

Fermi plug-in for TEMPO2 documentation

L. Guillemot

e-mail: guillemo at mpifr-bonn.mpg.de

v4.0, 24/09/2011

Introduction

The *Fermi* plug-in for TEMPO2 allows to calculate a pulsar rotational phase for each photon arrival time in a *Fermi Large Area Telescope* event file, using all functionalities provided by TEMPO2. The plug-in first calculates the position of the observatory for each photon time, using the same interpolation algorithm as implemented in the *Fermi* Science Tools (STs) `gtbary` and `gtpphase`¹. Knowing the satellite position as a function of time, it then converts the topocentric event dates to barycentric dates. It finally calculates pulse phases using the barycentric dates, converting for orbital motions if any.

The present document describes how to download, install and use the *Fermi* plug-in for TEMPO2. The following descriptions apply to Version 5.5 of the plug-in. Note that this documentation as well as the plug-in itself are in constant development. If you feel something needs improvement, please feel free to send your comments at `guillemo at mpifr-bonn.mpg.de`.

Documentation on TEMPO2 can be found in the official webpage:

<http://www.atnf.csiro.au/research/pulsar/tempo2/>

TEMPO2 and its timing model are described in detail in *Hobbs, Edwards & Manchester, MNRAS 369, 655–672 (2006)* and *Edwards, Hobbs & Manchester, MNRAS 372, 1549–1574 (2006)*. Please refer to these articles if you use TEMPO2 in a publication.

If you make use of the *Fermi* plug-in in your publications please refer to *Ray, et al., “Precise Gamma-Ray Timing and Radio Observations of 17 Fermi Gamma-Ray Pulsars”, ApJS 194, 17 (2011)*.

Downloading and installing the plug-in

Detailed TEMPO2 and *Fermi* plug-in installation instructions for Mac and Linux are provided at the end of this document (see **Appendix**). They describe how to download and install the latest version of the TEMPO2 package and the *Fermi* plug-in, as well as their dependencies.

¹See <http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/> for a description of *Fermi* Science Tools

Command-line arguments

The minimal command line is:

```
tempo2 -gr fermi -f PARFILE -ft1 FT1FILE -ft2 FT2FILE
```

In this command line:

- FT1FILE and FT2FILE are the photon and spacecraft files in fits format. **Make sure that the photon dates are covered by the spacecraft file!** Pulse phases of -1 are returned for photons not covered by the spacecraft file.
- PARFILE is the timing solution used for phase-folding the gamma-ray data. Detailed explanations about the content of TEMPO/TEMPO2 parameter files can be found in:

<http://www.atnf.csiro.au/research/pulsar/tempo2/index.php?n=Documentation.Parameters>.

Note: there are a few important differences between TEMPO and TEMPO2. One difference is for instance the assumed time system: TEMPO assumes “Barycentric Dynamical Time” (TDB) whereas TEMPO2 uses “Barycentric Coordinate Time” by default. Since the *Fermi* plug-in uses TEMPO2 functionalities, one needs to run TEMPO2 in “TEMPO emulation mode” in case a TEMPO timing solution is used. For that, edit the timing solution and add the following lines:

```
UNITS TDB
TIMEEPH FB90
DILATEFREQ N
PLANET_SHAPIRO N
T2CMETHOD TEMPO
CORRECT_TROPOSPHERE N
```

The following arguments are optional:

- “-h” displays help information.
- “-graph 0” disables the graphical output.
- “-output FILE” outputs barycentric event times and pulse phases in FILE.
- “-pos FILE” outputs (X,Y,Z) satellite positions in FILE.
- “-grdev FILE/DEVICE” sets the graphical output file to FILE, using the DEVICE pgplot graphical device. The available devices depend on the pgplot installation. Default device is X window.
- “-bins N” number of bins used in the phase histogram (if a graphical output is required). Default is 20.
- “-Hbins N” number of points in the *H*-test Test Significance (TS) vs time plot (if a graphical output is required). The *H*-test is described in *de Jager, Raubenheimer & Swanepoel, A&A 221, 180 (1989)*. Default is 20.
- “-phase” stores calculated pulse phases in the input FT1 file. By default they are not written.
- “-col COLNAME” stores calculated pulse phases in a COLNAME column in the input FT1 file (if the “-phase” option is used). Default column name is PULSE_PHASE.

