Thanks for your interest in PSUdoMRI:

A Digital Ostensive MRI system developed at
The Pennsylvania State University.

This document is designed to give an introduction to PSUdoMRI in its current form, first through the GUIs provided with insights to the operation of the engine, then with explicit description of the file structures and other information required to use the engine more directly.

PSUdoMRI was developed first to accurately predict how realistic distributions of the electromagnetic fields (DC, switched gradients, and radiofrequency) affect the MR images for arbitrary MRI pulse sequences.PSUdoMRI is capable of considering multiple RF coils in transmission and in reception, considering correlated noise between RF channels, and calculating the SAR distribution throughout the sample for arbitrary pulse sequences. Itcouldbe a valuable aid in the design of pulse sequences to overcome or compensate for imperfections in the field distributions due to field/tissue interactions or design constraints.It is our hope that the package will find value for other applications as well.

PSUdoMRI consists of a simulation engine that solves the Bloch equation, plus simple setup and reconstruction GUIs to allow some demonstrations for introductory purposes.

Inputs to the engine include files describing the sample geometry, tissue MR properties, field distributions, and pulse sequence.

Depending on the user's desires and the inputs given, the outputs of the engine can include files containing information on the MR signaland noise (in separate files) received through time, the location in k-space corresponding to each discrete acquisition time point, and the time-average SAR distribution for the entire sequence.

We hope you will find PSUdoMRI to be a versatile and valuable tool in your research. Please give us feedback to help us make improvements and also to help us justify our continued work on this project.

Sincerely,

Zhipeng Cao (zzc102@psu.edu)
Sukhoon Oh (suo2@psu.edu)
Chris Sica (csica@hmc.psu.edu)
and Chris Collins (cmcollins@psu.edu)

14 June 2010

Contents:

Installation

PSUdoMRI was developed and tested on work stations operating in a Microsoft Windows environment.

Currently the simulation engine ("Nutate") is programmed in C++ and compiled with Microsoft Visual Studio using OpenMP architecture to facilitate parallel processing. Thus in order to run simulations you will likely need to install the **x86** version of Visual Studio 2010 Runtime Environment. A free version can be found at "\Prerequisite\" folder.

The Setup and Reconstruction GUIs are written and executed with Matlab (http://www.mathworks.com). In order to operate them you will need to have Matlab 2007a or newer running on your system.

Once the above packages are in place, simply extract the compressed folder "PSUdoMRI" to the desired location on your hard drive. Thenadd a path in Matlab to the uncompressed PSUdoMRI folder (some components will not work properly unless you add the path in Matlab). The Setup GUI and Reconstruction GUI can then be launched using commands "Setup" and "Reco" in Matlab. Once all required input files are available, the simulator engine can be run either through the Setup GUI or by using a command line addressing the "\PSudoMRI\Engine\" directory, as described towards the end of this document.

Setup GUI: *Introduction*

Figure 1 shows the Setup GUI just as it appears after launching it. Notice that only certain buttons are active, and that it begins in the Signal calculation mode (red arrow). As fields are filled properly (through windows that pop up after pushing associated buttons, typing into fields directly, or using drop-down boxes), more buttons will become active, enabling access to more fields.

Figure 1. Setup GUI when first launched.

The simulator can run in any of three separate modes for calculating signal, noise, and SAR. Because calculation of these three entities requires very different information, running the calculations separately allows for larger problems to be solved with a given amount of memory.

In each mode you will need to provide different information before pressing the Run button to start the simulator.In order to run any simulation you will need to complete nearly all fields in the first three rows on the left, as indicated with the red bracket. This includes the study name, sample geometryfilename, sample tissue properties filename, number of transmit coils, and Pulse sequence file name. The Pulse sequence file name of a pre-existing file with the proper format (discussed later) can be provided, or the Imaging Parameter Setup and Geometry Editor window can be used to develop a sequence file for some simple demonstration cases.

Figure 2 shows the Setup GUI after the study name has been defined and the sample geometry and tissue property files have been selected. The sample geometry and tissue property files used here are available as "Models/Head 2mm 125MHz/bin/head2x2x2.smpl" and "Models/Head 2mm 125MHz/ Tissue_125MHz.prop" (respectively) within the dedicated folder used for PSUdoMRI installation.

