MRI Report

Yuzhou Chen

November 23, 2023

0.1 PART 1

The following equation was implimented, and we know $z_{\text{slice}} = 0$:

$$\omega_{\text{shift}} = \gamma G_z z_{\text{slice}} = 0$$

And we can also calculate the amplitude of RF pulse:

$$A = \frac{\frac{\pi}{2}}{\gamma \int_0^t A \cdot \text{RF}_{\text{shape}}(\tau)\, d\tau} = 7.7327 \times 10^{-6}$$

To simulate the magnetization, the following equation were implimented:

$$\begin{aligned} B_x &= RF(t)\cos(\omega_{\text{shift}} t) = RF \\ B_y &= RF(t)\sin(\omega_{\text{shift}} t) = 0 \\ B_z &= G_x x + G_y y + G_z z = 0 \end{aligned}$$

M_x: after a 90-degree RF pulse, we would expect the transverse magnetization M_x to be 0 if it was initially to be 0, which is consistent with the fact that the RF pulse is usually applied along the y-axis.

M_y: we would expect M_y to be 1 as the RF pulse tips the magnetization into the transverse plane, and then remains constant. Also, the timescale of the plot is too short to show decay from transverse relaxation T_2.

M_z: The decreasing from 1 to 0 could show the RF pulse is a 90-degree RF pulse tips the magnetization from the z-axis into the transverse plane. And M_z does not recover shown the long T_1.

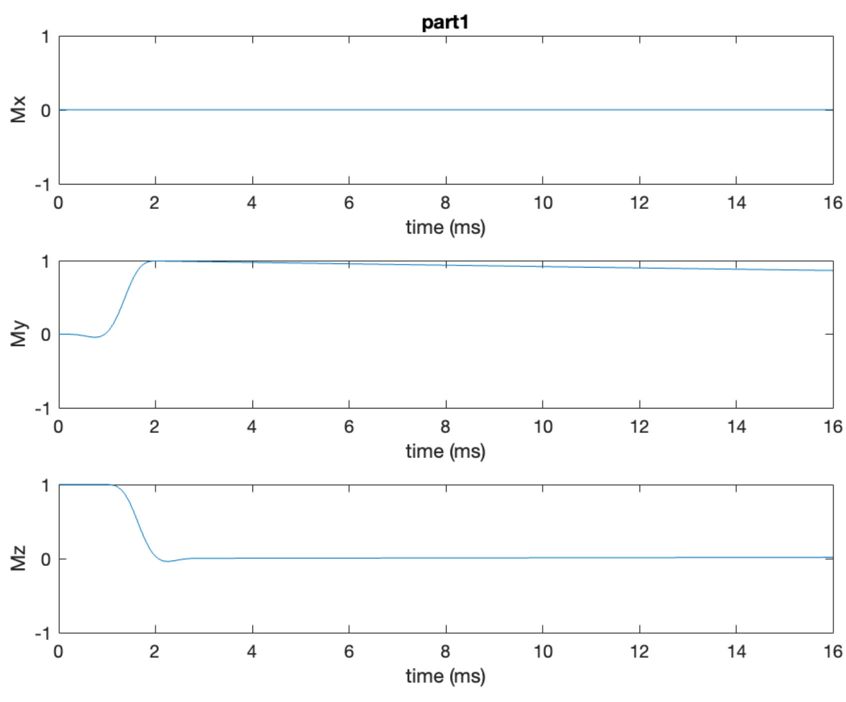


Figure 1: (T1, T2) = (1000,100) ms.

0.2 PART 2

Applied magnetic field remains the same:

$$B_x = RF(t)\cos(\omega_{\text{shift}}t) = RF$$
$$B_y = RF(t)\sin(\omega_{\text{shift}}t) = 0$$
$$B_z = G_x x + G_y y + G_z z = 0$$

M_x: M_x remains same.

M_y: Transverse relaxation T_2 is short enough to see the decreasing.

M_z: T_1 is also short enough to let M_z recover.

M_y component in part 2 shows a decay post-RF pulse, which is not observed in part1, due to the shorter T_2 relaxation time. For M_z, the recovery begins within the time frame shown in part2, unlike in part1, where M_z remains at its lower level, due to the much shorter T_1 relaxation time in part 2.

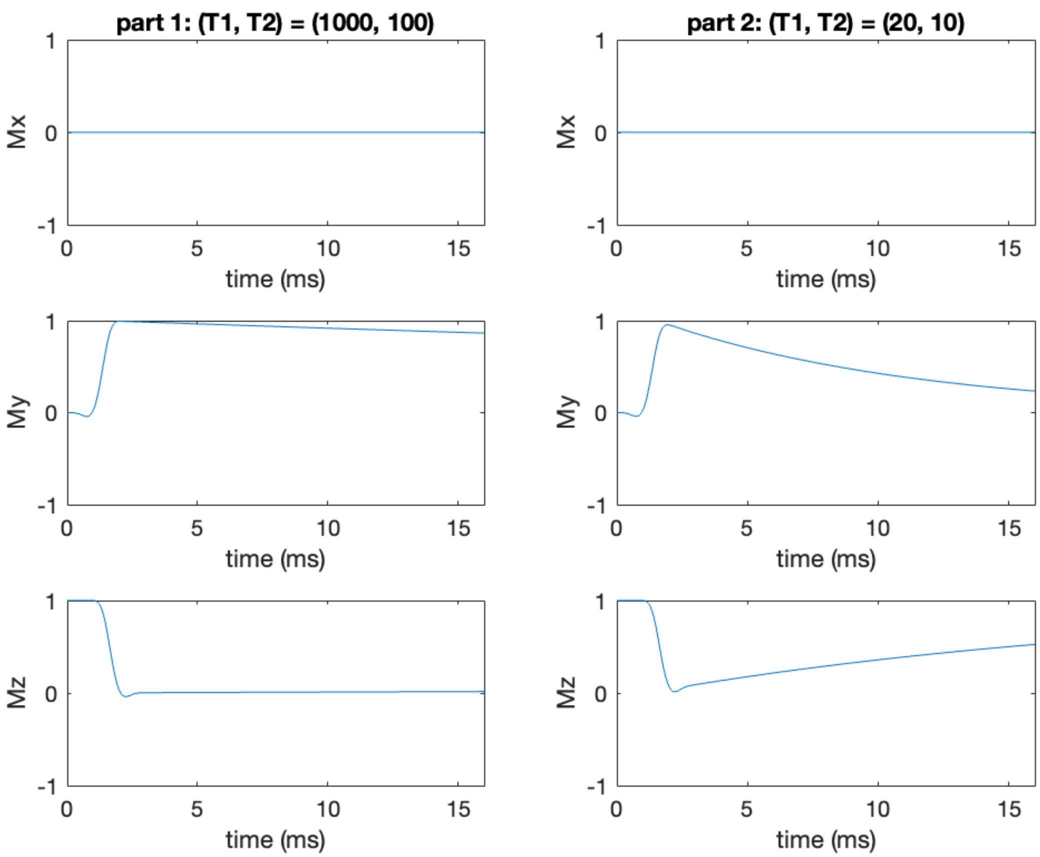


Figure 2: Campare part1 and part2.

0.3 PART 3

Impliment the followng equation and generate G_z first:

$$G_z\text{positive} = \frac{BW}{\left(\frac{\gamma}{2\pi}\right)\Delta z} = 3.1321 \times 10^{-6}$$

$$G_z\text{negative} = -G_z\text{positive} = -3.1321 \times 10^{-6}$$

Applied magnetic field remains the same (z remians to be 0, so that B_z remains unchanged):

$$B_x = RF(t)\cos(\omega_{\text{shift}}t) = RF$$
$$B_y = RF(t)\sin(\omega_{\text{shift}}t) = 0$$
$$B_z = G_x x + G_y y + G_z z = 0$$

M_x: M_x remains same as part1.

M_y: M_y remains same as part1.

M_z: M_z remains same as part1.

Due to the unchanged applied magnetic field, the plot looks exactly same as the plot in part1.