- “-profile FILE” stores the binned light curve in FILE (graphical mode only).
- “-tmin T” event times smaller than MJD T will be skipped. If this option is used in conjunction with the “-phase” option, skipped events will get a phase of -1 if the phase column is created. In the phase column exists already, the phases of skipped events will be left unchanged.
- “-tmax T” event times larger than MJD T will be skipped. The behavior for skipped events is the same as for the “-tmin T” option.

Examples

As mentioned above, ft1 (photon data), ft2 (spacecraft data) and par (pulsar ephemeris) files are required to use the *Fermi* plug-in. The following command will calculate phases and output a plot of results in an XW window:

```
tempo2 -gr fermi -ft1 FT1FILE -ft2 FT2FILE -f PARFILE
```

An example of graphical output is shown in Figure 1. It has three panels: an histogram of the calculated phases, a plot of the *H-test* Test Significance as a function of time and a plot of time versus phase for each event. The latter two plots can be used to monitor the peak stability and ephemeris accuracy (if a signal is detected!).

Note that **by default, the calculated phases will not be stored in the input ft1 file**. To do this, the “-phase” option is needed:

```
tempo2 -gr fermi -ft1 FT1FILE -ft2 FT2FILE -f PARFILE -phase
```

This will create a PULSE_PHASE column in the ft1 file if there is not one already, or overwrite the PULSE_PHASE column content otherwise. However, phases can be stored in another column by using the “-col COLNAME” option. This will create a COLNAME column containing calculated phases, or overwrite it if existing already. This option allows to have multiple phase column in a single ft1 file.

The graphical output can be turned off with the “-graph 0” option:

```
tempo2 -gr fermi -ft1 FT1FILE -ft2 FT2FILE -f PARFILE [other options] -graph 0
```

Or, the graphical display can be directed to another device than XW. The available devices depend on the pgplot installation. For instance, assuming that the CPS graphical device (color landscape postscript printer) is configured, a postscript output GRAPH.ps can be created by typing:

```
tempo2 -gr fermi -ft1 FT1FILE -ft2 FT2FILE -f PARFILE [other options] -grdev GRAPH.ps/cps
```

Finally, one can output barycentric event dates and interpolated (X,Y,Z) satellite positions as a function of time with the following commands:

```
tempo2 -gr fermi -ft1 FT1FILE -ft2 FT2FILE -f PARFILE [other options] -output FILE
```

The output FILE, in ascii format, contains three columns: row number, barycentric time in MJD format, and calculated pulse phase for each event:

