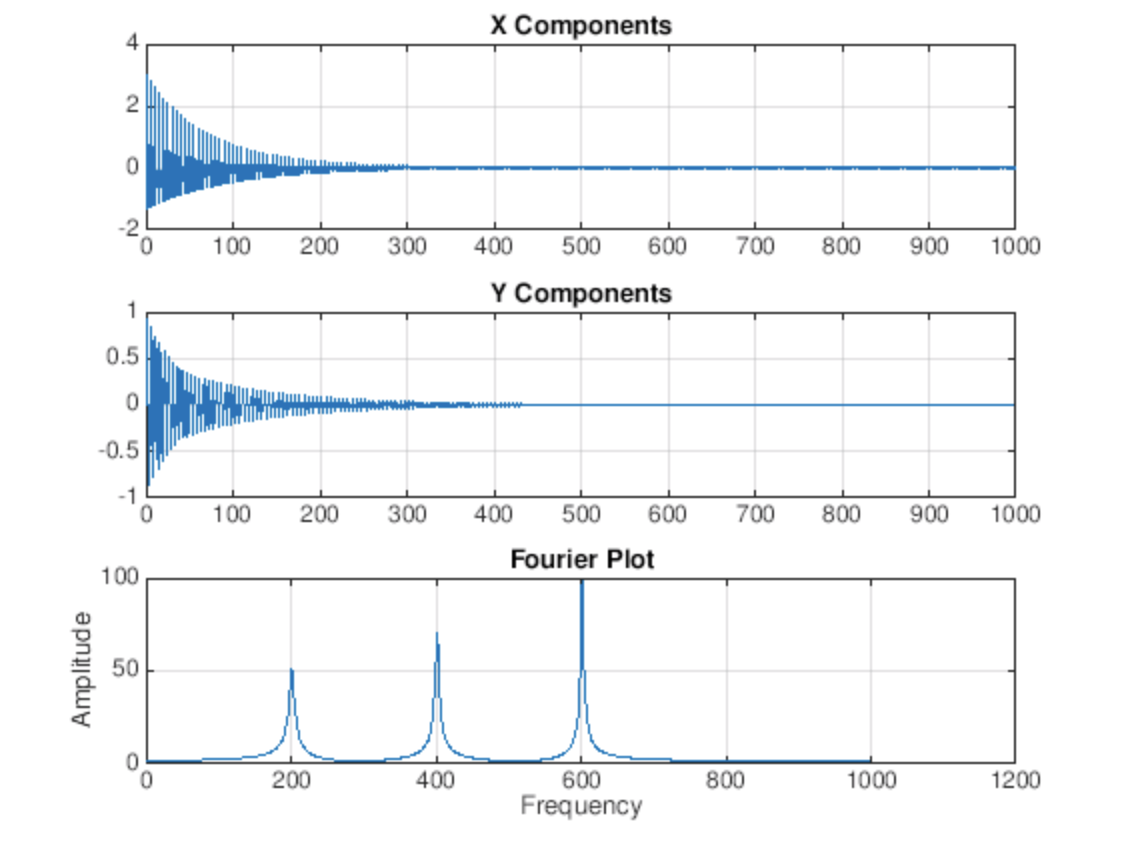
1. T2Contrast shows four graphs to understand T2 contrast. Run the program and enter the T2 for CSF, gray matter, white matter, and fat in msecs by typing in the integers and clicking enter. The program then calculates the signal magnitude at the start of acquisition using the equation and plots them on a graph of TR in msecs versus magnitude. The three graphs next to the first one are the contrasts relative to gray matter which are found by subtracting the magnitudes from the magnitude of gray matter and looking at the difference. On the contrast graphs, the maximum or minimum point is plotted to highlight the time where the difference is the greatest. Using T2 values of 150, 90, 50, and 20 for CSF, gray matter, white matter, and fat respectively, the graph looks like:



