BlochSolver CPU version User's Manual



Version 1.00, February 15, 2020 ©2020 MRI simulations Inc. All rights reserved Contact: info_at_mrisimulations.com 断りなく修正することがあります. 目次

- 1. はじめに
- 2. インストールと動作確認
- 3. 使用するための準備と画像再構成
- 4. シミュレーション例
 - 4-1 グラジエントエコー法
 - 4-2 スピンエコー法
 - 4-3 インバージョンリカバリ法
 - 4-4 プロトン密度強調高速スピンエコー法
 - 4-5 T2 強調高速スピンエコー法
 - 4-6 マルチプルスピンエコー法
 - 4-7 マルチスライスグラジエントエコー法
 - 4-8 マルチスライススピンエコー法
 - 4-9 3D グラジエントエコー法(32 スライス)
 - 4-10 3D グラジエントエコー法(128/256 スライス)

4-11 エコープラナーイメージング

- 5. シミュレーションの高速化のための工夫
- 6. 例題
- 7. むすび

1. はじめに

このマニュアルは、「MRI シミュレータを用いた独習パルスシーケンス [標準編]」に掲載されて いるパルスシーケンスを動作させるプログラム BlochSolver-CPU のために書かれたものである.

BlochSolver-CPUは、GPU (Graphics Processor Unit)を使用せずに、CPU (Central Processing Unit) だけで動作する MRI シミュレーションプログラムであり、既に開発されている GPU を用いた BlochSolver (BlochSolver-GPU) と全く同様に動作する.

BlochSolver-CPUは、CPUのすべてのコアを使用するため(図-1)、動作中は、他のアプリケーションの使用は、かなり制限される. BlochSolver-GPUに比べて、演算速度は数十分の一であるが、画素数やサブボクセル数などを適切に選択すれば、さまざまな MR 撮像のシミュレーションに利用可能である.

BlochSolver-CPUは、Windows10に対応し、さらに、高速化のための拡張命令セットである AVX2 命令が動作する 64 ビット CPU(第 4 世代 Intel Core プロセッサ(2013 年 6 月以降)より搭載)を必 須としている.このため、インストールにあたっては、その確認が必要である.

以下,インストール方法,動作確認,シミュレーションの準備と画像再構成法,シミュレーション例,シミュレーション高速化の工夫,例題の順番で説明する.

P 920 23-54-				-		<
ファイル(E) オプション(<u>O</u>) 表示(<u>V</u>)						
プロセス パフォーマンス アプリの履歴 スター	トアップ ユーザー 詳細	サービス				
へ 名前	状態	100% CPU	19% ×ਦ੫	0% ディスク	0% ネットワーク	
アプリ (5)						^
> Python		0%	28.2 MB	0 MB/秒	0 Mbps	
> 📷 Windows コマンドプロセッサ (3)		99.5%	103.5 MB	o MB/秒	0 Mbps	
> 🐂 エクスプローラー		0%	51.9 MB	o MB/秒	0 Mbps	
> 👰 タスク マネージャー		0.1%	19.5 MB	0.1 MB/秒	0 Mbps	
> 间 Xモ帳		0%	2.0 MB	0 MB/秒	0 Mbps	
バックグラウンド プロセス (75)						
> 💋 64-bit Synaptics Pointing Enhan		0%	0.3 MB	0 MB/秒	0 Mbps	
> 🔲 Adobe Acrobat Update Service (0%	0.3 MB	o MB/秒	0 Mbps	
> Antimalware Service Executable		0%	115.6 MB	0 MB/秒	0 Mbps	
Application Frame Host		0%	0.9 MB	0 MB/秒	0 Mbps	
> 🔳 Bonjour Service		0%	0.8 MB	0 MB/秒	0 Mbps	
COM Surrogate		0%	0.7 MB	0 MB/秒	0 Mbps	~
<					>	
○ 簡易表示(D)					タスクの終了(<u>E</u>)
িজি ৬৫০ এখ−এন-				_		<
ファイル(E) オプション(Q) 表示(V)	।ची ज +€ =∺om	u 12-				
ノロセス ハフォーマンス アノリの設定 スター	トアツノ ユーリー 詳細	9-EX				
CPU 100% 3.40 GHz	CPU Inte	l(R) Core(TI	M) i7-7700	HQ CPU @	2.80GHz	
メモリ メモリ	2.554				100%	
3.1/16.0 GB (19%)						
ディスク 0 (C:)						
ディスク 1 (D: E:)						
イーサネット イーサネット	60秒		其大速度,	2.81.6	0	
▲ 送信: 0 受信: 0 Kbps	100% 3.40	GHz	シャッシュン ソケット: コア:	1	12	
NVIDIA GeForce G 1%	プロセス数 スレッド数 188 2052 稼働時間 5:01:33:06	t ハンドル数 76039	論理プロセッ 仮想化: Hyper-Vサ; L1 キャッシュ L2 キャッシュ L3 キャッシュ	サ数: 8 無効 ポート: はい : 256 KB : 1.0 ME : 6.0 ME	1 3 8	
🔿 簡易表示(D) 🔕 リソース モニターを開	1<					

図 1 BlochSolver-CPU 実行中のタスクマネージャー(プロセス(上)とパフォーマンス(下))

2. インストールと動作確認

ダウンロードサイトから、ダウンロードされるのは、実行モジュールを含んだ zip ファイル (blochsolver-cpu.zip)と、ユーザーズマニュアル (blochsolver-cpu_users_manual.pdf:このファイル) である. blochsolver-cpu.zip は、解凍先のフォルダを指定して解凍する. 解凍には 2-3 分かかり、 blochsolver-cpu というフォルダが作成される. フォルダの内容は次のようになっている.

