Installation

Table of contents

Installing Python

I would highly recommend using Anaconda Python 3, which is a distribution of Python that comes pre-bundled with many of the most popular packages, including most of those used by GENFIRE. Choose the installer that matches your operating system, and make sure to choose Python 3.

If you are familiar with Python and have an existing installation you would like to use, that is fine, but be aware we will be less likely to provide troubleshooting if you deviate from the recommended installation procedure.

Virtual environments

A virtual environment is a sandbox environment where you can install specific versions of various software packages without affecting the rest of your system. This is not required for GENFIRE to work, but it is highly recommended. A smart practice is to have a separate python environment for all of your major projects.

You can use either Anaconda's command `conda create` or `virtualenv` to create a virtual environment. Open a terminal and navigate to a new folder.

Virtualenv

```
pip3 install virtualenv
virtualenv -p python3 yourenvname
source yourenvname/bin/activate
```

Anaconda

```
conda create -n yourenvname python=3.6 anaconda
activate yourenvname
```

More details can be found here or just use Google. If you are on Windows and the command `pip3` is not found, you will have to provide the full path, i.e.

```
C:\path\to\Anaconda\Scripts\pip3 install virtualenv
```

Or add the path prefix to your PATH environmental variable so that the OS can find the executable automatically.

Once you activate the virtual environment, your terminal prompt should change, which is a helpful indicator. If you would like to make this environment persistent on Mac or Linux, you can add the above `source` command to the end of you `~/.bash_profile` or `~/.bashrc` file, respectively.

Installing with pip

`GENFIRE` is hosted on PyPi, and thus can be easily installed with `pip`:

```
pip3 install genfire
```

You may need root privileges (sudo), depending on your environment. If this `pip3` installation is successful, you can now open the GUI by running the launch.py script inside of the package

```
python3 /path/to/genfire/gui/launch.py
```

Alternatively, on Mac/Linux you should have access to the shortcut `genfire` which does the same thing *(If anybody knows a reliable way to create a similar, installable shortcut command on Windows, I would love to hear about it)*

```
genfire
```

Troubleshooting pip

If an error occurs when using `pip3` to install the dependencies for GENFIRE, you may try manually installing that dependency with `pip` and then trying to install GENFIRE. For example, if you receive an error about PyQt5 not being installed, try manually installing it with `pip install PyQt5`. If this still fails, you can try instead installing with setup.py, as described below.

Installing with setup.py

If installing with `pip` fails, you can also try installing the package manually

Get the Source Code

To install `GENFIRE` as a python package, first download the source code. Then follow the procedure for your operating system.

Mac OS X

Python is preinstalled on Mac OS X, but it is generally a bad idea to alter the system python in /usr/bin as some programs depend on it. I would highly recommend using the package manager homebrew to create a python environment for you. The following commands will install homebrew and python3

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
$ brew install python3
```

Once that's done, navigate to the top source code directory. That's the same folder as requirements.txt and setup.py

```
$ cd /path/to/GENFIRE
```

Install (most of the) dependencies using pip3, you may need to prepend the command with sudo depending on your system

```
$ pip3 install --upgrade pip
$ pip3 install -r requirements.txt
```

You will also need sip and PyQt5

```
$ brew install sip
$ brew install pyqt
```

You should now have all the dependencies necessary to run `GENFIRE`. To install it, make sure you are in the directory with setup.py and enter the follow. Again, you may need sudo

```
$ python3 setup.py install
```

If everything worked, you can now launch the gui from the command line with

```
$ genfire
```

You can also use the code as any other python package. Here's an example of launching the gui from within python

```
$ python3
```

```
import GENFIRE
GENFIRE.gui.launch.main()
```

(Optional) Turbo-charge `GENFIRE` with `FFTW`

`GENFIRE` can make use of `pyFFTW`, which wraps the `FFTW` library. I have tested a number of FFT routines including those in `NumPy`, `SciPy`, and `pyFFTW`, and found this to be the fastest one by a factor of 2-3.

The following are details for installing `FFTW` from source, but recently pip has begun to support precomiled libraries for the package, so you should first try the easy way with

```
pip3 install pyfftw
```

if this fails, then proceed with building the `FFTW` libraries by hand and then reinvoking pip as described below

In order for `GENFIRE` to use `pyFFTW` you must install the `FFTW` libraries. Download the source code, decompress it, and navigate into the unpacked directory from your terminal. `pyFFTW` needs `FFTW` to be compiled for all precision types, so you have to compile it three times. Use the following commands

```
$ ./configure --enable-threads --enable-shared
$ make
$ sudo make install
```

```
$ ./configure --enable threads --enable-shared --enable-float
$ make
$ sudo make install
```