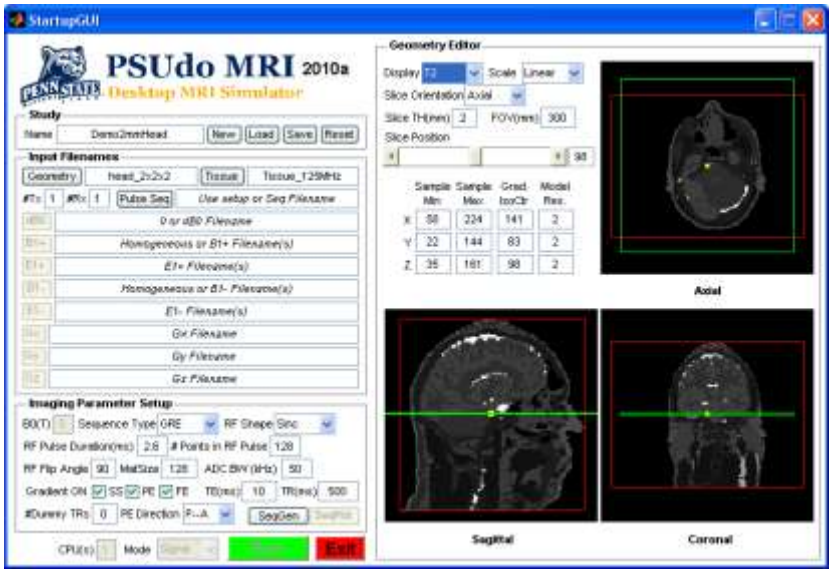

Figure 2. Setup GUI after sample Geometry and Tissue property files are identified.

In the "Geometry Editor" section, three orthogonal slices through the middle of the sample are shown. After the Tissue property file is loaded, the T1, T2, proton density, or tissue

identification integer can be shown by using the "Display" drop-down box. Currently T2 values are displayed.

Notice the red box in the Geometry Editor section. In each of the three modes this box indicates the portion of the model that is considered in the simulation. The size and shape of the box are controlled with the three Sample Min fields and three Sample Max fields. It can save time and memory to exclude unimportant parts of a given model for a given simulation. It may also be necessary to adjust the size of the red box for each mode separately. For example, if simulating a head in a volume coil with a sequence that only images one slice, if it can be assumed that tissue outside the slice makes negligible contribution to the signal, adjusting the red box to include only the portions of the model that include the imaging slice and some neighboring tissue can save a lot of time and memory during the signal calculation, but thorough consideration of noise and SAR will likely still require consideration of the entire model.

Figure 3 shows the displayafter adjusting the number of coils, Geometry Editor, and Imaging Parameters and pressing the "SeqGen" button. A popup window allows the user to set the relative magnitude and phase of each transmit coil. After this is done and the user presses "OK," if there are no errors a message will pop up indicating that the sequence file was generated and the Pulse Seq field will automatically change to the name of the file generated. The flip angle (FA) indicated in the popup window will match that in the Imaging Parameter Setup panel. For sequences generated by the GUI, this will correspond to the average flip angle induced by the individual coil in the portion of the sample in the red box. Thus if multiple coils are used and constructive interference is expected in the ROI, actual flip angle may be much higher. Also, for excitations with a single surface coil the flip angle near the coil will likely be much higher than this number.

Figure 3. Setup GUI after adjusting the number of coils, Geometry Editor, and Imaging Parameters and pressing the "SeqGen" button.

As mentioned previously, it is necessary to get at least this far in order to run any simulations. However, what type of simulations (or mode) can be run and what the results are will still depend on a variety of further inputs.

When all required information is entered for a given mode (as described below), pressing the green "Run" button will launch the appropriate portion of the engine through a separate command line window. Some information of the simulation progress is also given in this separate window. A simulation can be stopped by pressing CtrlC while in that separate window. While one simulation is running it is possible to continue using the Setup GUI to prepare and run other simulations.

Setup GUI: *Signal Calculation*

Figure 4. Setup GUI after pulse sequence is created and files for receive coil B1 distributions (B1-) have been selected.

