Prof. Mark D Shattuck
Physics 34100 Applied Differential Equations
May 14, 2026

Final Project

Magnetic Spin Echo

(1) ***Introduction:*** The total magnetic moment of a sample is the sum of all of the magnetic moments $\vec{m}$ or "spins" in the sample. In experiments, the magnetic moment per unit volume or magnetization $\vec{M}$, which is proportionals to $\vec{m}$, can be measured. In a constant magnetic field $\vec{B}$ the magnetization $\vec{M}(t)$ will align with the field exponentially with a time constant T_1. The equation for this relaxation is:

$$\frac{d}{dt}\vec{M}(t) = \dot{\vec{M}} = \frac{\mu\vec{B} - \vec{M}}{T_1} = \frac{\vec{M}_\infty - \vec{M}}{T_1}, \tag{1}$$

where $\vec{M}_\infty = \mu\vec{B}$ is the equilibrium magnetization.

(a) What are the fixed point(s) of (1)?
(b) Show that $\vec{M} = (1 - e^{-t/T_1})\vec{M}_0$ is a solution to (1).
(c) What is the null solution?
(d) What is the full solution for an initial condition of $\vec{M}(0) = \vec{M}_0$?
(e) Describe in words or pictures what happens to the magnetization if it starts in the x-direction $\vec{M}(0) = M_0\hat{x}$ in a magnetic field that points in the z-direction $\vec{B} = B_0\hat{z}$?

It is often convenient to fix the direction of the static magnetic field $\vec{B} = B_0\hat{z}$ in the z-direction. In that case, the magnetization perpendicular to the the field is called the transverse magnetization $\vec{M}_{xy}$.

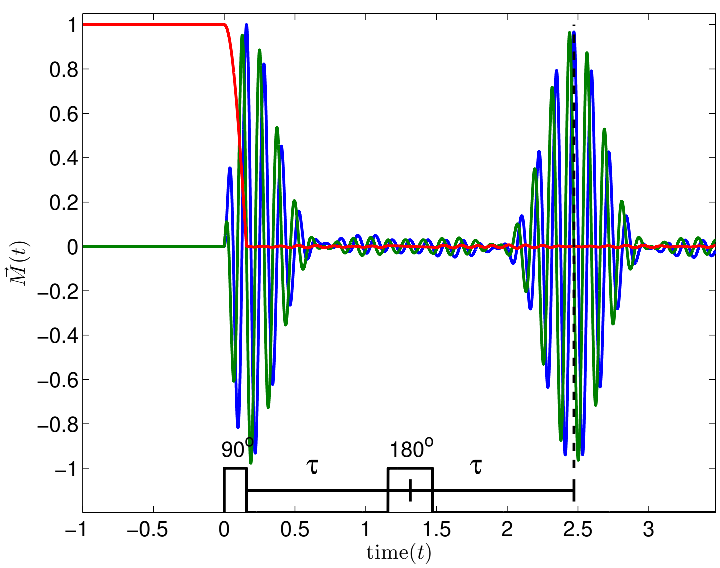

Figure 1. Plot of magnetization $\vec{M}(t)$ during a simulated spin echo using 1000 spins. M_z is red and $\vec{M}_{xy}$ are in blue and green. Timing of excitation the pulse $\vec{B}_{xy}(t)$ is shown below.

(2) ***Spin echo:*** A spin echo (see Figure 1) occurs when the transverse magnetization $\vec{M}_{xy}(t) = M_x(t)\hat{x} + M_y(t)\hat{y}$ seems to relax to zero after a time τ but by applying a special time-dependent transverse magnetic field $\vec{B}_{xy}(t) = B_x(t)\hat{x} + B_y(t)\hat{y}$ at $t = \tau$ the transverse magnetization returns at a time 2τ as an "echo". In this project we will simulate a magnetic "spin echo".

