Installation of Gadgetron-SPEN

1. **Install the dependencies**

The Linux installation of Gadgetron 4.xx is detailed here: <https://github.com/gadgetron/gadgetron/wiki/Linux-Installation-(Gadgetron-4)>

Follow the first step to install the dependencies, with or without CUDA depending of your computer.

**Before installing ISMRMRD,** I recommend to create a folder named “Code” inside your home/user/ folder to have all the important codes at the same place.

You can then Git clone everything (ISMRMRD, Gadgetron, Siemens\_to\_ISMRMRD…) in this folder.

Git clone, compile and install ISMRMRD as detailed in the Linux installation procedure.

1. **Git clone the SPEN Gadgetron**

I prepared a repository of the Gadgetron 4.1 including the modification required to embed the Matlab SPEN reconstruction. This repository is accessible here: [git@github.com:maximeYon/gadgetron-SPEN.git](mailto:git@github.com:maximeYon/gadgetron-SPEN.git)

You need to git clone this gadgetron-SPEN with the following command:

* git clone [git@github.com:maximeYon/gadgetron-SPEN.git](mailto:git@github.com:maximeYon/gadgetron-SPEN.git)

Then go inside the folder gadgetron-SPEN, create a build folder and builg and install gadgetron-SPEN:

* cd gadgetron-SPEN
* mkdir build
* cd build
* cmake ../
* make –j 8 (The –j “number of processor” allows you to use multiple processor with make)
* sudo make install

1. **Git clone the SPEN Matlab reconstruction code**

I prepared a repository containing the SPEN Matlab reconstruction code (Martins Otikovs, Lucio Frydman, Weizmann Institute of Science 07/2020). I just modified it slightly to avoid the *parfor* in case of less than 3 slices: the *parfor* is long to start and for few slices its use increase the reconstruction time.

Go into your Code folder:

* cd /home/user/Code/
* git clone [git@github.com:maximeYon/Gadg\_SPEN\_reco\_Matlab.git](mailto:git@github.com:maximeYon/Gadg_SPEN_reco_Matlab.git)

This folder contain all the Matlab functions and Siemens configuration files necessary for automatic SPEN reconstruction through the Gadgetron frameworks.

1. **Git clone Siemens to ISMRMRD**

We need to install the Siemens\_to\_ISMRMRD program to be able to convert Siemens Data to ISMRMRD format. To do that, go in your Code folder:

* cd /home/user/Code/
* git clone https://github.com/ismrmrd/siemens\_to\_ismrmrd.git
* cd siemens\_to\_ismrmrd
* mkdir build
* cd build
* cmake ../
* make -j 8
* sudo make install

1. **Add the SPEN parameter maps to Siemens to ISMRMRD**

The SPEN reconstruction required to retrieve some specific parameters from the sequence such as the time bandwidth product (Rvalue) of the Chirp pulse. These parameters need to be included during the conversion from Siemens to ISMRMRD. To allow that copy the two parameter maps: /home/user/Code/Gadg\_SPEN\_reco\_Matlab/Siemens\_Parameters\_Maps\_and\_Ice/IsmrmrdParameterMap\_Siemens\_SPEN.xml

/home/user/Code/Gadg\_SPEN\_reco\_Matlab/Siemens\_Parameters\_Maps\_and\_Ice/IsmrmrdParameterMap\_Siemens\_SPEN.xsl

To: /home/ user /Code/siemens\_to\_ismrmrd/parameter\_maps

Then go to:

/home/ user /Code/siemens\_to\_ismrmrd/

Open the CMakeLists.txt with a text editor and add following line after the one corresponding to EPI:

${CMAKE\_CURRENT\_SOURCE\_DIR}/parameter\_maps/IsmrmrdParameterMap\_Siemens\_EPI.xsl ${CMAKE\_CURRENT\_SOURCE\_DIR}/parameter\_maps/IsmrmrdParameterMap\_Siemens\_SPEN.xsl

${CMAKE\_CURRENT\_SOURCE\_DIR}/parameter\_maps/IsmrmrdParameterMap\_Siemens\_SPEN.xml

And again: ${CMAKE\_CURRENT\_SOURCE\_DIR}/parameter\_maps/IsmrmrdParameterMap\_Siemens\_EPI.xsl ${CMAKE\_CURRENT\_SOURCE\_DIR}/parameter\_maps/IsmrmrdParameterMap\_Siemens\_SPEN.xsl

Then compile and install again Siemens\_to\_ISMRMRD:

* cd build
* cmake ../
* make -j 8
* sudo make install

1. **Modify your bashrc file**

You need to add the localization of the Gadgetron, ISMRMRD and MATLAB to your .bashrc file.