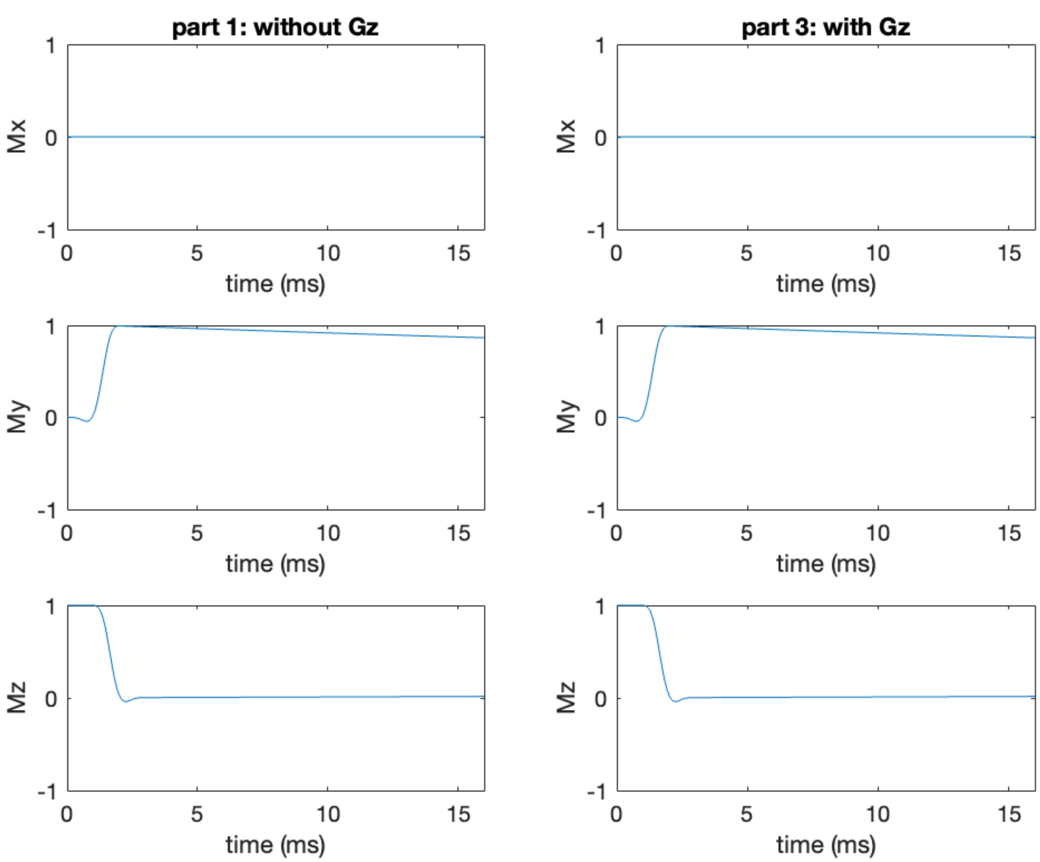


Figure 3: Campare part1 and part3.

0.4 PART 4

Applied magnetic B_z field was shown significant difference with the changing of z

$$B_z = G_x x + G_y y + G_z z = G_z \times z$$

M_x: the patterns of $z = 0.2$ and $z = 1.0$ are totally different than $z = 0$
$z = 0.2$: initialized with fluctuations, indicating that the gradient has caused some dephasing in the transverse plane.
$z = 1.0$: shows significant oscillations after the RF pulse, the dephasing is obviously.

M_y:
$z = 0.2$: there shown a huge fluctuations at the begining, followed with the same pattern as $z = 0$.
$z = 1.0$: also shows oscillations but appears to be damping out over time, suggesting the combined effects of the RF pulse and the gradient on spins outside the slice.

M_z:
$z = 0.2$: a same pattern as $z = 0$ was shown.
$z = 1.0$: increased extremly rapid at the begining, followed with a travial recovery, indicating the effect of relaxation was reduced.

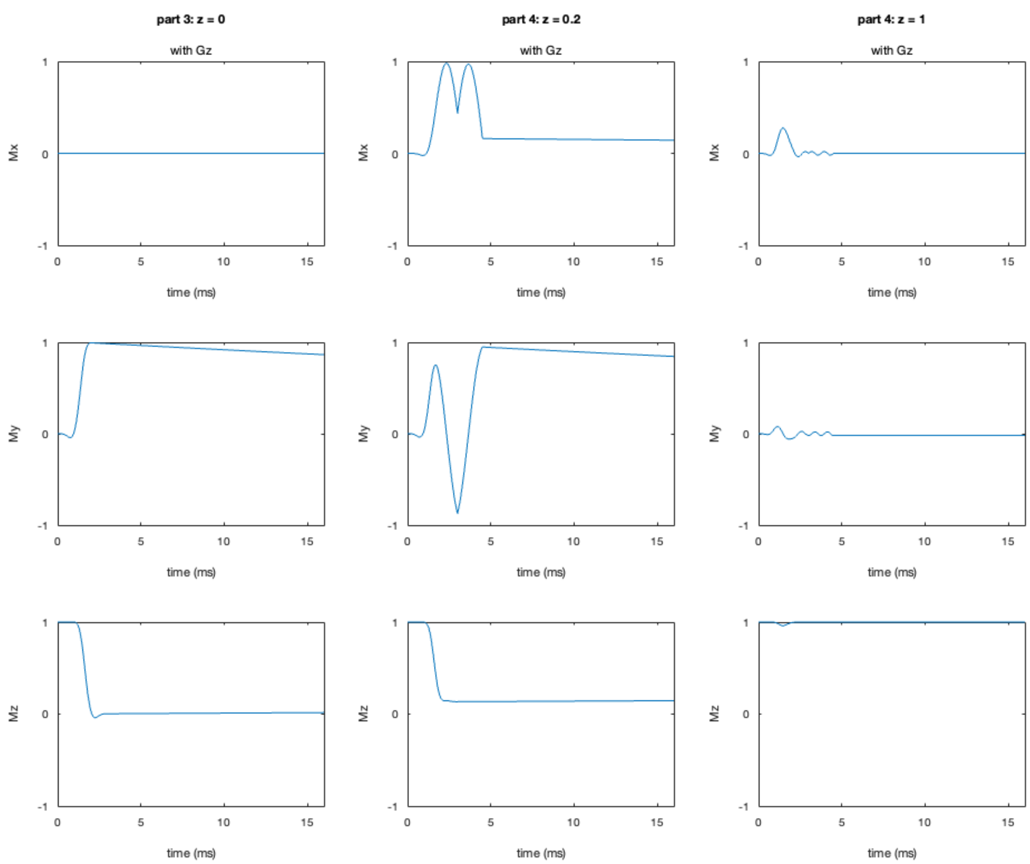


Figure 4: Campared with different z.

0.5 PART 5

The following equation was implimented to generate $G_x$positive first and $G_x$negative for symmetrical acquisition:

$$G_x\text{positive} = \frac{N_x \Delta k_x}{\frac{\gamma}{2\pi} T_{\text{read}}} = 1.4682 \times 10^{-5}$$

$$G_x\text{negative} = \frac{8ms \times -G_x\text{positive}}{2 \times 2ms} = -2.9363 \times 10^{-5}$$

Applied magnetic B_z field changed because of G_x

$$B_z = G_x x + G_y y + G_z z = G_x \times 4$$

M_x: M_x remains same bfore the G_x pluse, followed by significant oscillations.
M_y: M_y remains same bfore the G_x pluse, followed by significant oscillations.
M_z: M_z remains al most same than part1.

The introduction of the x-gradient has introduced dephasing in the transverse plane, which is evident from the oscillations in both M_x and M_y. This dephasing is a result of the Larmor frequency variation across the x-axis due to the x-gradient. The oscillations in M_x and M_y might be indicative of a spin echo sequence, where the x-gradient is switched rapidly to refocus the spins' dephasing caused by the z-gradient.

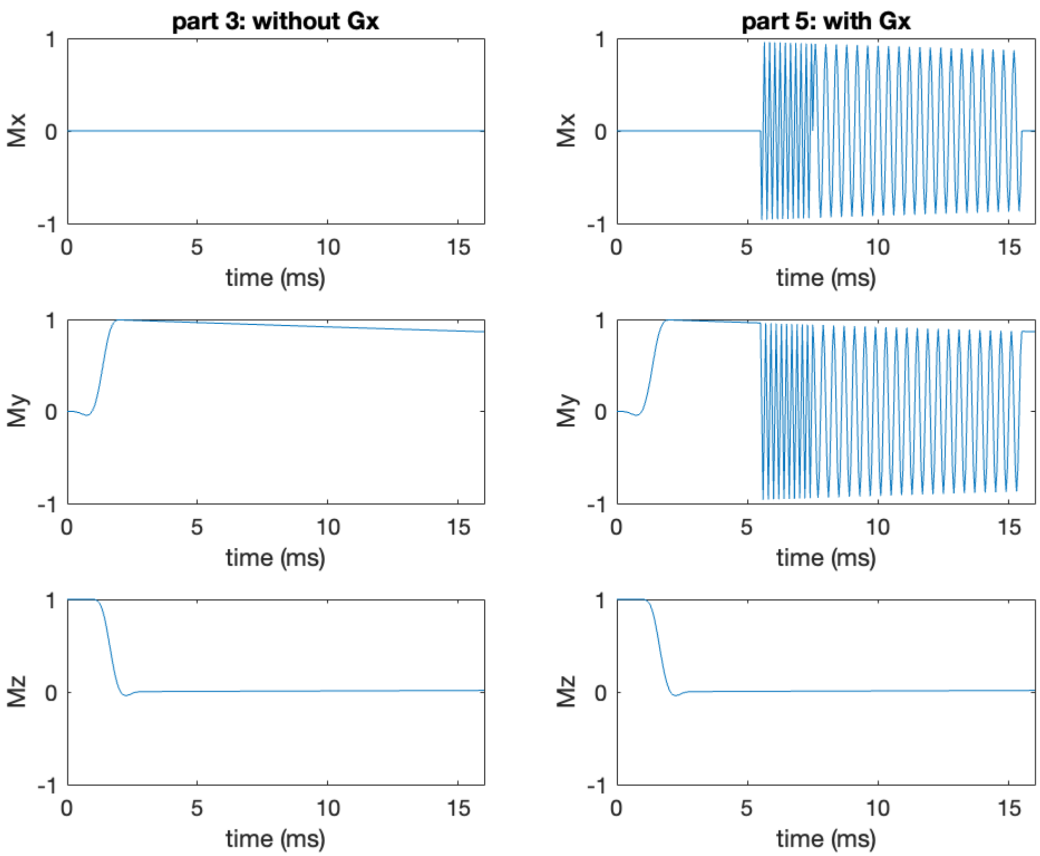


Figure 5: Caption for the figure.

