

# M219 – Winter 2018

## Homework Assignment #0 (Zero Points)

### *Introduction to Matlab*

This assignment is meant to provide an introduction to basic Matlab coding and functions that can form the basis for subsequent assignments in M219. This assignment is not graded nor due. If you have used Matlab before, then this assignment will likely prove relatively straightforward. If you have no familiarity with Matlab, then take this assignment seriously and work through the solutions. It is designed to give you some skills that you will need later in the course.

#### 1. M-file Scripts

Matlab enables saving the code you write as either a *script* or a *function* in the form of an *m-file* with a name like [filename].m. A script is simply a list of commands that are run in sequence when the file is *called* from the Matlab command prompt. A script does not accept *input* variables when called. To get going try the following after you open the Matlab application:

```
>> edit M219_HW0_svm.m
```

You'll be asked "Do you want to create it?" Yes! The Matlab editor will appear with an empty file. Type (or copy and paste) the following (not the line numbers):

```
1 % Filename: M219-2018-HW0_svm.m
2 %
3 % This script creates scalars, vectors, and matrices and performs some operations.
4 % Semi-colons (;) are important in Matlab.
5 %
6 %% Scalars
7 gamma=42.57e6           % 1H gyromagnetic ratio      [MHz/T]
8 B0=1.5;                 % Main magnetic field      [T]
9 M0=1;                   % Equilibrium magnetization [A.U.]
10 alpha=pi/2;            % Flip angle                [radians]
11 w0=gamma*B0;           % Larmor frequency          [MHz]
12
13 %% Vectors
14 M=[0 0 M0]'             % Bulk magnetization vector [A.U.]
15
16 %% Matrices
17 % Rotation by alpha about x-axis
18 Rx=[1 0 0;
19      0 cos(alpha) sin(alpha);
20      0 -sin(alpha) cos(alpha)]
21
22 % Rotation by alpha about y-axis
23 Ry=[cos(alpha) 0 -sin(alpha);
24      0 1 0;
25      sin(alpha) 0 cos(alpha)]
```

Return to the Matlab command prompt and type the following, which will to run your m-file:

```
>> M219_HW0_svm
```

You should see the value of `gamma`, `M`, `Rx`, and `Ry`, but none of the other variables returned to the command space. Note the importance of the “;” in Matlab. Convince yourself that `Rx` and `Ry` effect rotations by  $90^\circ$  about the x-axis and y-axis, respectively.

To run this file you need to either be in the same directory as the saved location of the m-file or you need to add the path of the file to Matlab. Not sure what that means? Try,

```
>> help path
>> help cd
```

In fact, for every function that is part of the Matlab language:

```
>> help [function-name]
```

will provide useful information. Google is your friend.

Within that function scalars, vectors, and matrices were each defined. Remember that the dimensionality of vectors and matrices is *really* important. A vector that is `[1x3]` is not equivalent to one that is `[3x1]`. Furthermore, if we multiply vectors and matrices, then their inner dimensions must match.

## 2. M-file Functions

An m-file *function* requires a function declaration. Functions can accept input variables and also produce outputs. The following demonstration creates a function that rotates the bulk magnetization vector. Note that the dimensions of the input vector must be consistent with the dimensions of the rotation matrix.

```
1 % Filename: M219_HW0_fnc.m
2 %
3 % This function rotates the bulk magnetization vector.
4 %
5 % Inputs:  alpha  - flip angle                [degrees]
6 %          rot_ax - 'x' or 'y' axis of rotation
7 %          M_i    - initial bulk magnetization [A.U.]
8 %
9 % Outputs: M_f    - final bulk magnetization [A.U.]
10 %
11 % Example: M_f=M219_HW0_fnc(pi/2,'y',[0 0 1]')
12
13 function M_f=M219_HW0_fnc(alpha,rot_ax,M_i)
14
15 % Define the rotation matrix
16 switch lower(rot_ax)
17     case 'x'
18         % Rotation by alpha about x-axis
19         R=[1 0 0;
20            0 cos(alpha) sin(alpha);
21            0 -sin(alpha) cos(alpha)];
22     case 'y'
23         % Rotation by alpha about y-axis
24         R=[cos(alpha) 0 -sin(alpha);
25            0 1 0;
26            sin(alpha) 0 cos(alpha)];
27     otherwise
28         error('Input_for_rot_ax_not_allowed');
29 end
30
31 % Rotate the bulk magnetization vector
```