Open you bashrc with your favorite text editor, such as gedit:

* cd /home/user/
* gedit .bashrc

Add the following line at the end of the bashrc

export GADGETRON\_HOME=/usr/local

export ISMRMRD\_HOME=/usr/local

export PATH=$PATH:$GADGETRON\_HOME/bin:$ISMRMRD\_HOME/bin

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$GADGETRON\_HOME/lib:$ISMRMRD\_HOME/lib

export MATLAB\_ROOT=/usr/local/MATLAB/R2018b/

export MATLAB\_HOME=/usr/local/MATLAB/R2018b/

export PATH=$PATH:/usr/local/MATLAB/R2018b/bin

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/MATLAB/R2018b/bin/glnxa64

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/opt/intel/mkl/lib/intel64:/opt/intel/compiler/lib/intel64:/opt/intel/lib/intel64

Restart your computer or use ● source bashrc to consider these modifications.

Now you can see if your Gadgetron is working by typing in a terminal: gadgetron

You should see a message like this:

07-10 12:42:29.055 INFO [main.cpp:49] Gadgetron 4.1.0 [5270c07c6d9760cfcad56a5e99ff08d03df6e685]

07-10 12:42:29.055 INFO [main.cpp:50] Running on port 9002

That mean you have an active Gadgetron instance ready for reconstruction.

