

# AggieSTAAR

## Python

## Bootcamp

Tutorial 5:  
Computing Resources and  
Remote Access



TEXAS A&M  
UNIVERSITY®



# Computing Resources in Astronomy



Astronomers sometimes deal with large data products or complicated models. Computing resources can be essential to do your research efficiently.

Thankfully, TAMU has High Performance Computing Resources (HPRC) available upon request.

# Remote Access

Depending on your advisor, you may need to access:

1. `Grace` or `FASTER` (HPRC)
2. `Saire` (Justin Spilker's personal machine)

which will require establishing a remote connection from your computer. There are two ways to do so:

1. Graphic User Interfaces (`GUIs`)
2. `SSH` (Secure SHell) connections through `terminal`

# GUIs – Graphic User Interfaces

GUIs are user-friendly ways to execute commands

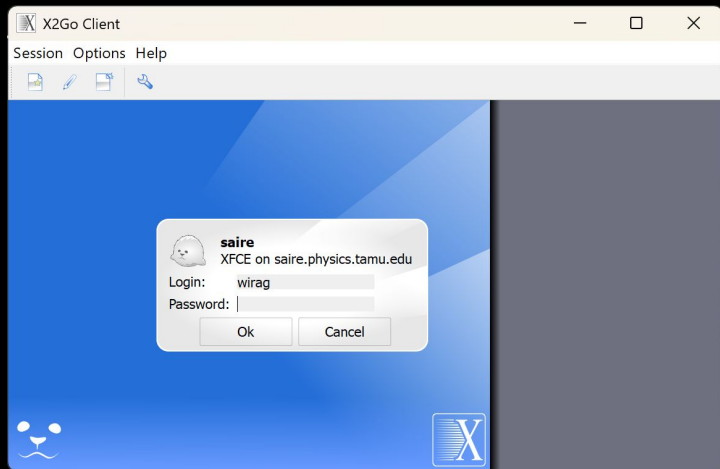
- File explorer (Windows) and Finder (Mac) are examples of these

