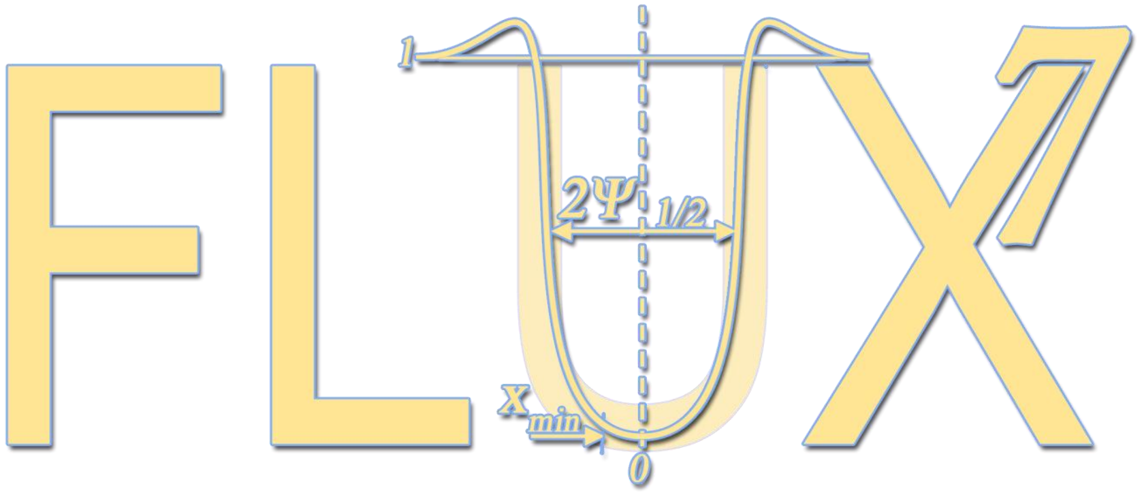

# Installation guide

## for Ubunto 10.10 (32-bit)

Norberto Catarino
27-03-2011

# Contents

# INSTALLATION

How to install the FLUX7 package on a Unix or Linux system.

## Fortran compiler

FLUX7 is written in Fortran77. Some of the auxiliary programs use modules written in C. Various Fortran compilers have been used in the past to compile the package. Notably the combination gcc - g77 has been found to work well and produce efficient code. This is also the compiler that was used to maintain and update Flux since around the year 2000.

However in recent versions of gcc the g77 compiler has been replaced with gfortran. This is a Fortran95 compiler, and although most Fortran77 statements also work in gfortran, some of the older constructs are obsoleted. As a result gfortran (and probably any other Fortran95 compiler) is not suitable to compile Flux7. It would be possible to replace the obsolete contructs with Fortran95 features, but is was found that the code so produced with gfortran was slower then the g77 version. Moreover the combination of Fortran and C input/output (used in the auxiliary programs) is handled much better in g77. Therefore for the time being we continue to only distribute the Fortran77 version.

Unfortunately g77 is no longer present in some recent Linux distributions, but it is still available. The next section explains how to install the g77 compiler.

## gcc-g77

Skip this section if you already have a suitable compiler available.

First check http://gcc.gnu.org/ for the current status of the GNU compiler direction. Then navigate via ``older news" to http://gcc.gnu.org/gcc-3.4/ . This page contains information about the last version (currently 3.4.6) containing the g77 compiler. Click on ``our mirror sites" go to one of those servers, and download the appropriate tar file. (e.g. http://gcc.igor.onlinedirect.bg/releases/gcc-3.4.6/gcc-3.4.6.tar.bz2 )

```
   # go to your home directory
cd
   # download the tarfile
wget web http://gcc.igor.onlinedirect.bg/releases/gcc-3.4.6/gcc-3.4.6.tar.bz2
   # extract the tarfile
tar jxvf gcc-3.4.6.tar.bz2
   # edit collect2.c
gedit gcc-3.4.6/gcc/collect2.c
   # replace (line ~ 1537):
    “redir_handle = open (redir, O_WRONLY | O_TRUNC | O_CREAT);”
   # by:
    “redir_handle = open (redir, O_WRONLY | O_TRUNC | O_CREAT, S_IRUSR | S_IWUSR);”
   # puts compilers under names gcc-3.4 and g77-3.4 in /usr/local/bin
   # (plus a lot of other stuff).
   # $HOME represente your home directory
mkdir gcc-3.4.6.objdir
cd gcc-3.4.6.objdir
$HOME/gcc-3.4.6/configure --program-suffix=-3.4 --disable-shared --enable-languages=c,c++,f77
   # (this will take a while, half an hour or so)
make
sudo make install
```

```
# ------------------------------ Optional part ----------------------------------

   #   It is not really necessary that g77 and gcc are the same version
   #   Create links to files gcc-3.4 and g77-3.4
 sudo rm /usr/bin/gcc
 sudo ln -T /usr/local/bin/gcc-3.4 /usr/bin/gcc
 sudo rm /usr/bin/g77
 sudo ln -T /usr/local/bin/g77-3.4 /usr/bin/g77

# --------------------------- End of optional part -------------------------------
```

