Tutorial 1: An Introduction to Some Useful Tools

1 Introduction

Throughout the semester you will use a virtual machine (VM), write in LaTeX, work with an image viewer like DS9, make use of the command line, and code in Python. Below are a series of exercises to get you familiar with these tools. I don't expect you to memorize the information, but rather gain a sense of what is possible and know where to go find specific syntax and additional help. Eventually tasks that you do repeatedly, will become second nature. Section 3 and the beginning part of Section 4 can be done without the VM, however for the remainder of the exercises you will need to have the VM running on your computer or will need to work in the computer lab in MPHY 330A.

2 Virtual Machine

All of the software that you need for this course is on a VM installed on the computers in the MPHY 330A computer lab or on a VM that you may download on a personal computer. If you choose to install the VM on your computer, there are instructions on Canvas. Additional information about working with the VM is below.

- For some of the tutorial activities, you will need internet access. The VM should automatically connect to the internet if the computer is connected to the internet. You will see an icon that looks like three rectangles at the top right of the VM (Figure 1) if the internet is connected. If you do not see the icon, try restarting the VM. To do that, click on the downward triangle at the top right of the VM (Figure 1), click on Power Off/Log Out, click on Power Off. Then start the VM again from the VM player.
- When you are done working, click on the Pause button (Figure 1), which will save the VM state (i.e., all of your new work since your last session), then quit the VM player (e.g., on a Mac, go to the menu bar, click on VMWare Fusion, click on Quit VMWare Fusion). Or, you just quit the VM player without hitting the Pause button, and the VM state will still be saved.
- The VM is portable. As discussed above, you can save your work on the VM (e.g., on your computer) and then copy the entire VM to your own Google drive and/or a USB flash drive. Then you can move to a different computer (e.g., a computer in MPHY 330A) download the VM, play the VM, and resume work where you left off.

Tutorial Report. (1) In your report, include a screen shot of the VM after you have logged on.

3 LaTeX

LaTeX is a typesetting program that is often used in science to produce professional documents, such as journal papers, telescope and grant proposals, resumes, and homework assignments/tutorial manuals like the ones you'll see in this class. With LaTeX, it is very easy to include equations and special symbols, and insert figures, tables, and references. Your tutorial reports should be written in LaTeX and we will use Texas A&M's Overleaf subscription to do so. Overleaf is an online LaTeX editor, with the advantage that you do not need to download LaTeX itself and multiple people can work on the same document simultaneously (although you will not need this feature for our class).

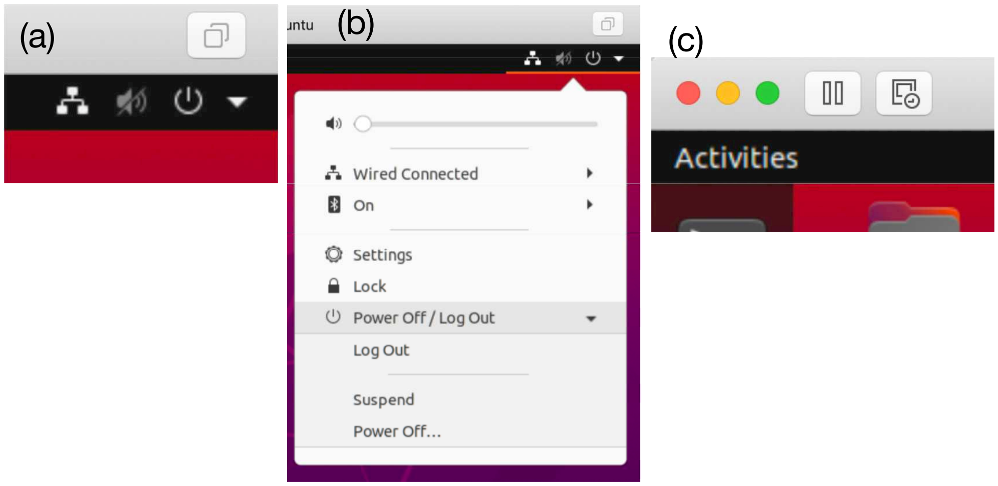

Figure 1: Zoomed-in views of the VM, including (a) the three-rectangle icon at the top right of the VM indicating that the internet is connected, (b) the downward triangle at the top right of the VM leading to the Power Off menu, and (c) the Pause button at the top left of the VM. If the internet is not connected, try powering off the VM and starting it back up. Clicking the Pause button will save your work/the state of the VM, then you can quit the VM player.