(3) ***Magnetization in a magnetic field:*** The magnetization $\vec{M}(t) = (M_x(t), M_y(t), M_z(t)) = M_x(t)\hat{x} + M_y(t)\hat{y} + M_z(t)\hat{z}$ of a sample in a constant magnetic field in the z-direction and a time-dependent field in the transverse direction $\vec{B} = (B_x(t), B_y(t), B_0) = (\vec{B}_{xy}(t), B_0)$ experiences a torque from the magnetic field in addition to the relaxation described above. It evolves according to the Bloch equations:

$$\dot{\vec{M}} = \gamma \vec{M} \times \vec{B} + \frac{M_0\hat{z} - \vec{M}}{T_1}, \tag{2}$$

where γ is the gyromagnetic ratio, $M_0\hat{z} = \left\langle \mu\vec{B} \right\rangle = \mu B_0 \hat{z}$ is the equilibrium magnetization for the time averaged magnetic field $\left\langle \vec{B} \right\rangle$. This requires that the time average of the transverse field is zero, $\left\langle \vec{B}_{xy} \right\rangle = \vec{0}$. $\vec{M} \times \vec{B}$ is the cross product of $\vec{M}$ and $\vec{B}$, and T_1 is the relaxation time.

(a) Rewrite the equation for $\vec{M}$ in this matrix form:

$$\dot{\vec{M}} = A\vec{M} + \vec{b} = \vec{F}$$

and find A and $\vec{b}$.

(b) Solve (2) using MATLAB for $\vec{B} = \hat{z}$, $T_1 = 10$,$M_0 = 1$,$\gamma = 1$, and initial condition $\vec{M}(0) = \hat{x}$ on the time interval $t = 0$–100.

(i) To setup for `ode45` make a function `blochEq.m` which returns `F`. Here is a start. I defined $\vec{\omega} = \gamma\vec{B}$ as `w=g*B`, which is called the Larmor frequency. I filled in `Fx`=F_x. You should add F_y and F_z.

```
function F=blochEq(M,B,T1,M0,g)
% blochEq <Bloch Equation>
% Usage:: F=blochEq(M,B,T1,M0,g)
%
% Return F the rhs of Bloch equation.
% M is 3x1 magnetization vector
% B is 3x1 magnetic field vector
% T1 is the relaxation time T_1
% M0 is the equilibrium magnetization M_0
% g is \gamma the gyromagnetic ratio

% revision history:
% 05/11/2026 Mark D. Shattuck <mds> blochEq.m

%% Main
w=g*B;  % (short cut) Larmor frequency vector

% rhs of Bloch equation
F=[-M(1)/T1+M(2)*w(3)-M(3)*w(2);...
   <fill in Fy here>;...
   <fill in Fz here>];
```

(ii) Use $\vec{M} = (1,2,3)$, $\vec{B} = (4,5,7)$, $T_1 = 3$, $M_0 = 2$, and $\gamma = 4$ to test if it is working:

$$3\vec{F} = 3\left(\gamma\vec{M}\times\vec{B} + \frac{M_0\hat{z} - \vec{M}}{T_1}\right)$$
$$= 3\left(4\begin{bmatrix}1\\2\\3\end{bmatrix}\times\begin{bmatrix}4\\5\\7\end{bmatrix} + \frac{1}{3}\left(\begin{bmatrix}0\\0\\2\end{bmatrix} - \begin{bmatrix}1\\2\\3\end{bmatrix}\right)\right)$$
$$= 12\begin{bmatrix}-1\\5\\-3\end{bmatrix} + \begin{bmatrix}-1\\-2\\-1\end{bmatrix} = \begin{bmatrix}-12\\60\\-36\end{bmatrix} + \begin{bmatrix}-1\\-2\\-1\end{bmatrix} = \begin{bmatrix}-13\\58\\-37\end{bmatrix}$$

```
>> 3*blochEq([1;2;3],[4;5;7],3,2,4)
ans =
  -13.0000
   58.0000
  -37.0000
```