In "Signal" mode, after the sequence file is generated or selected, the fields for B0 inhomogeneity, RF magnetic field, and gradient field distributions become available. If no field files are indicated, it will be possible to calculate the signal as if B0 and the fields of all RF coils are perfectly homogeneous throughout the sample, and all gradient fields are perfectly linear throughout the sample. If, on the other hand, it is desired to simulate images considering specific field distributions, it will be necessary to indicate the names of files containing the necessary information. Field distribution files for counter-clockwise and clockwise rotating components of the RF magnetic field for an 8 element array are provided in the folder "Models\Head 2mm 125MHz\bin." Multiple receive coil files are selected by using "Ctrl" or "Shift" on the keyboard with the left mouse button in the popup window launched with the B1- button. If multiple transmit coils with specific field distributions are used, they are indicated individually for each corresponding magnitude and phase (entered previously: figure 3) in the popup launched with the B1+ button.

In the example shown in Figure 4, it will be assumed that a single homogeneous coil will excite the spins and the 8 coils indicated will receive signal simultaneously. Pushing the green Run button will result in the generation of 8 k-space signal files (one for each coil) and one k-space map file, all written in the directory corresponding to the user-selected study name.

Setup GUI: *Noise Calculation*

When switching to "Noise" mode, all fields previously entered will be retained, and the E1- field will become available. In order to calculate the noise, the user must indicate which files describe the electrical field distributions for each of the receive coils. Multiple files are selected by using "Ctrl" or "Shift" on the keyboard with the left mouse button in the popup window launched with the E1- button. Once this is done, pressing the green "Run" button will result in the generation of one noise file for each receive coil, all written in the directory corresponding to the user-selected study name. Note that in Figure 5 the red box has been expanded to ensure that noise from the entire sample is considered.

Figure 5. Setup GUI in Noise mode.

Setup GUI: *SAR Calculation*

When switching to "SAR" mode, all fields previously entered will be retained, and the E1+ field will become available. In order to calculate the noise, the user must indicate which files describe the electrical field distributions for each of the receive coils. Multiple files are selected by using "Ctrl" or "Shift" on the keyboard with the left mouse button in the popup window launched with the E1- button. Once this is done, pressing the green "Run" button will result in the generation of one noise file for each receive coil, all written in the directory corresponding to the user-selected study name. Note that in Figure 5 the red box has been expanded to ensure that noise from the entire sample is considered.

Figure 6. Setup GUI in SAR mode.

Reconstruction GUI

The reconstruction GUI interface is shown in Figure 7. First, select the number of receive channels and hit "enter." Then if more than one channel is selected, choose between "Sum of square" or "Sum of magnitude" reconstruction. After pressing the "Load K-space Data" button, select the desired signal files in the window that pops up, and (if desired) the desired noise files in the ensuing window (otherwise, press cancel). By default the image magnitude data will be loaded into the active view port (indicated with the green check mark). Displaying other parts of the data and performing spatial rotations or flips of the data can be accomplished with the panel at the upper right. Adjustment of the gray scale for the active view port and some simple algebraic operations between images can be accomplished with the panel on the lower right.

