

Dust, Gas, and Metallicities of Cosmologically Distant Lens Galaxies

Phys 4310 - Capstone

By

Blake T Johnson

Under the guidance of

Dr. Xinyu Dai



Homer L Dodge Department of Physics and Astronomy,
University of Oklahoma, Oklahoma

NOVEMBER, 2023

CONTENTS

1	INTRODUCTION	1
2	Our Objective	2
3	How lensing helps to measure dust and gas	3
4	The Research	5
4.0.1	Obtaining the Data	5
4.0.2	Reprocess	5
4.0.3	Creating a PSF image	6
4.0.4	Creating a full, hard, and soft image	7
4.0.5	creating Sherpa texts	7
5	Next Steps	9
6	Conclusion	10
A	Q _1355 Spectrum	11

Chapter 1

INTRODUCTION

My capstone project was to analyze the X —ray observations of 10 gravitational lenses, HE 0047 —1756, QJ 0158—4325, SDSS 0246 —0805, HE 0435 —1223, SDSS 0924+0219, SDSS 1004+4112, HE 1104 —1805, PG 1115+080, Q 1355 —2257, and Q 2237+0305, to measure the X —ray absorption between images, the metallicity, and the dust —to —gas ratio of the lens galaxies. The objective is to measure how much dust and gas are in the interstellar medium of different galaxies. We used images from the Chandra X —ray telescope to obtain our data.

During my time in capstone, I collected 124 images from Chandra relating to our 10 observations. I loaded them into the school computer and then reprocessed the images to get a clear image. Then I created a psf file for the images and created a Sherpa script for three different energy levels in order to help me run a spectrum on the X-rays.

In this paper I will discuss the objectives of our research and why it is important, how I plan to make our measurements and why it is effective, the process taken up to this point for the research, and what we have left to do.

Chapter 2

OUR OBJECTIVE

The objective of our research is to determine the dust, gas, and metallicities in the interstellar medium. We want to calculate the ratio of dust to gas, obtain more research in this area, and obtain a better understanding of the activities in galaxies.

It is important to understand the interstellar medium (ISM) and the intergalactic medium (IGM). We use the ISM and IGM in studying many areas of astronomy, specifically in star and galaxy formation and studying evolution. Measuring the extinction by dust in the ISM is also important so that we can make corrections when using high redshift standard candles. These corrections will help us correctly determine the physical properties of sources and images. We measure extinction laws by comparing the spectra of reddened and un-reddened stars of the same spectral type so it is important for us to be able to correct for the dust and gas in our measurements.

Chapter 3

HOW LENSING HELPS TO MEASURE DUST AND GAS

Gravitational lensing happens when an extremely massive object - in our research a galaxy - causes the curvature of spacetime. The path of light flows through spacetime and appears visibly bent (much like light bends through a lens). The large mass that causes the bending of the spacetime is called a gravitational lens.^[1]

Einstein explained in his general theory of relativity that time and space are two sides of the same coin. He called the coin "spacetime." Very large objects cause space to curve. Gravity is the curvature of the spacetime. Light travels through spacetime and curves when spacetime curves. Gravitational lensing is a very large example of this theory^[1].

As light approaches these large galaxies, the light is curved in different directions around the galaxy. To an observer on the other side of the galaxy, it appears that there are multiple light sources, not just a single light.

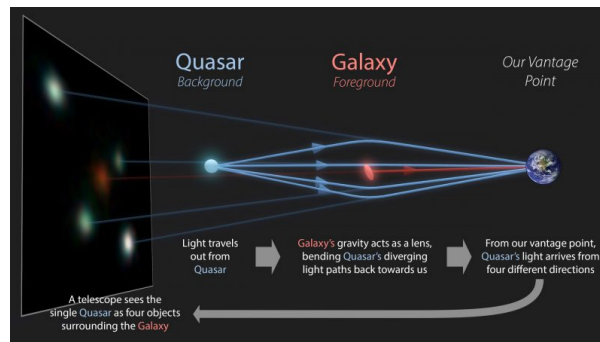


Figure 3.1: gravitational lens^[2]

Different rays of light cross through the lensed galaxy at different points. The amounts of dust and gas in the galaxy vary depending on the position. For instance, there is less

gas on the outskirts of the galaxy than there is at the center of the galaxy. As the rays pass through the galaxy they get obstructed by the gas. This causes a shift in their spectrum. Since each light ray passes through the lens at a different position, they each are interfered with differently by the gas. Each light ray has a different spectrum shift based on how much gas is interfering with it.