(iii) `ode45` needs a function `F(t,M)`. To convert `blochEq` to a function of only `t` and `M` use an autonomous function:
`F=@(t,M)blochEq(M,B,T1,Meq,g)`.
To use this function `B`, `T1`, `Meq`, and `g` must be defined earlier. Then `ode45` can be called with `F`. Here is an example for $\vec{B} = \hat{z}$, $T_1 = 10$,$M_0 = 1$,$\gamma = 1$, and initial condition $\vec{M}(0) = \hat{x}$.

```
B=[0;0;1];     % Magnetic field
T1=10;         % Relaxation time
Meq=1;         % Equilbrium magnetization
g=1;           % Gyromagnetic ratio
M0=[1;0;0];    % Initial Condition

F=@(t,M) blochEq(M,B,T1,Meq,g);
[t,M]=ode45(F,[0 100],M0);
plot(t,M)
```

You do not have to type the comments `%`. Use this to plot the solution two ways as shown in Figure 2.
For (a) use `plot3(M(:,1),M(:,2),M(:,3))`.
For (b) use `plot(t,M)`.
Plotting the vectors in (a) is optional but the full code to make the plots is in the appendix, if you are interested.

(iv) Change `T1`, `g`, and `M0` to get a feel for how they change the solution. It may be easier to include `F` in the `ode45` command directly so you do not have to keep redefining `F`. In this example `T1` is doubled:
`[t,M]=ode45(@(t,M)blochEq(M,B,2*T1,Meq,g),[0 100],M0);plot(t,M)`
This one changes `M0`:
`[t,M]=ode45(@(t,M)blochEq(M,B,T1,Meq,g),[0 100],[1;0;-1]);plot(t,M)`

(v) Explain in words or pictures how `T1` and `g` change the solution.

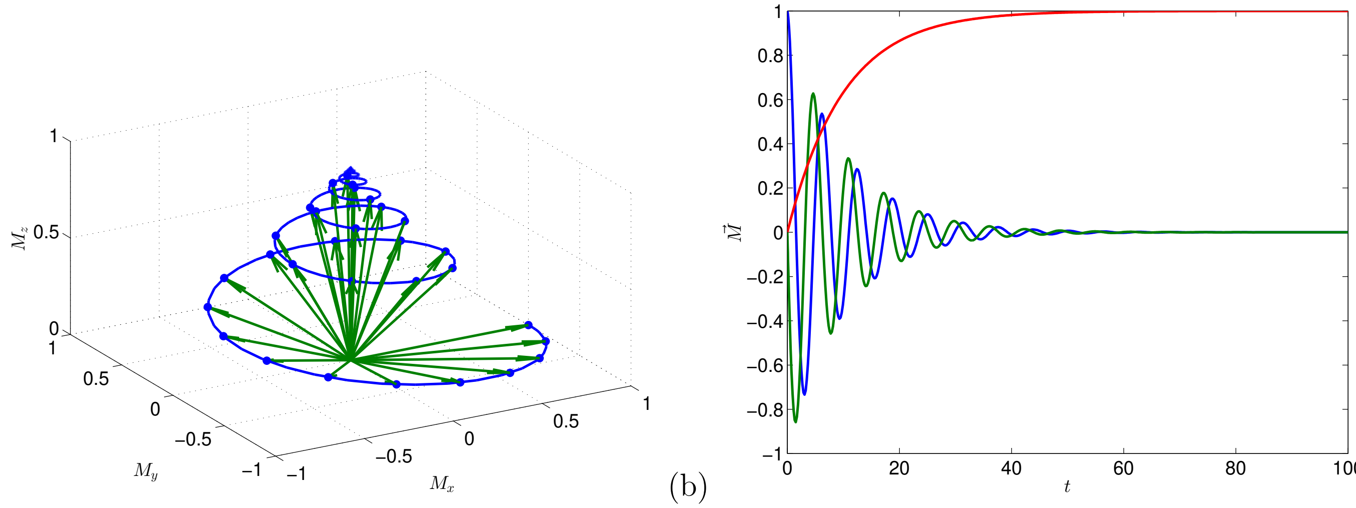

FIGURE 2. Solution to the Bloch equations for $\vec{B} = \hat{z}$, $T_1 = 10$,$M_0 = 1$,$\gamma = 1$, and initial condition $\vec{M}(0) = \hat{x}$. (a) 3D trajectory of $\vec{M}(t)$. (b) Plot of $\vec{M}(t)$ with $\vec{M}_x(t)$ blue, $\vec{M}_y(t)$ green, $\vec{M}_z(t)$ red.