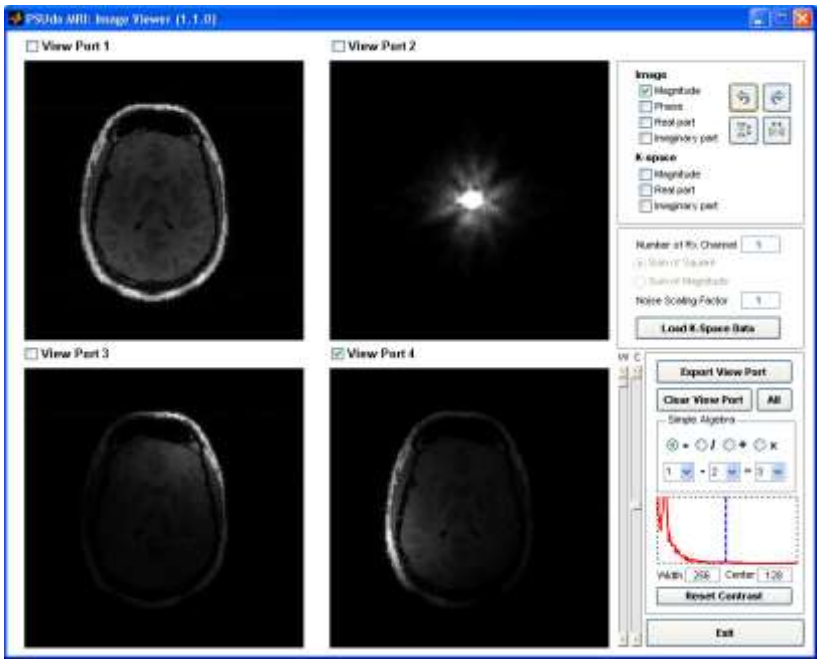

Figure 7.Reconstruction GUI.

SAR GUI

(In progress - Coming Soon!)

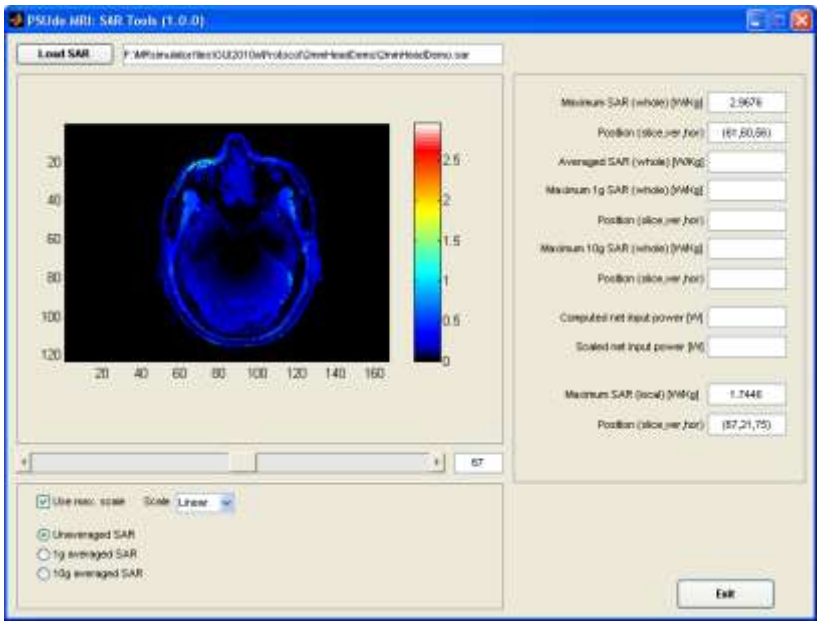

Figure 8.SAR GUI.

Requirements for the Simulator Engine: *Introduction*

We have tried to design the input files with a structure that is easily understood so that users who with to add their own models or field distributions to their library or simulate their own pulse sequences should be able to do so with some minimal effort.

The simulator engine consists of three separate executable files, one for each mode. Thus each one requires a different command and different input information. Understanding of the input information required for each mode can be seen by examining the setup GUI. Figure 9 shows which information is conveyed to the engine through the command line modifiers and through the sequence file.

Figure 9.Information that goes to the Command line (red boxes) and to the Sequence file (green boxes).

Requirements for the Simulator Engine: *File Structure for Spatially-organized Files*

Here we will describe the structure of the input files in text format. The engine reads files in binary format (except for the tissue property file, which is always text format). Some simple conversion tools (in the conversion tools folder) are provided to facilitate translation from text to binary format. Also, text versions of many file types are available in the folder "PSUdoMRI/Models/Head 2mm 125MHz/txt"

The sample geometry file in text format contains a brief header followed by four long columns of numbers. The 6-row header includes (on each row, in order of appearance) values for 1) The number of rows in the four-column array, 2)The minimum and maximum x coordinates where tissue is found, 3) The minimum and maximum y coordinates where tissue is found, 4) The minimum and maximum z coordinates where tissue is found, 5) The x, y, and z coordinates of the initial gradient isocenter, and 6) The grid resolution (in mm) in the x, y, and z directions. Each row of the four-column array contains integer x, y, and z coordinates and an integer tissue identifier corresponding to the "ID" in the tissue property file. After converting to binary, the sample geometry file should be given the extension "smpl".

