

3.39. we can use Bloch-equations.

(a) after  $180^\circ$  pulse

$$\begin{cases} M_x' = -M_0 \\ M_y' = 0 \\ M_z' = 0 \end{cases}$$

Bloch equations:

$$\begin{cases} \frac{dM_x}{dt} = \gamma M_z(t) \cdot B_y \\ \frac{dM_y}{dt} = -\gamma M_z(t) \cdot B_x \\ \frac{dM_z}{dt} = \gamma (M_x(t) \cdot B_y - M_y(t) \cdot B_x) - (M_z(t) - M_0)/T_1 \end{cases}$$

(b) after  $90^\circ$  pulse

$$\begin{aligned} M_x' &= \sqrt{M_x^2 + M_y^2} \\ &= |M_0| \end{aligned}$$

```
import numpy as np
import matplotlib.pyplot as plt

#Initial parameters
gamma = 42.58;
B1x = 1;
B1y = 1;
M0 = 1;
T1 = 1;
dt = 0.001;
t = np.arange(0, 10, dt);

Mx = np.zeros(len(t))
My = np.zeros(len(t))
Mz = np.zeros(len(t))

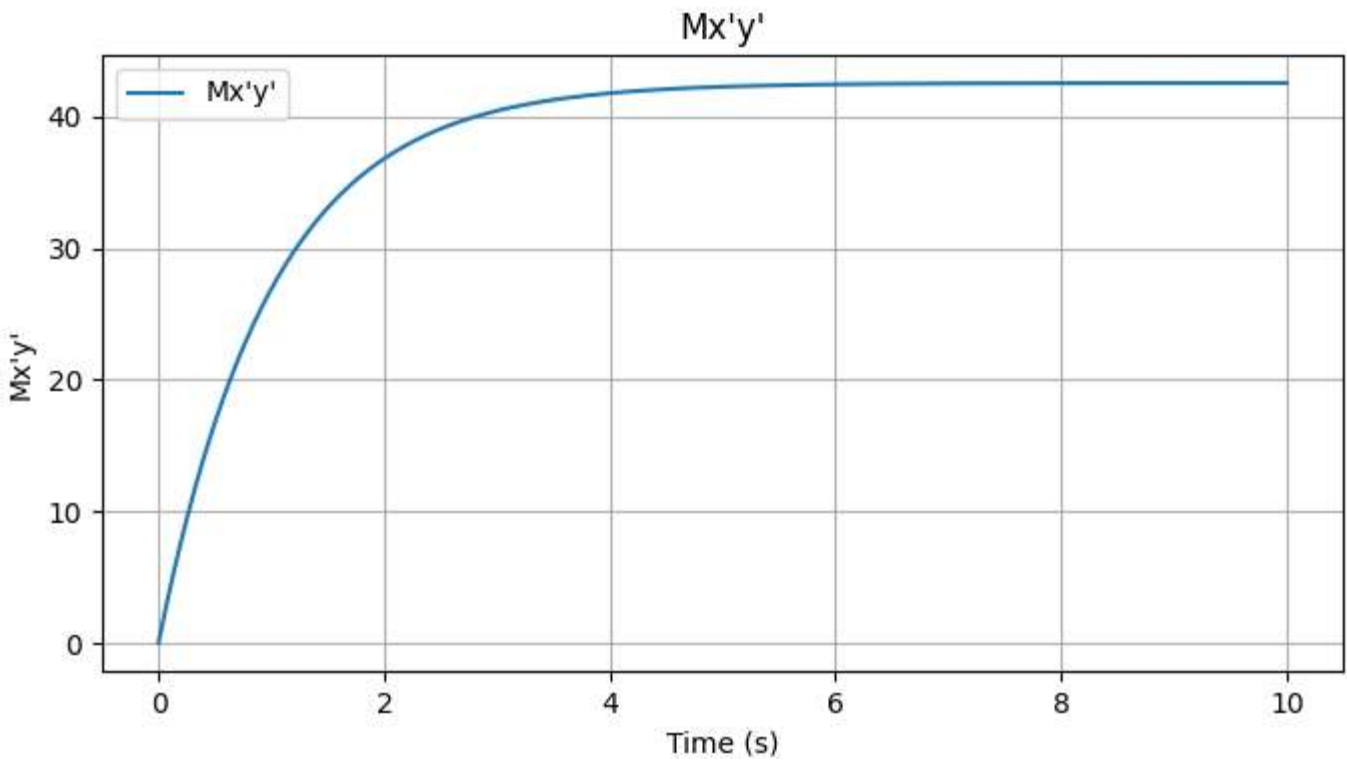
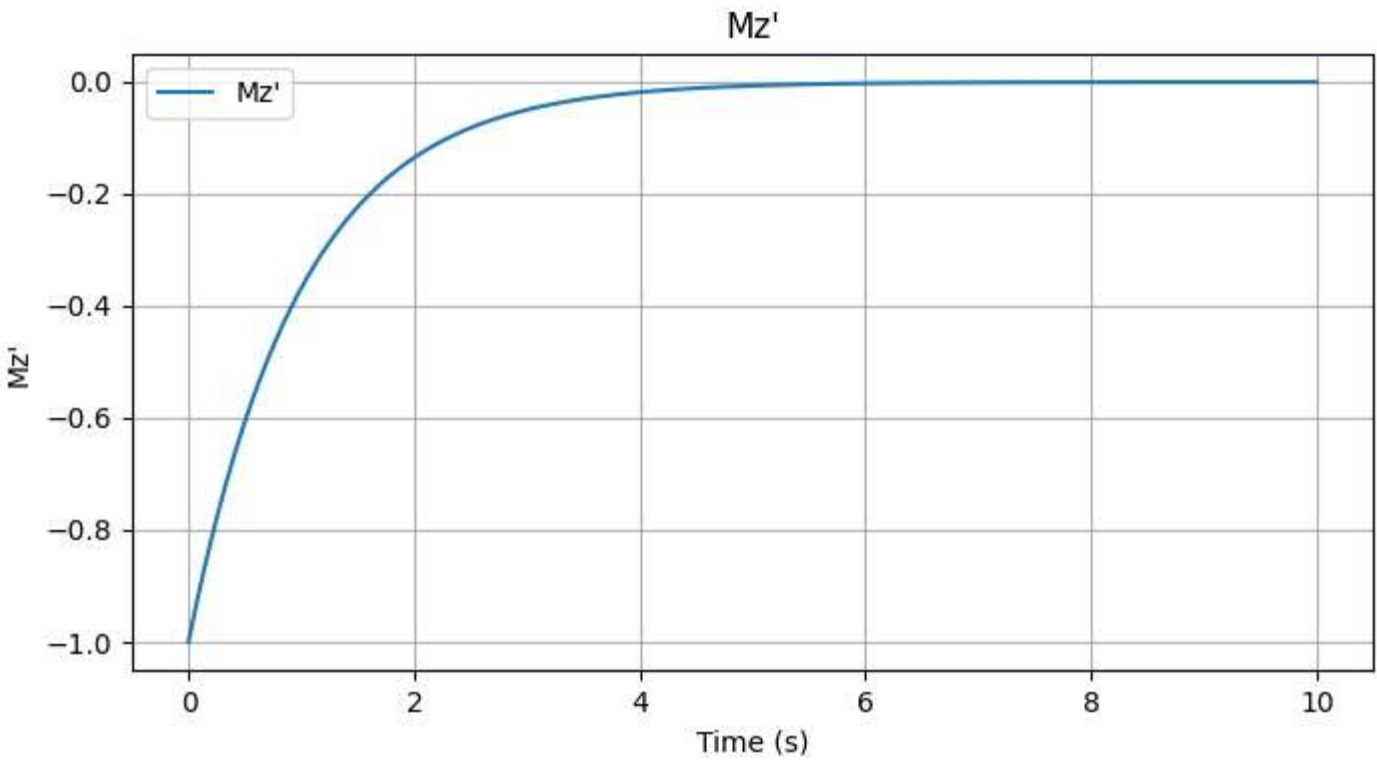
#After 180xpulse
Mx[0] = 0
My[0] = 0
Mz[0] = -M0

for i in range(1, len(t)):
    dMx = gamma * (Mz[i-1] * B1x)
    dMy = 0
    dMz = -Mz[i-1] / T1

    Mx[i] = Mx[i-1] + dMx * dt
    My[i] = My[i-1] + dMy * dt
    Mz[i] = Mz[i-1] + dMz * dt

plt.figure(figsize=(8, 4))