Use those names in the Flux7 makedefs.file. Also make sure to compile PGPLOT (see next) with the same Fortran compiler as the Flux package.

# PGPLOT

The following assumes you have installed the PGPLOT graphics package, although some programs, flux7 in particular, will work without it.

Skip this section if you already have a PGPLOT installed.

Make sure the devel package for X11 library is installed on your computer.

For instance, on a Ubunto distribution, check in the synaptics package manager, the **libx11-dev** package must be installed;

```
   # login as root to get clearance to operate in directories
 sudo bash
   # go to your distribution directory directory
 cd /usr/local/src/
   # download the tarfile
 wget web ftp://ftp.astro.caltech.edu/pub/pgplot/pgplot5.2.tar.gz
   # extract the tarfile
 gunzip -c pgplot5.2.tar.gz | tar xvof -
   # Create the target directory
 mkdir /usr/local/pgplot
   # Select device drivers
 cd /usr/local/pgplot
 cp /usr/local/src/pgplot/drivers.list .
   # edit device drivers, include only the drivers you plan to use
 gedit drivers.list &
```

Flux7 assumes that you have selected at least the X11 and PostScript drivers. Select (by removing the ! sign) the drivers to be used for pgplot outputs. I would suggest to uncomment the following in file drivers.list:

```
   NUDRIV 0 /NULL     Null device (no output)                    Std F77
   PSDRIV 1 /PS       PostScript printers, monochrome, landscape Std F77
   PSDRIV 2 /VPS      Postscript printers, monochrome, portrait  Std F77
   PSDRIV 3 /CPS      PostScript printers, color, landscape      Std F77
   PSDRIV 4 /VCPS     PostScript printers, color, portrait       Std F77
   TTDRIV 5 /XTERM    XTERM Tektronix terminal emulator          Std F77
   TTDRIV 8 /KRM3     Kermit 3 on IBM-PC                         Std F77
   XWDRIV 1 /XWINDOW  Workstations running X Window System       C
   XWDRIV 2 /XSERVE   Persistent window on X Window System       C
```

```
# ----------------------------- Optional part ----------------------------------

  # to increase the number of simultaneous devices from 8 to 32
cd /usr/local/src/pgplot/src
  # keep a copy of the original source file
cp grpckg1.inc grpckg1.inc_backup
cp pgplot.inc pgplot.inc_backup

  # this file needs to be edited
gedit grpckg1.inc &
                                  # Replace ‟ PARAMETER (GRIMAX = 8) ‟ in line 29
                                  #    by   ‟ PARAMETER (GRIMAX = 32) ‟

  # this file needs to be edited
gedit pgplot.inc &
                                  # Replace ‟ PARAMETER (PGMAXD=8) ‟ in line 7
                                  #    by   ‟ PARAMETER (PGMAXD=32) ‟
cd /usr/local/pgplot
# -------------------------- End of optional part -------------------------------

  #  prepares the makefile for linux system + gcc compiler
/usr/local/src/pgplot/makemake /usr/local/src/pgplot linux g77_gcc

  # edit makefile and replace de X11 directory library (find and replace)
gedit makefile &
                                  # Replace ‟ X11R6/include ‟
                                  #    by   ‟ include ‟

  # compiles the Fortran part
make
  # compiles the C part (binding to Fortran)
make cpg

  # this file needs to be edited
gedit /usr/local/src/pgplot/makehtml &
                                  # replace the first line by
                                  # ‟ #!/usr/bin/perl ‟

  # produces the html documentation
make pgplot.html

  # Remove the object files and other junk by typing
make clean

  # libraries for local installations
cd /usr/local/lib
  # f77 static library
ln -s /usr/local/pgplot/libpgplot.a libpgplot.a
  # c static library
ln -s /usr/local/pgplot/libcpgplot.a libcpgplot.a
  # dynamic (runtine) library
cp /usr/local/pgplot/libpgplot.so .

  # edit this file ld.so.conf
gedit /etc/ld.so.conf &
                                  # add the line:
                                  # ”include /usr/local/lib ‟

  # updates the path for runtime libraries
/sbin/ldconfig -v
```

```
ln -s /usr/local/pgplot/cpgplot.h /usr/local/include/cpgplot.h

  # You have to define a few environment variables (should typed in .bashrc file):
gedit &HOME/.bashrc
                                   # add the lines:
  PGPLOT_DIR=/usr/local/pgplot/
  export PGPLOT_DIR
  PGPLOT_DEV=/xwin
  export PGPLOT_DEV
  PGPLOT_BACKGROUND=white
  PGPLOT_FOREGROUND=black
  PGPLOT_XW_WIDTH=0.6
  # Restart the Terminal to apply the new environment variables
  # Some tests provided may be run:
cd /usr/local/pgplot
./pgdemo1
```