If you're working with Justin Spilker, you will need to install x2go

- X2go opens a window on your computer that is actually running on **Saire**

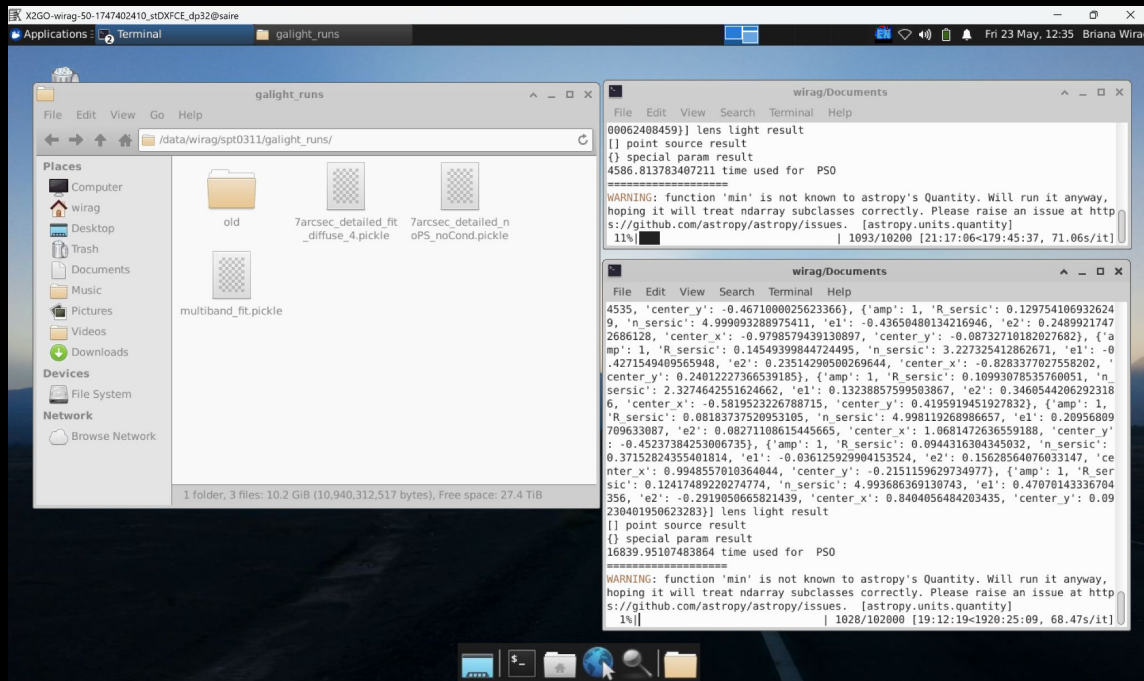


# GUIs – Graphic User Interfaces



<- Launch from here in x2go

The remote window ->



# SSH – Terminal

Another way to access **Saire**, and how you would connect to **Grace**, is through the **terminal** (Mac) or **powershell/command line** (Windows). You will initiate an **SSH** connection by typing the following:

- **Saire**: `<username>@saire.physics.tamu.edu`
- **Grace**: `<username>@grace.tamu.edu`

You will be prompted to enter your TAMU password (then DUO for **Grace**)

- **NOTE**: your password will NOT show up as you type, just trust the process

# SSH – Terminal

```
(base) PS C:\Users\briwi> ssh wirag@saire.physics.tamu.edu
```

Texas A&M University

This computer system and the data herein are available only for authorized purposes by authorized users. Use for any other purpose is prohibited and may result in disciplinary actions or criminal prosecution against the user. Usage may be subject to security testing and monitoring. There is no expectation of privacy on this system except as otherwise provided by applicable privacy laws. Refer to University SAP 29.01.03.M0.02 Acceptable Use for more information.

wirag@saire.physics.tamu.edu's password:

## Saire

```
(bhenv) PS C:\Users\briwi> ssh wirag@grace.tamu.edu
```

```
*****  
This computer system and the data herein are available only for authorized  
purposes by authorized users. Use for any other purpose is prohibited and may  
result in disciplinary actions or criminal prosecution against the user. Usage  
may be subject to security testing and monitoring. There is no expectation of  
privacy on this system except as otherwise provided by applicable privacy laws.  
Refer to University SAP 29.01.03.M0.02 Acceptable Use for more information.  
*****
```

(wirag@grace.tamu.edu) Password:

(wirag@grace.tamu.edu) Duo two-factor login for wirag

Enter a passcode or select one of the following options:

1. Duo Push to XXX-XXX-████
2. Phone call to XXX-XXX-████

Passcode or option (1-2): 1

Success. Logging you in...

## Grace

# Running code on Grace

Grace has two primary directories:

- `/home/user/<username>/`      `<-` the default upon login
- `/scratch/user/<username>/`      `<-` NEED to work from here
  - Always run code from your scratch directory.

Once logged in, you'll need to set up virtual environments

\*\*\* Grace Terminal \*\*\*

```
cd /scratch/user/<username>           # change working directory to scratch
module load Anaconda3/2024.02-1       # load Anaconda to use conda commands
conda create --name my_env             # create env. called my_env (or whatever name)
conda activate my_env                 # activate my_env before package installation
conda install python pip astropy...   # install necessary packages
### if conda DOESN'T work ###
pip install <package>                 # some packages are only available with pip
```



# Running code on Grace

**Important:** HPRC is a shared resource, so we cannot run code directly from the Grace terminal (or face the wrath of an angry email reminding you)

- Instead, we submit job requests so resources are allocated fairly. Your Grace account will be charged in SUs

```
#!/bin/bash

#SBATCH --job-name=ngc383          # Job name
#SBATCH --time=2:00:00            # Job time limit
#SBATCH --ntasks=48               # Task/core, max cores per node is 48
#SBATCH --ntasks-per-node=48      # Task/core per node (equal to ntasks for 1 node)
#SBATCH --mem=50G                 # Memory limit per node, max is 384GB
#SBATCH --output=out.%j           # Outfile -- %J for unique job number
#SBATCH --account=<##>             # Project account number associated with GRACE
```

Job specifications

```
module purge                      # Purge all modules
module load Anaconda3/2024.02-1  # Load the Anaconda3 module
cd /scratch/user/wirag/ngc383    # Change directory
source activate bhenv             # activate environment
python jam_optimization.py        # Run script
conda deactivate                  # deactivate environment
```

