Lecture-03A — Bloch Equation Matrix Simulations
Matrix Operations for Nutation, Relaxation and Precession

Brian Hargreaves
bah@stanford.edu
Learning Objectives

• Write the Bloch Equation in matrix form
• Write matrix operations for relaxation, nutation and precession
• Combine successive matrix operations
• Explain how to transform to other matrix formulations
Bloch Equation

• Basic Bloch Equation:

\[
\frac{d\vec{M}}{dt} = \vec{M} \times \gamma \vec{B} - \frac{\vec{M}_{xy}}{T_2} + \frac{\vec{M}_0 - \vec{M}_z}{T_1}
\]

- \(\vec{M}\) is magnetization, \(M_{xy}\) and \(M_z\) are transverse and longitudinal components
- \(\vec{B}\) is magnetic field, \(T_{1,2}\) relaxation times
- \(\gamma\) is the gyromagnetic ratio, \(M_0\) is the equilibrium magnetization
Matrix Bloch Equation

• Basic Bloch Equation:
  \[
  \frac{d\vec{M}}{dt} = \vec{M} \times \gamma \vec{B} - \frac{\vec{M}_{xy}}{T_2} + \frac{\vec{M}_0 - \vec{M}_z}{T_1}
  \]

• In Matrix form with:
  \[
  M = \begin{bmatrix}
  M_x \\
  M_y \\
  M_z
  \end{bmatrix}
  \]

  (usually set \( M_0 = 1 \))

• Becomes:
  \[
  \frac{dM}{dt} = \begin{bmatrix}
  -1/T_2 & \gamma B_z & -\gamma B_y \\
  -\gamma B_z & -1/T_2 & \gamma B_x \\
  \gamma B_y & -\gamma B_x & -1/T_1
  \end{bmatrix} M + \begin{bmatrix}
  0 \\
  0 \\
  M_0/T_1
  \end{bmatrix}
  \]

Standard right-handed cross product
Relaxation

- Over time period $\tau$

$$M' = \begin{bmatrix} E_2 & 0 & 0 \\ 0 & E_2 & 0 \\ 0 & 0 & E_1 \end{bmatrix} M + \begin{bmatrix} 0 \\ 0 \\ M_0(1 - E_1) \end{bmatrix}$$

$$E_1 = e^{-\tau/T_1}$$
$$E_2 = e^{-\tau/T_2}$$

$(\text{See relax.m})$

$$[AB] = \text{relax}(0.5,0.5,0.1,1)$$

$$AB = \begin{bmatrix} 0.0067 & 0 & 0 & 0 \\ 0 & 0.0067 & 0 & 0 \\ 0 & 0 & 0.3679 & 0.6321 \end{bmatrix}$$
Question 1

If \( T_1 = 1s \) and \( T_2 = 0.5s \), what are the relaxation matrix diagonal elements for \( \tau = 0.5s \)?

A. \[
\begin{bmatrix}
0.37 & 0.37 & 0.61
\end{bmatrix}
\]
B. \[
\begin{bmatrix}
0.61 & 0.61 & 0.37
\end{bmatrix}
\]
C. \[
\begin{bmatrix}
0.37 & 0.61 & 0.37
\end{bmatrix}
\]
D. \[
\begin{bmatrix}
0.61 & 0.61 & 0.78
\end{bmatrix}
\]

A. First, \( E_1 > E_2 \). Second, \( e^{-1} \sim 0.37 \). It is good to remember these numbers!
RF Rotations

- For a flip angle $\alpha$

\[
M' = R_x(\alpha)M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} M
\]

\[
M' = R_y(\alpha)M = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} M
\]

\[
\alpha = \gamma B_1 \tau
\]

Rotations are left-handed (also achieved with negative $\gamma$)
Question 2

Describe the (left-handed) rotation given by this matrix:

A. $30^\circ$ rotation about $m_y$
B. $30^\circ$ rotation about $-m_y$
C. $30^\circ$ rotation about $m_x$
D. $60^\circ$ rotation about $-m_y$
E. $60^\circ$ rotation about $m_y$
F. $60^\circ$ rotation about $m_x$
RF Rotations (Arbitrary $B_1$ phase, Axis in Transverse Plane)