<u>^</u>			
□ 名前	更新日時	種類	サイズ
128_sequence	2020/02/07 19:17	ファイル フォルダー	
256_sequence	2020/02/07 19:17	ファイル フォルダー	
📕 b0map	2020/02/09 19:58	ファイル フォルダー	
GUI_reconstruction_program	2020/02/09 15:39	ファイル フォルダー	
📜 Lib	2020/02/07 19:17	ファイル フォルダー	
📒 phantom	2020/02/07 19:18	ファイル フォルダー	
📜 Scripts	2020/02/07 19:18	ファイル フォルダー	
asyncio.pyd	2020/02/03 11:02	PYD ファイル	72 KB
_bz2.pyd	2020/02/03 11:02	PYD ファイル	88 KB
ctypes.pyd	2020/02/03 11:02	PYD ファイル	132 KB
decimal.pyd	2020/02/03 11:02	PYD ファイル	262 KB
_elementtree.pyd	2020/02/03 11:02	PYD ファイル	204 KB
_hashlib.pyd	2020/02/03 11:02	PYD ファイル	39 KB
_lzma.pyd	2020/02/03 11:02	PYD ファイル	252 KB
	2020/02/03 11:02	PYD ファイル	39 KB
_multiprocessing.pyd	2020/02/03 11:02	PYD ファイル	29 KB
overlapped.pyd	2020/02/03 11:02	PYD ファイル	44 KB
_queue.pyd	2020/02/03 11:02	PYD ファイル	28 KB
socket.pyd	2020/02/03 11:02	PYD ファイル	75 KB
sqlite3.pyd	2020/02/03 11:02	PYD ファイル	85 KB
ssl.pyd	2020/02/03 11:02	PYD ファイル	122 KB
blochsolver.bat	2020/02/03 11:06	Windows バッチ ファ	1 KB
blochsolver-cpu.exe	2020/02/04 9:27	アプリケーション	1,048 KB
command.txt	2020/02/07 12:19	テキスト ドキュメント	1 KB
ᆀ get-pip.py	2020/01/22 1:05	Python source file	1,735 KB
🗟 libcrypto-1_1.dll	2020/02/03 11:02	アプリケーション拡張	3,303 KB
🤹 libssl-1_1.dll	2020/02/03 11:02	アプリケーション拡張	671 KB
LICENSE.txt	2020/02/03 11:02	テキスト ドキュメント	13 KB
pyexpat.pyd	2020/02/03 11:02	PYD ファイル	195 KB
🌅 python.exe	2020/02/03 11:02	アプリケーション	98 KB
python3.dll	2020/02/03 11:02	アプリケーション拡張	58 KB
python37pth	2020/02/03 11:03	_PTH ファイル	1 KB
python37.dll	2020/02/03 11:02	アプリケーション拡張	3,664 KB
📕 python37.zip	2020/02/03 11:02	<u> 圧縮 (zip 形式) フォ</u>	2,340 KB
📑 pythonw.exe	2020/02/03 11:02	アプリケーション	97 KB
recon-0.1.0-cp37-cp37m-win_amd64	2020/01/22 1:49	WHL ファイル	247 KB
select.pyd	2020/02/03 11:02	PYD ファイル	27 KB
sqlite3.dll	2020/02/03 11:02	アプリケーション拡張	1,244 KB
unicodedata.pyd	2020/02/03 11:02	PYD ファイル	1,048 KB
s vcomp140.dll	2019/05/02 19:27	アプリケーション拡張	156 KB
🔄 vcruntime140.dll	2020/02/03 11:02	アプリケーション拡張	84 KB
🦻 viewer.bat	2020/01/22 2:15	Windows バッチ ファ	1 KB
viewer-0.1.0-py3-none-any.whl	2020/01/22 2:06	WHL ファイル	20 KB
winsound.pyd	2020/02/03 11:02	PYD ファイル	29 KB

図 2 blochsolver-cpu のフォルダの内容(拡張子は表示させておくこと!)

<blochsolver-cpu フォルダの説明>

このフォルダの中のファイルは、多くが、実行プログラム (blochsolver-cpu.exe) などが使用するものであるが、ユーザーが直接、使用するものもあり、それらについて以下に説明する.

まず,128_sequence というフォルダには,主に128×128×128 画素の数値ファントムを使用し, 二次元であれば,128×128 画素の画像およびそのマルチスライス像,三次元であれば,128×128× 32 画素および128×128×128 画素の画像を取得するためのパルスシーケンスが保存されている.

256_sequence というフォルダには、256×256×256 画素の数値ファントムを使用し、二次元であれば、256×256 画素の画像およびそのマルチスライス像、三次元であれば、256×256×32 画素および 256×256×256 画素の画像を取得するためのパルスシーケンスが保存されている.

b0map というフォルダには,静磁場不均一性を歳差運動周波数分布(Hz)で表す,数値ファント ムと同じフォーマットのファイル(b0map xy.fltなど)が保存されている.