The tissue property file is a text file with extension "prop". The first row of the file indicates the values in columns of following rows. Each row gives properties of a different tissue with a unique integer ID. After the ID, seven properties are given for each tissue, including T1 (in ms), T2 (in ms), proton density (fractional), chemical shift (in ppm), electrical conductivity (in S/m), mass density (in kg/m^3), and the name of the tissue.

The ΔB_0 file in text format contains a brief header followed by four long columns of numbers. The 1-row header contains only the integer number of rows in the table that follows. This number must match that for the sample geometry file, and the order of spatial information in the ΔB_0file should match that in the geometry file row by row. In each row of the table in theΔB_0 file is the magnitude deviation (in ppm) from a perfectly-uniform B_0 field, followed by the local gradient in B_0 (in ppm/m) in x, y, and z directions.

The B_1 field file for a given transmit or receive coil in text format contains a brief header followed by two long columns of numbers. The 1-row header contains only the integer number of rows in the table that follows. This number must match that for the sample geometry file, and the order of spatial information in the B_1 field file should match that in the geometry file row by row. In each row of the table in the B_1 field file is the complex (real and imaginary parts, in tesla) magnitude of the circularly-polarized component of the B_1 field produced by the coil. Note that for a given coil, circularly-polarized components with opposite sense of rotation are pertinent for transmission and reception, and that for quadrature coils the coil should be driven with opposite phases in calculating these two components. Which component is pertinent in transmission and in reception depends on the whether B_0 is parallel or antiparallel with the z axis.

The E_1 field file for a given transmit or receive coil in text format contains a brief header followed by six long columns of numbers. The 1-row header contains only the integer number of rows in the table that follows. This number must match that for the sample geometry file, and the order of spatial information in the E_1 field file should match that in the geometry file row by row. In each row of the table in the E_1 field file is the complex (real and imaginary, in V/m) magnitudes of the x, y, and z components of the electrical field produced by the coil when it is driven in the same way as to produce the corresponding pertinent B_1 field information.

The format of the Gradient field files not yet finalized.

Requirements for the Simulator Engine: *File Structure for Sequence Files*

The Sequence file consists of a header followed by a number of different blocks, each of which can be structured very differently. The header consists of only the integer number of time steps (Nsteps) in the entire sequence, the number of transmit coils (Ncoils), and the number of receive coils. In order of appearance, the blocks (5+Ncoils in number) describe 1) the time step sizes (in ms), 2) the receiver/kmap/ideal crushing events (all in one block with indicators 1, 0, -1, -2), 3) the real, imaginary, and frequency offset at each time for each transmit coil drive (number of blocks equal to Ncoils arranged in series, one block for each coil), 4) the x gradient strength (in mT/m), 5) the y gradient strength (in mT/m), and 6) the z gradient strength (in mT/m).

Once it is past the header information, the engine reads the sequence file data assuming it is always in a loop within a block until the number Nsteps is reached, at which time it assumes it is entering the next block. Thus each block must begin with structural information for the first loop in that block. Loop

structure is indicated with 2 integers: a number of elements per loop followed by a number of loops. For example, in a gradient block, the gradient strength 12 mT/m for one timestep followed by 4 timestepsat 0 mT/m, all looped 3 times could be represented as either

53120000

or

1112140111214011121400

or

151 12 0 0 0 0 12 0 0 0 0 12 0 0 0 0

where values for number of elements in a loop are shown in red and those for number of loops are shown in blue. Thus sequence file size can be reduced to some degree with strategic use of loops. Of course it is always possible to forego loops altogether by beginning each block with Nsteps followed by 1.

The data in the timestep block will contain a single value in ms for the size of the corresponding timestep. All field values are constant during a timestep, so timesteps will necessarily be much smaller during definition of a shaped RF pulse than during a rectangular phase encoding gradient or a recovery period.