From an observer on Earth, we view these bent lights as separate images. We can then take each image's spectrum and analyze its redshift to determine how much dust and gas are interfering with it. Then we can analyze the dust, gas, and metallicities at different points in the galaxy.

Chapter 4

THE RESEARCH

My specific part of this research involved preparing to analyze 10 different observations. The data I used came from the Chandra X-ray telescope. I then loaded these images into the University computer and then cleaned the images up to prepare them to be analyzed. I also prepared all the scripts necessary to run the spectral images. The next step is to run the spectrums for each image. I will discuss the steps I took in the next sections.

4.0.1 OBTAINING THE DATA

The 10 observations that we are analyzing are quasars that are lensed through galaxies. The Chandra X-ray telescope had already imaged these observations. So the first step in my research was gathering the data. I submitted a request to Chandra for the data on these images and they emailed me the files.

Once I had the files I loaded them into the University computer and sorted the files based on their observations. There were several images that they sent for each image. I ended up with a total of 124 images total after I gathered all 10 observations.

4.0.2 REPROCESS

Chandra sent a primary and secondary file (data distributions) for each image. Each folder had an evt2.fits file attached. Evt2.fits files are the physical photos that can be viewed in DS9. I needed to get one clean photo for each image. So I reprocessed the primary and secondary files to get a new clear evt2.fits. "The chandra _repro script takes the data from these data distributions and creates a new bad pixel file, a new level=2 event file, and a new level=2 Type II PHA file (grating data only) along with

the appropriate per-order and per-arm ARF and RMF files.^[3]"

Reprocessing was simple. Chandra has a simple script that does all the work. I just executed the script for each image. Now I had a single repro file for each image with a `repro_evt2.fits` image which combines the primary and secondary image. This is the photo that we want to use for our analysis.

4.0.3 CREATING A PSF IMAGE

Once the repro file was set up I created a `psf.fits` inside each repro file. Psf stands for point spread function. A point spread function is the infinite-resolution and infinite-signal-to-noise flux distribution from a point source on the detector, after passing through optics, dust, atmosphere, etc.^[4]. So it is important for us to apply this to our images.

"Chandra produces sharper images than any other X-ray telescope to date; and therefore, provides an opportunity for high-angular and spectral resolution studies of X-ray sources. Crucial to these studies is the knowledge of the characteristics of the PSF. The observed Chandra PSF is smeared with the blur introduced to the High-Resolution Mirror Assembly (HRMA) PSF by a combination of the telescope dithering motion, the limited size of detector pixels, and detector effects.^[4] The blurring of the Chandra PSF is introduced by the HRMA PSF, the aspect, the limited size of detector pixels and detector effects. Simulating the HRMA PSF using ChaRT is the first step in obtaining a good model of the Chandra PSF for a given observation. The shape and size of the HRMA PSFs vary significantly with source location in the telescope field of view (FOV), as well as with the spectral energy distribution of the source. The image quality is best in a small area centered about the optical axis. In fact, the mirrors were designed to produce images with better than one arc-second resolution; in particular to concentrate better than 85% of the energy at 0.277 keV within a 1 arcsec diameter^[3]"

4.0.4 CREATING A FULL, HARD, AND SOFT IMAGE

Once I applied the psf to each repro file I then needed to create images specifically focused on different energies. Chandra recommends that we filter the event files on energy before beginning the analysis. The energy ranges that I used were specifically given to me by Dr. Dai. I created images, from the evt2.fits image, that focuses on energies ranging from 200-2000, another from 2000-8000, and one from 200-8000. I created new fits for each one of these energy ranges using the 'dmcopy' command. This command creates a new evt2 file that only includes data in the specified energy range. The fits with energies from 200-2000 is labeled "soft," the fits with energies ranging from 2000-8000 is "hard," and the fits with the full range is labeled "full." These will be used in analyzing the shift.^[5]

