A Template for Theses and Dissertations

A dissertation presented to

the faculty of

the College of Arts and Sciences of Ohio University

In partial fulfillment

of the requirements for the degree

Doctor of Philosophy

Bhishan Poudel

2021

This dissertation titled

A Template for Theses and Dissertations

by

BHISHAN POUDEL

has been approved for

the Department of Physics and Astronomy

and the College of Arts and Sciences by

Douglas Clowe

Professor and Director of the Astrophysics Institute

Florenz Plassmann

Dean

Abstract

POUDEL, BHISHAN, Ph.D., 2021, Physics

A Template for Theses and Dissertations (70 pp.)

Director of Dissertation: Douglas Clowe

Insert your abstract here

Dedication

Insert your dedication here.

A double backslash can be used to force line breaks if desired.

Acknowledgments

Insert your acknowledgments here or comment out this line.

Table of Contents

Page

Abstract . . . 3

Dedication . . . 4

Acknowledgments . . . 5

List of Tables . . . 8

List of Figures . . . 9

List of Symbols . . . 10

List of Acronyms . . . 12

1 Introduction . . . 13
  1.1 Introduction . . . 14
    1.1.1 Einstein's Deflection Angle . . . 14
    1.1.2 Lens equation . . . 16
    1.1.3 Convergence κ and deflection potential ψ . . . 18
    1.1.4 Multiple Images . . . 19
    1.1.5 Magnification μ and shear γ . . . 21
    1.1.6 Examples of Gravitational Lensing . . . 24

2 Bulge Disk Decomposition . . . 25
  2.1 Bulge Disk Decomposition . . . 26

3 Generating Point Spread Function (PSF) Files . . . 34
  3.1 Creating PSF Using SED file . . . 35
    3.1.1 SED File For PhoSim . . . 35
    3.1.2 Background File for PhoSim . . . 36
    3.1.3 Instance Catalog For PhoSim . . . 37
    3.1.4 PSF Generation . . . 39

4 Galaxy Simulation . . . 40
    4.0.1 Creation of Scaled Bulge, Disk, and Monochromatic Images . . . 41
    4.0.2 PSF Creation for Bulge, Disk, and Monochromatic Images . . . 44
    4.0.3 Jedisim Simulations . . . 47
      4.0.3.1 Create the Catalogs for Jedisim . . . 47
      4.0.3.2 Transform the Galaxies . . . 49

4.0.3.3 Distort the Galaxies 50
4.0.3.4 Convolve the Galaxies with the PSF 50
4.0.3.5 Rescale the Galaxies from HST to LSST 50
4.0.3.6 Create Monochromatic LSST Image 51
4.0.3.7 Create Chromatic LSST Image 51
4.0.3.8 Rotated galaxies output from Jedisim 51

5 Star Creation 52
5.0.1 Create text file for star positions 53
5.0.2 Create star.fits file from text file 54
5.0.3 Convolve and re-scale star.fits with psf bdm files 54
5.0.4 Add stars to jedisim outputs 54

6 DMSTACK ANALYSIS 57
6.1 Introduction to LSST Pipeline 58
6.2 Preparation of Data for dmstack 61
6.3 Cleaning of dmstack outputs 61
6.3.1 Merge 4 catalogs into one catalog using IMCAT 63
6.3.2 Reduced Shear for monochromatic and chromatic 64

7 DMSTACK ANALYSIS 68

References 69

Appendix: An Appendix 70

List of Tables

Table Page

2.1 Tiny Tim Parameters 33

List of Figures

Figure Page

1.1 Simple sketch to gravitational lensing. 14
1.2 Solar deflection angle 16
1.3 Multiple images of single source 19
1.4 Gravitational Lensing in Abell 1689 24