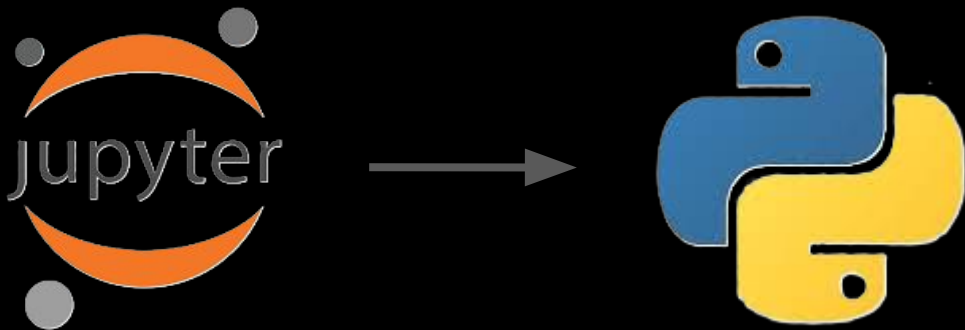
Example of executable terminal commands

# Running code on Grace

We learned about **Jupyter Notebooks** (`.ipynb` files) in a previous tutorial. However, you may have noticed I was running a `.py` file

- Since we access **Grace** through **SSH**, and to comply with the resource allocation system, we must run python scripts

Notebooks are great for testing code locally (on your computer) so you don't eat up your SUs on **Grace**



# Running code on Grace

After running your code, you'll need to transfer any outputs to your local computer for analysis.

```
*** Local Terminal ***
```

```
scp <username>@grace.tamu.edu:/scratch/user/<un>/path_to_files/files /Local/path
```

**OR**

```
*** Local Terminal ***
```

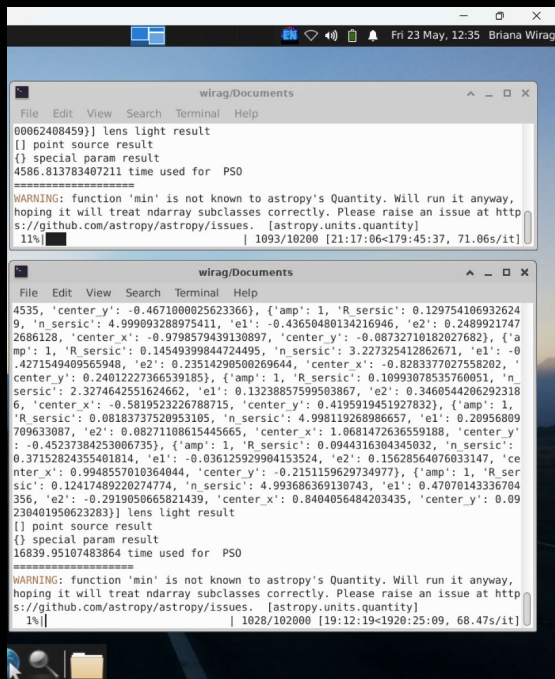
```
cd /Local/path
```

```
scp <username>@grace.tamu.edu:/scratch/user/<un>/path_to_files/files .
```

The period means “current directory”, you can double check what that is by typing `pwd` into the terminal