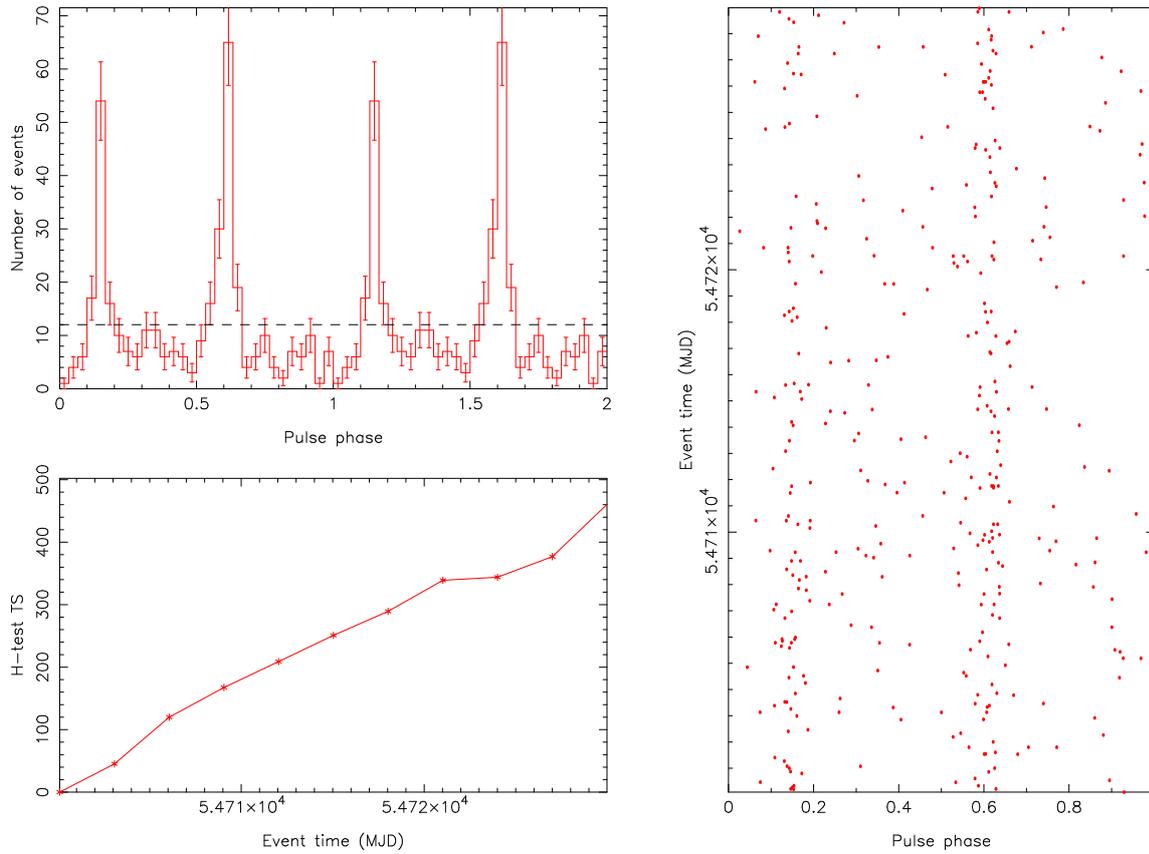


Figure 1: Example graphic output of the *Fermi* plug-in. *Top-left panel*: phase histogram of the analyzed gamma-ray data. The number of phase bins can be changed with the “-bins N” option. Two full rotations are shown for clarity. *Bottom-left panel*: *H*-test Test Significance (TS) as a function of time. *Right panel*: time versus pulse phase for each gamma-ray event.

```

1 54682.844241256756504 1.4444732666e-01
2 54686.166348854687634 5.4347991943e-01
3 54686.226616774200551 3.6297607422e-01
4 54689.550108410123467 3.0435180664e-01
5 54689.556175463143465 1.3642120361e-01
6 54691.006971309457938 6.4566040039e-01
7 54691.937456345703904 5.8476257324e-01
8 54691.940486247987661 1.4231872559e-01
9 54691.534279664343241 5.8385467529e-01
10 54691.807846674614492 1.7523956299e-01

```

```
tempo2 -gr fermi -ft1 FT1FILE -ft2 FT2FILE -f PARFILE [other options] -pos FILE
```

will produce an ascii file containing four columns: the first gives the topocentric (*i.e.*, before barycentric cor-

rections) event date in Mission Elapsed Time (MET) format², and the observatory (X,Y,Z) position at this date:

```
239573397.153821 6265542.553 1539570.532 2499252.255
239860404.492438 5489939.295 2974938.997 2980972.282
239865611.235304 6386484.459 -857632.783 2537514.316
240152739.121356 6043328.911 1670366.186 2937107.398
240153263.280684 4196876.206 5051772.729 2178861.295
240278602.844197 6207838.052 -750320.754 2984495.516
240358990.960060 6273190.798 -84211.554 2940328.621
240359252.724650 6109871.038 1870462.664 2673006.884
240324158.997595 5501439.744 -3127516.475 2828518.863
240347793.485891 6081151.109 1959930.260 2673812.452
```

Acknowledgements

Many thanks to Dave Thompson, Megan DeCesar and Paul Ray for providing the very detailed and useful installation instructions given in the appendix. Thanks to David Smith, Eric Grove, Dave Thompson and Elizabeth Ferrara for encouraging me to write this document! Thanks to Tom Glanzman for careful checks and useful discussions. Thanks to Anne Archibald for corrections and useful additions that helped improve the plug-in a lot. Finally, thanks to George Hobbs, Russell Edwards, and all TEMPO2 contributors for developing this powerful tool.

²See http://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone/Cicerone_Data/Time_in_ScienceTools.html for a description of the MET format.

Appendix A: downloading and installing TEMPO2 and the *Fermi* plug-in

This part applies to the latest versions of TEMPO2, for which an auto-configuration utility is provided with the package.

1. Get the TEMPO2 software:

```
cvs -z3 -d:pserver:anonymous@tempo2.cvs.sourceforge.net:/cvsroot/tempo2 co tempo2
```

2. Install the necessary support packages:

In order to install the *Fermi* plug-in, gfortran³, pgplot⁴ and cfitsio⁵ are needed.

On Linux machines, the packages can be set up automatically using package installers (synaptic, RPM, dpkg, ...) or manually.

On Mac machines, these packages can be installed using fink⁶, macports⁷, or also manually.