3.1 Flat SED 36

6.1 Squares of reduced shears 65
6.2 Squares of reduced shears 66
6.3 gsq vs gmdsq 67

List of Symbols

z - Redshift

H_0 - Hubble Constant

h - Dimensional Hubble Parameter

Λ - Cosmological Constant

Ω - Density Parameter

ρ_c - Critical Density

D_L - Luminosity Distance

σ_v - Dispersion Velocity

N'' - N Arcseconds

r_s - Scale Radius

r_{200} - Virial Radius

M_{200} - Virial Mass

Mpc - Mega Parsecs

α - Einsteins's Deflection Angle

ψ - Deflection Potential

θ_E - Einstein Angle

β - True Angular Position of the Source

θ - Observed Position of the Source

ξ - Impact Parameter

μ - Magnification

e_T - Tangential Ellipticity

e_X - Cross Ellipticity

γ - Shear

g - Reduced Shear

κ - Convergence

Σ_c - Critical Surface Mass Density

q - Ratio of Minor Axis to Major Axis of an Ellipse i.e. b/a

I - Intensity

R_e - Radius of an Isophote Containing the Half Of the Total Luminosity

List of Acronyms

ACS	Advanced Camera for Surveys
CCD	Charge-coupled Device
CDM	Cold Dark Matter
CMB	Cosmic Microwave Background
DE	Dark Energy
DM	Dark Matter
DMstack	Data Management stack pipeline of LSST
FWHM	Full Width Half Maximum
GR	General Relativity
HST	Hubble Space Telescope
IMCAT	Image and Catalog Manipulation Software
LSS	Large Scale Structure
LSST	Large Synoptic Survey Telescope
PSF	Point Spread Function
SED	Spectral Energy Distribution
UDF	Ultra Deep Field
WCS	World Coordinate System
WFC3	Wide Field Camera 3 of LSST
WFIRST	Wide-Field Infra-red Survey Telescope

1 Introduction

1.1 Introduction

Gravitational Lensing provides us a way to see how dark matter along with visible matter is distributed in the universe. This theory of gravitational lensing is supported by Einstein's general theory of relativity which predicts the deflection of light in a gravitational field if any massive object is present there ([Har03]).

1.1.1 Einstein's Deflection Angle

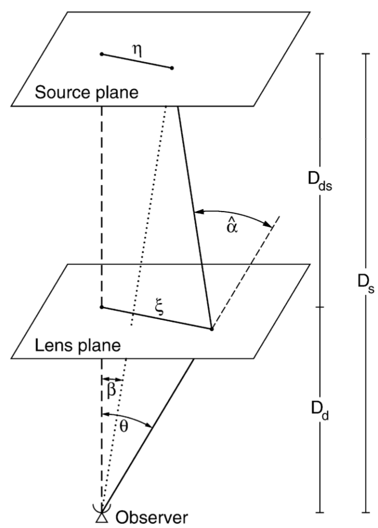


Figure 1.1: Simple sketch to gravitational lensing [BS01]

General Relativity predicts that when the beam of light passes through near the massive objects, the rays of light undergo deflection from their original path. This angle of deflection was predicted by Einstein's theory of General Relativity. According to this theory if the massive object of mass M is located at a perpendicular distance ξ (called **impact parameter**) from the line of sight of source and the observer then the deflection caused by that mass is given by [Sch07]

$$\hat{\alpha} = \frac{4GM}{c^2\xi}. \tag{1.1}$$

This equation (1.1) is valid only when the angle $\hat{\alpha} << 1$. In case of gravitational lensing, the product of mass of the deflector and the Gravitational Constant is always the much smaller than the squared of velocity of light, thus making the deflection angle very small.

For quantitative purpose, we can calculate the value of deflection angle for distant stars appearing near to the solar limb by setting the mass $M = M_{\odot}$ and radius $R = R_{\odot}$ in the above equation to obtain the angle of deflection $\hat{\alpha} = 1.74''$. This value was tested in famous solar eclipse experiment in May 29, 1991 which was conceived by Sir Frank Watson Dyson, Astronomer Royal of Britain in 1917 and led by Sir Arthur Stanley Eddington two years later in 1919 ([DED20]). The experiment was designed to test following hypotheses:

- The light path is uninfluenced by gravitation.

- The law of gravitation will follow Newtonian law and will produce 0".87 apparent displacement.
  The classical calculation for deflection angle was devised by Soldner in 1801 using Newtonian theory:
  $$\frac{2GM_{\odot}}{v^2 R_{\odot}} \tag{1.2}$$
  where Solar mass $M_{\odot} = 1.989 * 10^{30} kg$ and Solar radius $R_{\odot} = 6.96 * 10^{8}\ m$ and plugging the values gives deflection of 0.875 arc seconds.

- The course of ray of light will follow Einsteins generalized relativity and lead to apparent displacement of 1.74". Note: in 1915 Einstein used general relativity and derived a formula for gravitational deflection as
  $$\frac{4GM_{\odot}}{c^2 R_{\odot}} \tag{1.3}$$

The experiment concluded that the results were close to the Einsteins predicted deflection angle and thus supporting the gravitational lensing theory.

