**GalFIT with Parallel Processing**

Ben McColgan, Bailey Cross, Ammar Malik, Shah Hassan Syed

Department of Computer Science

Siena College

Loudonville, NY 12211

**Abstract**:

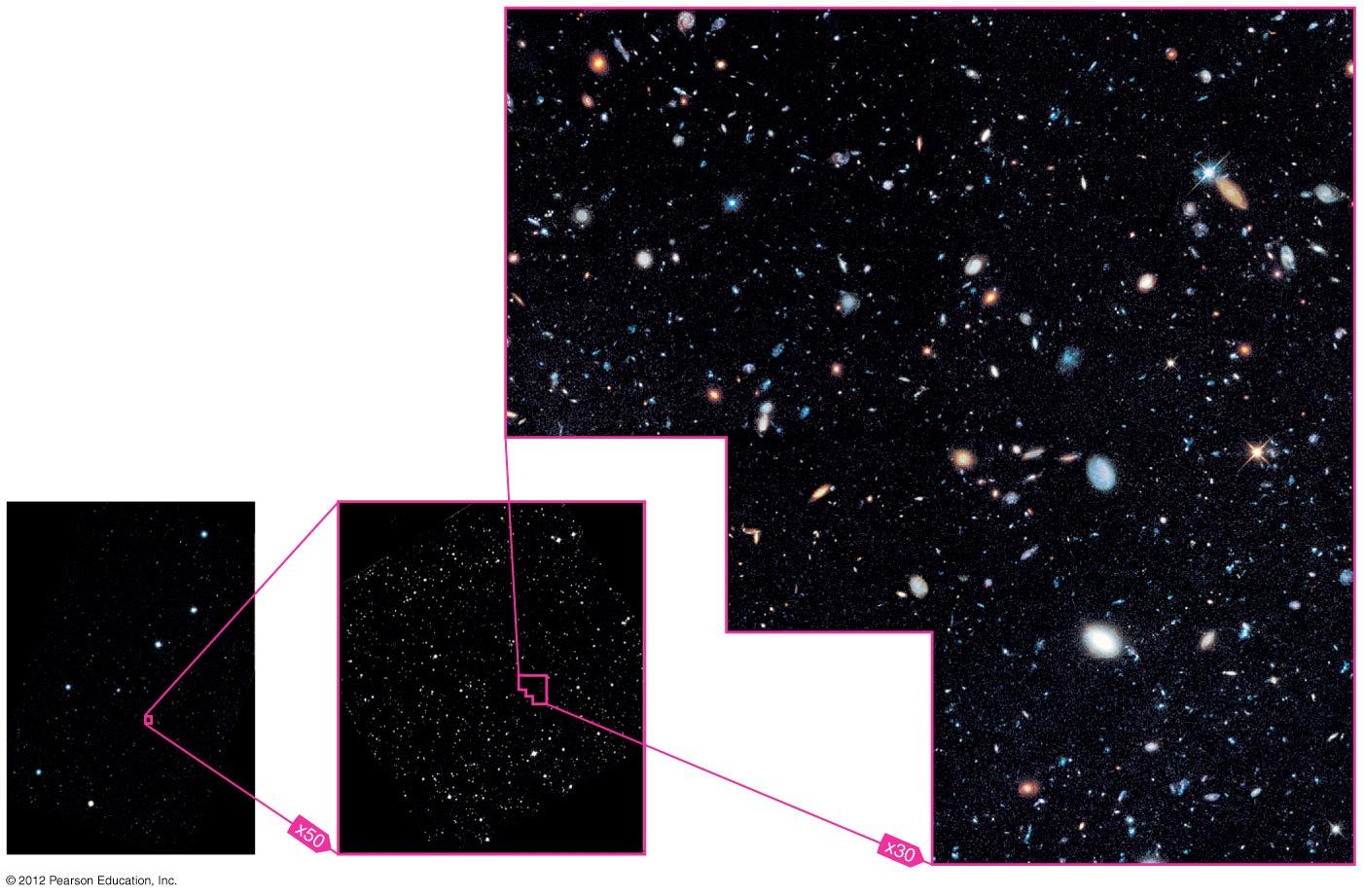
GalFIT or Galaxy Fitting is an algorithm developed first in 2002 and published in the Astronomical Journal. The entire concept of this algorithm was to “extract structural components from galaxy images, with emphasis on closely modeling light profiles of spatially well-resolved, nearby galaxies observed with the Hubble Space Telescope”[1]. The program calls an algorithm on a series of arguments. Our study was based upon parallelizing the calls so all the calls take place parallel to each other rather than serially. We would be using GNU Parallel to solve the time constraint and achieve significant efficiency in the GalFIT timing.

