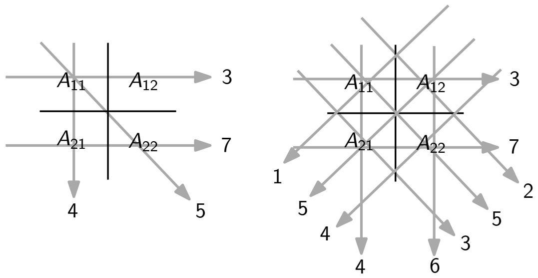
University of Salzburg | Lecturer: Roland Kwitt

Imaging Beyond Consumer Cameras – Proseminar (911.422)

Exercise sheet **B**

Exercise 1. 4 P.

Below are two examples of projections through a 2×2 grid: (*left*) from 4 directions and (*right*) from 10 directions.



Write Python code to compute the values for A_{ij} using direction reconstruction, i.e., solving the corresponding system of equations. Basically, you have to solve $\mathbf{F}\mathbf{x} = \mathbf{b}$ with $\mathbf{F}$ and $\mathbf{b}$ chosen appropriately (you can use `numpy.linalg` for that, in particular the `solve` or `lstsq` method).

Exercise 2. 2 P.

Consider *transverse relaxation* in MRI, i.e., the exponential decay (with T_2) of the the (x, y) component of the magnetization vector $\mathbf{M}(t)$. For the x-component, we know

$$M_x(t) = M_x(0)e^{-t/T_2}$$

where $M_x(0)$ denotes the value of the x-component at time $t = 0$. Assume $T_2 = 220$ [ms] and create a plot showing the time t (in [ms]) vs. $M_x(t)$ for $t \in [100, 300]$.

Exercise 3. 2 P.

For *transverse relaxation* now consider *both* the x- and y-component and $T_2 = 220$ [ms]. Start with

$$\mathbf{M}(0) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

and write $\mathbf{M}(t)$ as $\mathbf{A}(t)\mathbf{M}(0)$ where $\mathbf{A}$ is an appropriate 3×3 matrix. Obviously, $\mathbf{A}(t)$ needs to be such that this works for

$$\mathbf{M}(0) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

or any other initial magnetization vector as well.

Exercise 4. 2 P.

For *longitudinal relaxation*, the situation is slightly different than before. For the z-component of $\mathbf{M}(t)$, i.e., $M_z(t)$, we have

$$M_z(t) = M_0 + (M_z(0) - M_0)e^{-t/T_1}$$

where M_0 is the magnetization at *equilibrium*, i.e., when $T \to \infty$. Remember that without any relaxation, $\mathbf{M}(t)$ precesses around the axis of the static magnetic field $\mathbf{B}_0$. Taking relaxation into account, this means that longitudinal relaxation reaches M_0 as $T \to \infty$ (and transversal magnetization shrinks to 0). **Combine** both *longitudinal* and *transveral* relaxation such that you can write

$$\mathbf{M}(t) = \mathbf{A}(t)\mathbf{M}(\mathbf{0}) + \mathbf{b}(t)$$

where $\mathbf{A}(t)$ and $\mathbf{b}(t)$ are set appropriately (with, say $M_0 = 1$ and $T_1 = 500$ [ms], $T_2 = 120$ [ms]).

Exercise 5. 4 P.

In this exercise, we write Python code to simulate (1) the path of transversal relaxation and (2) how longitudinal magnetization approaches the equilibrium state. Use the template Python code in the provided Jupyter notebook `ExSheetB-Template.ipynb`.

Part A: Precession is rotation around the z-axis in our case. First write a simple function `rotZ(phi)` which returns a rotation matrix $\mathbf{R}$ around the z-axis.

Part B: Note that for the matrix $\mathbf{A}$ from *Exercise 4* and the rotation matrix $\mathbf{R}$ from *Part A*, we have

$$\mathbf{AR} = \mathbf{RA} \ ,$$

i.e., they commute. Write a function `freep`(T,T1,T2,f) which returns two matrices $\mathbf{A}$ and $\mathbf{B}$ such that we can write

$$\mathbf{M}(t) = \mathbf{A}(t)\mathbf{M}(0) + \mathbf{b}(t)$$

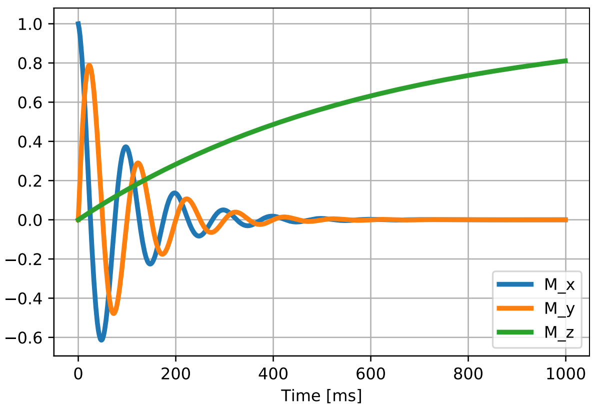
Note that `T` is the duration of the precession (here we look at the graph for 1000 [ms]), `T1,T2` are the longitudinal and transversal relaxation times (600 [ms] and 100 [ms]) and `f` is the frequency (10 [Hz]). Note that we can compute the rotation angle (around the z-axis) ϕ via

$$\phi = \frac{2\pi f T}{1000}$$

Part C: Finally, use the `freep` function to start at

$$\mathbf{M}(0) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

and plot $M_x(t)$, $M_y(t)$ as well as $M_z(t)$ over the time t (i.e., the 1000 [ms]). It's important to note that you only need to use `freep` once: start with $\mathbf{M}(0)$ and use `T=1` to get $\mathbf{M}(t_1)$. Then use $\mathbf{M}(t_1)$ to get $\mathbf{M}(t_2)$, etc. *Below is an example of what this plot should look like, as a sanity check.*



Exercise 6.

This last exercise is just for better understanding of MRI. Visit

http://drcmr.dk/BlochSimulator/index.Flash.html

and explore the Bloch simulator (the Flash version, **not** the newer one). There are a couple of Youtube videos on how to use it.

Total #points: 14 P.