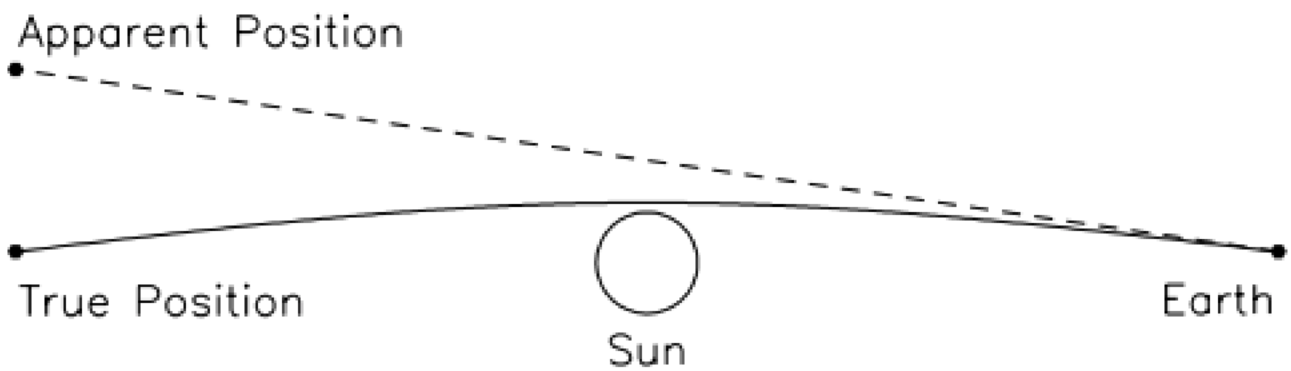


Figure 1.2: Solar deflection angle

This was the deflection due to point source. We can also define the deflection for the extended mass with certain surface mass density Σ as

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = \frac{4G}{c^2} \int d^2\boldsymbol{\xi\prime} \quad \Sigma(\boldsymbol{\xi\prime}) \quad \frac{\boldsymbol{\xi} - \boldsymbol{\xi\prime}}{|\boldsymbol{\xi} - \boldsymbol{\xi\prime}|^2} \tag{1.4}$$

Here, the *surface mass density* $\boldsymbol{\Sigma}$ is defined as,

$$\boxed{\Sigma(\boldsymbol{\xi}) \equiv \int dr_3 \; \rho(\xi_1, \xi_2, r_3)} \quad \text{(Surface Mass Density)} \tag{1.5}$$

which is the mass density projected onto the lens plane, perpendicular to the light ray, and r_3 is the coordinate along the line of sight, and ξ_1, ξ_2 the other two perpendicular coordinates.

1.1.2 Lens equation

We consider a simplistic lensing model in which lens and the source objects are point objects. The observer observes the rays of light from the source at a distance

D_s which pass through the gravitational influence field of a massive object of mass M and distance D_d located perpendicularly $\boldsymbol{\xi}$ distance away from line of sight (also called impact parameter).

Let $\boldsymbol{\eta}$ denotes the true, two-dimensional position of the source in the source plane and $\boldsymbol{\beta}$ is the true angular position of the source. This means in absence of light deflection we would have,

$$\boldsymbol{\beta} = \frac{\boldsymbol{\eta}}{D_s}. \tag{1.6}$$

The relation between position $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$ is given by,

$$\boldsymbol{\theta} = \frac{\boldsymbol{\xi}}{D_d}. \tag{1.7}$$

This means $\boldsymbol{\theta}$ is the observed position of the source on the sphere relative to the position of center of the lens which is the origin of the coordinate system with $\boldsymbol{\xi} = 0$. Where, D_{ds} is the distance of the source plane from the lens plane.

Here we adopt the relation,

$$D_{ds} = D_s - D_d. \tag{1.8}$$

This relation holds true as long as the relevant distances are much smaller than the radius of the universe (c/H_0) and this is always the case for distances within our Galaxy and in Local Group. However, this relation no longer holds true for cosmological distances between source and lenses.