GUI_reconstruction_program というフォルダには、各パルスシーケンスに対応した GUI の画像再構成・表示の実行プログラムが保存されている.

phantom というフォルダには、3 種類の 256×256×256 画素の数値ファントムと3 種類の 128×128 × 128 画素の数値ファントムが保存されている.ファントムは、スライス評価ファントム、NIST フ ァントム、Digital Brain Phantom の3 種類であり、詳細は教科書の第 I 部第3章に記載してある.

図2には表示されていないが、動作確認後には、result という実行結果を保存するフォルダと、実行ファイルのパラメタを保存する blochsolver.ini というファイルが新規に作られる.

フォルダ以外のファイルで,ユーザーが,直接使用するのは, blochsolver.bat, command.txt, viewer.bat の3個のファイルだけである. blochsolver.batは、シミュレーションプログラムを起動するためのバ ッチファイル, command.txtは、そのバッチファイルに与える各種パラメタを記載したテキストファ イル, viewer.batは、画像再構成およびファイル表示プログラムを実行するためのバッチファイルで ある.

<動作確認>

インストール直後に blochsolver.bat をダブルクリックすることにより,シミュレーションプログラ ムを起動する.そうすると,以下のようなコンソール画面が開いて,自動的に計算が始まる(図3). 計算の最中には,位相エンコードループが更新される毎にステップ数が出力されるが,このウイン ドウをクリックすると,一時的に計算が停止する.計算を再開するときは,リターンキーを押して 再開する.計算が終了すると,計算された MR 信号データが,自動的に作成された result フォルダの 中の,目付の名前のフォルダの中に保存される.



図 3 BlochSolver-CPU の実行画面(13 番目までの位相エンコードが終了)

計算が終了したら、この実行画面を、リターンキーを押して終了させ、GUI_reconstruction_program の中の256というフォルダの中のReconstruction_and_display_for_256x256_Cartesian.exeという画像再 構成プログラムを起動する.これを起動したら、メインメニューのFileを選んで、resultというフォ ルダの中の、日付の入ったフォルダの中に保存された MR 信号ファイルを読み込むと、自動的に画 像再構成が行われ、以下のように画像が表示される.



図4 再構成された画像. 上段は強度画像と位相画像. 下段は実部画像と虚部画像(負)

この画像は, 256×256×256 画素のスライス評価ファントムと, 256×256 画素のためのグラジエ ントエコーシーケンスを用いてシミュレーションしたものであり,後述するように,静磁場不均一 性がないため,サブボクセル数が1×1×1 (xyz)でも,アーチファクトは生じていない.

なお、このプログラムとパルスシーケンスの詳しい説明は、教科書を参照のこと.また、パルスシ ーケンスは、任意のエディタで編集できるが、Visual Studio Code を推奨する. ここで、サブボクセルについて、簡単に説明する.

サブボクセルとは、図5に示すように、一つの画素を,x,y,z方向にそれぞれ(等)間隔に分割したもので、被写体(連続体)のMRI現象を、Bloch方程式を用いてデジタル的に再現する場合に、画像化する画素よりもさらに小さい領域に分けて計算するときの、その領域のことである.

静磁場不均一性の影響の計算など、画像内の場所ごとに、異なる大きさのサブボクセルを設定す ることも考えられるが、設定方法や計算方法が煩雑になるので、BlochSolverでは、画像全体にわた って、同じようなサブボクセルへの分割を行っている.

実際に、サブボクセルへの分割が必要となるのは、後で示すように、スライスプロファイルを正確に再現したいとき、静磁場不均一性の影響を正しく再現したいとき、高速イメージングで高次エコーの効果を正しく再現したいときなどである.



図5 サブボクセルの概念

<BlochSolver を使用するための準備>

BlochSolverの実行は、前の節に書いたように、blochsolver.batの起動により行う.このバッチファイルの内容は、図6のようになっており、command.txtから、実行に必要なパラメタを読み込んでから実行ファイルが起動する.

1 blochsolver < command.txt
2 pause</pre>

図 6 BlochSolver を動かすためのバッチファイルの内容

初期設定の command.txt には、以下の内容が記載されており、それに従って詳細を説明する.

set dim 256 256 256 1 set subvoxel-dim 1 1 1 2 3 set actual-dim 256.0 256.0 256.0 set pd "./phantom/SlicePhantom/pd.flt" 4 set t1 "./phantom/SlicePhantom/t1.flt" 5 set t2 "./phantom/SlicePhantom/t2.flt" 6 7 set sequence "./256_sequence/2D_GradientEcho.seq.py" 8 start var PREFIX "./result/\${DATE}/\${SEQUENCE_NAME}-\${SERIAL_ID}" 9 save complex "\${PREFIX}-complex.flt" 10 quit 11

図 7 command.txt の例(初期設定のもの)

1 set dim 256 256 256

説明:数値ファントムのマトリクスサイズを,256×256 画素とする.xyzの順である.

2 set subvoxel-dim 1 1 1

説明:サブボクセル数を,x,y,z方向に,それぞれ1,1,1とする.

3 set actual-dim 256.0 256.0 256.0

説明:数値ファントムの物理的サイズを,x,y,zの順に256.0mm × 256.0mm × 256.0mm とする.

4 set pd "./phantom/SlicePhantom/pd.flt"

説明:プロトン密度分布のファイル名を指定する.

5 set t1 "./phantom/SlicePhantom/t1.flt"

説明:T1分布のファイル名を指定する.

6 set t2 "./phantom/SlicePhantom/t2.flt"

説明:T2分布のファイル名を指定する.