- Over time period $\tau$

$$R_{\phi}(\alpha) = \begin{bmatrix} \cos^2 \phi + \sin^2 \phi \cos \alpha & \cos \phi \sin \phi(1 - \cos \alpha) & -\sin \phi \sin \alpha \\ \cos \phi \sin \phi(1 - \cos \alpha) & \sin^2 \phi + \cos^2 \phi \cos \alpha & \cos \phi \sin \alpha \\ \sin \phi \sin \alpha & -\sin \alpha \cos \phi & \cos \alpha \end{bmatrix}$$

$$\phi = \tan^{-1}(B_y/B_x)$$

- Rotation axis in x-y plane

- Note: $R_{\phi}(\alpha) = R_z(-\phi)R_x(\alpha)R_z(\phi)$
RF Rotations (Completely Arbitrary B₁ Direction)

- \( R_{\theta,\phi}(\alpha) = R_z(-\phi) \ R_y(-\theta) \ R_z(\alpha) \ R_y(\theta) \ R_z(\phi) \)

\[ \phi = \tan^{-1}(B_y/B_x) \]
\[ \theta = \tan^{-1}(|B_{xy}|/B_z) \]
Gradient / $\Delta B_0$ Rotations

- Over time period $\tau$

$$M' = R_z(\theta)M = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} M$$

$$\theta = \gamma (G \cdot \vec{r} + \Delta B_0) \tau$$

Rotations are left-handed (also achieved with negative $\gamma$)

See zrot.m
Have Fun with MRI! The Lab and Rotating Frames

“Lab” View

“Rotating” View
Question 3

Describe what operations this matrix simulates?

A. 30° rotation about m_z
B. -45° rotation about m_z
C. None of these

E. This is rotation AND transverse relaxation. The rotation is -45° about m_z because of the position and relative magnitude of cosines/sines, and there is scaling of m_x and m_y based on the relaxation. Here there is no longitudinal relaxation (infinite T_1).
Perfect Spoiling

• Explicitly set transverse magnetization to zero

\[
M' = R_{\text{spoil}} M = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix} M
\]
Matrix Propagation

Rotations / Relaxation: \( M' = AM + B \)

- \( M_1 = A_1M_0 + B_1 \) \( (M_0 \) is starting \( M \), not equilibrium \( M \)!) \n- \( M_2 = A_2M_1 + B_2 \) \( \ldots \) \( M_n \) \( (n \) “operations”\)

Propagation: multiply A’s, sum B’s after multiplying by all successive A’s

- \( M_2 = (A_2A_1)M_0 + A_2B_1 + B_2 \)

\[
A = \prod_{i=n}^{1} A_i \quad B = \sum_{i=1}^{n} \left( \prod_{j=n}^{i} A_j \right) B_i
\]
Alternative “Homogeneous” 4x4 Matrices

- Eliminate the additive “B” vector

- Now model:

\[ M_{4x4} = \begin{bmatrix} M_x \\ M_y \\ M_z \\ 1 \end{bmatrix} \]

- Simple relationship for all operators:

\[ A_{4x4} = \begin{bmatrix} A_{3x3} & B_{3x1} \\ 0 & 1 \end{bmatrix} \]

- Rotations are block-diagonal
- Relaxation is now a simple multiplication
Relaxation: 3x3 to 4x4 Formulation

- **3x3:**
  \[
  \begin{bmatrix}
  M_x' \\
  M_y \\
  M_z
  \end{bmatrix} =
  \begin{bmatrix}
  E_2 & 0 & 0 \\
  0 & E_2 & 0 \\
  0 & 0 & E_1
  \end{bmatrix}
  \begin{bmatrix}
  M_x \\
  M_y \\
  M_z
  \end{bmatrix} +
  \begin{bmatrix}
  0 \\
  0 \\
  1 - E_1
  \end{bmatrix}
  \]

- **4x4:**
  \[
  \begin{bmatrix}
  M_x' \\
  M_y \\
  M_z \\
  1
  \end{bmatrix} =
  \begin{bmatrix}
  E_2 & 0 & 0 & 0 \\
  0 & E_2 & 0 & 0 \\
  0 & 0 & E_1 & 1 - E_1 \\
  0 & 0 & 0 & 1
  \end{bmatrix}
  \begin{bmatrix}
  M_x \\
  M_y \\
  M_z \\
  1
  \end{bmatrix}
  \]

- \( E_1 = e^{-\tau/T_1} \)
- \( E_2 = e^{-\tau/T_2} \)
Rotations

• **3x3:**

\[
\begin{bmatrix}
M_x' \\
M_y' \\
M_z'
\end{bmatrix} = R_x(\alpha)M =
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha \\
0 & -\sin \alpha & \cos \alpha
\end{bmatrix}
\begin{bmatrix}
M_x \\
M_y \\
M_z
\end{bmatrix}
\]

• **4x4:**

\[
\begin{bmatrix}
M_x' \\
M_y' \\
M_z' \\
1
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha & 0 \\
0 & -\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
M_x \\
M_y \\
M_z \\
1
\end{bmatrix}
\]
Question 4

If you do operations (in order) represented by A1, A2, A3, A4 in 4x4 homogeneous representation, what is the overall operation?