# Running code on Saire

Since we can access Saire through SSH or a GUI, we can run both scripts and notebooks.



```
wirag/Documents
File Edit View Search Terminal Help
00062408459]] lens light result
[] point source result
{} special param result
4586.813783407211 time used for P50
=====
WARNING: function 'min' is not known to astropy's Quantity. Will run it anyway,
hoping it will treat ndarray subclasses correctly. Please raise an issue at http
s://github.com/astropy/astropy/issues. [astropy.units.quantity]
11% | 1093/10200 [21:17:06-179:45:37, 71.06s/it]
```

```
wirag/Documents
File Edit View Search Terminal Help
4535, 'center_y': -0.467100025623366), {'amp': 1, 'R_sersic': 0.129754106932624
0, 'n_sersic': 4.999093288975411, 'e1': -0.43650480134216946, 'e2': 0.2489921747
2686128, 'center_x': -0.9798579439130897, 'center_y': -0.08732710182027682}, {'a
mp': 1, 'R_sersic': 0.14549399844724495, 'n_sersic': 3.227325412862671, 'e1': -0
.4271549409565948, 'e2': 0.23514290580269644, 'center_x': -0.8283377027558202,
'center_y': 0.24012227366539185}, {'amp': 1, 'R_sersic': 0.10993078535760051, 'n
_sersic': 2.3274642551624662, 'e1': 0.13238857599503867, 'e2': 0.3460544206292318
6, 'center_x': -0.5819523226788715, 'center_y': 0.4195919451927832}, {'amp': 1,
'R_sersic': 0.08183737520953105, 'n_sersic': 4.998119268996657, 'e1': 0.20956809
709633087, 'e2': 0.08271108615445665, 'center_x': 1.0681472636559188, 'center_y
': -0.45237384253006735}, {'amp': 1, 'R_sersic': 0.0944316304345032, 'n_sersic':
0.37152824355401814, 'e1': -0.036125929904153524, 'e2': 0.15628564076033147, 'ce
nter_x': 0.9948557010364044, 'center_y': -0.2151159629734977}, {'amp': 1, 'R_ser
sic': 0.12417489220274774, 'n_sersic': 4.993686369130743, 'e1': 0.47070143336704
356, 'e2': -0.2919050665821439, 'center_x': 0.8404056484203435, 'center_y': 0.09
230401950623283]] lens light result
[] point source result
{} special param result
16839.95107483864 time used for P50
=====
WARNING: function 'min' is not known to astropy's Quantity. Will run it anyway,
hoping it will treat ndarray subclasses correctly. Please raise an issue at http
s://github.com/astropy/astropy/issues. [astropy.units.quantity]
1% | 1028/10200 [19:12:19-1920:25:09, 68.47s/it]
```

After connecting with x2go, you can run scripts as normal

- Can run code from `/home/<username>`
- Store data/files in `/data/<username>`

Any code started in a terminal this way will continue to run in the background

# Running code on Saire

Running notebooks in x2go can be quite laggy. Thankfully, we can open a local browser that's running the remote notebook

```
*** Saire Terminal ***  
ssh <username>@saire.physics.tamu.edu  
*** enter password ***  
jupyter lab --no-browser --port=8889 #or another Saire port
```

```
*** Local Terminal ***  
ssh -N -f -L localhost:8888:localhost:8889 <username>@saire.physics.tamu.edu
```