7 set sequence "./256_sequence/2D_GradientEcho.seq.py"

説明:シーケンスファイル名を指定する.

8 start

説明:シミュレーションを開始する.

9 var PREFIX "./result/\${DATE}/\${SEQUENCE_NAME}-\${SERIAL_ID}"

説明: PREFIX という変数を指定する.

10 save complex "\${PREFIX}-complex.flt"

説明:シミュレーション結果(複素数の信号)を,指定したファイル名で保存する.

11 quit

説明:プログラムを終了する.

次に,静磁場不均一性を導入した command.txt の例を示す. set b0 以下には,静磁場の空間的分布 を歳差運動周波数分布(Hz)で表した,数値ファントムと同じマトリクスサイズのファイルを指定 する. factor は,マトリクスの内容の数の倍数, offset は,それに加える周波数を数値で与える.

```
1
     set dim 256 256 256
 2 set subvoxel-dim 2 1 4
     set actual-dim 256.0 256.0 256.0
 3
    set pd "./phantom/SlicePhantom/pd.flt"
 4
     set t1 "./phantom/SlicePhantom/t1.flt"
 5
     set t2 "./phantom/SlicePhantom/t2.flt"
 6
     set b0 "./b0map/b0map_xy.flt" factor=4.0 offset=0.0
 7
     set sequence "./256_sequence/2D_GradientEcho.seq.py"
 8
9
     start
     var PREFIX "./result/${DATE}/${SEQUENCE_NAME}-${SERIAL_ID}"
10
     save complex "${PREFIX}-complex.flt"
11
12
     quit
```

図8 静磁場不均一性を導入するための記述法

シミュレーション結果を以下に示す.画像の下に示すのは、サブボクセル数である.このよう に、静磁場不均一性があると、画素内にサブボクセルを設定しない場合((a))には、強度画像に顕 著なアーチファクトが発生するが、スライス方向に4個、リード方向に2個のサブボクセルをそれ ぞれ設定すると、アーチファクトが緩和され、同時に設定すると(d)に示すようにほぼ消失する.



(a) $1 \times 1 \times 1$





(c) $2 \times 1 \times 1$

(d) $2 \times 1 \times 4$

図 9 静磁場不均一性があるときのシミュレーション結果(2D グラジエントエコー) -12-

<画像再構成>

Blochsolver-cpuは、MR 信号を作成するためのプログラムであり、ここで提供する画像再構成プロ グラムは、計算された信号を検証するための必要最小限のものである.また、画像再構成プログラ ムとしては、個々のパルスシーケンスに対応したものと、比較的汎用なもの(viewer.bat)の二種類 を提供している.以下に、それぞれの画像再構成プログラムについて説明する.

GUI_reconstruction_program のフォルダには、128 と 256 のフォルダがあり、それぞれのフォルダ に、128² ないし 128³ 画素対応の画像再構成・表示プログラムと、256² ないし 256³ 画素対応の画像再 構成・表示プログラムが保存されている(図 10 と図 11). それぞれの名称にしたがって、それに対 応する MR 信号を入力することにより、1 枚の画像の場合には、その再構成画像を表示し、マルチス ライスや三次元画像の場合には、その中心の画像のみを表示する(図 4 と図 9). そして、このプロ グラムを閉じると、abs.flt、phase.flt、real.flt、imaginary.flt に、単精度浮動小数点数(32 ビット)で、 すべての画像データを出力するので、後述の viewer.bat などのプログラムで全画像を表示する.また、 ImageJ などのプログラムで、32-bit Real を指定して開くことができる.

名前	種類	サイズ
Reconstruction_and_display_for_128x128_2D_Cartesian.exe	アプリケーション	128 KB
Reconstruction_and_display_for_128x128_2D_EPI.exe	アプリケーション	137 KB
Reconstruction_and_display_for_128x128_FSE_PDW.exe	アプリケーション	129 KB
Reconstruction_and_display_for_128x128_FSE_T2W.exe	アプリケーション	129 KB
Reconstruction_and_display_for_128x128x16_2D_multiecho.exe	アプリケーション	129 KB
Reconstruction_and_display_for_128x128x24_2D_FSE_PDW.exe	アプリケーション	129 KB
Reconstruction_and_display_for_128x128x24_2D_FSE_T2W.exe	アプリケーション	129 KB
Reconstruction_and_display_for_128x128x24_2D_multislice.exe	アプリケーション	129 KB
Reconstruction_and_display_for_128x128x32_3D.exe	アプリケーション	129 KB
Reconstruction_and_display_for_128x128x128_3D.exe	アプリケーション	129 KB

図 10 128²ないし 128³ 画素対応の画像再構成プログラム・表示

名前	種類	サイズ
Reconstruction_and_display_for_256x256_Cartesian.exe	アプリケーション	129 KB
Reconstruction_and_display_for_256x256_FSE_PDW.exe	アプリケーション	129 KB
Reconstruction_and_display_for_256x256_FSE_T2W.exe	アプリケーション	129 KB
Reconstruction_and_display_for_256x256x16_2D_multiecho.exe	アプリケーション	129 KB
Reconstruction_and_display_for_256x256x24_2D_FSE_PDW.exe	アプリケーション	129 KB
Reconstruction_and_display_for_256x256x24_2D_FSE_T2W.exe	アプリケーション	129 KB
Reconstruction_and_display_for_256x256x24_2D_multislice.exe	アプリケーション	137 KB
Reconstruction_and_display_for_256x256x32_3D.exe	アプリケーション	130 KB
Reconstruction_and_display_for_256x256x256_3D.exe	アプリケーション	137 KB

