

# Simulating magnetic resonance experiments with Precession

D.C. Pizetta, V.H.M. Pessoa, F.F. Paiva

São Carlos Institute of Physics, University of São Paulo, São Carlos, Brazil

e-mail: daniel.pizetta@usp.br

**Abstract:** *Scientific tools for simulating magnetic resonance (MR) system are always a requirement from students to scientists. In this paper, we will present a new tool called Precession that will allow a creation of a MR experiment using the language Python and libraries available. Early results show the creation of a virtual sample for being used to an experiment in the future. This tool proved to be powerful for future development of simulation tools and integration with new features.*

**Palavras-chave:** magnetic resonance simulation, educational tool, Larmor precession.

## Introduction

Simulation tools play a role in any scientific field being a mechanism for study or professional use. In the learn process they are very important in the view of let students to apply their knowledge, test their hypothesis, and understand the concepts of the subject of study. In magnetic resonance, these tools have proven useful for certain niches. We can cite tools for educational applications such as PhET Simplified MRI [1] and Bloch Simulator [2], that brings an illustration of the dynamics of magnetic resonance phenomena. More complex system can be studied using tools for simulating specialized phenomena, such as DAMARIS [3], VeSPA [4] and SIMRI [5]. In general, these tools show some specific and a not complete case of study.

Our proposal is to offer a tool in which the user can easily create a simple sample with controlled spatial characteristics and properties such as T1, T2 and nuclei density. At this point we are not including other parameters such as chemical interactions, diffusion, etc. After that, users can set their system to an experiment, setting gradients – if necessary for creating an imaging experiment, RF pulses and so on. Including to that, the visualization can be changed to show details about which is approached at that time. As first application of this tool, named “Precession” is planned for being used by students to set and understanding an MR experiment. The main idea is to offer groups of sample/experiments that show the main effects of the MR phenomena. This tool could be used for educational purposes and scientific research. Soon,

after stabilization of code API (application programming interface), the code will be available on GitLab as open source code. This will allow the use by the scientific community, having the possibility of contributions and adaptations for other targets.

## Materials and Methods

Python [6] was chosen as the main language because of its simplified syntax and availability of many libraries for mathematical and signal processing procedures. Operations are made using common scientific libraries such as NumPy [7], SciPy [8] or more specialized one as NMRGlue [9]. This last one can deal with data from simulating signal acquisition to data processing. The Bloch equations is processed using Bloch Equations Simulator [10]. Graphical user interface (GUI) was planned to be highly reusable and versatile. The interface is implemented using a wrapper for the graphical framework Qt [11], in this case PyQt [12] from pyqtgraph [13], but another wrapper should work, i.e. PySide [14]. Viewers use pyqtgraph to plot data for each step of the experiment. Data structure is stored in HDF5 [15] files, a scientific data format. The code is managed using the GitLab [16], a system that integrates software development tools such as build, deploy, test, etc. Precession is planned to be multi operation system. The program has different contexts for different purposes – such as sample edition, system setup, sequence programming and processing, but the same basic interface simplifying the learning process and operation.

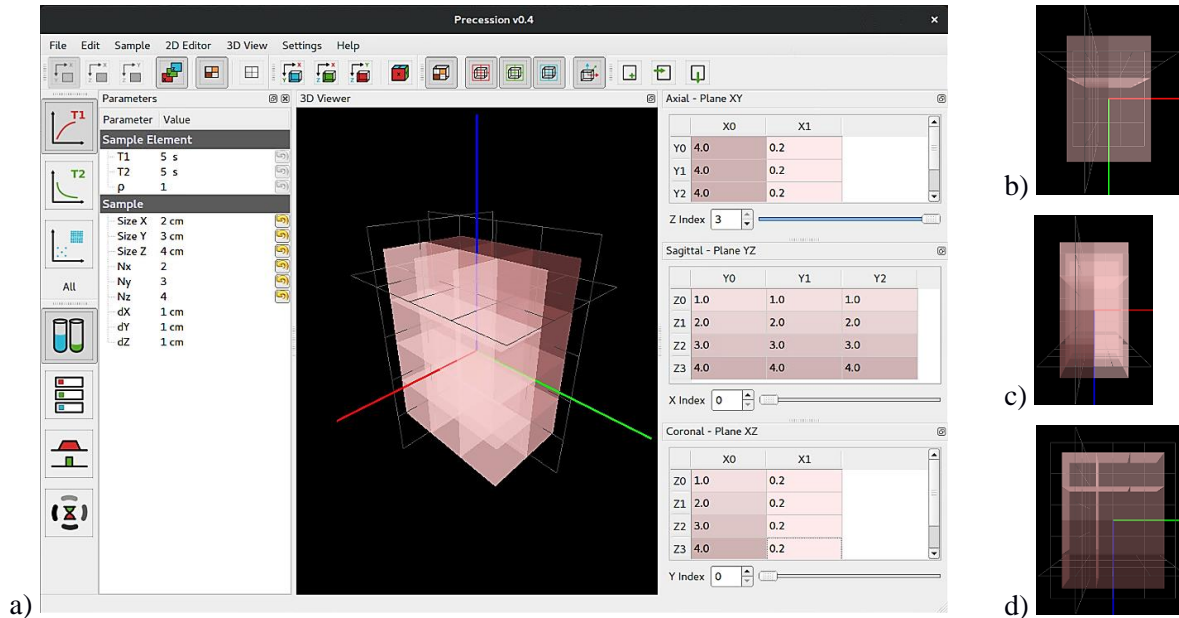
In the next section, we will show the main window of this program, a sample edition and its characteristics that will guarantee future applications.