4.0.5 CREATING SHERPA TEXTS

Once the full, soft, and hard photos of all 124 images were complete I was ready to run Sherpa. Sherpa is a command through a program called Ciao that pulls the spectrum of the images. Ciao is Chandra's software used to analyze the images.

Sherpa is designed to do several things with your data. "Sherpa is the CIAO modeling and fitting application. It enables the user to construct complex models from simple definitions and fit those models to data, using a variety of statistics and optimization methods. It lets you

- fit 1-D data sets (simultaneously or individually), including: spectra, surface brightness profiles, light curves, general ASCII arrays
- fit 2-D images/surfaces in the Poisson/Gaussian regime
- visualize the data with Matplotlib and DS9
- visualize the data with Matplotlib and DS9
- access the internal data arrays
- build complex model expressions

- import and use your own models
- choose appropriate statistics for modeling Poisson or Gaussian data
- import new statistics, with priors if required by analysis
- visualize a parameter space with simulations or using 1-D/2-D cuts of the parameter space
- calculate confidence levels on the best-fit model parameters
- choose a robust optimization method for the fit: Levenberg-Marquardt, Nelder-Mead Simplex or Monte Carlo/Differential Evolution
- perform Bayesian analysis with Poisson Likelihood and priors, using Metropolis or Metropolis-Hastings algorithm in the MCMC (Markov-Chain Monte Carlo)
- and use Python to create complex analysis and modeling functions, build the batch mode analysis or extend the provided functionality to meet the required needs." [4]

I want to use Sherpa to fit 1-D image for Spectra, to visualize the data with Matplotlib, and to use Python to create complex analysis and modeling functions.

We ran into some issues with Ciao during the semester. The University needed to update Ciao for me to be able to run my script. It took the University longer than expected to update the software and get me access. In the meantime, I created the scripts needed to run the spectroscopy. I want to run spectroscopy for each of the evt2.fits files, hard, soft, and full. I created a text file for each one of these and labeled it accordingly, Sherpa _full.txt, Sherpa _soft.txt, Sherpa _hard.txt

Chapter 5

NEXT STEPS

Once the University gets Ciao updated and gives me permission to access it, I can start running spectrum for all the images and then analyze the information. The Sherpa scripts are all ready, it is a matter of copying and passing them into Ciao. Although I have not run Ciao before, I did ask a grad student who has Ciao on his personal computer to run an example of the program. I have put that example in Appendix A so you can see what the spectrum should look like.

Chapter 6

CONCLUSION

I have organized the files so that, in the event I am unable to complete my research before graduation, any upcoming undergraduate can copy the files into their own log in and be able to pick up where I left off. I hope that the University can get ciao operational in the next week or so. This will allow me one more semester to try and complete my work and analysis. Either way, all of the data is organized and ready for analysis.