You can find Overleaf here[1]. If you already have an Overleaf account, click on 'Login' at the top right of the page. Otherwise, click on 'Register' and enter your Texas A&M email address and a password that you would like to use with your Overleaf account. Go through the LaTeX Part 1[2] and Part 2[3] tutorials. There are slides associated with each tutorial, so be sure to read the material and try out the examples by clicking on the relevant part of the slide to open a new browser window with Overleaf. Even if you have used LaTeX before, the tutorials are a good refresher and you may learn something new. Toward the end of the LaTeX Part 2 tutorial PDF, there is information about installing LaTeX on your own computer. There is no need to do that for this course – everything can be done through Overleaf. Besides the tutorials above, the Not So Short Introduction to LaTeX[4] and Overleaf Documentation Page[5] are good references.

On Canvas, under Home → Course Links → Tutorial Report Template, I have put a LaTeX template that you can use for reports. To start your report for this tutorial, download the template from Canvas but do not unzip the file. Open Overleaf in a new internet browser window, click on 'New Project' at the top left, and choose 'Upload Project' and then 'Select a .zip file' to upload the template zip file. After the project is created, rename the project by hovering your mouse over the project name (e.g., astr420_tutorial_template) at the top of the source text panel. You'll see a pencil (edit icon) appear. Click on the pencil and rename the project (e.g., astr420_tutorial_1). If you want to add figures to the project, you'll need to upload the figure to Overleaf and then use the proper syntax in the LaTeX file. To upload the figure to Overleaf, click on the Upload icon at the top left of Overleaf, underneath the Menu button. Also, you will notice that I created a number of LaTeX definitions near the top of the LaTeX template, which you may find helpful

[1] https://www.overleaf.com/edu/tamu#overview
[2] https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_1)
[3] https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_2)
[4] https://tobi.oetiker.ch/lshort/lshort.pdf
[5] https://www.overleaf.com/learn/latex/Main_Page

for your reports. To use, type the name of the defintion (including the backslash), which comes immediately after 'newcommand' in the LaTeX file.

Tutorial Report. (1) In your report, reproduce the following text exactly as shown below (including the text before and after the equation).

Under the assumption that the gas motions are isotropic in the r and z (cylindrical) coordinates, the correction is

$$v_c^2 \; - \; v_\phi^2 \; = \; \sigma_r^2 \left[- \frac{d \ln \nu}{d \ln r} \; - \; \frac{d \ln \sigma_r^2}{d \ln r} \; - \; \left(1 \; - \; \frac{\sigma_\phi^2}{\sigma_r^2} \right) \right] . \quad (1)$$

The above text comes from Walsh et al. 2013. The Astrophysical Journal paper described a black hole mass ($M_{\rm BH}$) measurement for M87 from gas-dynamical models of *Hubble Space Telescope* observations. Using the method, $M_{\rm BH} = (3.5^{+0.8}_{-0.6}) \times 10^9 \ M_\odot$ (1σ uncertainties).

4 An Image Viewer: DS9

DS9 is an image viewer often used in astronomy to examine and analyze fits files. Fits (Flexible Image Transport System) files are the standard format for storing astronomy data and you should read more about the format here[6]. DS9 is capable of displaying two-dimensional images and various cuts of three-dimensional data cubes. To start DS9, in the VM open a terminal by clicking on the black rectangle icon with white prompt symbol. The icon can be found on the left side of the VM near the top. In the terminal type `ds9 &`. If you wanted to open a particular fits file in DS9, you could type `ds9 filename_goes_here.fits &` in the terminal, or `ds9 filename_goes_here.fits[2] &` to open the second extension of the fits file. For this course, the most common DS9 tasks you'll need to know how to do include reading in fits files, examining a particular extension of a fits file, viewing the associated header, changing the scale/stretch/dynamic range of the image, blinking images against each other, overplotting contours, displaying a row or column cut through the image, measuring image statistics within a region, and visualizing a data cube. We will save the data cube visualization for Tutorial 3. As you go through the documentation linked below, pay particular attention to the other topics.