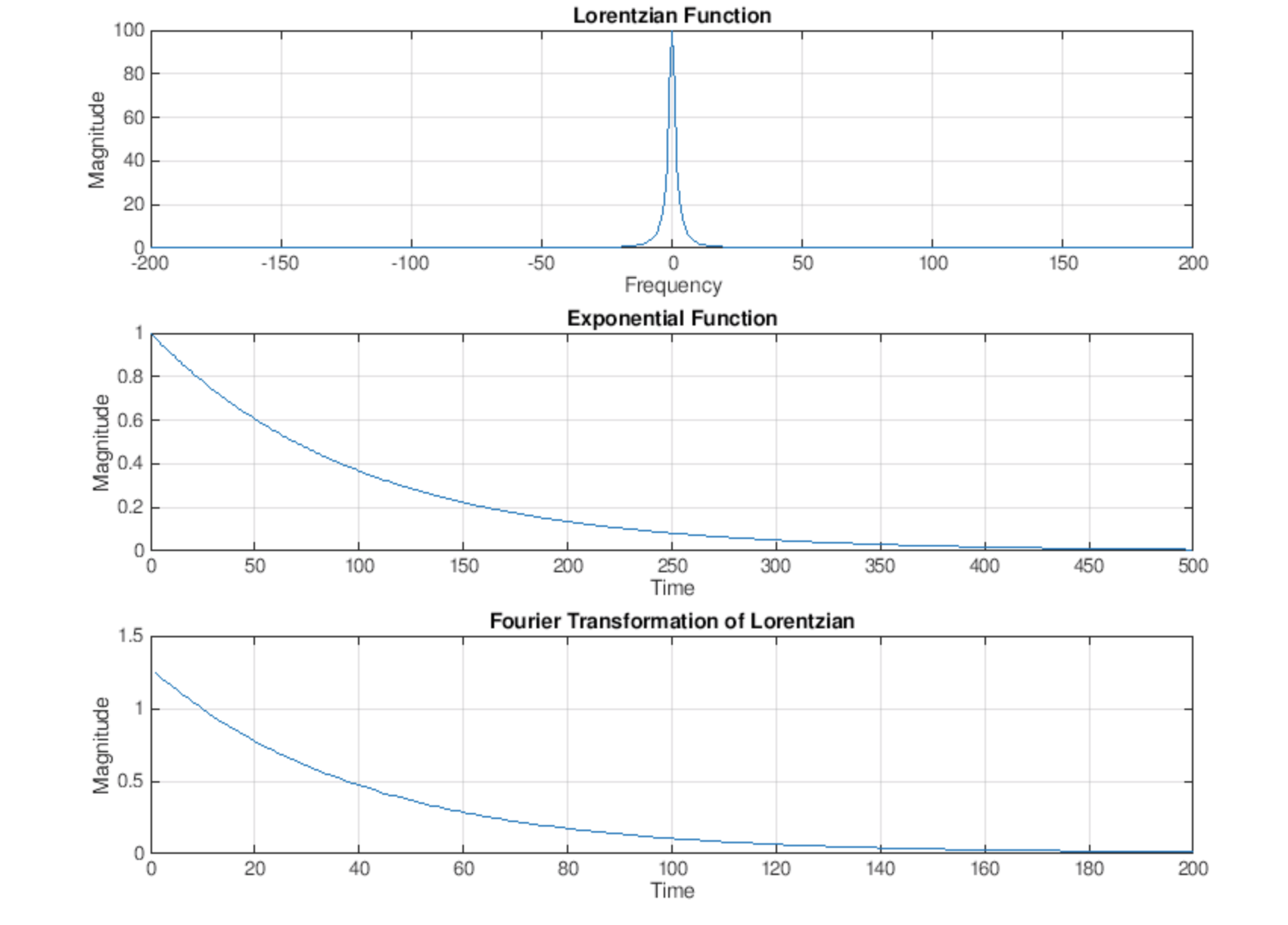
1. Weighted shows four graphs to understand T1 and T2 weighted contrast. Run the program and enter theT1 and T2 for CSF, gray matter, white matter, and fat along with the TE in msecs. The program then calculates the signal magnitude at the start of acquisition using the equation and plots them on a graph of TR in msecs versus magnitude. The three graphs next to the first one are the contrasts relative to gray matter which are found by subtracting the magnitudes from the magnitude of gray matter and looking at the difference. On the contrast graphs, the maximum or minimum point is plotted to highlight the time where the difference is the greatest. Inputting the following directions, the graph below is the result.
2. Weightedsurface (Matlab only) plots the T1 and T2 weighted contrast as a surface graph. Run the program and enter T1 and T2 values for two samples as integers. The program then uses the equation, to create a surface for magnetization as a function of TR and TE and another surface for contrast as a function of TR and TE. To see the graphs from other angle, click on the graphs and spin them around. A sample with a T1 of 1500 and T2 of 200 and a sample with a T1 of 500 and T2 of 100 creates these surfaces.
3. Chemicalshift (Matlab only) plots the magnetization after a chemical shift. Run the program and enter the desired frequency (in msecs) and T2 values. The program then plots the result on the real axis using the equation and the imaginary axis using . Finally, it plots the two on a three-dimensional graph versus time in msecs, which can move around when you click on it. A frequency of 100 and T2 of 50 gives the following graph:



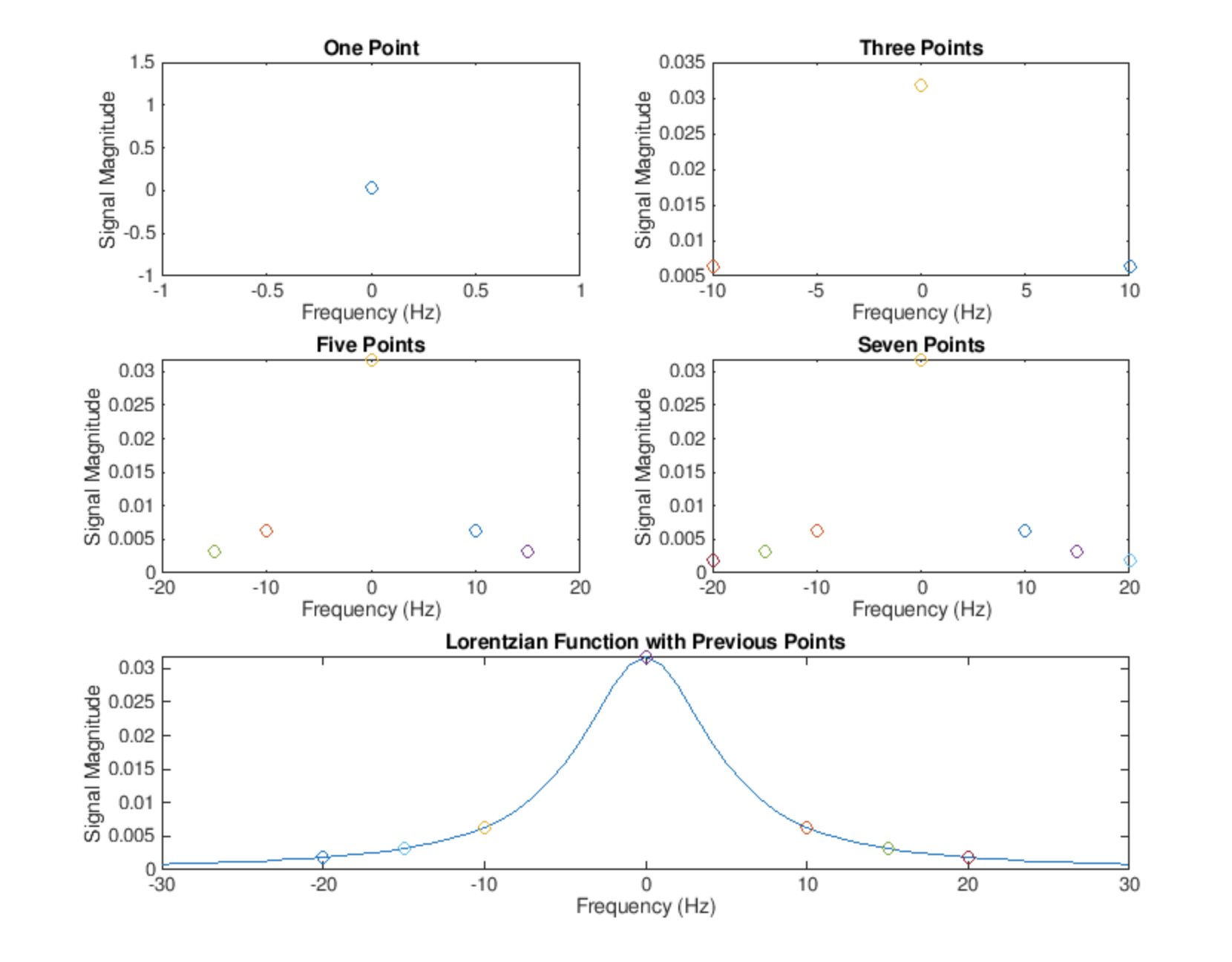
1. Fourier (Matlab only) plots the Fourier transform of signals to see the frequency and magnitude of them. Run the program and enter the number of samples. Next, enter the frequency in Hertz, Mo, and T2 of each sample. The program then adds the x-components of the samples with the equation, which are found with the equation Similarly, it adds the y-components with the equation . Finally, it adds the total components with the equation and plots the Fourier transform of it. Once the plots are graphed, click on the graphs to see the values. Here is an example of the commands and results for running this program with three samples.