(4) ***Excitation, the $\frac{\pi}{2}$-pulse:*** To create a spin echo in the transverse magnetization we need transverse magnetization. When we start, the system is in equilibrium and $\vec{M}(0) = M_0\hat{z}$. The first step is to use a transverse magnetic field to create transverse magnetization.
  (a) Show that $\vec{M}(0) = M_0\hat{z}$ is a fixed point for (2) with $\vec{B} = B_0\hat{z}$. (Note: $\vec{z} \times \vec{z} = 0$).
  (b) Use `ode45` to check the stability of the fixed point.
  (Hint: `ode45(@(t,M)blochEq(M,B,T1,Meq,g),[0 100],[0;0;1]+randn(3,1)/10)` will add `randn/10` to $\vec{M}(0)$.)

To create transverse magnetization we will need a significant perturbation to overcome the stability of the equilibrium solution.
  (c) Show that (2) reduces to the following if $\vec{B} = B_0\hat{z}$ and $T_1 = \infty$ (`T1=inf`):

$$\begin{aligned} \dot{M}_x &= \omega_0 M_y \\ \dot{M}_y &= -\omega_0 M_x \\ \dot{M}_z &= 0 \\ \omega_0 &= \gamma B_0 \end{aligned} \tag{3}$$

  (d) Show that $\vec{M} = (M_0\cos(\omega_0 t + \phi), -M_0\sin(\omega_0 t + \phi), 0)$ is a solution to (3).

With no relaxation $T_1 = \infty$ and no transverse magnetic field the initial magnetization $\vec{M}_0$ rotates around the z-axis at the Larmor frequency ω_0. This suggests that it might be useful to view $\vec{M}(t)$ in a frame of reference that is rotating at the same frequency ω_0. Here is a function to perform the rotation:

```
function [MR,Mxr,Myr,Mz]=rotFr(t,M,w)
% rotFr <Convert to rotating frame.>
% Usage:: MR=rotFr(t,M,w)
%

% revision history:
% 05/13/2026 Mark D. Shattuck <mds> rotFr.m

%% Main
Mx=M(:,1);
My=M(:,2);
Mz=M(:,3);
Mxr=Mx.*cos(w*t)-My.*sin(w*t);
Myr=My.*cos(w*t)+Mx.*sin(w*t);
MR=[Mxr Myr Mz];
```

Use the following to test the code:

```
B0=1;                         % Static magnetic field
B=[0;0;B0];                   % Magnetic field
T1=.5;                        % Relaxation time
Meq=1;                        % Equilibrium magnetization
g=50;                         % Gyromagnetic ratio
M0=[0;sin(pi/3);cos(pi/3)];   % Initial Condition

F=@(t,M) blochEq(M,B,T1,Meq,g);
[t,M]=ode45(F,[0 1],M0);
MR=rotFr(t,M,g*B0);
```

To see what is going on look at `M` and `MR` in two and three dimensions using `plot(t,M)` and `plot3(M(:,1),M(:,2),M(:,3))`. In the rotating frame the magnetization acts like eq. (1). If $T_1 = \infty$ then the magnetization is constant. If we apply a magnetic field

$$\vec{B}_{xy}(t) = (B_1 \cos(\omega_0 t), -B_1 \sin(\omega_0 t))$$

that is also rotating at the same frequency ω_0 then it will provide a constant torque the magnetization in the rotating frame. Here is an example plot:

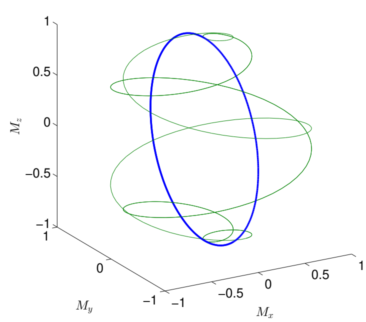

FIGURE 3. The magnetization rotates about the y-axis in the rotating frame (blue). The trajectory in the lab frame is shown in green.