Watch the IPAC Introduction to DS9 part 2[7] and IPAC Introduction to DS9 part 3[8] videos. These videos were recorded in 2013, and so additional features have been added to DS9 since then and some menu items may look a little different in your version of DS9. Nonetheless, the videos provide a good overview. Next, skim through a more recent reference in PDF form, Introduction to Astronomy Images and the DS9 Image Viewer[9]. There is overlap in the material presented in the videos and in the PDF, but you will need to know how to analyze specific regions of an image (pages 65 − 70 of the PDF) for this course.

For reference, you can find more documentation about DS9 here[10], including a couple of manuals and a frequently asked questions page. Finally, an important note about DS9 is that it starts counting at 1 and not at 0 like some computer languages, including Python. Therefore if you displayed a fits file in DS9 and read the same fits file into Python as a 2D array, pixel at x= 100, y= 100 in DS9 corresponds to x= 99, y=99 in the 2D Python array.

Tutorial Report. On page 17 of the DS9 PDF above, there is a link to a fits file of *Spitzer* 24μm image of M81. Click on the link and save the fits file to the VM. Open DS9 and display the image. In your tutorial report address the following:

[6] https://heasarc.gsfc.nasa.gov/docs/heasarc/fits_overview.html
[7] https://www.youtube.com/watch?v=Z1zic8msSM0
[8] https://www.youtube.com/watch?v=vVwW-8h2drw
[9] http://www.jb.man.ac.uk/ gbendo/Sci/Pict/DS9guide.pdf
[10] https://sites.google.com/cfa.harvard.edu/saoimageds9/documentation?authuser=0

1. When you first load the image, you'll see a small white dot. Let's change the range of values displayed so that we can actually see the galaxy. Keep the scale linear and the default colorbar stretch, but display values between -0.15 and 2.5 MJy/sr (which are the units of the pixel values in this image). Describe how you made the change and include a screen shot of the resulting image in your report.
2. Display the header associated with the image and include a screen shot of the header in the tutorial report.
3. At pixel x = 300 and y = 1075 (with counting starting at 1), what is the right ascension (RA) and declination (Dec) in units of hours, minutes, and seconds for RA and degrees, minutes, and seconds for Dec? Use DS9 to change from that sexagesimal format to a degree format, describe how you did so, and record the RA and Dec again for the same pixel.
4. At x = 700 and y = 1103 (with counting starting at 1) what is the pixel value (which will be in units of MJy/sr)?

Check point! Complete the tutorial, and write your report, up to this point. Submit a PDF of the report to Canvas for grading and feedback.

DS9 Continued...

Tutorial Report. Continue working with the M81 *Spitzer* 24μm image and use DS9 to answer the following in the tutorial report:

5. Open the same fits file but in a new frame. Show the second frame side-by-side with the first frame. Earlier you changed the range of values being displayed. Now, in the second frame only, change the scale to be anything other than linear and change the colorbar stretch. Therefore, you should have the image from Question 1 on the left and the modified image on the right. Describe how you implemented these changes in your report and take a screen shot to include in the report.
6. Delete the second frame and go back to a single frame showing the image from Question 1. Next, take a column cut from y = 600 to y = 1500 at x = 435 (with counting starting at 1) and plot the pixel values as a function of pixel location in DS9. Change the default plot so that the x-axis has a title of 'x, y pixel' and the y-axis has a title of 'MJy/sr'. Take a screen shot of the plot and include in your report.
7. What is the standard deviation of the pixel values in a circular region with a radius of 30 pixels centered on x = 780 and y = 920 (with counting starting at 1)? Describe what you set in DS9 to determine the standard deviation in this region.
8. Zoom into the star located at x = 261 and y = 1173 (with counting starting at 1). Use DS9 to overplot contours. Set the contours such that there are 10 levels from 0.15 to 3.5 MJy/sr with smoothness set to 3. Describe what you did in DS9 and include a screen shot of the star and contours.
9. Although not covered in the DS9 videos/PDF above, one of the DS9 regions is a ruler and it will measure the projected distance between two points on the image. Use the ruler region to determine the projected distance in arcseconds between a star x = 475, y = 1090 and a point near the galaxy nucleus at x = 538, y = 1020. Provide the projected distance in your tutorial report and outline how you made the measurement.