The receiver/kmap/crush block will contain a 1 at each timestep during which data is collected, a 0 at each timestep during which no data is collected, a -1 at timesteps (such as at the end of a TR) where the sequence returns to the center of k space (for use of the kmap files), and a -2 at timesteps (such as at the end of a TR) where the sequence returns to the center of k space and it is desired to use ideal crushing, setting the transverse magnetization to zero. (A current known issue with the simulator is that gradient and RF crushing can result in some aliasing-like banding in the signal, likely due to the discrete nature of the models.)

For each timestep in each of the Ncoils blocks describing the transmit RF waveforms, three values must be given: Real and Imaginary parts of the amplitude (given as a multiplier for the information in the corresponding B1 field file) and frequency offset (in Hz).

For each timestep in each gradient waveform block is the desired gradient strength (in mT/m).

Some examples of code in Matlab for generating some standard pulse sequences in the language of the simulator are given in the “GUI Toolbox/SeqGen” folder.

Requirements for the Simulator Engine: *Command line requirements*

Note that if all required files are created ahead of time it should be possible to launch any simulation through the GUI – even for arbitrary user-defined sequences (by selecting the appropriate sequence file rather than using the associated portions of the interface). Currently this is the recommended

procedure, but we provide the information below to aid the user in the event it is advantageous to launch the simulation engine without using the provided setup GUI.

Running the engine in each of the three modes requires different commands followed by different information. In general, quotation marks and an identifying string are used to group different portions of this information together and identify it to the engine. Examples of the commands required can be seen by examining the first lines appearing in the command window that appears after pressing the "Run" button with the setup GUI in the desired mode. Below is a brief explanation for each case.

Executing the Signal calculation requires the command "(path)/PSUdoMRI/Engine/NutateSignal" (where "(path)" would be replaced with the location of installation on your computer) followed by the number of threads for parallel processing (Thread), the strength of B0 (in Tesla), boundaries of the red box (integer values for xMin, xMax, yMin, yMax, zMin, zMax), the location of the gradient isocenter (grid location for xCtr, yCtr, zCtr), the desired resolution of the grid - if different than that in the geometry file(xWid, yWid, zWid in mm), the name of the tissue property file (TissueTypeFile), the name of the tissue geometry file (GeometryFile), the name of the ΔB0 file (DelB0File), the name(s) of the transmit B1 file(s) (B1PlusFile), the name(s) of the receive B1+ file(s) (B1MinsFile), the basis for the name(s) of the output signal file(s) (KSpaceFile), the name of the output k-space map file (KMapFile), and the name of the pulse sequence file (SequenceFile).

Executing the SAR calculation requires the command "(path)/PSUdoMRI/Engine/NutateSAR" (where "(path)" would be replaced with the location of installation on your computer) followed by the number of threads for parallel processing (Thread), boundaries of the red box (integer values for xMin, xMax, yMin, yMax, zMin, zMax), the name of the tissue property file, the name of the tissue geometry file, the name(s) of the transmit B1 file(s), the name(s) of the transmit E1 file(s) (E1PlusFile), the name of the pulse sequence file, and the name of the output SAR file (SARFile).

Executing the Noise calculation requires the command "(path)/PSUdoMRI/Engine/NutateNoise" (where "(path)" would be replaced with the location of installation on your computer) followed by boundaries of the red box (integer values for xMin, xMax, yMin, yMax, zMin, zMax), the desired resolution of the grid-if different than that in the geometry file(xWid, yWid, zWid), the name of the tissue property file, the name of the tissue geometry file, the name(s) of the receive E1 file(s) (E1MinsFile), the basis for the name(s) of the output Noise file(s) (KNoiseFile), and the name of the pulse sequence file.

Output File Structures

PSUdoMRI outputs four types of files: signal (extension "ksig"), noise (extension "nois"), k-space map (extension "kmap") and SAR (extension "sar"). Each is structured as a single array of binary "float" numbers. Each contains different information in a specific order.

The signal file contains the real part followed by the imaginary part of the signal received in the given coil at each acquisition time point in the order of occurrence during the sequence.

The noise file contains the real part followed by the imaginary part of the noise received in the given coil at each acquisition time point in the order of occurrence during the sequence.

The kmap file contains the location in k-space (in m^{-1}) in all three directions (k_x, then k_y, then k_z) at each acquisition time point in the order of occurrence during the sequence.