1. **GalFIT**

GalFIT is a two dimensional galaxy fitting algorithm developed by Peng et al. in 2002. The algorithm models galaxies and fetches the components that are contained in a galaxy. The modelling is based upon the light profile of galaxies that are able to be seen through a hubble telescope, and are provided as an image. Using this method, GalFIT is able to extract partial structures of a galaxy and provide other functionality over the galaxy, which includes: “standard modeling of global galaxy profiles; extracting bars, stellar disks, double nuclei, and compact nuclear sources; and measuring absolute dust extinction or surface brightness fluctuations after removing the galaxy model” [1].

**1.1 Classification of galaxies**

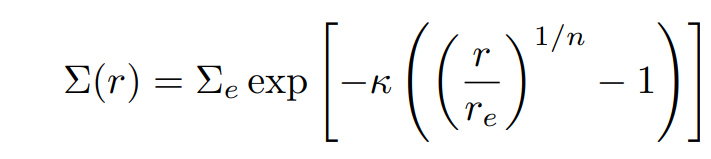
Since GalFIT works upon several galaxies, there are many types of galaxies that it comes across. All these galaxies are classified into different categories, such that each category holds different parameters. These variations in the galaxies lead to a different yield of results from the algorithm. Specifically, the categories are Spiral Galaxies, Elliptical Galaxies, Lenticular Galaxies, and Irregular Galaxies. Spiral Galaxies have thin disks with circular orbits which are all in the same directions with stars in the center. Elliptical Galaxies have stars in elliptical orbit that are in all directions. Lenticular Galaxies, which are like spiral galaxies, but with less external dust. Lastly, there are irregular galaxies, which have ongoing star formation. All these variations in the structures of the galaxies ultimately lead to different structural components and hence different results from the algorithm too.



In the above image we have several galaxies that are of a small size and can even be further narrowed down to a couple of pixels. If GalFIT is run on these galaxies with different parameters we can narrow down the specific galaxy type.

**1.2 Sersic Profile**

GalFIT models galaxies and classifies them using the sersic profile. The sersic profile of a galaxy is a measure of the variation of the intensity of the galaxy in accordance to its radius, the distance from its center. The following is the formula used to derive the sersic profile of a certain galaxy;



Here Σe is the surface brightness of the at re, n is the concentration, such that when n is large, the profile tends to be steep. re is the effective radius that encloses half of the galaxy light and k is a control that changes based on n to make re enclose half the light.

1. **Project Overview:**

The goal of the project is to help Dr. Rose Finn and Kimberly Conger in her Ph.D thesis to run GalFIT on the large amount of galaxies to accelerate the research. The target set of data to be classified contains 14,000 galaxies in a TAR file. Hence, an iterative run over the data would take days of processing on GalFIT, given that one galaxy takes approximately 10 seconds to be modelled. The path taken to overcome such time complexity here is parallel processing of any kind to cut the time as much as possible.

**2.1 Initial Plans:**

Since GalFIT is written in C, the initial plan was to get to the algorithm and divide the work. The method of choice devised was pthread, and the scheme was to divide different parts of the algorithm on different pthreads. As complex as it seems to divide a Galaxy fitting algorithm into pthreads, we never got a chance to test that, since we were not able to access the source code. We also got in contact with the creator of GalFIT, Chieng Y Peng, to ask if he has knowledge about others that are working on Parallel GalFIT running or offers information on such projects. Unfortunately he did not know anyone, neither was he ever involved in such development himself. However he plans to get into this path of research himself soon.