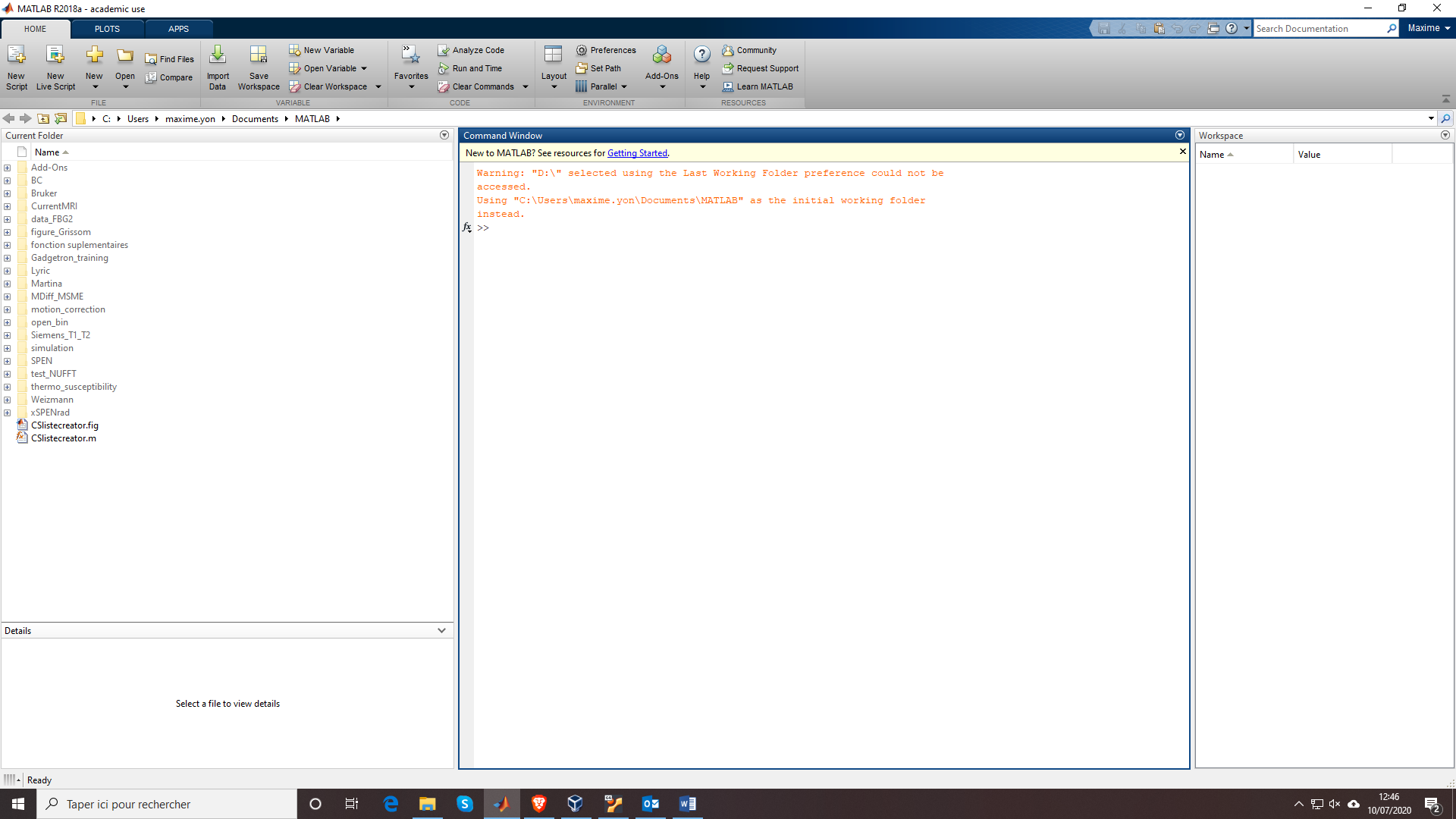
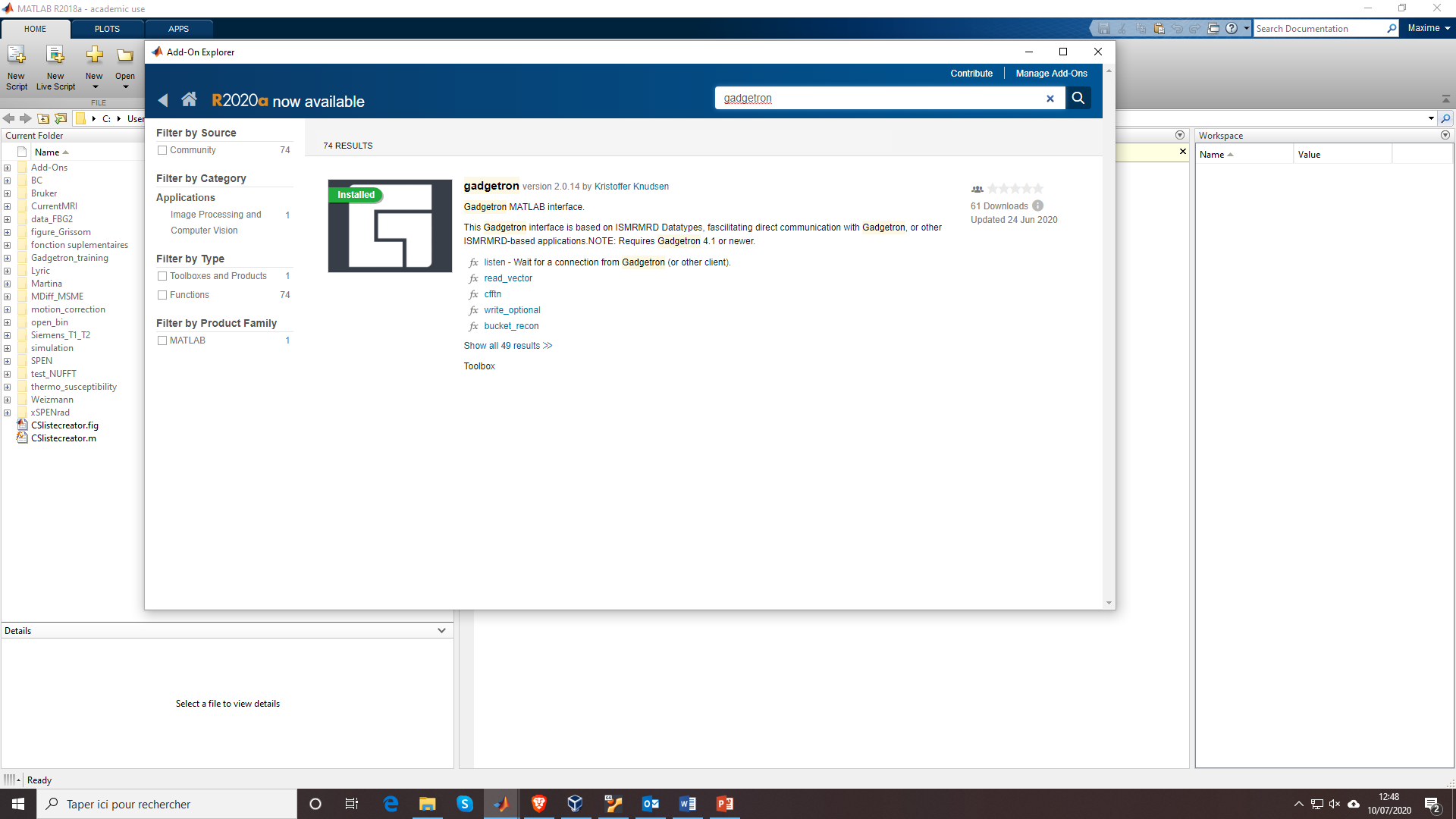
1. **Insure compatibility with matlab**

