Tutorial on using FFTW

Xubo Liu

Feb, 2024

1 Linking FFTW to Matlab

The FFTW is a C++ library that designed for FFT and its inverse. It is not set up in general so one needs to download it first from FFTW. After installing the FFTW library, we can use the header file fftw3.h in C++ and compile functions in Matlab through MEX. The C++ source code is written in C language and passes objects into Matlab with the help of some commands like ”mxCreateDoubleMatrix”, ”mxGetPr”, etc. Users can learn the syntax by themselves. Here, we give a tutorial for linking the external library to Matlab and make it located in the Matlab enviornment.

1.1 Windows

Computers with a Windows system can visit the installed FFTW library directly by specifying the path to the folder where FFTW is downloaded.

1.2 MacOS with any Intel chips (before 2021)

Users who have the OS system based on the X86_64 architecture can easily link the library to Matlab. The FFTW will generally create files in the root folder on an OS system. In general the header file “fftw3.h” is under the path */usr/local/include* and the compiled dynamical library with a suffix *_x86_64.dylib* is saved under */usr/local/lib*. To compile a MEX file in Matlab (for instance, the “log_density.cpp” file in the package), one can simply do:

```
mex -I/usr/local/include -L/usr/local/lib -lfftw3 log_density.cpp -output log_density
```

which will compile a file “log_density.mexmaci64” that can be used by OS users. And users only need to compile a C++ function in Matlab the first time they use it.

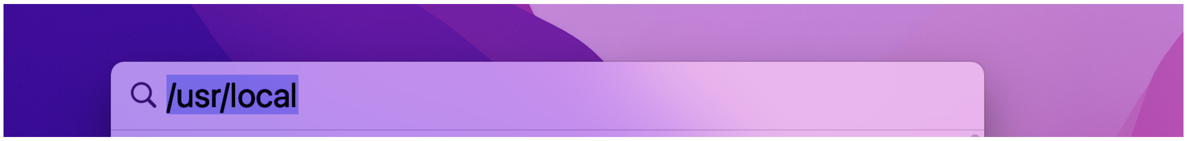
1.3 MacOS with any M chips

1.3.1 Approach 1

After 2021, the MacOS will be based on new M chips. Users may encounter compatibility issues when locating FFTW because the system is built on ARM architecture and an ARM version of Matlab is not available yet to detect a dynamic lib file that ends with “.a”. We spent time on solving this issue and successfully compiled all necessary MEX files. Here we summarized the procedures.

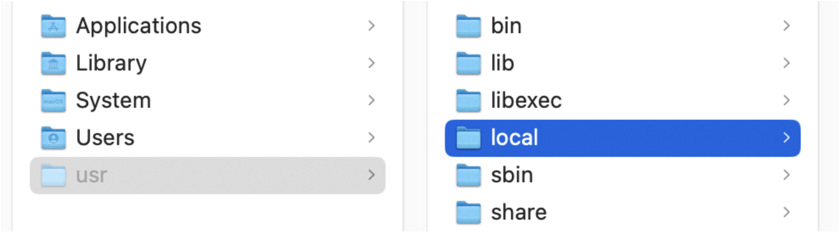
One can try to build their own “mexmaci64” files via MEX (the way that we recommend).

1. Put the “fftw3.h” header file we provided in the folder “/usr/local/include”. If the path doesn’t exist, please create it first. Macbook users can locate the path by searching “/usr/local” in the Spotlight Search (usually in the toolbar in the upper right corner of the desktop):



/usr/local

Then we can find:



2. Open or create the path: "/usr/local/Cellar/" and put the "fftw_x86_64" folder inside and get the path "/usr/local/Cellar/fftw_x86_64". Refer to Step 1 to find the path and folders.

3. To compile a C++ function via MEX, for instance, the "log_density.cpp", we can execute the following code in the Matlab Command Window:

   ```
   mex -I/usr/local/include -L/usr/local/Cellar/fftw_x86_64/lib -lfftw3 log_density.cpp -output log_density
   ```

   then we have a "log_density.mexmaci64" function saved in the current folder of Matlab that can be directly used. For first-time users, please complete the compilation of all C++ functions by running the "compile_cpp.m" file in the package.

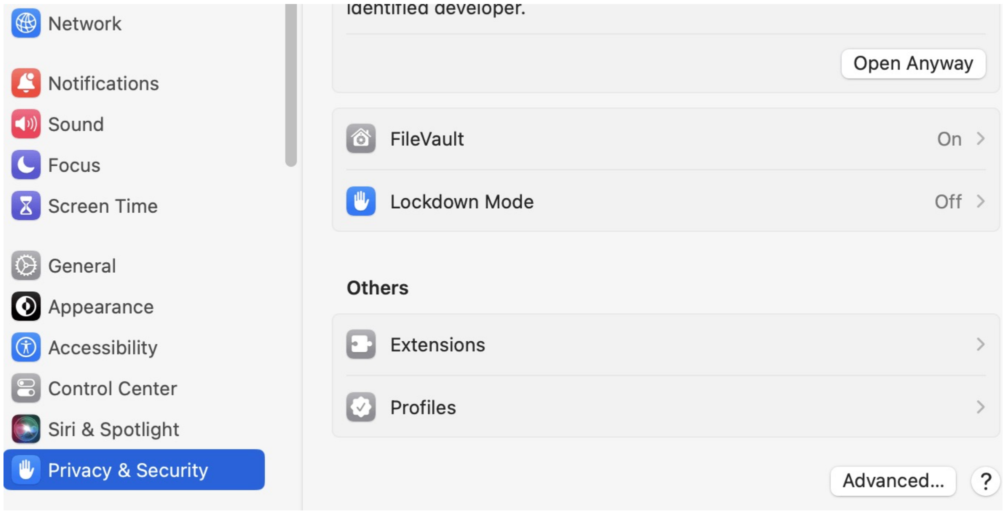
1.3.2 Approach 2

For convenience, we export the "libfftw3.3.dylib" file, a current version of fftw3.h that can be loaded by users. Please put the "libfftw3.3.dylib" under the path:

/usr/local/opt/fftw/lib/libfftw3.3.dylib

If the folder "opt", "fftw" or "lib" don't exist for new users, please create them to complete the path. When directly calling all the "mexmaci64" functions, some security or privacy restrictions might come into play. When restricted, we suggest users to follow the steps below:

1. Open System Settings. Click Privacy & Security, scroll down, and click the "Open Anyway" button to confirm your intent to open the functions.


identified developer.

FileVault On

Lockdown Mode Off

2. A warning message will appear. Click "cancel" to close it.

3. For each "mexmaci64" function, users need to repeat step 1 and step 2. Repeating changing the privacy setting until all functions can work without warnings and errors.