10. Play with DS9, and describe one new feature/tool that you learned about that was not previously covered in the DS9 videos or the PDF.

One additional measurement often made is the full width at half maximum (FWHM) of a star's light profile. As far as I know this cannot be done with DS9 alone, although it can be performed using the combination of DS9 and PyRAF, which is a collection of astronomical software for reducing and analyzing data. We'll use PyRAF more in Tutorial 3. For now, follow the steps below to use the `imexamine` package in PyRAF to measure a star's FWHM.

The image should already be loaded in DS9. To open PyRAF, in a terminal type `conda deactivate`, hit return, type `conda activate iraf27`, and finally type `pyraf`. Once in PyRAF, load the `imexamine` package by typing `images`, hitting return, and typing `tv`. Next, type `epar imexam` and fill in the `input` field with the image name; everything else can be left at their default values. Click on `Execute`. You will see a flashing cursor when placing your mouse over the image in DS9.

Let's look at the star at x = 314 and y = 883 (with counting starting at 1). Place the blinking cursor over that location in DS9 and type `r`. You will see a radial profile of the star's light and numbers at the bottom. The last number is the FWHM in pixels; make a note of this. If you want to print the FWHM without showing the radial plot, place the blinking cursor over the star in the DS9 image and type `a`. Again the last number (under DIRECT) is the FWHM in pixels. With `imexam` you can also perform tasks like doing a line (i.e., a row) cut at the position of your cursor by typing `l`, a column cut by typing `c`, making a contour plot of the region around the cursor by typing `e`, and displaying a surface plot of a region around the cursor by typing `s`. Try these functions out. When you are done, place your cursor anywhere on the DS9 image and type `q`. This will bring you back to the terminal. Type `.exit` to quit PyRAF. Go back to the `astroconda` environment that runs Python 3 for the remainder of the tutorial. In the terminal, type `conda deactivate`, hit return, type `conda activate astroconda`.

Tutorial Report. (11) What is the FWHM of the star located at x = 314 and y = 883 (with counting starting at 1)?

5 The UNIX Shell

The most common way people interact with computers is by using a graphical user interface (GUI). With a GUI, we give instructions by clicking on an icon or selecting an option from a drop-down menu. GUIs make learning how to use computers easy, but giving instructions to a computer though a GUI scales very poorly. For example, if you had to copy the fifth line of two thousand text files in two thousand different directories and paste them into a single file, using a GUI would be extremely time consuming and tedious and there is a high probability that you will make a mistake. This is the advantage of the UNIX shell, which is both a command-line interface (CLI) and a scripting language. With certain commands, the UNIX shell can repeat tasks, with or without some modification, as many times as we want, and the example task described above can be accomplished in seconds.

The shell is a program where users can type commands. With the shell, it's possible to invoke complicated programs or simple commands, like deleting a file. The most popular UNIX shell is Bash (the Bourne Again SHell), which is the default shell on most modern implementations of UNIX. The class VM runs UNIX, specifically Ubuntu Linux, and the default shell is Bash. Using the shell takes some time and effort to learn; it is similar to learning a new language. While a GUI presents you with options to select, the command line choices are not given to you. However, knowing a few "words" (i.e., commands) really goes a long way, and we'll focus on those basic commands now. Sequences of commands can be written into a script too.

The command line is also the easiest way to interact with remote machines and supercomputers. Thus, familiarity with the shell is essential, and is becoming more important as high performance computing systems and cloud computing systems gain popularity for working with large amounts of scientific data.