Here is the code to rotate $\vec{M}(0)$ about the $y-axis$:

```
B0=1;         % Static magnetic field strength
Bs=[0;0;B0]; % Static Magnetic field vector
B1=1/5;       % Strength of the rotating Bxy field
T1=inf;       % Relaxation time
Meq=1;        % Equilibrium magnetization
g=50;         % Gyromagnetic ratio
M0=[0;0;1];   % Initial Condition

ts=[0 1];

opts=odeset('reltol',1e-4);  % Increase accuracy

w=g*B0;
B=@(t) [B1*cos(w*t);-B1*sin(w*t);0]+Bs; % Magnetic field
[t,M]=ode45(@(t,M) blochEq(M,B(t),T1,Meq,g),ts,M0,opts);
```

In this code, we need better accuracy so I have included `odeset('reltol',1e-4)`.

(e) `Plot3(M(:,1),M(:,2),M(:,3))`. Compare to `MR=rotFr(t,M,g*B0)`. It should look like Figure 3.

(f) `Plot(t,M)` for the code above. Notice that M_z is oscillating with frequency $\omega_1 = \gamma B_1$ and a period of $\frac{2\pi}{w_1}$.

Since the magnetization is rotating at a fixed rate, we can stop it when all of the magnetization is in the transverse plane. This occurs after a rotation of $\frac{\pi}{2}$ or 90°.

(g) Find the time t_{90} (`t90`) so that $\omega_1 t_{90} = \frac{\pi}{2}$. Set `ts=[0 t90]`. Plot the result like Figure 3.

This code represents a $\frac{\pi}{2}$-excitation about the x-direction. A function `aPulse.m` to generalize this to an α-excitation about an arbitrary direction $\hat{\phi}$ is in the appendix and in the zip file spinecho.zip.

(h) Look over `aPulse.m` and make sure you understand it.

(5) ***Simulating the spin echo:*** We now have all of the parts needed to simulate a spin echo. We use a $\frac{\pi}{2}$ excitation to rotate the initial equilibrium magnetization, which points z-direction, by 90° about the x-direction to produce transverse magnetization pointing in the y-direction. Then we let the spins evolve for a time `Te` with $\vec{B}_{xy} = 0$. To make the echo we then rotate by 180° about the y-axis. Then we let the system evolve again with $\vec{B}_{xy} = 0$. So the outline of the code is:

(a) Set all parameters.

(b) Apply a 90° rotation about x using `aPulse`.

(c) Evolve freely with $\vec{B}_{xy} = 0$ for a time `Te`.

(d) Apply a 180° rotation about y using `aPulse`.

(e) Evolve freely with $\vec{B}_{xy} = 0$ for a time `2*Te`.

(f) Concatenate the times `t` and magnetization `M` from each step.

After each step we will use the final state of the system for the initial conditions for the next step. A script `spinecho.m` to implement this is in the appendix and in the zip file spinecho.zip.

(a) Test `spinecho`. Set `Ns=1` and `noise=0` to simulation one spin with no noise. The result should look like Figure 4.

This is not a very interesting echo. With only one spin we rotate from z to y then rotate about y which does nothing. You can look at `MR` in the rotating frame as well to see that the 180° pulse does not do anyting.

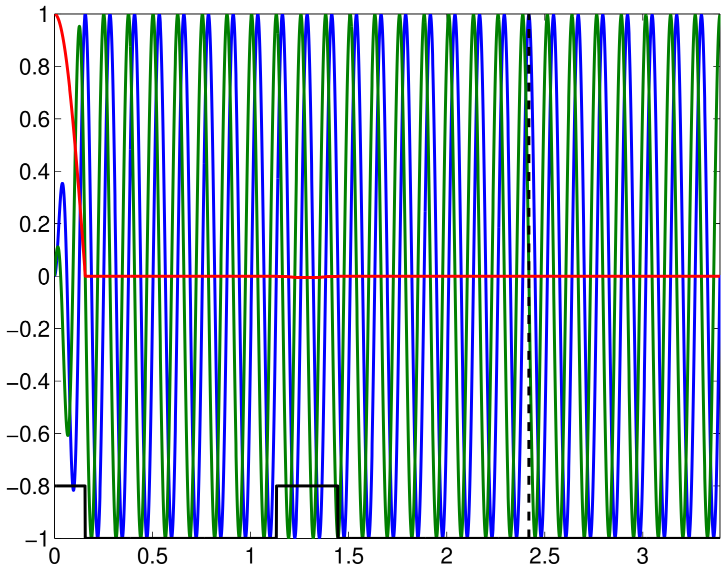

FIGURE 4. One spin, no noise.