0.6 PART 6

Axis x was generated to show the variation between signal and x:

$$n_{\text{read}} = 80, \Delta x = \frac{FOV_x}{n_{\text{read}}} = 0.2cm \Rightarrow$$
$$x = [-FOV_x/2 + 0\Delta x, -FOV_x + 2\Delta x, ... - FOV_x + (n_{\text{read}} - 1)\Delta x, -FOV_x + n_{\text{read}}\Delta x]$$

From the plot, we can find that the only signal was generated at location $x = 4$. The other location remain 0, which correctly describe the step we set up from part1 to part5.

Compared to part3, which did not involve the x gradient, this step in the simulation process includes both the z and x gradients. The presence of the x gradient allows for spatial encoding along the x-axis, which is necessary for image reconstruction.

The main difference is that now the signal can be localized in space due to the spatial encoding provided by the x gradient. In part3, without the x gradient, such spatial localization would not be possible.

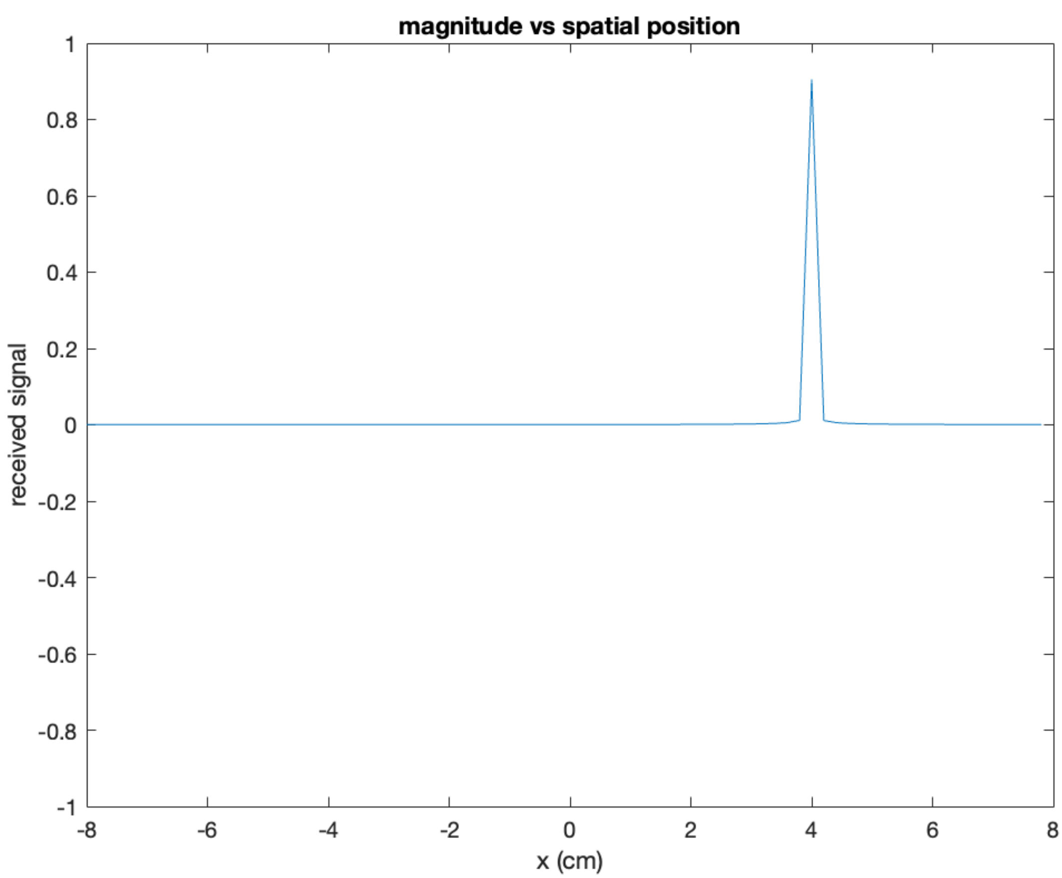


Figure 6: signal vs x

0.7 PART 7

The following equation was implimented to generate $G_y$max first:

$$G_y\text{max} = \frac{N_y \Delta k_y}{\frac{\gamma}{2\pi} 2T_y} = 2.9363 \times 10^{-5}$$

And then phase decoding parameter ΔG_y:

$$\Delta G_y = \frac{2\pi}{2\gamma FOV_y} = 9.7878 \times 10^{-7}$$

Applied magnetic B_z field changed because of G_y

$$B_z = G_x x + G_y y + G_z z = G_x \times 4 + G_y \times 4 + G_y \times 0$$

The M_{xy} was also generated:

$$M_{xy} = m_x + i m_y$$

Finally, axis k_x and k_y were generated to use "image" commend in MATLAB:

$$W_x = \Delta k_x \times N_X \Rightarrow k_x = [-W_x + 0\Delta k_x, -W_x + 2\Delta k_x, ... - W_x + (N_x - 1)\Delta k_x, -W_x + N_x \Delta k_x]$$
$$W_y = \Delta k_y \times N_y \Rightarrow k_y = [-W_y + 0\Delta k_y, -W_y + 2\Delta k_y, ... - W_y + (N_y - 1)\Delta k_y, -W_y + N_y \Delta k_x]$$

From the plot, we can see there was shown uniform stripes in the k-space, indicating the single signal was generated inside k-space. At the same time, the imaginary part was also shown uniform stripes. But they will change significantly in part 10 with $object_{23}$.
The real and imaginary parts of the k-space data can be combined to form a complex image, which after inverse Fourier transforming, provides both the magnitude and phase images in the spatial domain.

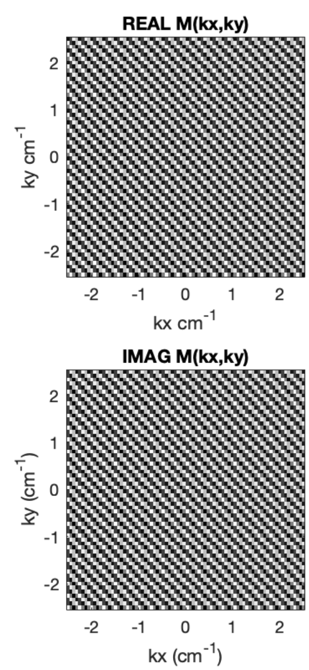


Figure 7: Real part and imaginary part.

0.8 PART 8

The following equation was implimented (2D inverse Fourier transform), and the image was generated:

$$m(x, y) = F_{2D}^{-1}\{M(k_x(t), k_y(t))\}$$

Axis x and y were generated to use "image" commend in MATLAB:

$$\begin{aligned}
n_{\text{read}} = 80, \Delta x = \frac{FOV_x}{n_{\text{read}}} = 0.2cm \Rightarrow \\
x = [-FOV_x/2 + 0\Delta x, -FOV_x + 2\Delta x, ... - FOV_x + (n_{\text{read}} - 1)\Delta x, -FOV_x + n_{\text{read}}\Delta x] \\
n_{\text{pe}} = \frac{G_y \text{max}}{\Delta G_y} = 60, \Delta y = \frac{FOV_y}{n_{\text{pe}}} = 0.2cm \Rightarrow \\
y = [-FOV_y/2 + 0\Delta y, -FOV_y + 2\Delta y, ... - FOV_y + (n_{\text{pe}} - 1)\Delta y, -FOV_y + n_{\text{pe}}\Delta y]
\end{aligned}$$

From the plot, it is obviously that only location $(x, y) = (4, 4)$ has pixel image, and the value is equal to 1 which is pure white. For the other parts of the image, they are all black which means no signal was collected. The single bright spot indicates that the simulation has correctly localized the signal from the point object in space. This spot is the result of the MRI sequence encoding spatial information into the phase and frequency of the signal, which is then decoded through the Fourier transform.