Appendix A

Q _1355 SPECTRUM

```
In [10]: from sherpa.astro.ui import * #import sherpa function
         from ciao_contrib.runtool import * #import CIAO tools
         import matplotlib.pyplot as plt
```

```
In [4]: #we can define the source and bkg files to be used by specextract:
specextract.infile = "../acisf09514_repro_evt2.fits[sky=region(srcA.reg)]"
specextract.bkgfile = "../acisf09514_repro_evt2.fits[sky=region(bkg.reg)]"
# define the directory to load the outputs
specextract.outroot = "srcA/srcA"
#run the specextract function
specextract()
```

```
Out[4]: Running specextract
Version: 18 January 2023

Aspect solution file ../pcadf09514_000N001_asol1.fits found.

Bad pixel file ../acisf09514_repro_bpix1.fits found.

Mask file ../acisf09514_000N003_msk1.fits found.

Extracting bkg spectra

Extracting src spectra

Using mkacisrmf...

Using mkacisrmf...

Creating bkg RMF

Creating bkg ARF

Creating src RMF

Creating src ARF
```

```
In [5]: #we can define the source and bkg files to be used by specextract:
specextract.infile = "../acisf09514_repro_evt2.fits[sky=region(srcB.reg)]"
specextract.bkgfile = "../acisf09514_repro_evt2.fits[sky=region(bkg.reg)]"
# define the directory to load the outputs
specextract.outroot = "srcB/srcB"
#run the specextract function
specextract()
```

```
Out[5]: Running specextract
Version: 18 January 2023

Aspect solution file ../pcadf09514_000N001_asol1.fits found.

Bad pixel file ../acisf09514_repro_bpix1.fits found.

Mask file ../acisf09514_000N003_msk1.fits found.

Extracting src spectra

Extracting bkg spectra

Using mkacisrmf...

Using mkacisrmf...

Creating src RMF

Creating src ARF

Creating bkg ARF

Creating bkg RMF
```

```
In [7]: load pha(1,"srcA/srcA.pi")
load pha(2,"srcB/srcB.pi")

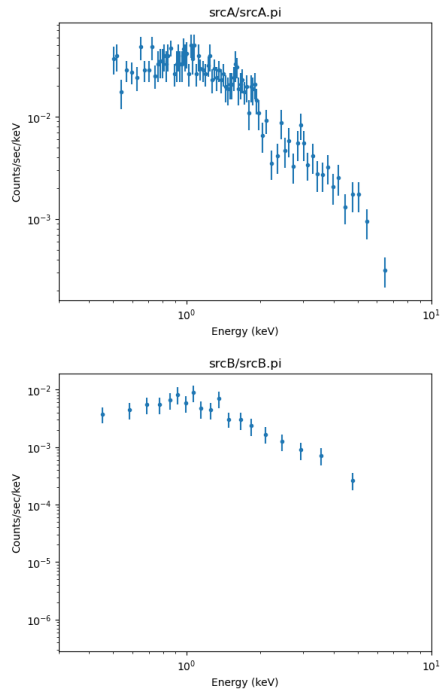
read ARF file srcA/srcA.arf
read RMF file srcA/srcA.rmf
read ARF (background) file srcA/srcA_bkg.arf
read RMF (background) file srcA/srcA_bkg.rmf
read background file srcA/srcA_bkg.pi
read ARF file srcB/srcB.arf
read RMF file srcB/srcB.rmf
read ARF (background) file srcB/srcB_bkg.arf
read RMF (background) file srcB/srcB_bkg.rmf
read background file srcB/srcB_bkg.pi
```

```
In [8]: #we constrain our energy range from 0.5-7 keV
notice(0.5,7)

dataset 1: 0.0073:14.9504 -> 0.4964:7.008 Energy (keV)
dataset 2: 0.0073:14.9504 -> 0.4964:7.008 Energy (keV)
```

```
In [14]: #we now plot the regrouped, background subtracted spectrum of daset 1
plot_data(1,xlog=True,ylog=True)
plt.xlim(0.3,10)
plt.show()

#we do the same for dataset 2
plot_data(2,xlog=True,ylog=True)
plt.xlim(0.3,10)
plt.show()
```



we can now define our models. We will use

- 1 - `xszpowerlw` : redshifted power law photon spectrum. The default parameters are `PhoIndex`, `redshift`, and `norm`.
- 2 - `xszphabs` : photoelectric absorption of lens. Paramaters are `nH` and `redshift`.
- 3 - `xsphabs` : photoelectric absorption from line of sight of observer. Parameters are `nH`.

models from <https://cxc.cfa.harvard.edu/sherpa/models/> (<https://cxc.cfa.harvard.edu/sherpa/models/>)

structure for defining model is `set_source(index,model name.variable model 2.variable 2 etc)`.