Before using the function, try:

```
>> help M219_HW0_fnc
```

Now try using the function as follows:

```
>> M_f=M219_HW0_fnc(pi/2,'y',[0 0 1]')
```

Again, convince yourself that that the chosen axis of rotation effects rotations by  $\alpha$  about the x-axis or y-axis, respectively. What happens if you try different input vectors (e.g.  $[0 \ 1 \ 0]'$ ).

### 3. For-Loops and Figures

For-Loops are a programming method used to repeat a common function many times. Here we use a for-loop to repeatedly rotate the bulk magnetization vector by calling the M219\_HW0\_fnc.m function. Then we can plot the bulk magnetization's trajectory and generate a figure that plots the results. Try running the following code:

```
>> M_f=M219_HW0_rot_mag(pi/2,'y',[0 0 1]',100)
```

```

1 % Filename: M219_HW0_rot_mag.m
2 %
3 % This function rotates the bulk magnetization vector by several small
4 % steps, then plots a figure of the results.
5 %
6 % Inputs:  alpha  - flip angle                [degrees] [1x1]
7 %          rot_ax - 'x' or 'y' axis of rotation
8 %          M_i   - initial bulk magnetization [A.U.]   [3x1]
9 %          N     - Number of steps           [#]       [1x1]
10 %
11 % Outputs: M      - final bulk magnetization [A.U.]   [3xN]
12 %
13 % Example: M=M219_HW0_rot_mag(pi/2,'x',[0 0 1]',100);
14 function M=M219_HW0_rot_mag(alpha,rot_ax,M_i,N)
15
16 dalpha=alpha/N; % Delta alpha (step size)
17
18 %% Create a bulk magnetization vector array and store the results
19 M=zeros(3,N); % Initialize the matrix to hold the magnetization vectors
20 M(:,1)=M_i; % Store the initial magnetization in the first time slot
21 for j=2:N % For-loop of N-steps, but start with the second time point
22     M(:,j)=M219_HW0_fnc(dalpha,rot_ax,M(:,j-1));
23 end
24
25 %% Create a matlab figure to show the results
26 fig_handle=figure; hold on;
27 p=plot(M');
28 set(p,'LineWidth',3);
29 title('Incremental_Bulk_Magnetization_Components');
30 xlabel('Time_Step_[#]');
31 ylabel('Bulk_Magnetization_Components_[A.U.]');
32 legend('X-Magnetization','Y-Magnetization','Z-Magnetization');
33
34 %% Create a graphic file from the figure
35 M219_Print2Graphic(fig_handle,['./'],'M219_HW0_Sample_Figure',[8 5]);

```

This function generates a figure like this one:

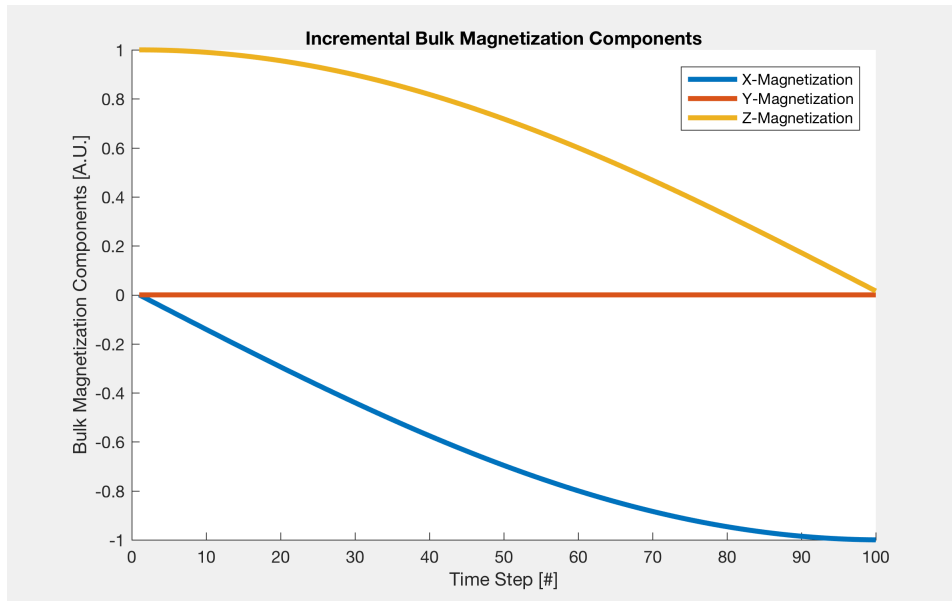


Figure 1: Bulk magnetization components after rotation about the y-axis by a  $90^\circ$  RF pulse.

#### 4. Images

When making figures and plots *always* use appropriate titles, axis labels, colorbars, etc. We can use matlab to load, manipulate, and view images. Let's try doing so below:

```

1 % Filename: M219_HW0_Images.m
2 %
3 % This function loads a default set of MRI data, performs a Fourier
4 % Transform and generates a figure.
5 %
6 % Inputs: None.
7 %
8 % Outputs: IM      - Default MR image array [A.U.] [128x128x1x27]
9 %           KS      - k-space array [A.U.] [128x128x1x27]
10 %
11 % Example: [IM,KS]=M219_HW0_Images;
12 function [IM,KS]=M219_HW0_Images
13
14 %% Load Matlab's default set of MR image
15 load mri; % Load Matlab's default set of MR images
16 IM=D; % The default data is stored in "D".
17
18 %% Compute the Fourier transform
19 KS=zeros(size(IM)); % Allocate memory for k-space
20 for i=1:size(IM,4) % Compute the Fourier transform of each image
21     KS(:,:,,i)=fftshift(fft2((IM(:,:,,i))));
22 end
23
24 %% Display the image and k-space data
25 f1=figure; hold on; colormap(gray);
26 subplot(2,2,1); imagesc(squeeze(IM(:,:,, 1))); axis square;
27 subplot(2,2,2); imagesc(squeeze(IM(:,:,, 6))); axis square;
28 subplot(2,2,3); imagesc(squeeze(IM(:,:,,11))); axis square;
29 subplot(2,2,4); imagesc(squeeze(IM(:,:,,16))); axis square;