図 11 256² ないし 256³ 画素対応の画像再構成・表示プログラム

図 10 と図 11 に示したすべての画像再構成(ただし, EPI を除く)を,一つのプログラムで実施 できるのが,viewer.bat というプログラムである.このプログラムは,Cartesian sampling の MR 信号 の再構成にも,Non Cartesian sampling の MR 信号の画像再構成にも対応しているが,ここで提供し ているのは,Cartesian sampling に対応したものだけである.以下に,すべての MR 信号に共通した 使用方法を示し,各 MR 信号に対応したものに関しては,シミュレーション例のセクションで紹介 する. Viewer.bat をダブルクリックして立ち上げると、以下のようなウインドウが開く.



図 12 Viewer の初期画面

File メニューには、画像再構成や、表示したいファイルを Open する機能、表示した画像を保存 する機能などがある.また、Window メニューには、ウインドウを整列させたりする機能がある. 以下に、基本的な使用方法を解説する.

まず, File → Open と選ぶと, 次のようなダイアログボックスが開く.

🔳 Open				?	×
File					
Width			256		÷
Height			256		÷
Slice			256		•
Offset			0		-
Data Type			float32		~
Transform	1				
Ad	ld	Insert	Remove	Cle	ar
		OK	Cancel		

図 13 Viewer 使用の基本となるダイアログボックス

一番上のテキストボックスには、右のファイルアイコンを押すことによってファイルを開くダイ アログが開くので、そこから、処理(画像再構成やファイル表示)をしたいファイル名を選択す る.

次の3個のテキストボックスは、データのマトリクスサイズを示すもので、三次元データを基本 とし、ディフォルトのサイズは、width、height、slice ともに、256、256、256 となっているが、特 に、サイズの制約はない.また、width、height、slice の名称は、便宜的なもので、データファイル 内のデータの並び方 (C言語の配列の表現では array[slice][height][width]) に対応しているに過ぎな い.ただし、三次元データの表示を行う場合には、最初の表示面は、width × height の面であり、 slice で表示される枚数の面が、マウスローラーによってスクロールして表示されるようになってい る. 次のOffset と書かれたテキストボックスには、ファイルの先頭から、画像データまでのバイト数 を入力し、画像ヘッダーなどがある場合に使用する.次のData Type と書かれたボックスは、右の ボタンを押して出てくるメニューから、データタイプを選択する.BlochSolver-CPU で出力される データタイプは、単精度浮動小数点数(float32)を実数部と虚数部とする complex64 である.数値 ファントムや静磁場不均一性ファイルを入力する場合には、float32 を選択する.

その下の Transform で示されるボックスには、入力データに対する処理方法を、その中の Add を 押すことによって追加して選択し、選択が終わったら OK を押して処理を実行する. この処理方法 には、色々なものがあるので(詳しくは説明しないが、ぞれぞれの項目を実行して動作確認をする こと)、これらを複数組み合わせることにより、複雑な処理(マルチスライス高速スピンエコーの 画像再構成など)も実施可能である. 次のセクションで、個々の MR 信号データに関して、処理方 法と画像再構成結果を紹介する.

4. シミュレーション例と画像再構成

図 14 に、付属するパルスシーケンスを用いてシミュレーションを行ったときの計算時間を示す. 使用したハードは、図 15 に示す PC の CPU (Core i7 7700HQ (4 core)) と GPU (GTX 1070 (2560 core))である.また、図 16 に、CPU と GPU の計算時間を比較したグラフ、図 17 に、CPU で、128² 画素と 256² 画素の画像をシミュレーションしたときの各計算時間を比較したグラフを示す.

Hardware		CPU (Core i7 7700HQ, 4 core)		GPU (GTX 1070, 2560 core)	
Sequence	pixel subvoxel	128 ² / 128 ³	256 ² / 256 ³	128 ² / 128 ³	256 ² / 256 ³
2D Gradient Echo	$1\times1\times1$	15.8	220.7	0.58	7.2
2D Spin Echo	$1 \times 1 \times 1$	24.8	401.7	1.0	12.7
2D Inversion Recovery	$1\times1\times1$	50.1	783.1	2.0	24.0
2D PDW FSE	$1\times1\times16$	445.5	7082.7	13.7	207.6
2D T2W FSE	$1\times1\times32$	879.2	14148.0	26.1	403.0
2D multiple SE	$\texttt{1}\times\texttt{1}\times\texttt{4}$	887.9	14772.1	29.4	442.1
2D multislice GRE	$1\times1\times4$	1505.1	21002.0	41.8	631.8
2D multislice SE	$\texttt{1}\times\texttt{1}\times\texttt{4}$	2396.7	38066.0	77.6	1143.9
2D multislice PDW FSE	$1\times1\times8$	5256.0	-	163.1	2506.4
2D multislice T2W FSE	$\texttt{1}\times\texttt{1}\times\texttt{8}$	5304.9	-	162.6	2504.5
3D GRE (32 slice)	$\textbf{1}\times\textbf{1}\times\textbf{1}$	76.1	1735.5	4.2	58.5
3D GRE (128 / 256 slice)	$\textbf{1}\times\textbf{1}\times\textbf{1}$	300.8	13877.1	16.1	451.3
2D Gradient echo EPI	$\textbf{1}\times\textbf{1}\times\textbf{1}$	1.58	11.6	0.1	0.4

図 14 シミュレーションの計算時間の比較(CPU vs GPU). 数字は秒.