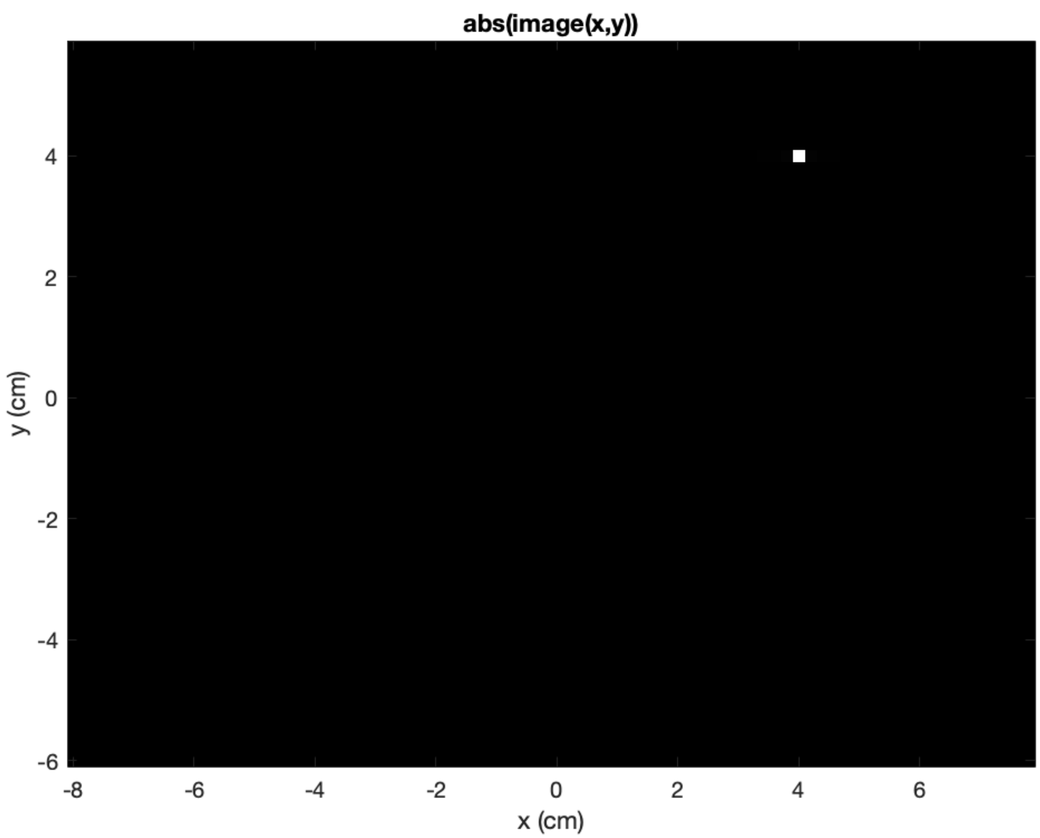


Figure 8: Image after 2D inverse Fourier transform.

0.9 PART 9

By contrast to the part8, the only thing changed is the position of y, which caused the change of applied magnetic B_z:

$$B_z = G_x x + G_y y + G_z z = G_x \times 4 + G_y \times 10 + G_y \times 0$$

The FOV is not wide enough to cover the position at +10 cm, but the plot still showed us somthing interesting. The spin wrap around to the opposite side of the image due to the aliasing effect, instead of going outside the image and disappear. And the location of new pixel of (x, y) is (4,-2).

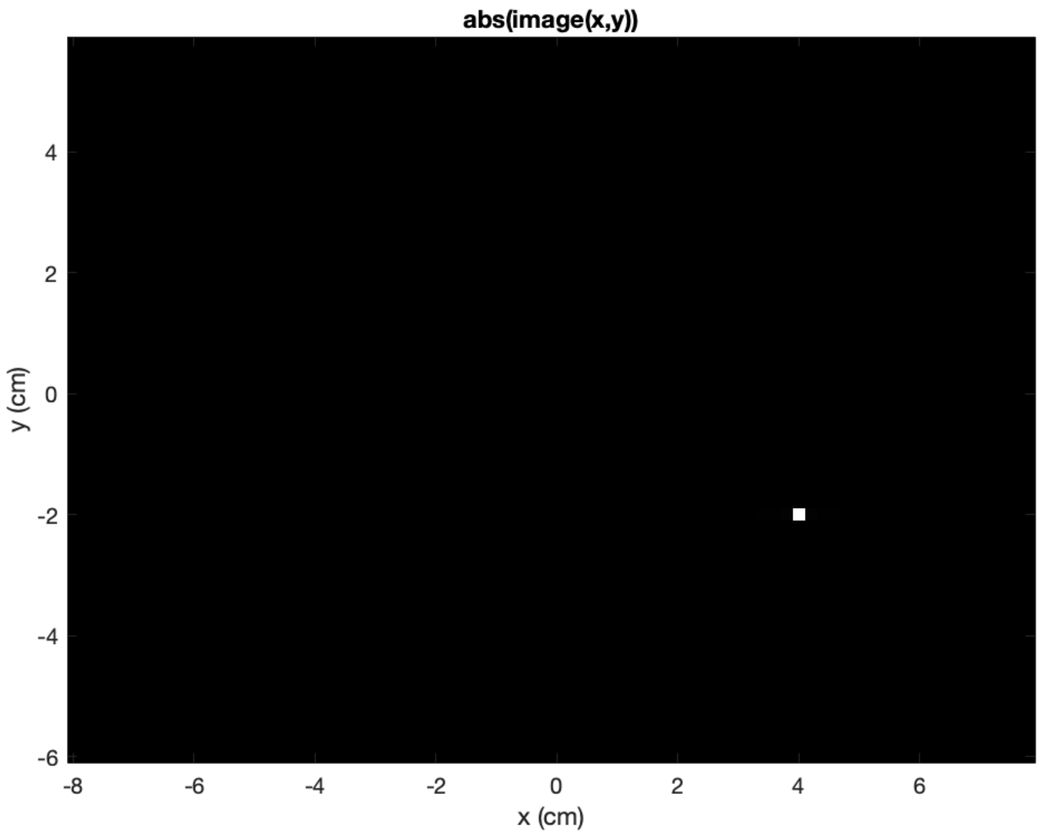


Figure 9: (x,y,z) = (4,10,0).

0.10 PART 10

The real and imaginary parts of the k-space data typically contain both the amplitude and phase information of the signal. The center of the k-space ($k_x = 0, k_y = 0$) contains the bulk of the signal power, and the surrounding k-space encodes the higher spatial frequencies (details and edges) of the image.

The magnitude of the k-space data is a monochrome image showing the distribution of signal intensity across different spatial frequencies. Bright spots in this image correspond to areas with higher signal power, which, after inverse Fourier transformation, contribute to the contrast and details in the spatial domain image.

The object looks like the slice of brain.

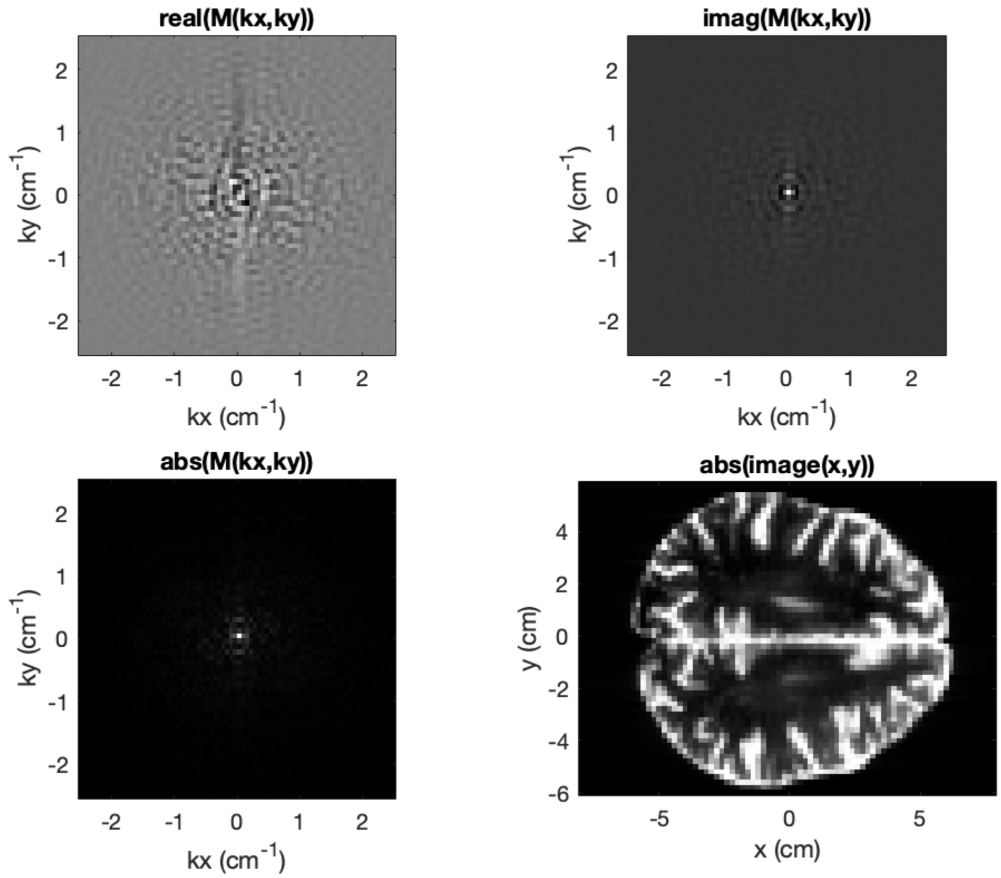


Figure 10: Image for $z_{\text{slice}} = 0$.

