Week 6 Exercises

Physics 91SI Spring 2011 Handout 17

Alex Ji and Zahan Malkani

May 5, 2011

Halo Rotation Curves

Introduction and Motivation

For this part of the exercise, we're going to build up a profile of galaxies, their spatial and velocity distributions in a set of halos. These halos are from the same simulation data that was shown in class by Matt this Tuesday. All you need to understand about it at this point is that it simulates the effect of gravity on N galaxies (hence called an N-body simulation) over exteremely long periods of time. We will go into the Physics in a later section. There are two pieces of data, the final positions and velcoties of galaxies after the simulation has been run, and the final positions of the center of mass of dark matter halos. If the notion of galaxies in halos is confusing, don't worry about it. Just think of them as clumps of dark matter.

We're essentially going to read in the data, pick a halo, identify the galaxies within that halo, and then plot a couple of statistical summaries of the data for those galaxies.

Reading in the data

Time to flex that (extremely simple) file reading capability that you already know. In the `vel_distr.py` file we've set up the framework of what you should need.

Fill out the code in the `VelDistr` constructor, reading each file into the appropriate list. Store a dictionary (also called a map) for each halo / galaxy with meaningful keys like `xpos` and `yvel`. Your final data structure should be a list of dictionaries. Makes life a whole lot easier when someone else is reading your code. Cool, yeah?

Identify the Interesting Galaxies

Next move your attention to the `galaxies_id()` function. This function is responsible for picking out those galaxies that fall within this halo. Use the same outline as the code we deciphered in class. Hint: the euclidean distance needs to be less than `MIN_DIST`. It might be useful to store the calculated distance from the halo centre in your galaxy info map if it satisfies the conditions. Return a list of these galaxies.

Make Plots Summarizing Data on Galaxies

Here we are going to implement the `halo_stats()` function, taking a list of galaxy infos assumed to be within the same halo. First, make scatterplot of the galaxy positions in the X-Y, and other planes. Take a look at it. Is it what you would expect from a cluster of massive bodies

under the influence of gravity. Next make a histogram of the density of galaxies as a function of (euclidean) distance from the halo center. Now lets, focus on the velocities. Make a histogram of (the magnitude of) the velocities of the galaxies. And for our final materpiece, make a scatterplot of the velocity of galaxies as a function of distance from the halo centre. Is this what you expected to see from elementary notions of orbital velocity, using Newtonian Gravity? Here lies compelling proof for the prescence of dark matter.

Installing and Running New Software

This is a real-life example I helped someone with last weekend. She wanted to run her program on `corn` because it was taking a long time on her lab's computer, so I helped her install it and run it in the background. It turns out this is a pretty useful thing to be able to do yourself, so we're going to have you try it too.

Read Script Documentation

Clone over the repository `exercise6install` from the class /src folder. You've been told before that you can run things by typing: `./run_dammin.py 1 M01PB.90.out M01PB`. Read through `run_dammin.py` and make a list of all the things you need to do in order to get this to run on a new computer **BEFORE YOU READ THE NEXT PARTS**. If you're having trouble, you can look ahead, but try to restrict yourself as much as possible.

If you're feeling confident, try installing everything yourself. If you're worried, follow the handout to get an idea of what you have to do.

Download and Unzip ATSAS

The comments tell you that you need working copies of DAMAVER and SUPCOMB. It gives you two links to download stuff, but you can download the file linked here: `http://www.stanford.edu/~alexji/`

Get the file onto `corn` using `scp`, `ftp`, or an FTP client (Filezilla, SecureFX, Fetch). Move it into your work directory then type `tar zxvf atsas-2.3.2-1.i686.tar.gz` and it will automatically unzip into a directory called `ATSAS-2.3.2-1`. (This directory is 44MB, so if your disk space is very full you might have to delete some stuff.) The `/bin` folder contains all of the necessary executables.

Create an environment variable and modify PATH

The script assumes that there is an environment variable called `DAMAVER_exec` that points to the executable `damaver`. In your `.cshrc`, add a line that says `setenv DAMAVER_exec 'path_of_damaver_executable'`. For me, the line is: `setenv DAMAVER_exec '~/work91SI/ATSAS-2.3.2-1/bin/damaver'`

There's a line in the comments that say: "In addition, you need to make sure that the SUPCOMB executable is in the path". In your `.cshrc`, there should be a place where you set the PATH variable (described in the section on environment variables on a previous exercise). Add something like `~/work91SI/ATSAS-2.3.2-1/bin` to that.

Remember that the `.cshrc` is only run when you log in, so you can either logout/log back in or type that `setenv` line into the shell. Check by doing `echo $DAMAVER_exec` and `echo $PATH`.

Edit the actual script

The script itself defines a variable called `dammin_exec`. You should change this to link to the correct file in YOUR directory.

Since the script referenced another file, which is `dammin_reconstruction_series.pl` you should look through that file and make any necessary changes to the directories it references.

Run the script

If you've been following the handout, you should be ready to run the script, so try running it (command is in the script documentation section of this exercise). It should spit a whole bunch of output to the console. Type `Ctrl-Z` to suspend this process, then type `jobs` to see that it's suspended. Kill the run_dammin process with `kill %1` (or whatever other job number it is on your computer), then delete all the extra files it made. (All the `*.log` files and the `run.M01PB*` file.

This program actually takes about a day to run on one file. However, it will only run as long as you are logged into the `corn` computer. In order to circumvent this problem, we would like to run it in the background. You can do this by typing `bg` once you've suspended the process, or you can just add an ampersand to the end of the first line you call.

Unfortunately, as far as I can tell, it doesn't seem possible to logout and have it continue running when you're working on `corn`. On other computers, you can use the command `nohup` to run processes that will continue after you've logged out, but corn seems to go through and kill processes whose owners log out. Oh well.

However, we can simulate this "background process' thing by logging in to the same computer from a new terminal. So open a new terminal and login to corn, then log in to the specific corn machine that your run_dammin process is running on. For instance, if I started the run_dammin process on corn25, when I log into corn from a separate terminal I might get something like corn29. I can just type `ssh corn25.stanford.edu` and I'll log into corn25.

You can now type `ps -ef | grep (your username)` to search for all processes on this computer that you own. This allows you to get the process id of any background running jobs and kill them with `kill`. Go ahead and kill the run_dammin process from this other terminal.

You're Done!

Push the exercise6 repository, but don't push the exercise6install repository. You can actually go ahead and delete the ATSAS folder, if you'd like to free up some disk space.