3. Compile and install the program:

In the TEMPO2 directory, run:

```
./bootstrap
```

Then set the TEMPO2 environment variable to point to the subdirectory named T2runtime. For instance:

```
setenv TEMPO2 PATH_TO_TEMPO2_DIR/T2runtime
```

You can now run the auto-configuration utility:

```
./configure
```

Depending on your installation, pgplot and cfitsio (which are needed to compile the plug-in) may not be found properly. You should get:

```
checking for PGPLOT installation... yes
checking for CFITSIO installation... yes
```

If not, make sure that the PGPLOT_DIR points to the directory where pgplot was installed, and that this directory is included in the LD_LIBRARY_PATH.

³<http://www.gfortran.org/download/>

⁴<http://www.astro.caltech.edu/~tjp/pgplot/>

⁵<http://heasarc.gsfc.nasa.gov/fitsio/>

⁶<http://www.finkproject.org>

⁷<http://www.macports.org/>

Concerning cfitsio, try to add the following options to the “configure” command:

```
./configure --with-cfitsio-dir=PATH_TO_CFITSIIO_DIR
--with-cfitsio-include-dir=PATH_TO_CFITSIIO_DIR/include
--with-cfitsio-lib-dir=PATH_TO_CFITSIIO_DIR/lib
```

When the “configure” script has run successfully, you can build TEMPO2 with the following commands:

```
make
make install
```

This should create an executable named “tempo2” in the TEMPO2 directory, that you may want to include in the PATH.

Note: in recent versions the cfitsio headers and libraries are sometimes not found when compiling the *Fermi* plug-in, even if cfitsio was successfully found by the “configure” script. If so, you can try re-running the above “configure” command, adding the following arguments:

```
CXXFLAGS=-IPATH_TO_CFITSIIO LDFLAGS=-LPATH_TO_CFITSIIO LIBS=-lcfitsio
```

4. Some quick first tests to check that the code is working:

```
tempo2 -h
tempo2 -gr fermi -h
tempo2 -gr <any other plug-in> -h
```

These commands will give a help menu. The first just gives help for TEMPO2 in general, while including “-gr <plug-in>” will give help for using a specific plug-in.

5. Further tests:

To test TEMPO2 further, download the .par and .tim files available at:

```
http://www.atnf.csiro.au/research/pulsar/tempo2/index.php?n=Main.Data
```

After downloading, go to the directory where they are stored and type:

```
tempo2 -f psr1.par psr1.tim
```

or:

```
tempo2 -f psr1.2.par psr1.tim
```

These will result in output to the screen. Try:

```
tempo2 -gr plk -f <psr1.par OR psr1.2.par> psr1.tim
```

to inspect the residuals with an interactive plot. NOTE: These tests are explained in more detail at:

<http://www.atnf.csiro.au/research/pulsar/tempo2/index.php?n=Documentation.Introduction>

6. Usage of the *Fermi* plug-in:

Finally, to use the *Fermi* plug-in to add phases to a pulsar data file, type:

```
tempo2 -gr fermi -ft1 <FT1 file> -ft2 <FT2 file> -f <.par file> -phase
```

To learn about other options available from this plug-in, get the help menu by typing:

```
tempo2 -gr fermi -h
```

Appendix B: older versions of TEMPO2

This part applies to older versions of TEMPO2, that did not come with the auto-configuration utility.

Installing TEMPO2 on a Mac machine

The following installation instructions were compiled by Dave Thompson, with help from Paul Ray, Dave Davis and David Friedlander.

1. Make sure needed support packages are on the system

gfortran and pgplot and cfitsio are all needed.

TEMPO2 requires several packages including a FORTRAN compiler (gfortran is the preferred one nowadays), CFITSIO, and PGPLOT. I install those using “fink” (<http://www.finkproject.org/>), but there are other ways to get/install them, depending on your preferences. If you don’t use fink or some other package that helps you port linux software to the Mac (e.g. darwinports, etc...), you certainly should!

2. Set the TEMPO2 environment variable

In the directory where TEMPO2 was installed, there is a subdirectory called T2runtime. Use setenv to set TEMPO2 to point to the T2runtime directory.