The SAR file contains the integer grid coordinates (x, then y, then z) of a location in tissue followed by the corresponding tissue ID and SAR (in W/kg) for all locations in tissue.

About Source Code

If you've made it this far you might also want to see how the code works, modifying it for your specific needs, or maybe even improving it for everyone. Right now we don't have a policy for sharing the code – which is still in development – but we're not against talking about possibilities for the future. Give us a call.

Known Current Issues (and recommended workarounds)

No realistic Gradient Field Files (currently only ideal, linear gradients).

SNR too high (scale noise to achieve reasonable results).

Appearance of banding when applying only realistic RF&gradient crushing pulses (use ideal crushing flag -2 in receiver block of sequence file where appropriate).

Future Plans

(Besides working to resolve the known issues above, and depending on our resources …)

Hopefully before too long we'll have spatial-averaged (e.g., 10g) SAR calculations, and eventually temperature calculations, available through the SAR GUI.

We'd like to expand the library of input files to include fields at other frequencies and portions of the body other than the head only.

We'd like to develop a ΔB_0 field calculator producing the needed input for the simulator from given geometry files and susceptibility information.

We've discussed ways to simulate sample motion.

Acknowledgements
In addition to those listed on page 1 of this document, many people contributed significantly to the development of this simulator either directly (especially John McGarrity, Tim Horan, and Busik Park), through making code freely available online which helped to get us started in early Matlab-based versions of the engine (especially Brian Hargreaves), or through published works (especially Thies Jochimsen and James Bankson, references 2 and 3). Below is a list of references that were useful in some stage or another in our development, or that give important and/or interesting examples of prior approaches and applications, followed by references to our simulator in various stages of development.

Funding for this work was provided by the National Institutes of Health through R01 EB006563 and R01 EB000454.

References about some prior MR simulators

1. Brian Hargreaves, "Bloch Equation Simulator" http://mrsrl.stanford.edu/~brian/blochsim/
2. Jochimsen*et al*., JMR 2006;180:29-38
3. Bankson and Wright. MRM 2002;47:777-86
4. Sharp *et al*., 2006 ISMRM, p. 1351
5. Yoder *et al*., MRI 2004;22:315-28
6. Wei *et al*., JMR 2005;172:222-30
7. Jochimsen*et al*., JMR 2004;170:67-78
8. Benoit-Cattin*et al*., JMR 2005;173:97-115
9. Stöcker*et al*., Adv. Parall. Comp. 2008;15:122-64
10. Shkarin and Spencer, IJIST 1997;8:419-426
11. Marshall, Concepts MagnReson, 1999;11(6):379-392
12. Maccotta et al., NMR Biomed 1997;10:216-221
13. Summers et al., MagnReson Med 1986;3:363-376
14. Kwan et al., IEEE Trans Medical Imaging 1999;18(11):1085-1097
15. Ludecke et al., MagnResonImag 1985;2:329-343
16. Malik, Concepts MagnReson B 2009;35B(4):232-237
17. Pauly et al., IEEE Trans Medical Imaging 1991;10(1):53-65
18. Wei et al., J MagnReson 2005;172:222-230

References about PSUdoMRI

1. Cao Z, Sica CT, Oh S, McGarrity J, Horan T, Park B, Collins CM. Development of a Versatile MRI Simulator Considering Static, Gradient, and RF Field Distributions, Including SAR and Noise Correlation for Arrays. University of Minnesota Workshop on High Field Imaging and Spectroscopy, Minneapolis, MN, October 9-12, 2009.

2. Cao Z, Oh S, Sica CT, McGarrity J, Horan T, Park B, Collins CM. Development of a Versatile MRI Simulator Including SAR and Noise Correlation for Multiple Transmit and Receive Coils in 3D. ISMRM Third International Workshop on Parallel MRI.Santa Cruz, October 23-26, 2009.

3. Cao Z, Sica CT, Oh S, McGarrity J, Horan T, Park B, Collins CM. An MRI Simulator for Effects of Realistic Field Distributions and Pulse Sequences, Including SAR and Noise Correlation for Array Coils.Proc. 18thISMRM, Stockholm, 2010, p. 1457