# FLUX7

First check http://members.home.nl/p.j.m.smulders/FLUX/HTML/ for the current status of the FLUX program. Then navigate via `` the download page’’ to http://members.home.nl/p.j.m.smulders/FLUXBIN/ . This page contains the last version of FLUX (currently 7.8.9).

Downloading and Unpacking the files

```
cd
  # download the tarfile from the repository
wget http://members.home.nl/p.j.m.smulders/FLUXBIN/flux7.8.9.tar.bz2
  # extract the tarfile
tar jxvf ./flux7.8.9.tar.bz2
  # This should produce directory tree FLUX7.
Cd flux7.8.9/FLUX7/
```

# Make FLUX7 your current directory.

In the following this directory is indicated as ‘ ./ ‘

# Make Makefiles

```
  # edit the Makefiles in this directory, and in the various subdirectories, type:
./makemake
  # or
./makemake makedefs.linux
```

If you don’t like the result, edit the file makedefs.xxx for your installation, rather then the individual Makefiles.

Things that might need to be adapted are BINDIR, and the system dependent parameters FFLAGS, CFLAGS, etc. The option –Df77_UNDERSCORE_AFTER in CFLAGS should be set if your Fortran appends an underscore to subroutine names. The makedefs file also selects a few system dependent subroutines.

```
 'makedefs' files are included for the following systems:
  makedefs.hp
  makedefs.linux
  makedefs.sgi
  makedefs.sun
  makedefs.gcc (HP-UX Fortran in combination with gcc)
  makedefs.OSF1
  makedefs.mingw
```

If you make one for another system, or have suggestions for improvements to the above, please send me a copy.

## Attention PC users (Linux as well as MSWindows)

It may be necessary to manually change the makedefs file. Check the value of **HOSTTYPE** in makedefs.linux and makedefs.linux.chk, or makedefs.mingw. If you are not sure about the cpu-type delete the '-march=' option altogether. Users may profit from the –march=*cpu-type* commandline option of gcc to somewhat speed up calculations.

## Environmental variable FLUX

Define an environment variable with name FLUX and value the full pathname to directory FLUX7.

```
  # You have to define a few environment variables
  # (should typed in one  of the startup files: .profile, .bash or .bashrc file)
gedit &HOME/.bashrc
                                  # add the lines:
  FLUX=/home/peter/flux7.8.9/FLUX7
  export FLUX
  PATH=$FLUX/BIN:$PATH

# Restart the Terminal to apply the new environment variables
```

## make

```
cd /flux7.8.9/FLUX7/
   # type:
make
   # This will leave the compiled programs in various directories
   # and also in directory ./BIN.
```

## Tests

```
  # Some tests provided may be run by typing:

ini -t fluxtest

cd $FLUX/TEST
make test

cd $FLUX/MISC
make test

cd $FLUX/RBSIM
make test
```

```
cd $FLUX/YIMP
make test

cd $FLUX/FITYIM
make test

cd $FLUX/INPUT
./fluxvelo.bat                        # (takes a while, have a coffee break)

cd $FLUX/PLOTXY
make test
```

Observe the output for error and warning messages!