Another plan that was devised was to write an external C program and use pthreads to call the algorithm in its entirety on each pthread. The reason that this didn’t work is because without the source code we were left with only an executable file. Without access to source code and only an executable file we were stuck. We came to the conclusion that we were going to have to use something else to run the executable file in parallel. After a few days of research we found something called GNU Parallel.

**2.2 GNU Parallel:**

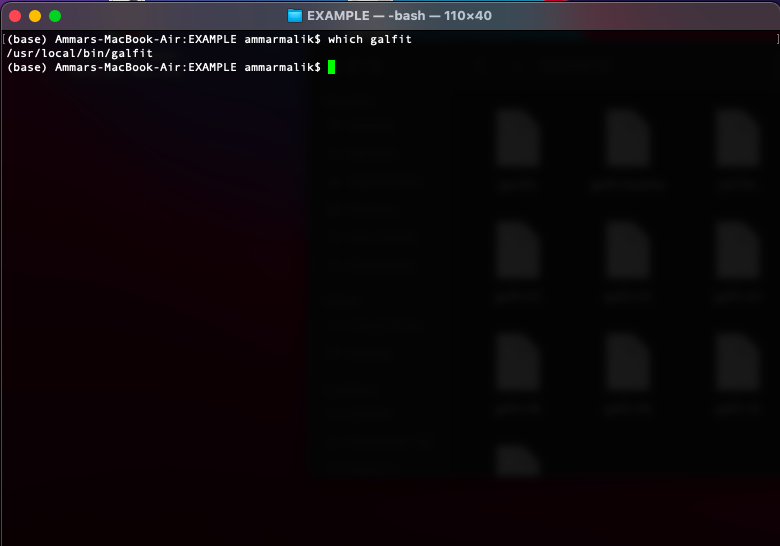
Now that we were not able to introduce Pthread or OMP into the algorithm, we turned towards another mode of parallelism that is structured upon the bash command line itself, and that is the GNU Parallel. GNU Parallel allows you to run galfit without the need to access the source code and use the command line required to run galfit in parallel. GNU Parallel is a utility that comes with linux and unix operating systems. It provides the ability to run scripts and commands on the command line in parallel. A better way to describe it is, “GNU Parallel is a shell utility for executing jobs in parallel. It can parse multiple inputs, thereby running your script or command against sets of data at the same time.” [2]

The plan of calling the algorithm on different pthreads inside an external C program was perhaps replaced by the GNU Parallel, which provides a similar solution. The following GNU command line was used to run GalFIT in parallel;

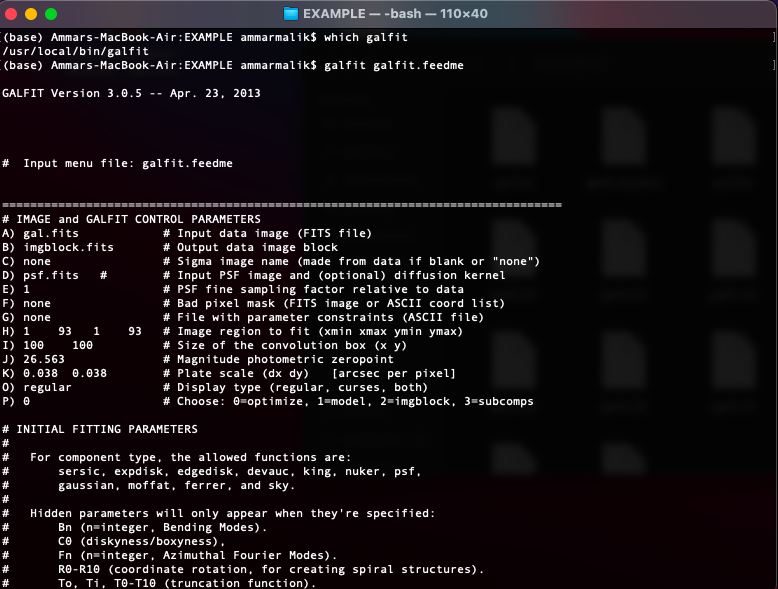
**time parallel -j+0 --eta ‘cd {} && galfit galfit.feedme > output.dat && cat ../fit.log >> allFits.log’ ::: \*/**