0.11 PART 11

The following equation was implimented, and $z_{\text{slice}} = 1$, which is different than part1 - part10:

$$\omega_{\text{shift}} = \gamma G_z z_{\text{slice}} = 8.3776$$

To simulate the magnetization, the following equation were implimented and changed other than same as part1 - part10:

$$B_x = RF(t)\cos(8.3776t)$$
$$B_y = RF(t)\sin(8.3776t)$$
$$B_z = G_x x + G_y y + G_z z$$

By Impliment same loop in part10, we can generate the following plot, which is different than part10. By adjusting the slice position and subsequently modifying B_1 would result in an image that reflects the anatomy or object structure at the new slice location.

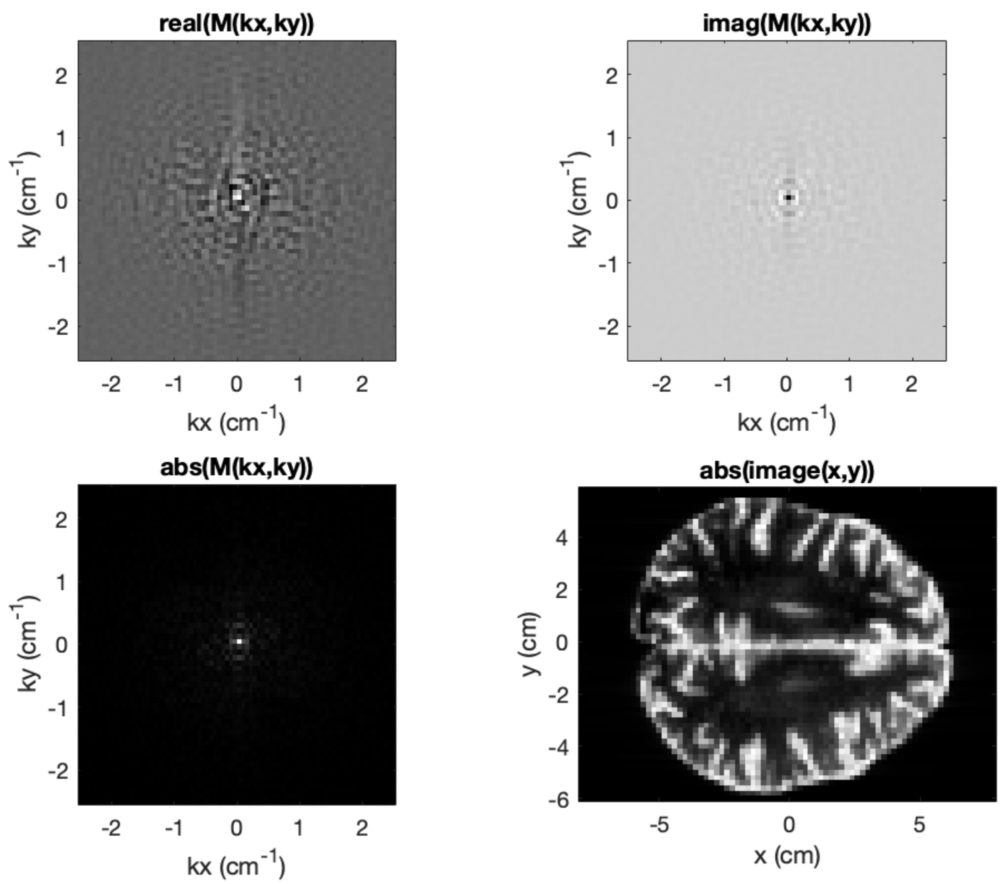


Figure 11: Image for $z_{\text{slice}} = 1$.

Contents


```
% BME/EECS516
% MRI Project Template
clc
clear all
% Other m-files required: ift2, ift, ft2, ft, blochsim_516
% Subfunctions: none
% MAT-files required: object18.mat

%Programme: YUZHOU CHEN
%DATA:November 23, 2023
%Oct 2022;
%Last revision: Oct-30-2022
```

Select whether to load complex 2D object or create simple point object

```
complexobj = 0;
if complexobj
    % 2D Object for reconstruction
  load object23;
else
  % Single point object at (x,y,z) = (2,2,0) cm;
  % Point object has T1 of 1000 ms, T2 of 100 ms
  obj_x = 4;
  obj_y = 4;
  obj_z = 0;
  obj_T1 = 1000;
  obj_T2 = 100;
end

FOVx = 16;
FOVy = 12;
Nx = 80;
Ny = 60;
T_read = 8;
T_y = 2;
obj_n = length(obj_x); % Determine number of objects
```

Define simulation constants

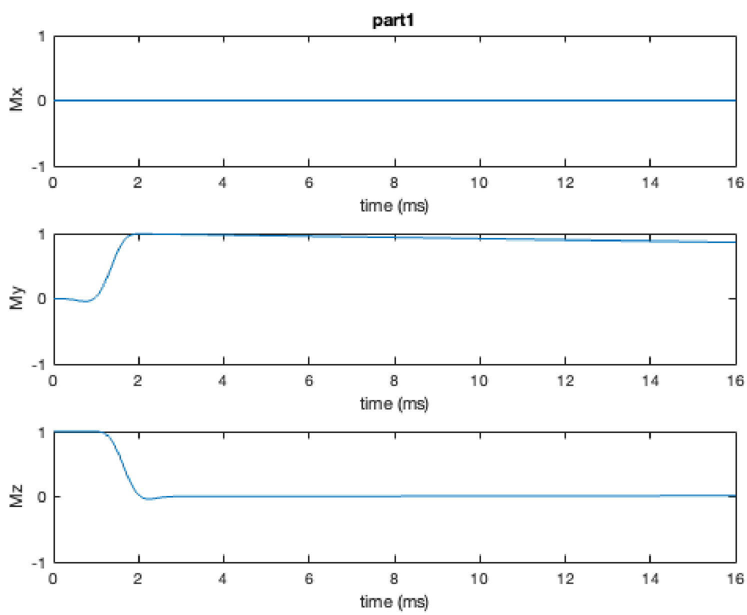
Physical constants

```
gambar = 42570;               % Gamma/2pi in kHz/T
gam = gambar*2*pi;            % Gamma in kiloradians/T

% Simulation values
dt = 0.05;                    % Time step for simulation, ms (50 us step size)
te = 10.0;                    % Echo time, ms
endtime = 16;                 % Total runtime of simulation, ms
time = [0:dt:endtime]';       % Vector containing each time step, ms (size #timepoints x 1)
totalTimepoints = length(time);           % Number of time points for simulation
```

```
% Initialize B vectors, the effective (x,y,z) applied magnetic field
% Vectors define applied magnetic field at time tp_n for object obj_n

bx = zeros([totalTimepoints obj_n]);
by = zeros([totalTimepoints obj_n]);
bz = zeros([totalTimepoints obj_n]);

% Define a 90 RF pulse
rf90pw = 3;                  % Pulse width in ms
sincper = rf90pw/4;          % in ms (this is the sinc stretch parameter)
                             % e.g. sinc(time/sincper) as shown below
rf_timepoints = rf90pw/dt;   % Number of simulation steps for RF
rf_time = [-(rf_timepoints-1) / 2 : (rf_timepoints-1) / 2]'.*dt; % Time vector for creating sinc, centered at 0
rf_shape = hanning(rf_timepoints) .* sinc(rf_time./sincper); % Sinc waveform shape with hanning window, with amplitude 1
rf_amplitude90 = (pi / 2)/(gam * dt * sum(rf_shape));        % REPLACE 0 with amplitude of the RF pulse here, in T
% Scale rf_shape by a_rf90 (amplitude), then fill the remainder of the time with zeros
b1_90 = rf_amplitude90.*[rf_shape; zeros([totalTimepoints-rf_timepoints 1])];
m0 = [0; 0; 1];
gz = 0;
omega_shift = gam * gz * 0;
```

