# 2011 Sagan Summer Workshop

## Software Installation Manual for Hands-on Sessions

**Required:**
Each participant is required to adopt one of the following two ways to remotely display graphical programs that are running on the server.

1. The preferred way is to install an X11 server and display/run the programs through SSH Tunneling.

*Linux:*
you are good to go, everything works! You will be able to ssh to the server (with graphical X11 tunneling. eg.: *ssh -Y username@server*) and any graphical program you start on the server will be opened on your own screen.

Mac:
there are plenty of tutorials on how to install X11 on the Mac. For example:
http://www.janerigby.net/JRR/osx.html#essentials
The ssh client allows you to set-up SSH Tunneling to the server and run agraphical program: *ssh -Y username@server*. If you do not have *openssh* already installed, you should get it.

Windows:
*Xming* is the X sever for Windows. If you install it and run, it will run in the background (see the system tray icon with X) and is able to display any X11 program that is run via the network (through SSH tunneling). The best way to setup the SSH tunnel is to use the PuTTY program Xming which can be found here: http://www.straightrunning.com/XmingNotes/

PuTTY can be downloaded from:
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Another very useful tool is SCP, this allows you to copy files between your computer and the server:
WinSCP (http://winscp.net/eng/download.php)

2. Another method (as a backup) to display the graphical programs is to use a remote desktop through the VNC protocol.

Linux:

there are quite a few VNC clients for Linux, for example you can try: vncviewer. This should already be in your distribution's repository, or already installed. The command:

```
>  which vncviewer
```

will tell you if it is installed. If you are using Fedora yum you can use yum to install VNC. Just type (as root):

```
> yum search vncviewer
```

This command will tell you in which package the vncviewer is located. You can install the package with yum like this:

```
>  yum install tigervnc
```

If you are using Ubuntu, use program apt-get, which is similar to yum. You have to prep every run of apt-get with sudo, like this:

```
> sudo apt-get install tigervnc
```

Later, during the workshop, to run vncviewer you will have to know the name of the computer which you have to run vncserver on, and you will have to know which display your vncserver occupies - we will tell you when you run it. Then just type:

```
> vncviewer  hostname.caltech.edu:20
```

where hostname.caltech.edu is the name of the computer (change this accordingly), and 20 is the number of the display your server took when you started it (change this as well).

Mac:

The program called Chicken of the VNC works well for a Mac and can be downloaded here: http://sourceforge.net/projects/cotvnc/ Download and install the package.

To connect to a server using VNC you will have to give it the name of the server and a display number on which your vncserver is running. During the workshop you will ssh to the server (*ssh username@server*), start *vncserver* on it and remember the display number. Than you have to give this information to your Chicken of the VNC client.

Windows:

TightVNC (free, all Windows): http://www.tightvnc.com/
RealVNC (for Windows XP only, for other Windows versions is commercial): http://www.realvnc.com/products/download.html
Download one of the packages and install. To connect to a server using VNC you will have to give it the name of the server and a display number on which your vncserver is running. During the workshop you will ssh to the server (for example using PuTTy), start *vncserver* on it and remember the display number. Then, you have to give this information to your VNC client.

## Basic Linux Commands:

The modeling software will be operating on a Linux server. It will be very helpful for the participants to get familiar with some basic Linux commands (e.g., *ls, cd, cp, rm, mkdir, pwd*) and be able to edit text files with a Linux text editor (e.g., *nano*, *vi, emacs*). There are plenty of Linux tutorials on the web, for example:

Basic Linux Commands from Google University:
http://code.google.com/edu/tools101/linux/basics.html

## Optional software:

### 1. Fortran compiler

The light curve modeling programs are in Fortran 90. They will be run on the server remotely through SSH. However, if a participant wants to have the fitting program on his/her own machine, it is important to have a Fortran 90 compiler. The one that is free and available for all platforms is: *gfortran*.

### 2. Plotting program

A plotting program such as gnuplot or SuperMongo will be useful to plot the results of the modeling. Choose your favorite.
If you do not have any plotting program installed, we will provide the SuperMongo script to be run on the server, which will provide the light curve plots in the Postscript format.