(b) In a system of spins the local magnetic environment differs from spin to spin. To see this set `noise=1/10`. Describe what you see.

If the local magnetic field is different from the average magnetic field then the local Larmor frequency will vary and the spin will not be stationary in the rotating frame. They will slowly rotate with an angular frequency Ω proportional to the difference between the local and average magnetic field. If we add a lot of spins all slowly rotating in the transverse plane then the net signal will cancel out. Instead of all spins pointing in the same direction they will point in all directions and the net transverse magnetization will be ~ 0.

(c) Set the number of spins `Ns=100` and run `spinecho`. As it runs it adds each spin in with a different random magnetic field. Describe what you see.

When we apply the 180° pulse about the y-axis it flips $M_x \to -M_x$. Imagine a spin that is moving clockwise with a frequency Ω starting from the y-axis at 12 on the clock. After a time T it is at 2. When it is flipped by the 180° pulse, then it goes across to 10 but it is still moving at a clockwise speed Ω, so after another interval T it will return to 12. This is true for all Ω so every spin will come back in a echo at the same time.

(d) Change some of the parameters and report the effects. If you change things too much then you may violate some of our assumptions and things will not work.

(6) ***Conclusion:*** Put all of your answers, figures, drawing, etc into a single pdf with enough explanation for me to understand what you did. Add the pdf and all of your MATLAB files into a single zip file and upload it to Brightspace.