図 15 シミュレーションに使用した PC (OMEN by HP Laptop 17-an0xx) -18-



図 16 CPU 計算時間の GPU の計算時間に対するプロット(二次元撮像のみ)



Calculation time : 128 vs 256

図 17 CPU を用いた計算時間: 128² 画素に対する 256² 画素のプロット

4-1 グラジエントエコー法

図 13 に示すダイアログボックスで、再構成するファイルを選んだ後に、256×256 画素の場合に は、図 18 のように、データサイズを、256、256、1 とし、BlochSolver の出力データの標準形式で ある complex64 を選択する. 次に、図 19 に示すように、Add を押して、処理として FFT を選び、 その configure のダイアログボックス(図 20)で、Z のチェックボックスを外して(Z 方向には FFT をしないという意味)OK を押し、その後、Transform ボックスの下の OK を押して再構成を行 う (図 21). ただし、1 slice の場合には、チェックボックスを外さなくても結果は同じになる.

🔳 Open	? ×	Open ? ×
File	2020-02-06/2D GradientEcho 2pai-17-complex.flt	File 2020-02-06/2D GradientEcho 2pai-17-complex.flt
Width	256	Width 256
Height	256	Height 256
Slice	1	Slice 1
Offset	0	Offset 0
Data Type	complex64 🗸	Data Type complex64 ~
Transform	int64 uint8 uint16 uint32 uint64 float32 float64 <u>complex64</u>	Transform Absolute Absolute FFT IFFT Imaginary Phase Real Reoder FSE Signal Reshape Swap Axes
Ac	ld Insert Remove Clear	Add Insert Remove Clear
	OK Cancel	OK Cancel

図 18 Data type の選択

図 19 処理の選択

python	?	×
х		
Y		
Z		
Normalization		
🥔 ОК	💥 Cancel	

図 20 FFT に対する configure -20-



図 21 再構成された画像の viewer における表示

図 22 に, 128×128 画素と 256×256 画素で,それぞれグラジエントエコーシーケンスを用いてシ ミュレーションを行った結果を示す.



128² (15.8 s) 256² (306.8 s)

図 22 グラジエントエコー画像における画素数の比較

4-2 スピンエコー法

シングルスピンエコー法も、画像再構成法は、グラジエントエコー法と同一である.



128² (24.8 s)

256² (401.7 s)

図 23 スピンエコー画像における画素数の比較

4-3 インバージョンリカバリ法

インバージョンリカバリ法も、画像再構成法は、グラジエントエコー法と同一である.



128² (50.1 s) 256² (900.4 s)

図 24 インバージョンリカバリ画像における画素数の比較

4-4 プロトン密度強調高速スピンエコー法

シングルスライスのプロトン密度強調高速スピンエコー法によるシミュレーション信号の画像再 構成では、まず、図 25 のように設定して、ファイルを開く.

🔳 Open			×
File	sult/2020-02-07/2D FastSpinEcho-26-comple	ex.flt	
Width	2048	÷	
Height	1	•	
Slice	32	-	
Offset	0	•	
Data Type	complex64	\sim	
	Id Insert Remove C	Par	
Ac	id Insert Remove C	lear	
	OK Cancel		

図 25 プロトン密度強調高速スピンエコー信号ファイルを開くダイアログボックス

🔳 Open		×		
File	sult/2020-02-07/2D FastSpinEcho-26-complex.fl	t 🕒		
Width	2048	3		
Height	1			
Slice	32	3		
Offset	0	3		
Data Type	complex64 ~	1		
Transfor	n			
	Swap Axes v configure Reshape v configure		III python	
			Width	256
			Height	256
A	dd Insert Remove Clear		Slice	1
	OK Cancel		-	OK 🛛 🗶 Cancel

図 26 Swap Axes と Reshape を追加し, Reshape の configure は上のように設定

🔳 Open	?	×		
File	sult/2020-02-07/2D FastSpinEcho-26-complex	flt 🕒		
Width	2048	•		
Height	1	÷		
Slice	32	*		
Offset	0	* *		
Data Type	complex64	~		
Transform	n			
	Swap Axes v configure			
	Reshape v configure			
Re	eoder FSE Signal 🗸 configure		E python	? >
			Phase Encoding Order	PDW ~
			Number of Phase Encoding [32 🔹
Ad	ld Insert Remove Clea	ar	Number of Echos	8
	OK Cancel		🛹 ОК	X Cancel

図 27 Reorder FSE signal を追加し、この configure は上のように設定

🔳 Open	? ×		
File	sult/2020-02-07/2D FastSpinEcho-26-complex.flt		
Width	2048		
Height	1		
Slice	32 🜩		
Offset	0		
Data Type	complex64 ~		
Transform	1		
	Swap Axes V configure		
	Reshape v configure	python ?	×
Re	oder FSE Signal V configure	× 🗹	
	FFT V configure	Y 🗹	
		z 🗆	
Ad	d Insert Remove Clear	Normalization 🗹	
	OK Cancel	K Cancel	

図 28 FFT を追加し, configure は上のように設定

上記のように設定して, OK ボタンを押すと再構成画像が表示される.