* time measures the time for the command to execute
* parallel tells the computer to use GNU Parallel to divy up the tasks to the cores available.
* -j+0 this command tells GNU Parallel how many jobs to assign per core. Ie j+0 assigns one job per core.
* --eta gives information about the command in real time, as well as the completion information. Ie the computer the command is running on, the number of cores, the number of jobs in total, jobs running in real time, jobs completed, % of started jobs, and average seconds to complete.
* cd {} changes directories into all of the subfolders that we are running the command on. The way that Dr. Finn set it up so that every galaxy is in it’s own directory and the script is running in the parent directory. The \*/ runs the command after the && symbol in each subdirectory.
* galfit galfit.feedme runs galfit on each galaxy using the input parameters in galfit.feedme and outputs the last iteration to a file called fits.log.
* output.dat is there so that galfit doesn’t print to the screen.
* cat fit.log >> ../allFits.log appends the fit.log in each galaxy subdirectory to a parent directory file called allFits.log

1. **Running GalFIT**

Running GalFit requires an install in the root directory of the system where it is stored. After installation we can run the command ***which galfit*** to check if it installed correctly or not. 

**3.1 Extracting Galaxies**

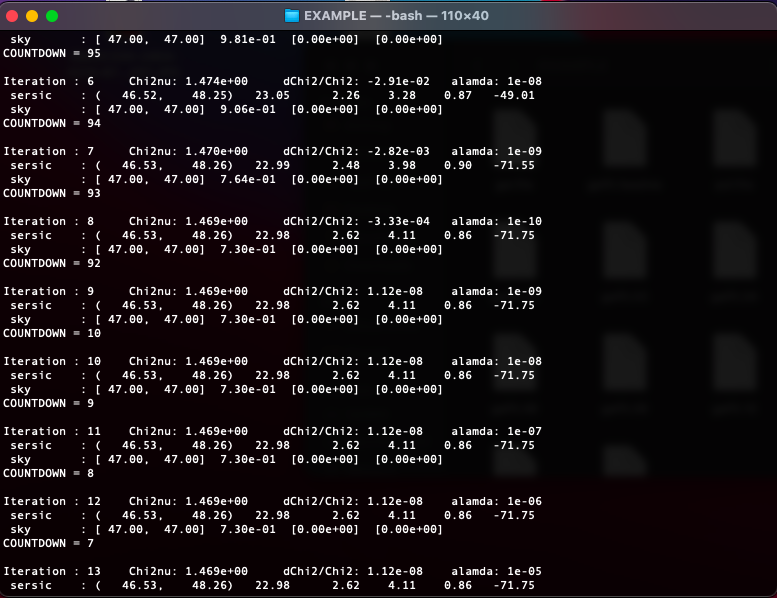
After checking if GalFIT is properly installed we can begin to extract information from our galaxies. After going into the correct directory we can run the command ***galfit filename.feedme*** and require some additional files to run properly. The feedme file encloses all the required information, variables and controls needed to run GalFIT. The following is the image of the feedme file:

**3.2 Parameters and Controls**

As it comes with no surprise parameter and control variable provided to the program yield results in several different manners. To run GalFIT, it requires the user to provide certain parameters, which include an input file, that encloses other information and parameters too. Along with the input file, an image, a noise image, a PSF image and a mask image must also be given.