A. \( M' = A_1 A_2 A_3 A_4 M \)

B. \( M' = A_4 A_3 A_2 A_1 M \)

C. \( M' = A_1 A_2 A_3 A_4 M + B \)

D. \( M' = A_4 A_3 A_2 A_1 M + B \)

E. None of these

B. The matrices are combined in reverse order, so that they multiply M in increasing order. There is no B vector in the 4x4 system!
Complex M Vector Format

- So far we simply use a **real-valued vector**
- Advantages to a **complex vector**:
  - Complex signal is simply first row
  - Easy to apply gradient-induced phase to $M_{xy}$
  - Can express rotation, relaxation
  - Simple transformation

\[
M_r = \begin{bmatrix}
M_x \\
M_y \\
M_z
\end{bmatrix}
\]

\[
M_c = \begin{bmatrix}
M_x + iM_y \\
M_x - iM_y \\
M_z
\end{bmatrix}
\]

\[
M_c = TM_r = \begin{bmatrix}
1 & i & 0 \\
1 & -i & 0 \\
0 & 0 & 1
\end{bmatrix} M_r
\]

\[
M_r = T^{-1}M_c = \begin{bmatrix}
0.5 & 0.5 & 0 \\
-0.5i & 0.5i & 0 \\
0 & 0 & 1
\end{bmatrix} M_c
\]

*mr2mc.m and mc2mr.m*
Rotations, Relaxation in Complex Format

• So we can use the simple transformation matrix $T$:
  – Relaxation is unchanged (block-diagonal)

Rotations can be “transformed”

\[
R_{x,c}(\alpha) = T R_x(\alpha) T^{-1}
\]
\[
R_{y,c}(\alpha) = T R_y(\alpha) T^{-1}
\]
\[
R_{\phi,c}(\alpha) = \begin{bmatrix}
  \cos^2(\alpha/2) & e^{2i\phi} \sin^2(\alpha/2) & -i e^{i\phi} \sin \alpha \\
  e^{-2i\phi} \sin^2(\alpha/2) & \cos^2(\alpha/2) & i e^{-i\phi} \sin \alpha \\
  -i/2 e^{-i\phi} \sin \alpha & i/2 e^{i\phi} \sin \alpha & \cos \alpha
\end{bmatrix}
\]
Gradients in Complex Format

- Make $M_c \cdot 3xN$, to simulate $N$ spins at different positions
- Make a “linear phase” matrix
- Element-wise multiplication

```matlab
% -- Initialize
z = [-1:0.05:1]; % Z locations
M = [ones(size(z)); 0*z; 0*z]; % Excited magnetization.

% -- Phase Twist
ph = exp(pi*i*z); % One cycle of phase.
phmult = [ph; conj(ph); 0*ph]; % 3xN phase Multiplier

% -- Apply and Plot
M = mc2mr(phmult.*mr2mc(M)); % Convert to Mx+i*My,
% add phase, convert back
plotm(M,1,[0*z;0*z;z]); % Plot twist!

More on this later!
```
Question 5

If the real-valued M vector is as shown, what is the complex vector?

A. $M_c = [0.4, -0.3i, 0.5]^T$
B. $M_c = [0.4, 0.3i, 0.5]^T$
C. $M_c = [0.4+0.3i, 0.4-0.3i, 0.5]^T$
D. $M_c = [0.4-0.3i, 0.4+0.3i, 0.5]^T$
E. $M_c = [0.7, -0.7i, 0.5]^T$

D. The first element of $M_c$ is $m_x + im_y$. 
Matlab Code in Rad229

• Most functions and examples in GitHub:
  – Matlab/ folder: general functions
  – Matlab/lectures: lecture examples

• arrow3d - download from Mathworks
  https://www.mathworks.com/matlabcentral/fileexchange/71994-arrow-3d

• Animations (optional): saveframe.m
  – Uses global framenum and filestem
  – saveframe captures figure to image sequence
  – combine to movies in QuickTime or other software
• $M$ is modeled as a 3x1 vector
• The Bloch Equation and solutions can be written in matrix form
  – Rotations are simple multiplications
  – Relaxation is a multiplication and addition
• A 4x4 formulation eliminates the additive term
• A complex representation eases application of gradients
How do we use these matrix operations to simulate sequences?