By passing the same variable name to multiple sources, we are linking the model (they will be the same value for all).

```
In [15]: set_source(1,xszpowerlw.p1*xszphabs.labs1*xsphabs.gabs)
set_source(2,xszpowerlw.p2*xszphabs.labs2*gabs)
```

we can now redefine some of the parameters to their known/observed value. The redshift of the source and the lens is found in the [CASTLES survey](https://web.cfa.harvard.edu/castles/) (<https://web.cfa.harvard.edu/castles/>).

The H I column density is found using [HEASARC](https://heasarc.gsfc.nasa.gov/cgi-bin/Tools/w3nh/w3nh.pl) (<https://heasarc.gsfc.nasa.gov/cgi-bin/Tools/w3nh/w3nh.pl>).

```
In [16]: #we fix the redshift of the source since it is known.
#we Link it for all models since it is same z for all.
p1.redshift = 1.37
link(p2.redshift,p1.redshift)

#similarly, the redshift of the lens is known.
#we fix it and Link it to all the models.
labs1.redshift = 0.48
link(labs2.redshift,labs1.redshift)

#We fix the galactic nH value.
#Again, it will be the same for all models.
# 5.25 x 10^20 so
gabs.nh = 0.0525 #unit: 10^22 atoms / cm^2
freeze(gabs.nh)

#the photon index is expected to be the same for all sources:
link(p2.PhoIndex, p1.PhoIndex)

#we can look at our models:
show_model()

Model: 1
apply_rmf(apply_arf((31188.838169386 * ((xszpowerlw.p1 * xszphabs.labs1) * xsphabs.gabs))))
Param      Type      Value      Min      Max      Units
-----
p1.PhoIndex thawed      1          -3         10
p1.redshift frozen     1.37      -0.999      10
p1.norm     thawed      1           0      1e+24
labs1.nh    thawed      1           0      1e+06 10^22 atoms / cm^2
labs1.redshift frozen    0.48      -0.999      10
gabs.nh     frozen     0.0525      0      1e+06 10^22 atoms / cm^2

Model: 2
apply_rmf(apply_arf((31188.838169386 * ((xszpowerlw.p2 * xszphabs.labs2) * xsphabs.gabs))))
Param      Type      Value      Min      Max      Units
-----
p2.PhoIndex linked      1      expr: p1.PhoIndex
p2.redshift linked     1.37      expr: p1.redshift
p2.norm     thawed      1           0      1e+24
labs2.nh    thawed      1           0      1e+06 10^22 atoms / cm^2
labs2.redshift linked    0.48      expr: labs1.redshift
gabs.nh     frozen     0.0525      0      1e+06 10^22 atoms / cm^2
```

```
In [17]: #once our parameters are adjusted, we can simultaneously fit all 4 models.
#we fit three times so the fit converges.
fit(1,2)
fit(1,2)
print('-----')
fit(1,2)
```

```
Datasets      = 1, 2
Method        = levmar
Statistic      = chi2gehrels
Initial fit statistic = 4.15646e+11
Final fit statistic = 66.0583 at function evaluation 322
Data points   = 184
Degrees of freedom = 99
Probability [Q-value] = 0.995587
Reduced statistic = 0.667174
Change in statistic = 4.15646e+11
p1.PhoIndex   1.98249 +/- 0.0868248
p1.norm       0.000407501 +/- 5.61505e-05
labs1.nh      0.00202197 +/- 0.0545696
p2.norm       7.71839e-05 +/- 1.13953e-05