3. Edit the TEMPO2 files to incorporate the *Fermi* plug-in and accessories

Go to the tempo2/plugin subdirectory and edit Makefile:

Change:

```
plugin_SRCS += basic_plug.C delays_plug.C plk_plug.C splk_plug.C
```

To:

```
plugin_SRCS += basic_plug.C delays_plug.C plk_plug.C splk_plug.C fermi_plug.C
```

Back up to the TEMPO2 directory. Edit: Makefile.common:

Change:

```
CXX := MACOSX_DEPLOYMENT_TARGET=10.3 $(CXX)
```

To:

```
CXX := $(CXX)
```

In the same directory, Edit Makefile.config:

Change:

```
ICPGPLOT = -I$(PGPLOT_DIR)
```

To:

```
ICPGPLOT = -I$(PGPLOT_DIR) -I/sw/include
```

Change:

```
LCPGPLOT = -L$(PGPLOT_DIR) -lcpgplot -lpgplot -L/usr/X11R6/lib -lX11 -lpng -lg2c -pg
```

To:

```
LCPGPLOT = -L$(PGPLOT_DIR) -L/sw/lib -lcpgplot -lpgplot -L/usr/X11R6/lib -lX11 -Wl,  
-framework -Wl,Foundation -lpng -lz -laquaterm -pg
```

When linking, use `-L/usr/X11R6/lib -lX11 -Wl,-framework -Wl,Foundation -L/sw/lib -lpng -lz -laquaterm -L$PGPLOT_DIR/lib -lpgplot`

Change:

```
FC = g77
```

To:

```
FC = gfortran  
LF = -L/sw/lib/gcc4.4/lib -L/sw/lib -lgfortran
```

Near the top of this file, add these two lines:

```
CXXFLAGS += -I/sw/include  
LDFLAGS += -L/sw/lib lcfitsio
```

4. Compile and install the program

In the TEMPO2 directory,

```
make
```

After it compiles,

```
make install
```

5. Test the radio portion of the program

The TEMPO2 documentation has directions for downloading and testing a sample of radio data.

`http://www.atnf.csiro.au/research/pulsar/tempo2/index.php?n=Documentation.Introduction`

6. Should be ready for use with *Fermi* LAT data.

You need to have the binary TEMPO2 program in your path, or else explicitly give it the path to the executable. For example if you are in the TEMPO2 directory, type “./tempo2”. Making sure the *Fermi* plug-in is installed:

```
tempo2 -h
```

should show a list of plugins, including *Fermi*. If something goes wrong (like the typos that I made), then after doing the editing, use “make clean” before “make” and “make install”.

7. Check your system

After doing the install, I discovered that the links to the *Fermi* Science Tools on my machine were gone. This could have been operator error, but it might have been associated with the TEMPO2 install. To recover the links to the *Fermi* STs, type:

```
setenv FERMI_DIR <path to the science tools>
```

and then

```
source $FERMI_DIR/fermi-init.csh
```

On a Linux machine

The following installation instructions were compiled by Megan DeCesar, with help from Paul Ray.

1. Add the following to your `.cshrc` in your home directory:

```
setenv PGPLOT_DIR <directory containing pgplot>
setenv LD_LIBRARY_PATH ${PGPLOT_DIR}
```

To find the directory containing `pgplot`, type `locate pgplot`. If you already have a `LD_LIBRARY_PATH` variable defined, just add `${PGPLOT_DIR}` to it. For example, if you currently have the following in your `.cshrc`:

```
setenv LD_LIBRARY_PATH /Software/lib
```

then change it to:

```
setenv LD_LIBRARY_PATH /Software/lib:${PGPLOT_DIR}
```

2. Assuming `TEMPO2` was installed in a directory named `/T2DIR`, add the following line in `.cshrc`:

```
setenv TEMPO2 /T2DIR/tempo2/T2runtime
```

This sets the `TEMPO2` environment variable, which the configuration scripts need in order to compile the code.

3. To update your environment, type:

```
source ~/.cshrc
```

Check that the new variables have been set by typing:

```
echo $PGPLOT_DIR
echo $LD_LIBRARY_PATH
echo $TEMPO2
```

Each “echo” command should result in the display of the directories to which you’ve assigned the variables in your `.cshrc` file.