256² (7082.7 s)

図 29 プロトン密度強調高速スピンエコー画像における画素数の比較

128×128の画素の画像再構成を行う場合には、図 25 において 2048 → 1024, 32 → 16, 図 26 の configure において 128, 128, 1, 図 27 の configure において, 16, 8とする.

4-5 T2 強調高速スピンエコー法

シングルスライスの T2 強調高速スピンエコー法によるシミュレーション信号の画像再構成では, まず、図25と同様に設定して、ファイルを開く.その他は、プロトン密度強調高速スピンエコーの 場合と同様であるが, Reorder FSE signal の configure は,図 30 のように T2W を選択する.

💵 python		?	×
Phase Encoding Order	T2W		~
Number of Phase Encoding	32		▲ ▼
Number of Echos	8		*
🥔 ОК	🗶 Cancel		

図 30 Reorder FSE signal の configure



 $128^2 (879.2 \text{ s}) \qquad 256^2 (14148.0 \text{ s})$

図 31 プロトン密度強調高速スピンエコー画像における画素数の比較

128×128の画素の画像再構成を行う場合には,図25において2048 → 1024,32 → 16,図26の configure において 128, 128, 1, 図 27 の configure において, 16, 8とする.

4-6 マルチプルスピンエコー法(16枚)

図 32 のように設定してファイルを読み込み,最初に swap axes の処理を行い, configure で, OK ボ タンを押す.次に,図 33 のように設定して OK を押せば,再構成画像が出力される.

🔳 Open		×		
File	2020-02-05/2D_multiple_SpinEcho-697-complex.flt			
Width	256			
Height	16			
Slice	256			
Offset	0			
Data Type	complex64 ~			
	Swap Axes V configure		🔲 python	?
			Axis1	0
A	ld Insert Remove Clear		Axis2	1
	OK Cancel		🥔 ОК	🗶 Cancel

図 32 マルチプルスピンエコー信号ファイルを開くダイアログボックスと swap axes の選択

Open	? X		
File	2020-02-05/2D_multiple_SpinEcho-697-complex.ftt		
Width	256		
Height	16		
Slice	256		
Offset	0		
Data Type	complex64 V		
- Transfor	m		
	Swap Axes V configure		
		E python	? ×
	FFI V contidure		_
		×	\checkmark
		Y	
		z	
A	dd Insert Remove Clear	Normalization	
	OK Cancel	🥔 ОК	🗶 Cancel

図 33 FFT を追加し, configure で z のチェックを外す



図 34 128 × 128 画素のマルチプルスピンエコー画像(887.9 秒)

128×128の画像では、図 32 において、256→128とする.



図 35 256 × 256 画素のマルチプルスピンエコー画像(14772.1 秒)

4-7 マルチスライスグラジエントエコー法(24枚)

マルチエコーの場合と、画像再構成法は同一である.ただし、図 32 における height の数字を 16 から 24 に変更して、同様の操作を行う.



図 36 128 × 128 画素のマルチスライスグラジエントエコー画像(1505.1 秒)

128×128の画像では、ファイル入力ダイアログボックスにおいて、256→128とする.



図 37 256 × 256 画素のマルチスライスグラジエントエコー画像(21002.0 秒)

4-8 マルチスライススピンエコー法(24枚)

画像再構成法は、マルチスライスグラジエントエコーの場合と同一である.



図 38 128 × 128 画素のマルチスライススピンエコー画像(2396.7 秒)

128×128の画像では、ファイル入力ダイアログボックスにおいて、256→128とする.



図 39 256 × 256 画素のマルチスライススピンエコー画像(38066.0 秒)

4-9 3D グラジエントエコー法(32 スライス)

3D グラジエントエコーシーケンスのシミュレーションによって得られた信号の画像再構成では, 入力のダイアログボックスを,図40のように設定し,OK ボタンを押して再構成を行う.FFT は, x, y, zのすべての方向に関して行う.また,128×128×32 画素の画像の再構成には,図40 におい て,256 → 128 とする.

🔲 Open	?	×		
File	pu/result/2020-02-06/3D GRASS-23-complex.	flt 🕒		
Width	256	÷		
Height	256	▲ ▼		
Slice	32	+		
Offset	0	* *		
Data Type	complex64	~		
- Iranstorn	n		python	?
	FFT v confiaure		×	\checkmark
			Y	\checkmark
			z	
Ad	ld Insert Remove Clea	r	Normalization	\square
	OK Cancel		🥔 ОК	💥 Cancel

図 40 3D グラジエントエコー信号ファイルを開くダイアログボックスと FFT の configure



図 41 128 × 128 × 32 画素の 3D グラジエントエコー画像(76.1 秒)



図 42 256 × 256 × 32 画素の 3D グラジエントエコー画像(1735.5 秒)

4-10 3D グラジエントエコー法(128/256 スライス)

上記の 3D グラジエントエコーシーケンスのシミュレーションによって得られた信号の画像再構成では,図40の入力のダイアログボックスにおいて,sliceを32から128ないし256に変更し,OK ボタンを押して再構成を行う.FFTは,x,y,zのすべての方向に関して行う.



図 43 128 × 128 × 128 画素の 3D グラジエントエコー画像(300.8 秒)



図 44 256 × 256 × 256 画素の 3D グラジエントエコー画像(13877.1 秒)

4-11 エコープラナーイメージング

エコープラナーイメージングの画像再構成は, viewer.bat は対応していないので, GUI のプログラ ムを使用する. 図 45 の画像は, パルスシーケンスは同一であるが, 使用する数値ファントムのサイ ズが, 128³ と 256³ と異なっている.