part 1

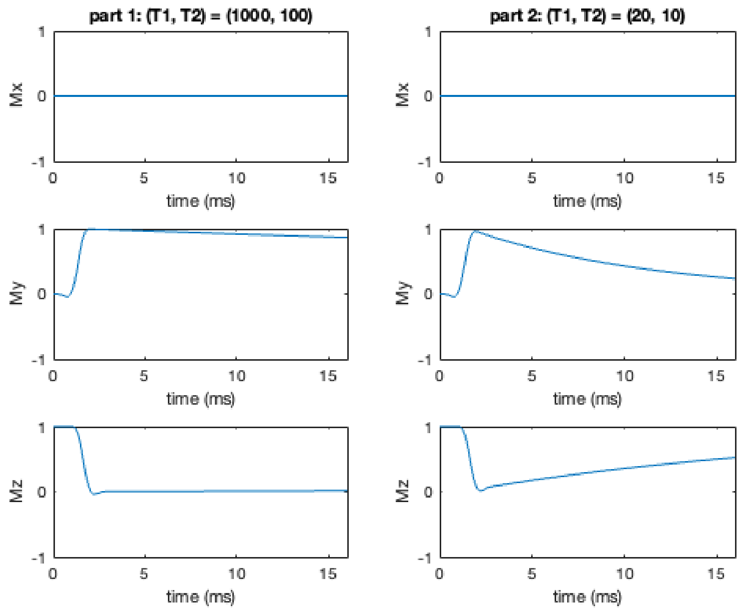
```
bx = b1_90;
[mx,my,mz] = blochsim_516(m0,bx,by,bz,obj_T1,obj_T2,dt);
figure (1)
subplot(3,1,1)
plot(time,mx);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part1');
subplot(3,1,2)
plot(time,my);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,1,3)
plot(time,mz);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
```



part 2

```
obj_T1_1 = 20;
obj_T2_2 = 10;
[mx_2,my_2,mz_2] = blochsim_516(m0,bx,by,bz,obj_T1_1,obj_T2_2,dt);

figure (2)
subplot(3,2,1)
plot(time,mx);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 1: (T1, T2) = (1000, 100)');
subplot(3,2,3)
plot(time,my);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,2,5)
plot(time,mz);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
```

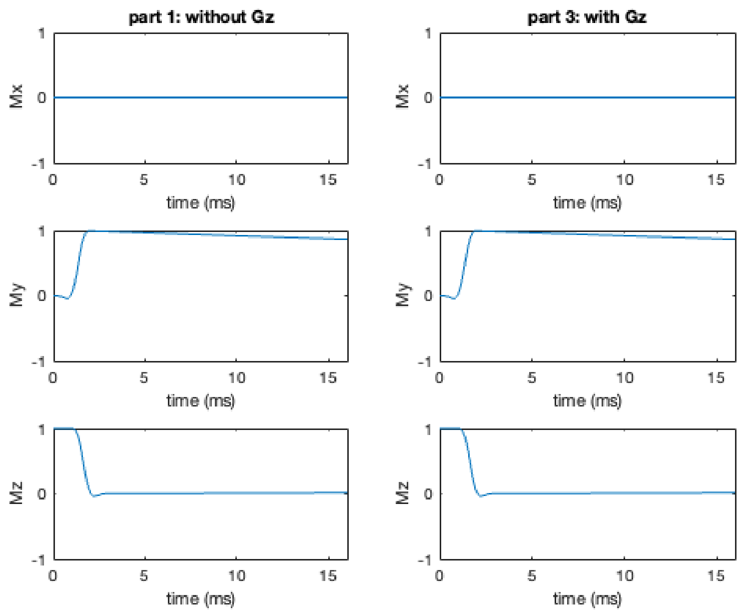
```
subplot(3,2,2)
plot(time,mx_2);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 2: (T1, T2) = (20, 10)');
subplot(3,2,4)
plot(time,my_2);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,2,6)
plot(time,mz_2);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
```



part 3

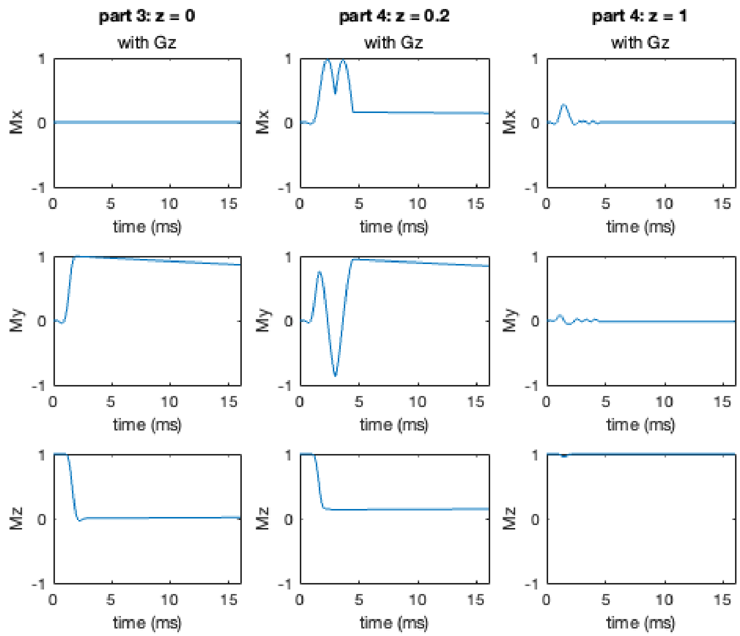
Create gradients Create gz

```
rf90bw = 1 / sincper;     %bandwith of RF
slThick = 1;              % Slick thickness in cm
gz1_a = 2*pi*rf90bw/gam/slThick;                  % REPLACE 0 with amplitude of gz1 in T/cm
gz1_pw = rf90pw;            % Match the width of gz1 to the RF pulse
gz2_a = -gz1_a;              % REPLACE 0 with amplitude of gz2 in T/cm
gz2_pw = rf90pw/2;
% Create gz with positve area gz1_a*gz1_pw, followed by negative area gz2_a*gz2pw
% gz step size is dt, with amplitude values in T/cm
gz =  (time < gz1_pw) .* gz1_a ...
       + (time >= gz1_pw).*(time < (gz1_pw+gz2_pw)) .* gz2_a;
bz_3 = gz * obj_z;
[mx_3,my_3,mz_3] = blochsim_516(m0,bx,by,bz_3,obj_T1,obj_T2,dt);

figure (3)
subplot(3,2,1)
plot(time,mx);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 1: without Gz');
subplot(3,2,3)
plot(time,my);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,2,5)
plot(time,mz);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
subplot(3,2,2)
plot(time,mx_3);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 3: with Gz');
subplot(3,2,4)
plot(time,my_3);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,2,6)
plot(time,mz_3);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
```



part 4

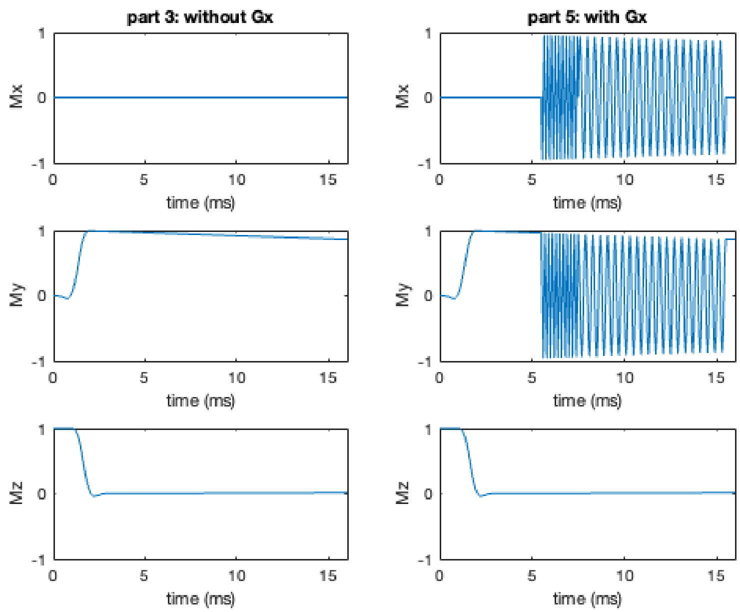
```
obj_z_41 = 0.2;
obj_z_42 = 1;
bz_41 = gz * obj_z_41;
bz_42 = gz * obj_z_42;

[mx_41,my_41,mz_41] = blochsim_516(m0,bx,by,bz_41,obj_T1,obj_T2,dt);
[mx_42,my_42,mz_42] = blochsim_516(m0,bx,by,bz_42,obj_T1,obj_T2,dt);
figure (4)
subplot(3,3,1)
plot(time,mx_3);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 3: z = 0');subtitle('with Gz');
subplot(3,3,4)
plot(time,my_3);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,3,7)
plot(time,mz_3);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
subplot(3,3,2)
plot(time,mx_41);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 4: z = 0.2');subtitle('with Gz');
subplot(3,3,5)
plot(time,my_41);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,3,8)
plot(time,mz_41);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
subplot(3,3,3)
plot(time,mx_42);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 4: z = 1');subtitle('with Gz');
subplot(3,3,6)
plot(time,my_42);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,3,9)
plot(time,mz_42);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
```