Appendix

```
function [t,M,ta]=aPulse(a,M,B1,w,ax,B0,T1,M0,g,dt)
% aPulse <alpha-pulse>
% Usage:: [t,M,ta]=aPulse(a,B1,B0,T1,M0,g)
%
% rotates M about the ax-axis by an angle a using a resonant
% field (B1*cos(w*t+p),B1*sin(w*t+p),B0);
% M is the current 3x1 magnetization vector
% B1,w rotating transverse magnetic field strength vector
%      and frequency
% ax Axis of rotation x-axis [1 0] y-axis [0 1] etc
% B0 is static magnetic field vector
% T1 is the relaxation time T_1
% M0 is the equilibrium magnetization M_0
% g is \gamma the gyromagnetic ratio
% dt time step for output

% revision history:
% 05/13/2026 Mark D. Shattuck <mds> aPulse.m

%% Main
p=atan2(ax(2),ax(1)); % phase angle to set axis of rotation
ta=a/g/B1;            % Time for a alpha-pulse
ts=0:dt:ta;           % Time points for output

opts=odeset('reltol',1e-4);  % Higher accuracy

% Magnetic field
B=@(t) [B1*cos(w*t+p);-B1*sin(w*t+p);0]+B0;

% Solve blochEq
[t,M]=ode45(@(t,M) blochEq(M,B(t),T1,M0,g),ts,M,opts);
```

```
%% mds spin echo simulation for final project
% 5/13/2026

%% Parameters
Ns=100;       % Number of spins
B0=1;         % Static B-field in z-dir
g=50;         % Gyromagnetic ratio
Meq=1;        % Equilbrium magnetization
T1=inf;       % Relaxation time
B1=1/5;       % Excitation strength
M0=[0;0;Meq]; % Initial magnetization in equilbrium
Te=1;         % Approx time between 90 and 180
noise=1/10;   % Strength of noise in B

%% Calculations
w1=g*B1;       % Excitation frequency
w0=g*B0;       % Larmor frequency
T0=2*pi/g/B0;  % Larmor period
dt=T0/100;     % 100 time steps per Larmor period

T90=pi/g/B1/2; % Time for a 90-pulse

% Te must be a multiple of T0 for phase coherence
Te=fix((T90+Te)/T0)*T0-T90;


xa=[1 0];     % x-axis
ya=[0 1];     % y-axis
M=0;  % Initialize M

%% Loop over all spins
for n=1:Ns
  disp(n);  % User feedback

  Bv0=[0;0;B0];            % Average B vector
  Bs=Bv0+noise*randn(3,1); % Add noise to each spin

  % Set blochEq to solve during free evolution
  Free=@(t,M) blochEq(M,Bs,T1,Meq,g);

  % Apply 90-deg rotation about the x-axis
  % this will rotate (0,0,Meq) into (Meq,0,0)
  % Note: use Bv0 to approximate fast rotation ie T90<<Te
  [t90,M90,ta]=aPulse(pi/2,M0,B1,w0,xa,Bv0,T1,Meq,g,dt);

  % Let evolve from last point of M90 for Te time
  ts=0:dt:Te;
  [t1,M1]=ode45(Free,ts,M90(end,:));


  % Apply 180-deg rotation about the y-axis
  % this will rotate flip spins causing them to reverse
  % Note: use Bv0 to approximate fast rotation ie T90<<Te
```

```
  % Start from last point of M1
  [t180,M180,tb]=aPulse(pi,M1(end,:),B1,w0,ya,Bv0,T1,Meq,g,dt);

  % Let evolve from last point of M180 for 2*Te time
  ts=0:dt:2*Te;
  [t2,M2]=ode45(Free,ts,M180(end,:));

  % Concatenate all of the times and M's together
  t=[t90;ta+t1;ta+Te+t180;ta+Te+tb+t2];
  M=M+[M90;M1;M180;M2];

  tau=ta+Te;        % time from center of 180 to echo

  % Nice plot
  MR=rotFr(t,M,g*B0);
  h=plot(t,M/n,...
    t,.2*((t<ta)+(t>ta+Te & t<ta+Te+tb))-1,'k',...
    ta+tb/2+Te+tau*[1 1],[-1 1],'k--');
  axis([0 t(end) -Meq Meq]);
  drawnow;

end
```

```
%% 3D plot of M
% mds 5/2026

%% Parameters
B=[0;0;1];   % Magnetic field
T1=10;       % Relaxation time
Meq=1;       % Equilbrium magnetization
g=1;         % Gyromagnetic ratio
M0=[1;0;0];  % Initial Condition
ts=[0 100];  % time span

N=30;                    % Number of arrows
t=-T1*log(1-(0:N-1)/N);  % Equal spacing

%% Trajectory
[¬,M]=ode45(@(t,M) blochEq(M,B,T1,Meq,g),ts,M0);
plot3(M(:,1),M(:,2),M(:,3),'linewidth',2);

%% Vector evolution
hold all;

[t,M]=ode45(@(t,M) blochEq(M,B,T1,Meq,g),t,M0);
h=quiver3(0*M(:,1),0*M(:,2),0*M(:,3),M(:,1),M(:,2),M(:,3),0);
set(h,'linewidth',2);
h=plot3(M(:,1),M(:,2),M(:,3),'b.');
set(h,'linewidth',2,'markersize',20)

%% Look nice
daspect([1 1 1]);        % Axes equal
grid on;                 % Grid
set(gca,'fontsize',15);  % Fontsize larger

xlabel('$$M_x$$','interp','latex'); % XYZ labels
ylabel('$$M_y$$','interp','latex');
zlabel('$$M_z$$','interp','latex');
view([-30 20])           % Orientation

hold off;
```

```
%% 2D plot of M vs t
% mds 5/2026

%% Parameters
B=[0;0;1];    % Magnetic field
T1=10;        % Relaxation time
Meq=1;        % Equilbrium magnetization
g=1;          % Gyromagnetic ratio
M0=[1;0;0];   % Initial Condition

%% M vs t
[t,M]=ode45(@(t,M) blochEq(M,B,T1,Meq,g),ts,M0);
plot(t,M,'linewidth',2);

%% Look nice
set(gca,'fontsize',15);  % Fontsize larger
xlabel('$$t$$','interp','latex');          % XY labels
ylabel('$$\vec{M}$$','interp','latex')
```