## Results

Precession has a main window with a menu that allow the user access all the features, Figure 1.a-top, and a context menu in the Figure 1.a-left. Also, a parameter window that allows the user to change all the parameters for all contexts, Figure 1.a-left. A 2d view in which user can edit data for each element,

Figure 1.a-right, and a 3d view that shows a volumetric view as shown in Figure 1.a-center. Each plane is represented in a table as axial, coronal and sagittal cuts of the sample (since this notation is usually used in magnetic resonance imaging to a

sample aligned with the z, y and x axis, respectively). The plane is represented by the grids on the 3d view, Figure 1.a-center, and the views can be adjusted to show each direction as shown in Figure 1.(b-d).



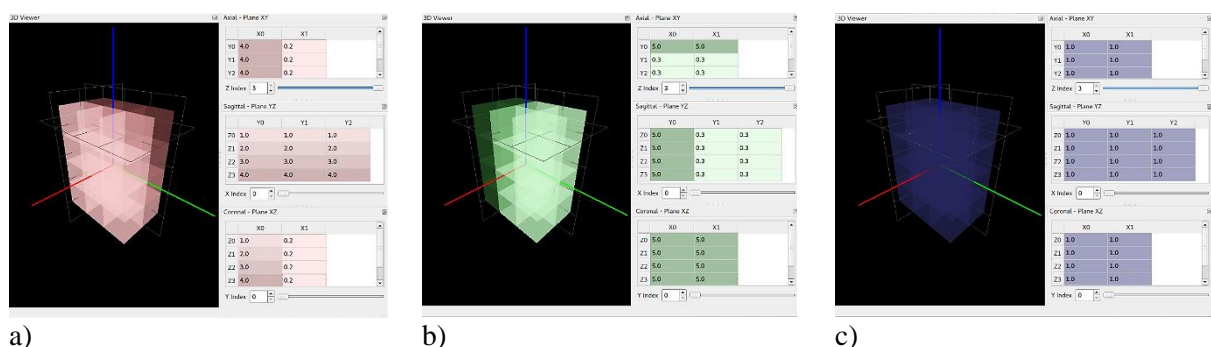
**Figure 1.** Precession’s main interface. **a)** Sample edit context showing (top/left) menu and tool bar, (left) parameters view/editor, (right) 2D view/editor (center) 3D view. **b)** Axial view (XY plane). **c)** Coronal view (XZ plane). **d)** Sagittal view (YZ plane).

Precession offers a possibility to begin a magnetic resonance experiment by creating a sample. In this step, the user can change the sample characteristics to fit his/her needs. There are macroscopic characteristics that may be edited at this stage, as the number of sample elements in each direction (Nx, Ny, Nz) and its size (dx, dy, dz), or the size of the whole sample (Size X, Size Y, Size Z). Those parameters are shown in the Figure 1.a-left.

Each sample element can be individually edited to modify its value of T1, T2 and nuclei density. The values can be changed by direct selecting and editing the elements in the 2d editor as in Figure

2.(a-c)-right or using the parameter table. The values can be individually changed, batch changed using a single value or setting a gradient along rows and columns.