part 5

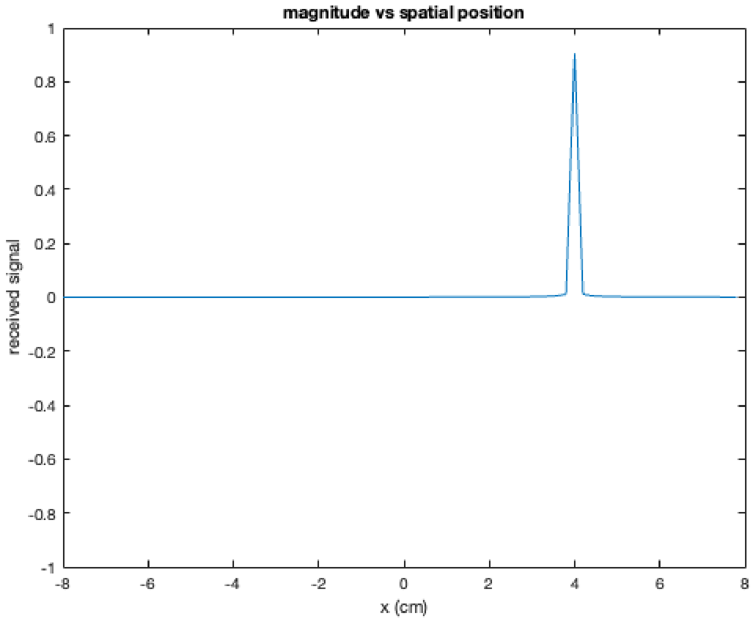
Create gx

```
k_x = 1/FOVx;
gx_5b = Nx* k_x / T_read / gambar;
gx_5a = -2 * gx_5b;

t_gx_1 = 5.5;
t_gx_2 = 7.5;
t_gx_3 = 15.5;

gx_5 =  (time >= t_gx_1).*(time < t_gx_2) .* gx_5a ...
       + (time >= t_gx_2).*(time < t_gx_3) .* gx_5b;

bx_5 = b1_90;
by_5 = zeros([totalTimepoints obj_n]);
bz_5 = gx_5 * obj_x;

[mx_5,my_5,mz_5] = blochsim_516(m0,bx_5,by_5,bz_5,obj_T1,obj_T2,dt);
figure (5)
subplot(3,2,1)
plot(time,mx_3);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 3: without Gx');
subplot(3,2,3)
plot(time,my_3);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,2,5)
plot(time,mz_3);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
subplot(3,2,2)
plot(time,mx_5);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 5: with Gx');
subplot(3,2,4)
plot(time,my_5);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
subplot(3,2,6)
plot(time,mz_5);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
```



part 6

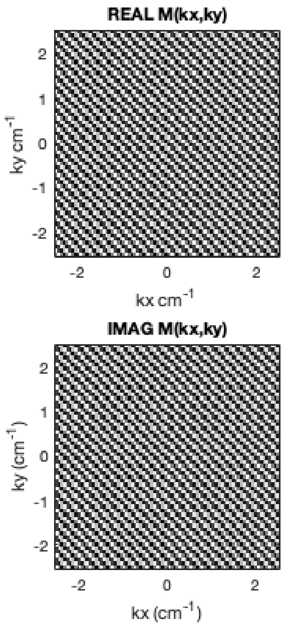
```
nread = 80;
npe = 60;
by_6 = zeros([totalTimepoints obj_n]);
sig_6 = zeros([nread 1]);
gy_6 = zeros([totalTimepoints 1]);
bz_6 = gx_5 * obj_x;
[mx_6,my_6,mz_6] = blochsim_516(m0,bx_5,by_6,bz_6,obj_T1,obj_T2,dt);
M_6 = mx_6 + 1i * my_6;
index_1 = 1;
for index = 1 : totalTimepoints
    if (index >= 151 && index <= 310) && mod(index,2) == 1
    sig_6(index_1,1) = M_6(index,1);
    index_1 = index_1 + 1;
    end
end
xpos = [-nread/2:nread/2-1]/nread.*FOVx;
ypos = [-npe/2:npe/2-1]/npe*FOVy;
figure(6)
plot(xpos,abs(ift(sig_6)));xlabel('x (cm)');ylabel('received signal');axis([-8 8 -1 1]);title('magnitude vs spatial position');
```



part 7

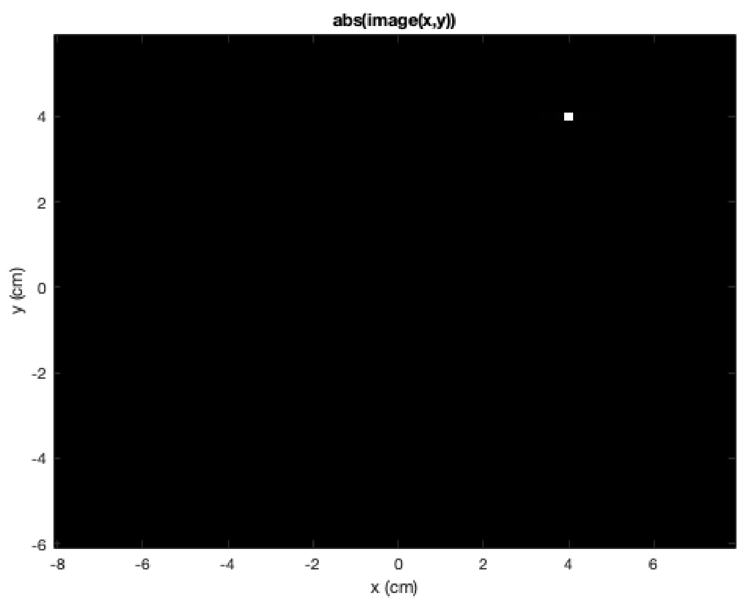
Create gy

```
k_y = 1/FOVy;
gy_max = Ny * k_y / gambar / T_y / 2;
delta_gy = 2 * pi / (gam * 2 *FOVy);
by_7 = zeros([totalTimepoints obj_n]);
sig_7 = zeros([nread npe]);
M_7 = zeros([totalTimepoints npe]);
for pe = 1:npe
    gy_7 = (time >= t_gx_1).*(time < t_gx_2) .* (delta_gy * (pe - 1) - gy_max);
    bz_7 = gx_5 * obj_x + gy_7 * obj_y + gz * obj_z;
    [mx_7,my_7,mz_7] = blochsim_516(m0,bx_5,by_7,bz_7,obj_T1,obj_T2,dt);
    M_7(:,pe)= mx_7 + 1i * my_7;
    index_1 = 1;
    for index = 1 : totalTimepoints
        if (index >= 151 && index <= 310) && mod(index,2) == 1
        sig_7(index_1,pe) = M_7(index,pe);
        index_1 = index_1 + 1;
        end
    end
end

% show images for parts 7-10
W_kx = k_x * Nx;
W_ky = k_y * Ny;
kxpos = linspace(-W_kx/2, W_kx/2, Nx);  % vector of kx locations
kypos = linspace(-W_ky/2, W_ky/2, Ny);  % vector of ky locations

figure(7)
subplot(2,1,1)
imagesc(kxpos,kypos,real(sig_7)'); colormap gray; axis('image'); axis('xy')
xlabel('kx cm^{-1}');
ylabel('ky cm^{-1}');
title('REAL M(kx,ky)')

% disp 'Press any key to continue...'; pause

subplot(2,1,2)
imagesc(kxpos,kypos,imag(sig_7)'); colormap gray; axis('image'); axis('xy')
xlabel('kx (cm^{-1})');
ylabel('ky (cm^{-1})');
title('IMAG M(kx,ky)')
%disp 'Press any key to continue...'; pause
```