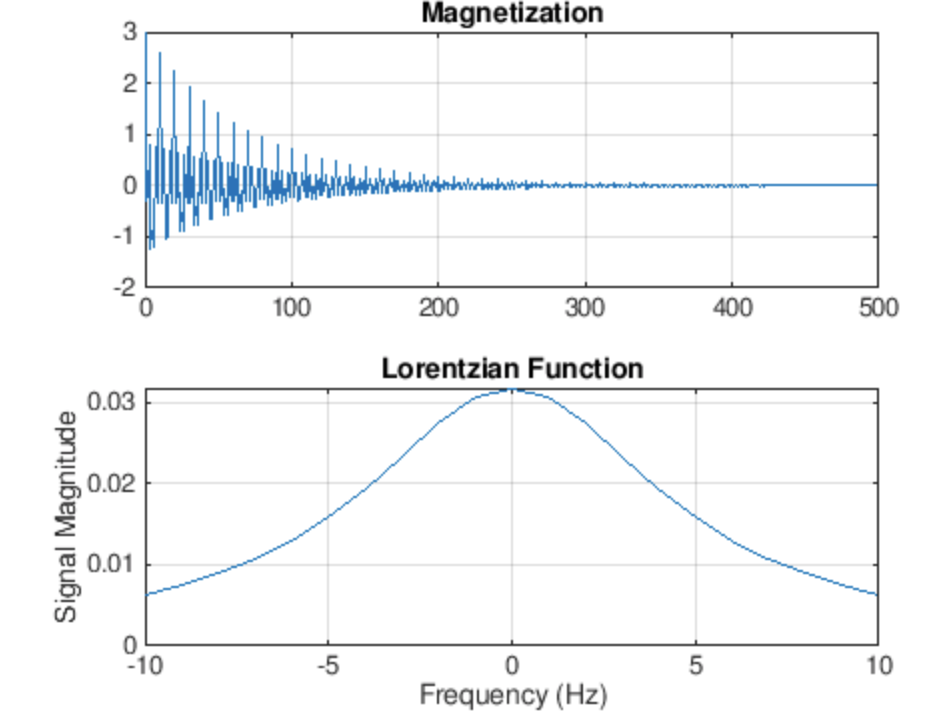
1. Lorentzian (Matlab only) plots the Lorentzian function, exponential function, and Fourier transform of the Lorentzian function for a signal with a chosen T2 value. Run the program and enter the T2 value in msecs. The program then uses the equation to plot the normal Lorentzian equation. Next, it uses the T2 to plot the exponential function, , from 0 to 500 msecs. Finally, the program takes the inverse Fourier transform of the Lorentzian function and plots it from 0 to 500 msecs. As a result, the two graphs of the exponential function and Fourier transform should look the same. Below is an example for T2 = 100 msecs.