4. Change directories (`cd`) to `/T2DIR/tempo2/plugin`. Edit `Makefile` as follows:

Change:

```
plugin_SRCS += basic_plug.C delays_plug.C plk_plug.C splk_plug.C
```

to:

```
plugin_SRCS += basic_plug.C delays_plug.C plk_plug.C splk_plug.C fermi_plug.C
```

Change:

```
CXXFLAGS += $(ICPGPLOT)
LDFLAGS += $(LCPGPLOT)
```

to:

```
CXXFLAGS += $(ICPGPLOT) -I/usr/local/include/
LDFLAGS += $(LCPGPLOT) -L/usr/local/lib/ -lcfitsio
```

5. Go back one directory, to /T2DIR/tempo2. Edit Makefile.config:

Change:

```
LCPGPLOT = -L$(PGPLOT_DIR) -lcpgplot -lpgplot -L/usr/X11R6/lib -lX11 -lpng -lg2c -pg
```

to:

```
LCPGPLOT = -L$(PGPLOT_DIR) -L<directory containing libgcc.a> -lcpgplot -lpgplot
-L/usr/X11R6/lib -lX11 -lpng -lg2c -pg
```

For example, in my Makefile.config, I have:

```
LCPGPLOT = -L$(PGPLOT_DIR) -L/usr/lib/gcc/i386-redhat-linux/3.4.6/ -lcpgplot
-lpgplot -L/usr/X11R6/lib -lX11 -lpng -lg2c -pg
```

6. This step may or may not be needed; I needed to do it. I would advise skipping it at first, and see if the code will compile correctly as in step 8. If you get an error that says “fitsio.h: No such file or directory”, then come back and do step 7, redo step 8, and continue on.

Go into /T2DIR/tempo2/plugin and edit fermi_plug.C:

Change:

```
#include <fitsio.h>
```

to:

```
#include <<directory containing fitsio.h>/fitsio.h>
```

7. In the same directory (/T2DIR/tempo2), type:

```
make
make install
```

This should compile the code. After the first command, you’ll see a bunch of files ending in “.o” and “.h” that weren’t there before, you’ll see a “tempo2” executable file, and in the /plugin directory there

will be files named the same as the .C plugins but ending in “.so”. After the second command, the T2runtime/plugins directory will be populated by all the plug-in codes ending in “.so”.

8. Open your .cshrc again, and add the line:

```
alias tempo2 /T2DIR/tempo2/tempo2
```

Then type:

```
source ~/.cshrc
```

Now, if you type tempo2 from any directory, the tempo2 executable located in /T2DIR/tempo2 will be run.

9. Some quick first tests to check that the code is working:

```
tempo2 -h
tempo2 -gr fermi -h
tempo2 -gr <any other plug-in> -h
```

These commands will give a help menu. The first just gives help for TEMPO2 in general, while including “-gr <plug-in>” will give help for using a specific plug-in.

10. To test TEMPO2 further, download the .par and .tim files available at:

```
http://www.atnf.csiro.au/research/pulsar/tempo2/index.php?n=Main.Data
```

After downloading, go to the directory where they are stored and type:

```
tempo2 -f psr1.par psr1.tim
```

or:

```
tempo2 -f psr1.2.par psr1.tim
```

These will result in output to the screen. Try:

```
tempo2 -gr plk -f <psr1.par OR psr1.2.par> psr1.tim
```

to inspect the residuals with an interactive plot. NOTE: These tests are explained in more detail at:

```
http://www.atnf.csiro.au/research/pulsar/tempo2/index.php?n=Documentation.Introduction
```

11. Finally, to use the *Fermi* plug-in to add phases to a pulsar data file, type:

```
tempo2 -gr fermi -ft1 <FT1 file> -ft2 <FT2 file> -f <.par file> -phase
```

To learn about other options available from this plug-in, get the help menu by typing:

```
tempo2 -gr fermi -h
```

12. Note that to use a plug-in, you use “-gr” followed by the plug-in name; -h will give help for whatever plug-in, if you decide you want to use others in your work.
13. Additional notes: This guide was written assuming you have access to all the required software, and that is likely the case if you are logging into a university or government machine to do your work. It may not be the case if you are using your own Linux machine. So, there are a few packages you need in order to use TEMPO2 in the first place. These are PGPLOT, CFITSIO, and a Fortran compiler (in this case, one that uses g77, but you may need to get gfortran instead; see below).

PGPLOT is available from here:

```
http://www.astro.caltech.edu/~tjp/pgplot/
```

CFITSIO is available here:

```
http://heasarc.nasa.gov/docs/software/fitsio/fitsio.html
```

You probably have a Fortran compiler. If not, you may be able to download it here:

```
http://gcc.gnu.org/fortran/
```

(A google search may yield more download options, but this was the most promising one that I found.) I think that if you use this, you'll need to edit /T2DIR/tempo2/Makefile by changing FC = g77 to FC = gfortran, and you may need to change the LF line as well; an example is in Dave Thompson's notes, but I don't know if changing LF is specific to the Mac or not. This will be updated if/when I try using the GNU version of Fortran.