```

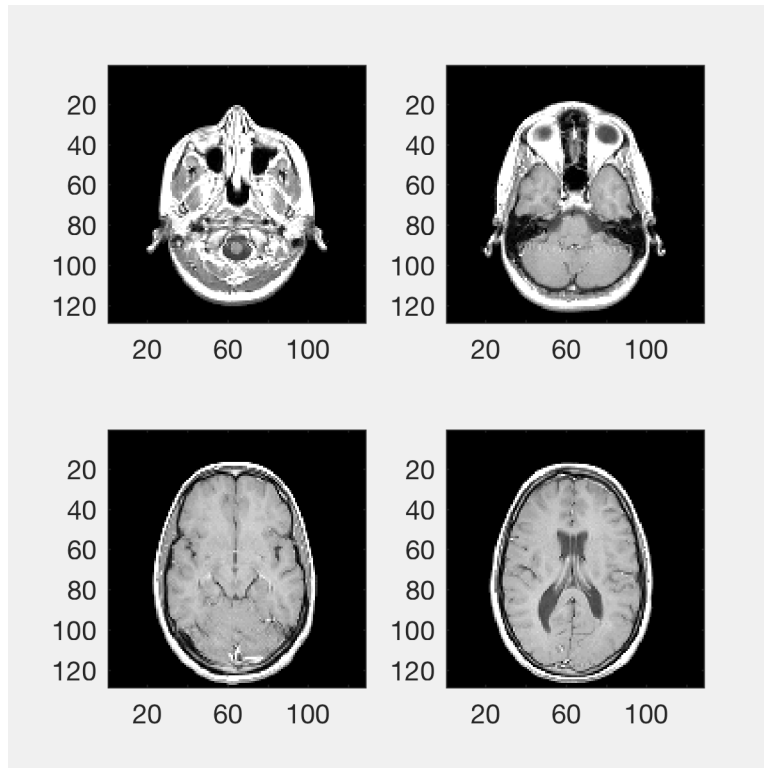


Figure 2: Images from Matlab's default set of MR images.

```

30
31 f2=figure; hold on; colormap(hot);
32 subplot(2,2,1); imagesc(squeeze(abs(KS(:,:,, 1)))); caxis([0 10000]); axis square;
33 subplot(2,2,2); imagesc(squeeze(abs(KS(:,:,, 6)))); caxis([0 10000]); axis square;
34 subplot(2,2,3); imagesc(squeeze(abs(KS(:,:,,11)))); caxis([0 10000]); axis square;
35 subplot(2,2,4); imagesc(squeeze(abs(KS(:,:,,16)))); caxis([0 10000]); axis square;
36
37 %% Create a graphic file from the figure
38 M219.Print2Graphic(f1,['./'],'M219.HW0.Sample.Image',[4 4]);
39 M219.Print2Graphic(f2,['./'],'M219.HW0.Sample.k-space',[4 4]);
40
41 return

```

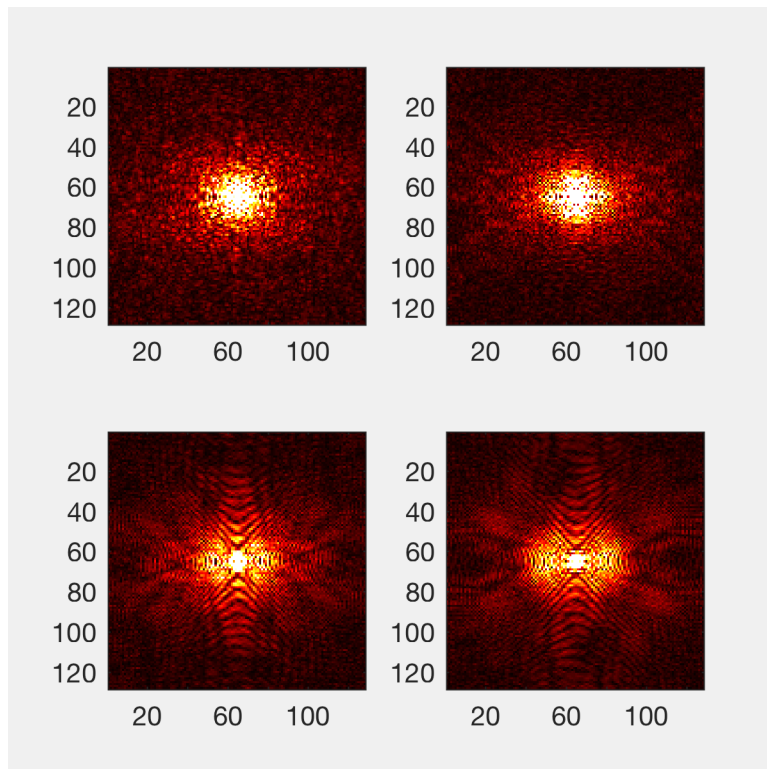


Figure 3: Magnitude Fourier transform of the images in Fig. 2.