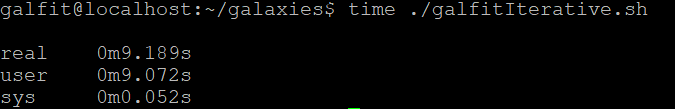
The following are the extended parameters that go into GalFIT, defining the type of the galaxy.

| **Parameter** | **Min** | **Max** | **Meaning** |
| --- | --- | --- | --- |
| n | 0.5 | 4 | Concentration of the light profile steepness; the ratio of light to the radius |
| Re | 1 | Image size | Radius that encloses half the light |
| B/A | 0.1 | 1 | Ratio of semi minor to semi major axis |
| PA | 0 | 180 | Position angle of the ellipse |
| (xc,xy) | 1 | Image size | Center x and center y coordinate |
| magnitude | 8 | 20 | Brightness of galaxy in log scale |

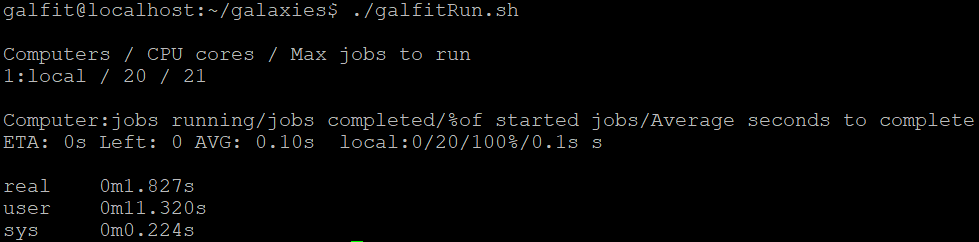
After running the command we are met with the following output based on the amount of iterations in the parameters. Each of the parameters affect our output as such they approximate the data from an image. We can change around the parameters to see if we get any major changes in our output. An example of this output can be shown in this picture below.

**4. Results**

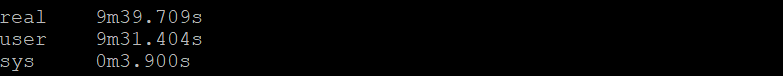
GALFIT iteratively on 20 galaxies:

****

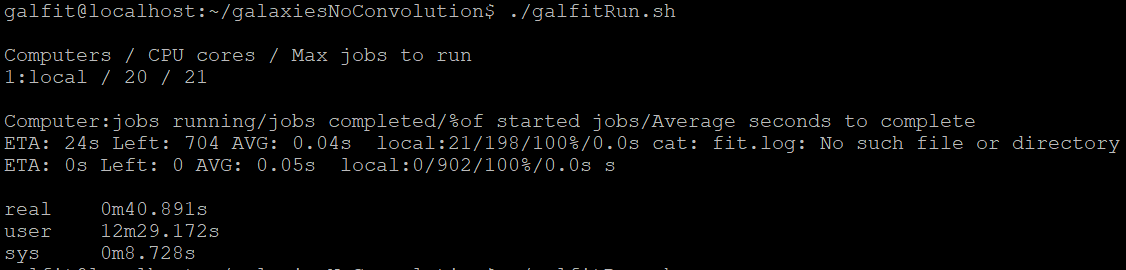
GALFIT in parallel on 20 galaxies:



GALFIT iteratively on 900 galaxies:



GALFIT in Parallel on 900 galaxies:



**5. Conclusion**

Starting with an initial plan of implementing Pthreads to the algorithm, we were halted in our route due to the unavailability of the source code. After various days of researching and talking with Dr. Finn, we realized that the only way forward was to somehow parallelize the command line executable for the program. We were able to find the GNU library which helped us achieve this goal by writing a bash script which allowed us to run the command line in parallel. By implementing this bash script we were able to run GalFIT on various datasets, ranging from just 10 galaxies to over 900 galaxies. We were also able to see a significant speedup compared to running the program serially, with efficiency reaching around a factor of 18x.

**References:**

[1][**https://ui.adsabs.harvard.edu/abs/2002AJ....124..266P/abstract**](https://ui.adsabs.harvard.edu/abs/2002AJ....124..266P/abstract)

[2] [**https://opensource.com/article/18/5/gnu-parallel**](https://opensource.com/article/18/5/gnu-parallel)

[3][**https://users.obs.carnegiescience.edu/peng/work/galfit/GFAQ.html**](https://users.obs.carnegiescience.edu/peng/work/galfit/GFAQ.html)