labs2.nh      0.0076188 +/- 0.0710819

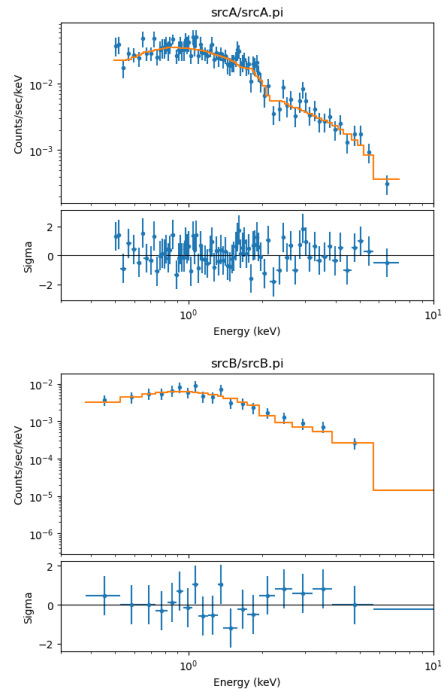
Datasets      = 1, 2
Method        = levmar
Statistic      = chi2gehrels
Initial fit statistic = 66.0583
Final fit statistic = 65.5828 at function evaluation 291
Data points   = 184
Degrees of freedom = 99
Probability [Q-value] = 0.996122
Reduced statistic = 0.662453
Change in statistic = 0.467438
p1.PhoIndex   1.98204 +/- 0.0884685
p1.norm       0.000404742 +/- 5.96491e-05
labs1.nh      0 +/- 0.0541931
p2.norm       7.30239e-05 +/- 1.20793e-05
labs2.nh      0 +/- 0.0698041
WARNING: parameter value labs1.nh is at its minimum boundary 0.0
WARNING: parameter value labs2.nh is at its minimum boundary 0.0

-----
Datasets      = 1, 2
Method        = levmar
Statistic      = chi2gehrels
Initial fit statistic = 65.5828
Final fit statistic = 65.5665 at function evaluation 278
Data points   = 184
Degrees of freedom = 99
Probability [Q-value] = 0.996139
Reduced statistic = 0.662288
Change in statistic = 0.0162759
p1.PhoIndex   1.98221 +/- 0.0883648
p1.norm       0.00040539 +/- 5.96707e-05
labs1.nh      0 +/- 0.0541192
p2.norm       7.30838e-05 +/- 1.20771e-05
labs2.nh      0 +/- 0.0697446
WARNING: parameter value labs1.nh is at its minimum boundary 0.0
WARNING: parameter value labs2.nh is at its minimum boundary 0.0
```



```
In [18]: #we can plot our fit for dataset 1 and include residuals
set_xlog()
set_ylog()
plot_fit_deichi(1)
plt.xlim(0.3,10)
plt.show()

#we can plot our fit for dataset 1 and include residuals
set_xlog()
set_ylog()
plot_fit_deichi(2)
plt.xlim(0.3,10)
plt.show()
```



```
In [19]: #we estimate the parameter confidence intervals
conf(1,2)
```

```
labs1.nH lower bound:  -----
labs2.nH lower bound:  -----
p1.norm lower bound:   -2.00181e-05
labs2.nH upper bound:   0.049471
p1.norm upper bound:    4.90178e-05
labs1.nH upper bound:   0.0301181
p1.PhIndex lower bound:  -0.0379315
p1.PhIndex upper bound:   0.0766384
p2.norm lower bound:    -5.74022e-06
p2.norm upper bound:    1.08565e-05
Datasets                = 1, 2
Confidence Method        = confidence
Iterative Fit Method     = None
Fitting Method           = levmar
Statistic                 = chi2gehrrels
Confidence 1-sigma (68.2689%) bounds:
Param      Best-Fit  Lower Bound  Upper Bound
-----
p1.PhIndex    1.90221  -0.0379315  0.0766384
p1.norm      0.00040539  -2.00181e-05  4.90178e-05
labs1.nH      0          -----  0.0301181
p2.norm      7.30838e-05  -5.74022e-06  1.08565e-05
labs2.nH      0          -----  0.049471
```

```
In [ ]:
```

BIBLIOGRAPHY

- [1] H. Information@eso.org, “Gravitational lensing,” 2020.
- [2] E. earthsky.com, “What is gravitational lensing?,” Sep 2021.
- [3] C. Chandra, “Create a new level=2 event file - ciao 4.16,” Oct 2023.
- [4] J. Doe 1, “Sherpa: Ciao’s modeling amp; fitting application - ciao 4.16 sherpa,” Dec 2023.
- [5] J. Doe, “Filtering data,” Oct 2023.