### 3. Interactive images and light curve generation tools.

The interactive tool (written in Python) will be run on the server through SSH. However if participants would like to run their own copy of the program and have a chance to use it after the Workshop or in-between the sessions, some additional packages are required:

*Linux:*
To have Python graphical programs working on your personal computer these packages have to be installed:

- *python* (python interpreter, sometime executables can be called python2.7 or similar)

- *numpy* (package for python, can be called python-numpy as well, eg. on Ubuntu)
- *matplotlib* (plotting library for python)
- *pygtk2* (or pygtk, this should contain modules gtk and gobject for python)
- *f2py* (a program to compile fortran modules for python, it is possible that this requires python-dev or python-devel package as well)
- *gfortran* (Fortran compiler)
- *scipy* (science package for python)

To install packages in Fedora use yum command or Add/Remove Software from the Menu.
To search, from the command line, which packages are available do *yum search*, eg. *yum search numpy* or *yum search gfortran*, to learn more about package use *yum info*, eg. *yum info matplotlib.*

On Ubuntu use *apt-get* program. This works very similar to yum on Fedora, see *apt-get –help* or *man apt-get*. You can also use *Synaptics*, a graphical program to install and search for packages.

*Mac:*
On a Mac you can install the same packages as on Linux (although the names can be slightly different, thus one has to search). One of the free software repositories for Mac are *fink* and *MacPorts*. If you can install *fink* you then can use it similarly to *yum* on Fedora or *apt-get* on Ubutu to install programs.

Example of the packages' names mentioned above in fink would be for example: numpy-py26 , matplotlib-basemap-py26, matplotlib-py26, pygobject2-py26, pygtk2-gtk-py26, etc. (if you decide to have python 2.6 installed, you can try py27 packages as well). gfortran should be in package gcc45.

*Windows:*
see section at the end of this document

## 4. IDL interactive light curve generation tool

A similar interactive tool is also provided in IDL. If participant has IDL and fortran compiler installed on his machine, he will be able to run this tool as well. The only requirement is to have IDL connected to a licensed server. The name of the server is required and the port to connect to.

## Python Installation Tutorial on Windows

In order to run the microlensing GUI programs, you need to have Python
and several additional packages installed on your machine.
We are using Python v2.7

The necessary Python packages are:

- scipy
- numpy
- matplotlib
- pygtk2 (or pygtk)
- gtk
- gobject

You also need

- f2py
- gfortran

If you have all of these, and they work together, good for you. If not, this tutorial will go through the installation process for Windows step-by-step.

1. Go to http://www.enthought.com/products/getepd.php to get the basic Python installation.

Don't be intimidated by the listed prices. They make special dispensation for us poor academics. At the very bottom of the page, there is a section for "Trial and Academic versions". Click on the "Academic" link.

Select the Windows icon and fill out the requested information.

Click "Download".

Open and Run the installer.

I chose all the default options for the installer.

You know have Python 2.7 installed.

You also have the packages:

- numpy

- scipy

2. Get the other Python packages from
http://ftp.gnome.org/pub/GNOME/binaries/win32/pygtk/2.24/

Download and run "pygtk-all-in-one-2.24.0.win32-py2.7.msi"

Note these are version specific. Make sure you get the one for Python 2.7.

I again chose all the default options for the installer. It automatically put these files in the same place as the Python installation from Step 1.

You now have the packages:

- pygtk
- gtk
- gobject

3. Get gfortran from http://gcc.gnu.org/wiki/GFortranBinaries#Windows

Under "MinGW build", download and run the installer (3rd bullet point)

It is recommended to put this in C:\ . There can be problems if you put it in a directory with spaces in the name, e.g. "Program Files".

4. Get f2py from http://cens.ioc.ee/projects/f2py2e/#download

Download and run "2.x/F2PY-2-latest.win32.exe" (3rd link).

Again, I chose all the default options in the installer and it put these files in with the original Python installation.

This webpage http://www.scipy.org/F2PY_Windows has some other comments on f2py including the tip about not installing gfortran in "Program Files".

5. Download the matplotlib package

Go to http://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.0.1/

Download file matplotlib-1.0.1.win32-py2.7.exe (to match the installed Python version).
The installer should locate the Python directories on its own.

------------

You should now have all the necessary programs.

Download the ulens GUI and unpack it.

Open a Windows command line.

cd to the directory ulens-common-files/

execute the compile script. If using Windows, this will be: compile.bat.

Note: if compile.bat fails, open the file in a text editor. The Linux/Windows conversion may have introduced some unidentified characters instead of carriage returns.

This script will take a while to run; f2py should by compiling the required modules.

When it is done, go up one directory and try running the programs.

For example, enter into the command line

```
> python ulens-binary-lc.py
```