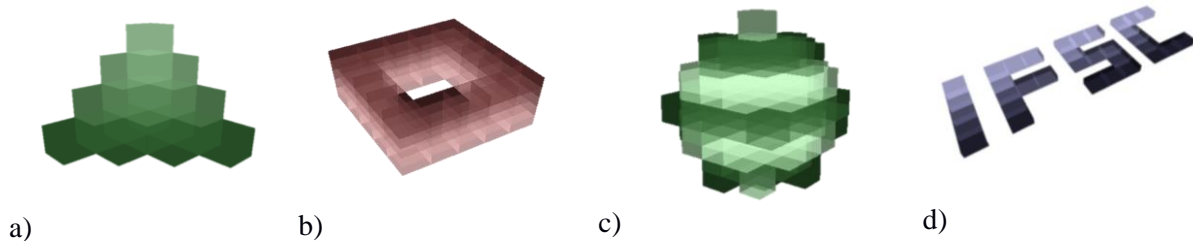
Beyond the simple creation of a sample, there is also the possibility of view the sample appearance in a three-dimensional space that can be rotated by the user freely or adjusted to the canonical views in MRI, as shown in Figure 1.b-c. The sample's color expresses the intensity of the selected characteristic, in both 3d or table views, being T1 in red, T2 in green and density of nuclei in blue (also weighing the opacity of each cell, since a null density means that there is no sample in that region).



**Figure 2.** Visualization of different parameters of the same sample. **a)** T1 values (red). **b)** T2 values (green). **c)** Proton density (blue).

Using these three parameters together with the sample element position, added to future system setup and sequence programming the calculus of a

simulated magnetic resonance signal can be generated. In the Figure 3 we show some created samples to show the possibilities of the Precession sample editor.



**Figure 3.** Set of samples created using Precession sample editor. **a)** A pyramid sample describing T2 values. **b)** A square sample torus representing T1 values. **c)** A sphere showing T2 values. **d)** Our institute acronym showing proton density.

The same structure will be used to set the gradients inside the magnet, where a 3d view should be generated. To the next steps, the sequence of MR pulses will also be edited and viewed by the same interface. Finally, the signal of the simulated sample and its spectrum or image will be show at the same place. So, the program and its interface was improved to being reusable.

## Discussion and Conclusion

These early results show the capacity of this tool for simulating sample construction, even complex ones, with a simple interface. The structure of the main window will permit that this tool grown up with many others features to create a complete tool for simulating a magnetic resonance experiment.

## References

- [1] Simplified MRI. Available from: <https://phet.colorado.edu/en/simulation/legacy/mri>. Accessed: 15/09/2017.
- [2] HANSON, L. G. Bloch simulation and MR fundamentals visualized. Available from: <http://www.drcmr.dk/bloch>. Accessed: 15/09/2017.
- [3] GÄDKKE, A. et al. DAMARIS - a flexible and open software platform for NMR spectrometer control. *Magnetic Resonance Imaging*, v. 25, n. 4, p. 576, doi:10.1016/j.mri.2007.01.081, 200.
- [4] VeSPA: Versatile simulation, pulses and analysis. Available from: <https://scion.duhs.duke.edu/vespa/>. Accessed: 20/03/2014.
- [5] BENOIT-CCATTIN, H; et al. The SIMRI project: a versatile and interactive MRI simulator. *Journal of Magnetic Resonance*, v. 173, n. 1, p. 97-115, 2005.
- [6] Python programming language: official website. Available from: <http://www.python.org/>. Accessed: 02/02/2016.
- [7] NumPy. Available from: <http://www.numpy.org>. Accessed: 25/09/2017.
- [8] SciPy, Scientific Computing Tools for Python. Available from: <http://www.scipy.org>. Accessed: 25/09/2017.
- [9] HELMUS, J.J. nmrglue, a module for working with NMR data in Python. Available from: <https://www.nmrglue.com/>. Accessed: 25/09/2017.
- [10] AMALKANTI, N. bloch-simulator-python. Available from: <https://github.com/nej49/bloch-simulator-python>. Accessed: 25/09/2017.
- [11] Qt - cross-platform application and UI development framework. Available from: <http://qt.digia.com/>. Accessed: 05/03/2016.
- [12] What is PyQt? Available from: <https://riverbankcomputing.com/software/pyqt/intro>. Accessed: 05/03/2016.
- [13] PyQtGraph - Scientific Graphics and GUI Library for Python. Available from: <http://www.pyqtgraph.org/>. Accessed: 06/11/2017.
- [14] PySide - Python for Qt. Available from: <https://wiki.qt.io/PySide>. Accessed: 05/11/2017.
- [15] The HDF5 group. Available from: <https://www.hdfgroup.org/HDF5/>. Accessed: 07/05/2014.
- [16] Gitlab Inc. Available from: <https://about.gitlab.com/>. Accessed: 25/09/2017.