part 8

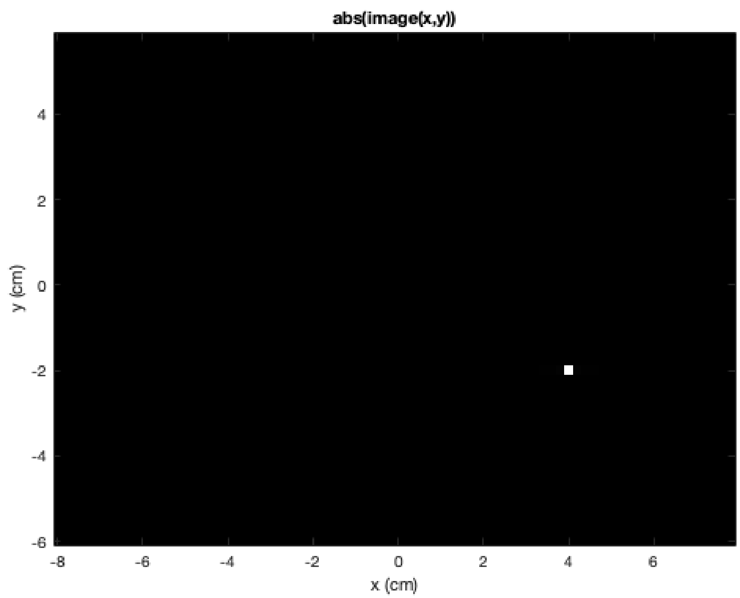
```
figure(8)
imagesc(xpos,ypos,abs(ift2(sig_7))'); colormap gray; axis('image'); axis('xy')
xlabel('x (cm)');
ylabel('y (cm)');
title('abs(image(x,y))')

%disp 'Press any key to continue...'; pause
```



part 9

```
by_9 = zeros([totalTimepoints obj_n]);
sig_9 = zeros([nread npe]);
index_1 = 1;
M_9 = zeros([totalTimepoints npe]);
for pe = 1:npe
    gy_9 = (time >= t_gx_1).*(time < t_gx_2) .* (delta_gy * (pe - 1) - gy_max);
    bz_9 = gx_5 * obj_x + gy_9 * 10 + gz * obj_z;
```

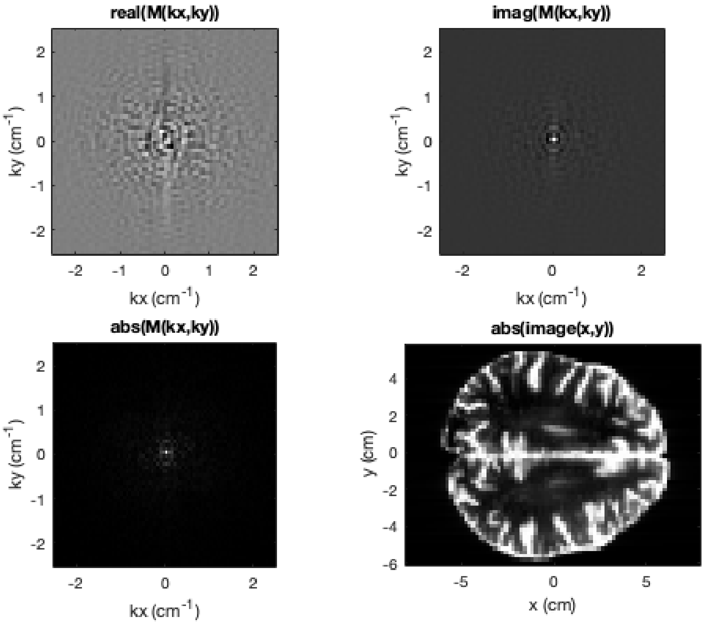
```
    [mx_9,my_9,mz_9] = blochsim_516(m0,bx_5,by_7,bz_9,obj_T1,obj_T2,dt);
    M_9(:,pe)= mx_9 + 1i * my_9;
    index_1 = 1;
    for index = 1 : totalTimepoints
        if (index >= 151 && index <= 310) && mod(index,2) == 1
        sig_9(index_1,pe) = M_9(index,pe);
        index_1 = index_1 + 1;
        end
    end
end

figure(9)
imagesc(xpos,ypos,abs(ift2(sig_9))'); colormap gray; axis('image'); axis('xy')
xlabel('x (cm)');
ylabel('y (cm)');
title('abs(image(x,y))')

%disp 'Press any key to continue...'; pause
```

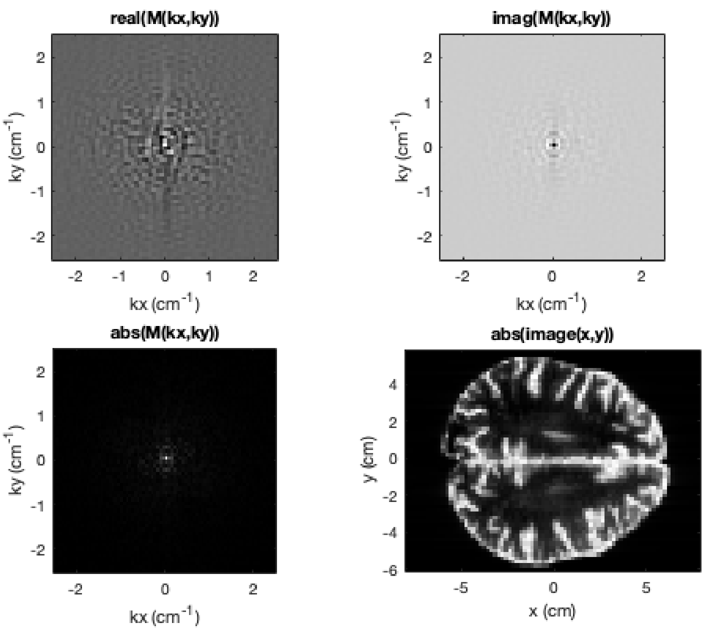


part 10 11

```
load object23.mat
b1_90_x = zeros([totalTimepoints 1]);
b1_90_y = zeros([totalTimepoints 1]);
obj_n = 2850;
m0_10 = [zeros([2 obj_n]);
         ones([1 obj_n])];
M_10 = zeros([totalTimepoints npe]);
sig_10 = zeros([nread npe]);
for slice = 1:2  % slice loop
    if slice == 1
        z_slice = 0;
    else
        z_slice = 1;
    end

    omega_shift = gam * gz1_a * z_slice;
    for tt = 1 : totalTimepoints
        b1_90_x(tt,:) = b1_90(tt,:) * cos(omega_shift * tt * dt);
        b1_90_y(tt,:) = b1_90(tt,:) * sin(omega_shift * tt * dt);
    end
        bx_10 = b1_90_x * ones([1 obj_n]);
        by_10 = b1_90_y * ones([1 obj_n]);
        gx_10 = gx_5;
        gz_10 = gz;
        %figure (10)
    for pe = 1:npe
```

```
        gy_10 = (time >= t_gx_1).*(time < t_gx_2) .* (delta_gy * (pe - 1) - gy_max);
        bz_10 = gx_10 * obj_x + gy_10 * obj_y + gz * obj_z;
        [mx_10,my_10,mz_10] = blochsim_516(m0_10,bx_10,by_10,bz_10,obj_T1,obj_T2,dt);
%         subplot(3,1,1)
%         plot(time,sum(mx_10,2)/obj_n);xlabel('time (ms)');ylabel('Mx');axis([0 endtime -1 1]);title('part 10');
%         subplot(3,1,2)
%         plot(time,sum(my_10,2)/obj_n);xlabel('time (ms)');ylabel('My');axis([0 endtime -1 1]);
%         subplot(3,1,3)
%         plot(time,sum(mz_10,2)/obj_n);xlabel('time (ms)');ylabel('Mz');axis([0 endtime -1 1]);
%         pause(0.01);
        M_10(:,pe) = sum(mx_10,2)+ 1i.*sum(my_10,2);
        index_1 = 1;
        for index = 1 : totalTimepoints
            if (index >= 151 && index <= 310) && mod(index,2) == 1
                sig_10(index_1,pe) = M_10(index,pe);
                index_1 = index_1 + 1;
            end
        end
    end

    %close(figure(10))

    if slice == 1
        figure(11)
    else
        figure(12)
    end

    subplot(2,2,1)
    imagesc(kxpos,kypos,real(sig_10)'); colormap gray; axis('image'); axis('xy')
    xlabel('kx (cm^{-1})');
    ylabel('ky (cm^{-1})');
    title('real(M(kx,ky))')


    subplot(2,2,2)
    imagesc(kxpos,kypos,imag(sig_10)'); colormap gray; axis('image'); axis('xy')
    xlabel('kx (cm^{-1})');
    ylabel('ky (cm^{-1})');
    title('imag(M(kx,ky))')

        subplot(2,2,3)
    imagesc(kxpos,kypos,abs(sig_10)'); colormap gray; axis('image'); axis('xy')
    xlabel('kx (cm^{-1})');
    ylabel('ky (cm^{-1})');
    title('abs(M(kx,ky))')

    subplot(2,2,4)
    imagesc(xpos,ypos,abs(ift2(sig_10))'); colormap gray; axis('image'); axis('xy')
    xlabel('x (cm)');
    ylabel('y (cm)');
    title('abs(image(x,y))')
end
```





Published with MATLAB® R2022a