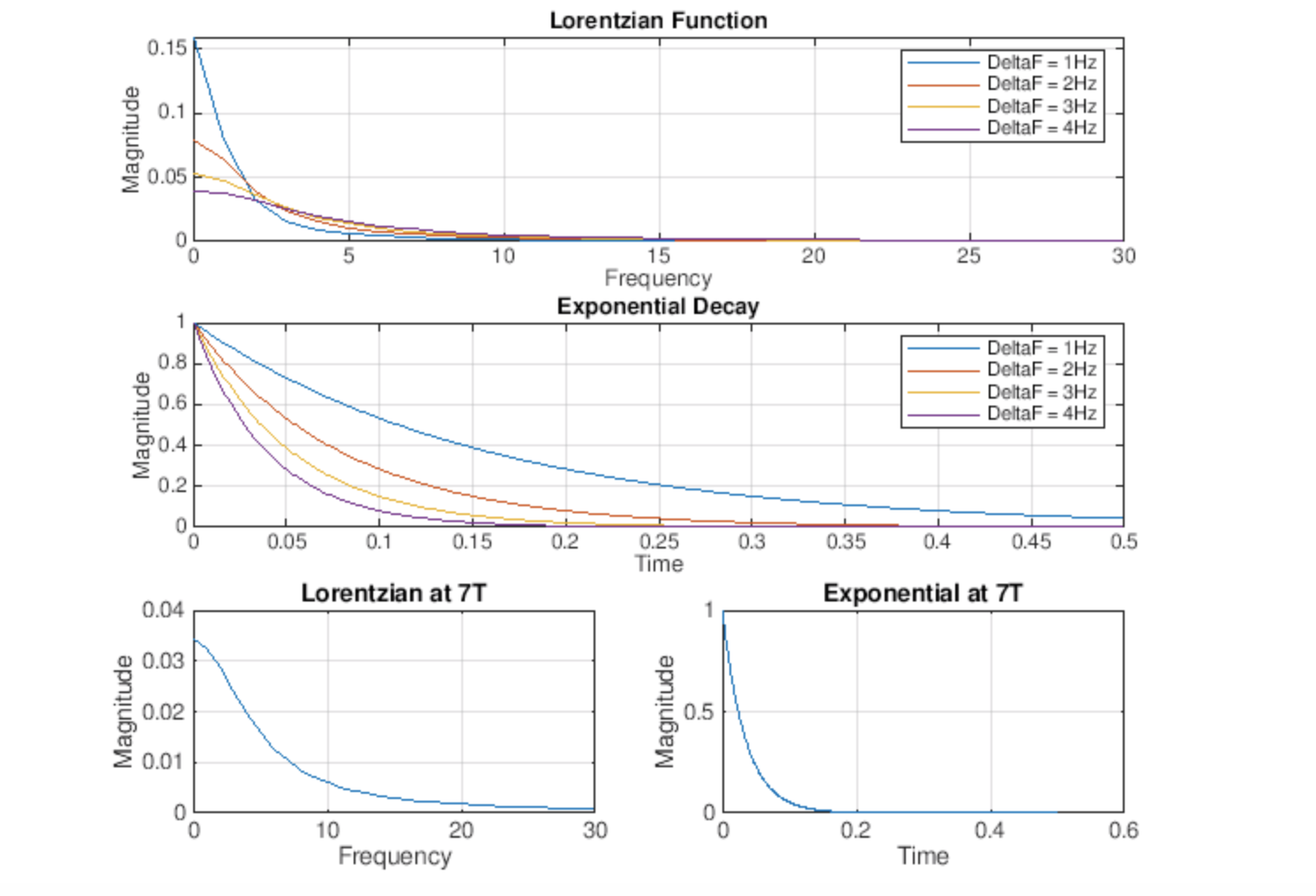
1. Dephasing (Matlab only) shows signal decay due to dephasing. After running the program, type in the T2 value in msecs for the signal. It will then ask for four frequencies in Hz. The program first takes the first frequency and using the equation, to plot the single points. It then creates a new graph to add in the second frequency along with its negative value, and it repeats this step for the third and fourth frequency. The results of this are shown on four graphs each adding two more points until it looks like a Lorentzian function when it has seven points. On a fifth graph, the points are on the actual Lorentzian function, which is . In the picture below, the T2 is 2/(2\*pi\*10) and the frequencies typed into the command window are 0, 10, 15, and 20, respectively.