Now that you add those libraries to the path, you might have an incompatibility with one Matlab library. You can check that by trying to recompile the Gadgetron. If you have an error (xml-sethashsalt), you need to change the name of the “libexpat.so.1” library of Matlab. If the compilation works, jump to the next step (8).

* cd /usr/local/MATLAB/R2019b/bin/glnxa64
* mv libexpat.so.1 libexpat.so.1.NOFIND

Matlab will still works as it find the version of this library inside the Gadgetron library files.

1. **Allows communication between Gadgetron and Matlab (at least R2018b)**

To insure communication between Gadgetron and Matlab you need to install the gadgetron add-on.

In Matlab go to add-Ons and install the gadgetron :

The add-ons folder should be located in your main folder (/home/username/) or in documents.

Go in this folder and follow the folders until you find +gadgetron. In my case its : /home/user/MATLAB Add-Ons/Toolboxes/gadgetron/+gadgetron

Cut the folder: /home/user/Code/Gadg\_SPEN\_reco\_Matlab/ gadgetron4.1\_Matlab\_functions/+SPEN

Past it to: /home/user/MATLAB Add-Ons/Toolboxes/gadgetron/+gadgetron/+SPEN

1. **Set up the correct path in the matlab code**

In the file: /home/user/MATLAB Add-Ons/Toolboxes/gadgetron/+gadgetron/+SPEN /reconstruct\_SPEN.m and /home/user/MATLAB Add-Ons/Toolboxes/gadgetron/+gadgetron/+SPEN /reconstruct\_SPEN\_diff.m

Change the addpath line to be able to retrieve the reconstruction functions:

Change the user name in the line:

addpath(genpath('/home/user/Code/Gadg\_SPEN\_reco\_Matlab/ gadgetron4.1\_Matlab\_functions /SPEN\_Siemens\_Recon/'))

1. **Test the reconstruction**

The test of the reconstruction required two steps: The first one is to convert the Siemens .dat SPEN data to ISMRMRD format .h5. The second is to perform the reconstruction through the Gadgetron.

In home/user/ create a Data folder, inside of this folder create a RAW folder.

* cd /home/user/
* mkdir Data
* cd Data
* mkdir RAW

Copy the two script in /home/user/Code/Gadg\_SPEN\_reco\_Matlab/test\_scripts to /home/user/Data/

Add some SPEN test data in the RAW folder.

Open with a text editor the script *conversion\_spen* and change the file name according to your data.

Execute *conversion\_spen*:

● ./conversion\_spen

You should see to .h5 file appearing in the RAW, it is the data and noise corresponding to your SPEN data.

Now start a Gadgetron instance:

● gadgetron

In another terminal launch: *launch\_reco\_spen*:

● ./ launch\_reco\_spen

You should see the Gadgetron terminal displaying some log, calling matlab and display the matlab disp messages such as:

SCCing

Duration of SPEN reco = xxx.xx s

At the end of the reconstruction, you should have the following lines:

03-03 13:29:08.546 INFO [GadgetStreamController.cpp:164] Shutting down stream and closing up shop...

03-03 13:29:08.546 INFO [GadgetStreamController.cpp:190] Stream is closed

You should also see an out.h5 script inside your Data folder. This file contains the reconstructed SPEN images in ISMRMRD format. That mean that the reconstruction works.