```
$ ./configure --enable-threads --enable-shared --enable-long-double
$ make
$ sudo make install
```

Now, install pyFFTW with pip and and test that it worked

```
$ pip3 install pyfftw
$ python -c "import pyfftw"
```

If you don't receive an error, then it was successful. If you get an error along the lines of "ImportError: libfftw3l.so cannot open shared object file" then you need to set your `DYLD_LIBRARY_PATH` environmental variable so that `pyFFTW` can find the libraries

```
$ export DYLD_LIBRARY_PATH=/usr/local/lib:$DYLD_LIBRARY_PATH<
```

If the `FFTW` .so files are somewhere other than /usr/local/lib, then you should replace that part appropriately. To make this change permanent add the above line to the end of your ~/.bash_profile

Linux (Ubuntu 14.04)

Navigate to the source code directory. That's the same folder as requirements.txt and setup.py

```
$ cd /path/to/GENFIRE
```

Install (most of the) dependencies using pip

```
$ pip3 install --upgrade pip
$ sudo pip3 install -r requirements.txt
```

You will also need sip and PyQt5 this command should install both.

```
$ sudo apt-get install python3-pyqt5
```

You should now have all the dependencies necessary to run GENFIRE. To install it, make sure you are in the directory with setup.py and enter

```
$ sudo python setup.py install
```

If everything worked, you can now launch the gui from the command line with

```
$ genfire
```

You can also use the code as any other python package. Here's an example of launching the gui from within python

```
$ python3
```

```
import GENFIRE<br>
GENFIRE.gui.launch.main()
```

(Optional) Turbo-charge `GENFIRE` with `FFTW`

`GENFIRE` can make use of `pyFFTW`, which wraps the `FFTW` library. I have tested a number of FFT routines including those in `NumPy`, `SciPy`, and `pyFFTW`, and found this to be the fastest one by a factor of 2-3.

The following are details for installing `FFTW` from source, but recently pip has begun to support precomiled libraries for the package, so you should first try the easy way with

```
pip3 install pyfftw
```

if this fails, then proceed with building the `FFTW` libraries by hand and then reinvoking pip as described below

In order for `GENFIRE` to use `pyFFTW` you must install the `FFTW` libraries. Download the source code, decompress it, and navigate into the unpacked directory from your terminal. `pyFFTW` needs `FFTW` to be compiled for all precision types, so you have to compile it three times. Use the following commands

```
$ ./configure --enable-threads --enable-shared
$ make
$ sudo make install
```

```
$ ./configure --enable-threads --enable-shared --enable-float
$ make
$ sudo make install
$ ./configure --enable threads --enable-shared --enable-long-double
$ make
$ sudo make install
```

If you don't receive an error, then it was successful. If you get an error along the lines of "ImportError: libfftw3l.so cannot open shared object file" then you need to set your `LD_LIBRARY_PATH` environmental variable so that `pyFFTW` can find the libraries

```
$ export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH<
```

If the `FFTW` .so files are somewhere other than /usr/local/lib, then you should replace that part appropriately. To make this change permanent add the above line to the end of your ~/.bashrc

Windows 10

For python3 on Windows 10 I would recommend using Anaconda from Continuum Analytics. It's a distribution of python that contains most of the packages used by `GENFIRE` all wrapped into a simple-to-use installer. *Note you must use python3.* Once you have python setup, open up a cmd prompt and navigate to the source directory. That's the same folder as requirements.txt and setup.py

```
$ C:\path\to\Anaconda\python3 setup.py install
```

If everything worked you can now launch the gui.

```
$ C:\path\to\Anaconda\python GENFIRE\gui\launch.py
```

You can also use the code as any other python package. Here's an example of launching the gui from within python

```
$ C:\path\to\Anaconda\python3
```

```
import GENFIRE
GENFIRE.gui.launch.main()
```

(Optional) Turbo-charge `GENFIRE` with `FFTW`

`GENFIRE` can make use of `pyFFTW`, which wraps the `FFTW` library. I have tested a number of FFT routines including those in `NumPy`, `SciPy`, and `pyFFTW`, and found this to be the fastest one by a factor of 2-3.

The following are details for installing FFTW from source, but recently pip has begun to support precomiled libraries for the package, so you should first try the easy way with

```
$ C:\path\to\Anaconda\Scripts\pip3 install pyfftw
```

if this fails, then consult the `FFTW` documentation

Installation Troubleshooting

If you have trouble installing `PyQt5` or `sip`, consult their documentation If you have some problem with the "pip3 install -r requirements.txt" step, you can view the requirements.txt file to see the packages that are necessary, and try to install them one-by-one.