Examples of input files may also be found in subdirectory *INPUT*. Various directories have subdirectories named *TESTOUTPUT*. These give an idea of what to expect when running the test programs.

## Clean-up

```
cd /flux7.8.8/FLUX7/
   # Remove the object files and other junk by typing
make clean
```

# TUNING PGPLOT.

Although PGPLOT is well documented, some details are hard to find.

## PGPLOT environmental variables, general

```
 PGPLOT_DIR     directory where to find pgplot stuff (/usr/local/pgplot/)
 PGPLOT_DEV     default device driver (/xwin)
 PGPLOT_BACKGROUND=white     (default is the other way around!)
 PGPLOT_FOREGROUND=black
 PGPLOT_FONT     specify font file, override $PGPLOT_DIR/grfont.dat
 PGPLOT_RGB     specify color-name database, override $PGPLOT_DIR/rgb.txt
```

## Environment variables for the Xwindows driver

```
 PGPLOT_XW_WIDTH (value between 0 and 2)
      may be used to scale the size of the window. Only takes effect
      after restarting the server (icon 'pgxwin').
      A good starting value is something around 0.6.
      This variable has no effect if the pgxwin.Win.geometry is defined !?!
      (See below)
```

# Environment variables for the PostScript driver:

```
PGPLOT_PS_WIDTH    default  7800
PGPLOT_PS_HEIGHT   default 10500
PGPLOT_PS_HOFFSET  default   350
PGPLOT_PS_VOFFSET  default   250
PGPLOT_PS_BBOX
     If this variable has value MAX,  PGPLOT puts standard (full-page)
     bounding-box information in the header of the PostScript file. If
     the variable is unset or has some other value, PGPLOT puts the
     correct (smallest) bounding box information in the trailer of the
     PostScript file.
     (Says the doc's. It does not work for me PJMS)
PGPLOT_PS_MARKERS
     Specify "NO" to suppress use of a PostScript font for the graph
     markers; markers are then emulated by line-drawing.
PGPLOT_PS_COLOR     if set, use color mode
PGPLOT_IDENT        if set write user name etc on each page
PGPLOT_PS_DRAW_BBOX If set, draw bounding box
```

# X resources (may be put in your file ~/.Xdefaults):

```
pgxwin.Win.geometry: WIDTHxHEIGHT+X+Y
pgxwin.Win.iconGeometry:  +X+Y
pgxwin.Win.iconize: If true, iconize inactive windows if persistent
pgxwin.Win.acceptQuit: True if WM_DELETE_WINDOW events are to be obeyed
pgxwin.Win.minColors: Min number of colors per colormap
pgxwin.Win.maxColors: Max number of colors per colormap
pgxwin.Win.visual: default|monochrome|pseudocolor|directcolor|
                  staticcolor|truecolor|grayscale|staticgray
pgxwin.Win.crosshair: If true show crosshair cursor
pgxwin.server.iconGeometry:  +X+Y
pgxwin.server.visible: True|False
pgxwin.server.display: display_name
```

```
# ---------------------------------- Unix Command's ---------------------------------

  #copies a file
cp filename1 filename2

  #create links (link) to files
ln file_name link_name

  #lists your files
ls

  #removes a file
rm filename

  #online manual for program
man progamname

  #sudo allows a permitted user to execute a command as the superuser
sudo command
sudo -i
sudo bash
sudo -s
whoami
sudo passwd root
su root

  #Kill PROCESS (however some processes may resist).
kill

  #Clear shell window.
clear


# ------------------------------ INTERACTIVE FEATURES ------------------------------

TAB                          #Command completion  !!! USEFUL !!!
UPARROW                      #Command history     !!! USEFUL !!!

CTRL-C                       #Interrupt current process.
CTRL-D                       #If a program/command waits for input, end input.
```