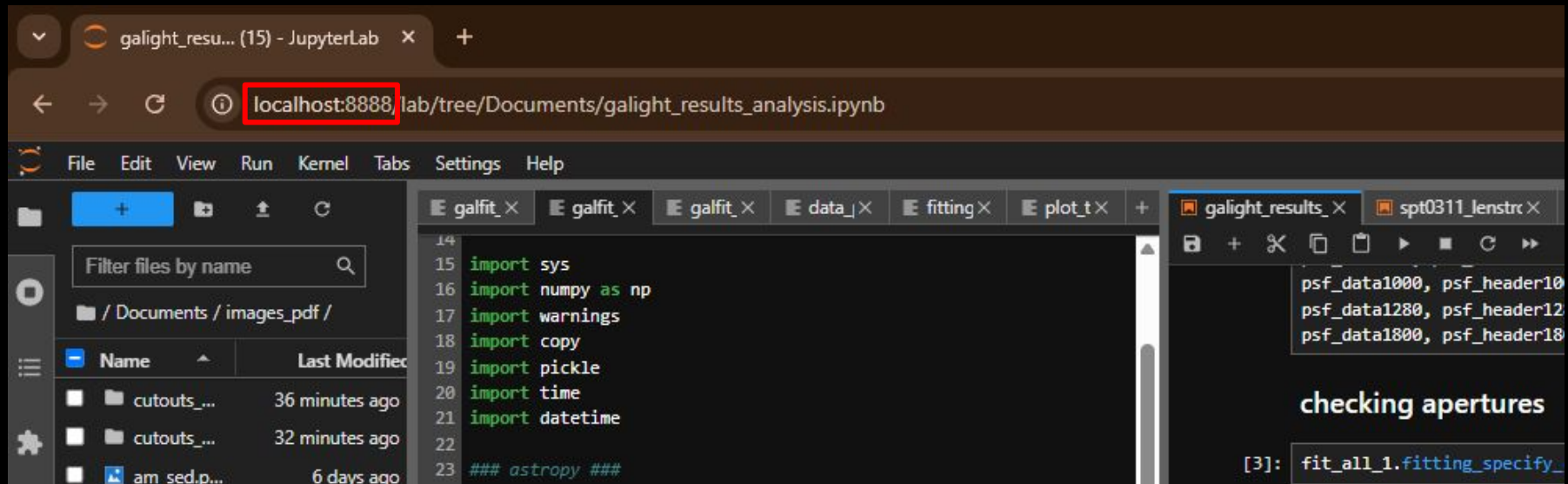
Local port



Saire port

Then type `localhost:8888` into a browser on your computer (it may ask for a token, which will be in the `Saire Terminal`)

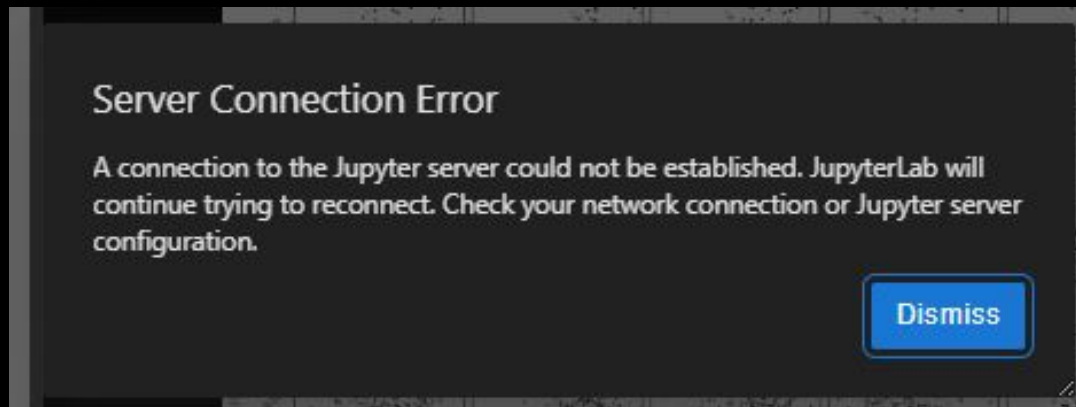
# Running code on Saire



This will look the exact same as normal Jupyter, but all of this is actually running on Saire

- Configuration will be preserved after disconnecting

# Running code on Saire



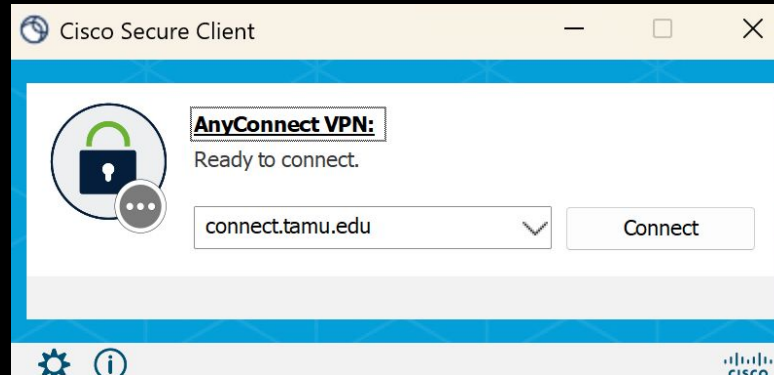
Unlike running scripts in x2go, if the `ssh` connection breaks, the notebooks do NOT continue running, so you will need to re-establish the connection and restart the notebook

- This happens if you're inactive for too long or lose internet connection

# Remote Access From Home

Everything mentioned thus far is assuming you are connected to TAMU wifi

- If you need to access these resources from home, you will need to install the TAMU VPN ([here](#))
- Follow the instructions under “**Getting Started**”





# Additional Resources

- Check out the [TAMU HPRC website](#) for more information about **Grace**
  - Includes more detailed explanations and examples than what was presented here
- If working with Justin Spilker, go to [this Evernote](#) page for more information about **Saire**
- When in doubt, look it up or ask for help :)