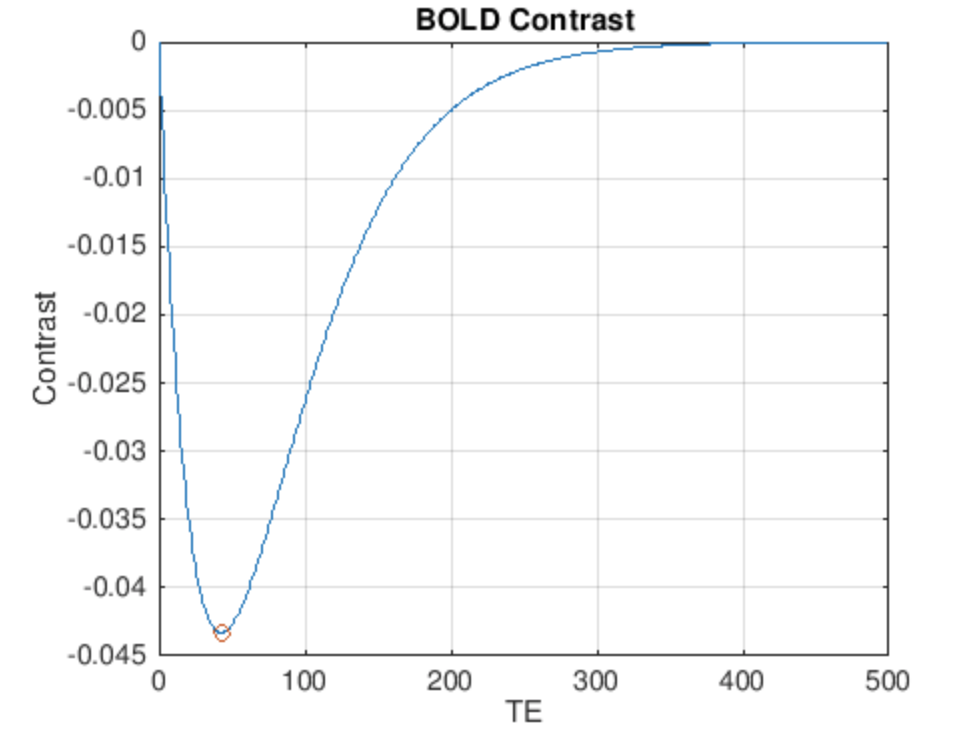
1. Dephasing2 (Matlab only) plots the magnetization along with the continuous Lorentzian function of signal decay due to dephasing. Run the program and enter how many samples you would like. Next, enter the frequency in Hz, Mo,and T2 in msecs for each sample. The graph for magnetization adds together the samples using the equation . Since the Lorentzian form of signals ends up looking like it has a T2 of 2/(2pi10), the second plot shows that continuous function. A test run with one signal that has a frequency of 100, Mo of 1, and T2 of 50, another that has a frequency of 300, Mo of 1, and T2 of 70, and a third with a frequency of 600, Mo of 1, and T2 of 90 looks like the graph below.



1. Signaldecay (Matlab only) shows the signal decay due to a continuous frequency distribution. Run the program and it will show four different graphs. The first two are Lorentzian and exponential decay functions for deltaF = 1, 2, 3, and 4 Hz. The Lorentzian equation is and the exponential is . In the next two graphs, the program plots the Lorentzian and exponential functions if the deltaF was 7T instead of the typical 3T value.



1. Boldcontrast (Matlab only) plots the BOLD contrast of a signal at t=0 as a function of TE. Run the program and then enter the T2 and T2(Bo, Delta Chi) for the first and second signal. Using these, the program calculates the T2\*, which is and the magnitude, which is . In order to find the contrast, the program subtracts the magnitude of the second signal from the first and then plots the result as a function of TE. Finally, the maximum or minimum is highlighted. One signal with a T2 of 50 and T2(Bo, Delta Chi) of 200 (T2\* of 40) and another signal with a T2 of 50 and T2(Bo, Delta Chi) of 450 (T2\* of 45) creates the graph below.



How Run the Python Programs

* Download and install the latest Python for your operating system from <https://www.python.org>
* Search your applications for IDLE (Python’s user interface) which is installed with the language
* Open your terminal and type “pip install matplotlib” then “pip install NumPy”
* Open the desired files in IDLE
* Click Run then Run Module
* Follow the program’s instructions

How to Run Matlab Programs

* Download Matlab from Mathworks <https://www.mathworks.com/downloads/> or use

Matlab online <https://matlab.mathworks.com>

* Open program
* Click run
* Follow the program’s instructions