plt.plot(t, Mz, label="Mz'")
plt.title("Mz'")
plt.xlabel("Time (s)")
plt.ylabel("Mz'")
plt.legend()
plt.grid()
plt.show()

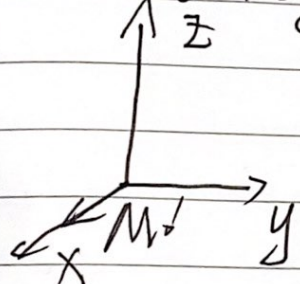
#$Mx'y'=\sqrt{Mx^2+My^2}$
Mxy = np.sqrt(Mx[0]**2 + My[0]**2)
```





4.11 (a) after a  $90^\circ_x$ -pulse.

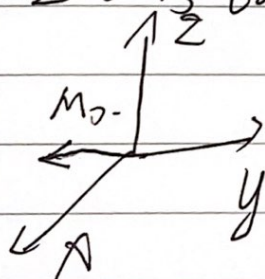
The  $90^\circ_x$  pulse flips the magnetization from the z-axis to the x-axis. Immediately after this pulse, the magnetization vector lies in the x-axis.



$$\text{while } \begin{cases} M'_x = M_0 \\ M'_y = 0 \\ M'_z = 0 \end{cases}$$

after a  $90^\circ_y$ -pulse.

The  $90^\circ_y$  pulse flips the magnetization from the z-axis to the minus y-axis like



$$\begin{cases} M'_x = 0 \\ M'_y = -M_0 \\ M'_z = 0 \end{cases}$$

(b).  $S_1(t)$  and  $S_2(t)$  are the FID signals generated after the two different pulses.

On the same ~~in~~ initial state. The magnitude of them is the same while they have a phase difference of  $90^\circ$ . It's  $S_1(t) = S_2(t) \cdot e^{-\frac{\pi}{2}i}$  or  $S_1(t) = S_2(t - \frac{\pi}{2\omega})$ .