Read and work through the six tutorials on Canvas, in the UNIX Tutorials Module. Anything highlighted in gray in the tutorials is information that you should type into the terminal on the VM (or is information that the terminal will display after you issue a command).

Tutorial Report. In your report, include responses for all 9 exercises (labeled as 'Tutorial Report' in the tutorials).

6 A Text Editor: Emacs

There are many plaintext text editors available, any of which can be used to write codes and scripts. One such text editor is emacs. You can use emacs as a GUI or within the terminal along with combination of keystrokes. (You can also use those same keystrokes within the GUI.) To open emacs as a GUI type `emacs filename_goes_here.txt &` where `filename_goes_here.txt` is the name of the file. If the file does not yet exist in the directory from which you issued the command, then a new file will be created. If the file already exists, then emacs will load it. The `&` will run emacs in the background and you will still be able to use the terminal from which you initially issued the emacs command. When emacs opens, there will be two panels/buffers in the window. To remove the bottom buffer that provides emacs help, click on the top buffer, go to File → Remove Other Windows. Then you'll have an entire window to type your code.

You can use emacs as a GUI, but over time as you get used to emacs, it may be faster to use a series of keystrokes to carry out actions like moving to the end of a sentence or paragraph, delete the rest of the line, copying, pasting, saving, quitting, etc. On Canvas there is a cheatsheet for emacs keystrokes. You can also learn keystrokes by looking the pull-down menu options at the top of the emacs GUI. For the keystrokes, 'C' corresponds to the Control key and 'Meta' corresponds to the Option key. I recommend using the emacs GUI (with or without keystrokes), but if you become very comfortable with using only keystrokes then you could consider using emacs within the terminal. To open emacs without the graphical interface, in a terminal type `emacs -nw filename_goes_here.txt` where the `-nw` flag tells emacs not to create a graphical frame.

If you already have a favorite text editor (e.g., vi, nano), by all means use that. LibreOffice Writer (Ubuntu's equivalent of Microsoft Word) is not a text editor.

7 A Python Primer

Python 3 is a common programming language used in astronomy, and will be the language we use in this course. For those that have no prior Python experience, the tutorials below will introduce you to the language. They are, of course, not meant to be comprehensive but will be enough to get you started. You may also find the following references useful: Python for Astronomers[11] and the Python Data Science Handbook[12]. Once you understand the basic concepts in one programming language, learning a new language is mostly a matter of figuring out the proper syntax. Therefore, even if you never use Python again, you will have laid the necessary groundwork for learning any other programming language.

[11]https://prappleizer.github.io/textbook.pdf
[12]https://jakevdp.github.io/PythonDataScienceHandbook/index.html

There are three Python tutorials below and they are written as Jupyter notebooks. If you aren't familiar with Jupyter notebooks, read this[13] brief introduction. Note, Jupyter is already installed on the VM and you can simply start the Jupyter notebook server from the terminal. You may want to use Jupyter notebooks for this course, or you may want write Python code using a text editor (like emacs), save the file as a `.py` file, and run the code by typing `python nameofcode.py` in the terminal. Either way works.

The Python tutorials can be found in `/home/student/python_primer/` on the VM. Start with `tutorial_part_1.ipynb`, then work on `tutorial_part2.ipynb`, and finally `tutorial_part3.ipynb`. The `plotting.ipynb` Jupyter notebook is not a tutorial, but is a guide for producing nicer looking plots in Python. You should read through the document now, and may find it helpful as a reference in the future. To run the first tutorial, open a terminal on the VM, navigate to the `python_primer` directory, and type `jupyter notebook tutorial_part_1.ipynb`. The `tutorial_part_1.ipynb`, `tutorial_part_2.ipynb`, and `plotting.ipynb` were written by Imad Pasha & Christopher Agostino as part of the Python for Astronomers: An Introduction to Scientific Computing textbook, and were modified for our purposes.

Tutorial Report. In your report, include responses for all 11 exercises (labeled as 'Tutorial Report' in the notebooks).

Congrats! This concludes Tutorial 1. Write the remainder of your report (by updating the report from the check point onward) and submit a PDF of the report to Canvas.

[13] https://realpython.com/jupyter-notebook-introduction/