図 45 128³の数値ファントムを使った場合と 256³の数値ファントムを使った場合の EPI 画像

5. シミュレーションの高速化の工夫

シミュレーションを高速化するためには, GPU の使用の他, **画像マトリクス数**と**サブボクセル数** の選択が重要である.

<画像マトリクス数>

画像マトリクス数に関しては、128²ないし128³画素の画像を使用するか、256²ないし256³画素の 画像を選択する二つの選択肢がある.もちろん、64²ないし64³画素の画像、もしくは512²ないし512³ 画素の画像の使用も、仕様上は可能であるが、現実的な選択肢ではない(特別な場合には選択肢と なるかも知れない).また、2のべき乗ではない画像マトリクスサイズを使用することも可能である が、MR 画像の標準サイズは、256²ないし512²であるため、あまり現実的ではない.

図 17 に示したように、同様のパルスシーケンスで、128² 画素の画像のシミュレーションを行った 場合と、256² 画素の画像のシミュレーションを行った場合には、15 倍程度の処理速度の差があるの で、シーケンスの調整や、パラメタの探索などには 128² 画素の画像などを使用し、最終的な評価に は 256² 画素の画像を使用するなど、目的に応じて使い分けると便利である.

<サブボクセル数>

アーチファクトのない画像を取得したり、実験データなどと比較するためのシミュレーション画 像を得るためには、サブボクセル数を正しく設定する必要がある.複数のサブボクセルを確保しな ければならないのは、次の場合である.

(1) スライスプロファイルを正しく表現したい場合

スライスに使用する選択励起パルスの種類にも依存するが、一般的な選択励起パルスを印加した 場合、核磁化は、画素内で、スライス面に垂直な方向に複数回回転するので、その現象を正しく表現 するためには、その方向に、複数個のサブボクセルが必要である.提供しているパルスシーケンス では、±2πの hamming windowed sinc パルスを使用しているが、この場合には、経験上、スライス方 向のサブボクセル数は 4 以上が望ましい.また、マルチスライスのように、スライス厚方向にグラ ジエントを印加して,繰り返し RF パルスを印加する場合には,スライスプロファイルの正確な再現 が必要となるため,さらに多数のサブボクセルが必要になる.その最も顕著な例が,T2 強調高速ス ピンエコー法であり,8エコーの場合でも,16~32 個のサブボクセルが必要となる.

(2) 静磁場不均一性が存在する場合

静磁場不均一性が存在する場合,核磁化の位相は,画素内で大きく変化することがあり,このような場合には,信号強度が低下したり失われる場合がある.このような現象を正しく記述するためには,画素内に複数のサブボクセルを確保する必要がある.図9に示した例は,その典型例であり,シングルスライスの画像の場合には,リード方向に2個,位相エンコード方向に1個,スライス厚方向に4個のサブボクセルを確保すれば,多くの場合,アーチファクトを解消することができる.

(3) T₂ << TR でない場合

RF パルスで励起する場合に,前の RF パルスによる横磁化が残留している場合.特に, RF スポイ ル型グラジエントエコー法などで,ボクセルの中に,過去の RF パルスにより発生した横磁化成分が 残っている場合には,信号読み出し方向に,T₂と TR の関係に依存して決まる多数のサブボクセル 数を確保する必要がある.

6. 例題

以下の例などが、シミュレータを使った MRI の学習に役立つと思われる.

1. TR, TE, TI などと共に変化する画素強度変化の計測と T1, T2, PD などの算出

シングルスライススピンエコー法で、TR や TE を変化させた場合、ファントムの画像コントラストがどのように変化するかを理解することは、MR 撮像の基本である.そこで、T1 強調画像、T2 強調画像、PD 強調画像がどのようなものとなるか、ファントムと Digital Brain Phantom などで確認する.また、信号強度式を用いて、T1、T2 などを逆算する.

2. 静磁場不均一性の MR 画像への影響

静磁場不均一性が存在する場合に,スピンエコー法,グラジエントエコー法,EPI などで,どのような画像の変化が生じるかなどを体験する.ただし,サブボクセル数不足に伴うアーチファクトが 出やすいので,十分な数のサブボクセル数を確保する.

3. グラジエントエコー法における FA, TR の変化による画像コントラストの変化

グラジエントエコー法で, TR と FA を変化させた場合, 信号変化がどのようになるかを体験する. 本マニュアルでは, 計算時間などの都合で, RF スポイル型グラジエントエコー法や, バランス型定 常グラジエントエコー法には, 触れることができなかったが, 教科書にはシーケンスのソースコー ドが記載してあるので, それらを参考にして, グラジエントエコーの多彩な画像コントラストを体 験する.

 フリップ角を小さくした極限, TR や TE を短くした極限, また長くした極限でどのような画像が 出現するかを調べる.

あまり現実的ではないが、現実的な制約がない場合に、どのような画像が得られるか興味深い.

5. 新しいパルスシーケンスの開発

自分で、これまでに提案されていないパルスシーケンスを開発する.

7. むすび

教科書と合わせて、このマニュアルを使いながら、MRI シミュレーションが行えるように本マニ ュアルを作成しました.ただし、情報不足の面などがありますので、ご不明な点や、ご提案などは、 ご遠慮なく、

info_at_mrisimulations.com (_at_は, @)

まで、ご連絡下さい. できるだけ速やかに、回答したいと思います.