From the figure 1.1 , we can relate $\boldsymbol{\eta}$ with deflection angle $\boldsymbol{\alpha}$ as

$$\boldsymbol{\eta} = \frac{D_s}{D_d}\boldsymbol{\xi} - D_{ds}\hat{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{1.9}$$

Using equation (1.6) we can write $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \boldsymbol{\theta} - \frac{D_{ds}}{D_s}\ \hat{\boldsymbol{\alpha}}(D_d\boldsymbol{\theta}). \tag{1.10}$$

In this equation (1.10) we have some factor multiplying the deflection angle, so we define *reduced deflection angle*

$$\boxed{\boldsymbol{\alpha}(\boldsymbol{\theta}) = \frac{D_{ds}}{D_s}\,\hat{\boldsymbol{\alpha}}(D_d\boldsymbol{\theta})} \quad \text{(Reduced Deflection Angle)} \tag{1.11}$$

and then rewrite the lens equation (1.10) as

$$\boldsymbol{\beta} = \boldsymbol{\theta} - \boldsymbol{\alpha}(\boldsymbol{\theta}). \tag{1.12}$$

For a point mass object, using the equations (1.4) and (1.7) the equation of reduced deflection angle (1.11) becomes

$$\boldsymbol{\alpha}(\boldsymbol{\theta}) = \frac{D_{ds}}{D_s}\,\frac{4GM}{c^2 D_d \boldsymbol{\theta}}. \tag{1.13}$$

1.1.3 Convergence κ and deflection potential ψ

Then convergence is defined as the ratio of the surface mass density of the lens and the critical surface mass density as like

$$\boxed{\kappa(\boldsymbol{\theta}) \equiv \frac{\Sigma(D_d\boldsymbol{\theta})}{\Sigma_{cr}}} \quad \text{(Convergence).} \tag{1.14}$$

Where, the critical surface mass density is given by

$$\Sigma_{cr} = \frac{c^2}{4\pi G}\frac{D_s}{D_d D_{ds}}. \tag{1.15}$$

The critical density depends on the redshift of source and lens. If the convergence $\kappa \geq 1$ i.e. surface density is less than critical surface density then we can see the multiple images of the source. If the value of κ is very large it is called “strong gravitational lensing” and if it is only slightly greater than one, it is called “weak gravitational lensing”. So, the value of critical mass density plays the role in distinguishing weak vs. strong gravitational lensing.

Moreover, we can define scaled deflection angle in terms of convergence as

$$\boldsymbol{\alpha}(\boldsymbol{\theta}) = \frac{1}{\pi} \int d^2\theta\prime \ \kappa(\boldsymbol{\theta}\prime) \ \frac{\boldsymbol{\theta} - \boldsymbol{\theta}\prime}{\boldsymbol{\theta} - \boldsymbol{\theta}\prime^2} \ . \tag{1.16}$$

and the *deflection potential* can be defined as

$$\psi(\boldsymbol{\theta}) = \frac{1}{\pi} \int d^2\theta\prime \ \kappa(\boldsymbol{\theta}\prime) \ ln\boldsymbol{\theta} - \boldsymbol{\theta}\prime. \tag{1.17}$$

1.1.4 Multiple Images

We can see the multiple images of the source at different places $\boldsymbol{\theta}_i$ if the equation (1.12) holds true for different values of the deflection angles.

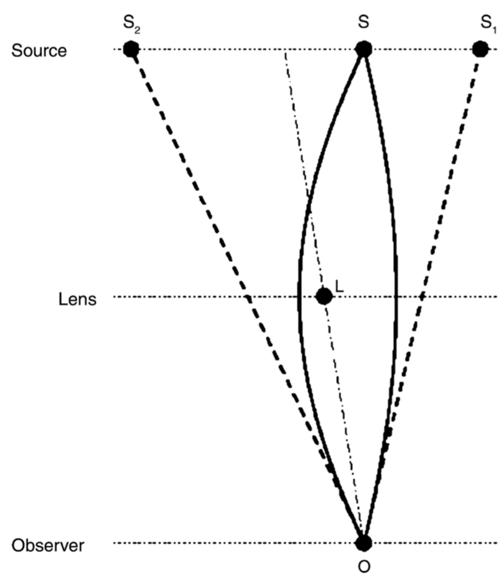


Figure 1.3: Multiple images of single source [Sch07]

This figure illustrates a typical situation in which there are two images S_1 and S_2 of the single source S which is lensed by the point massive object L.

Here, if we take direction of deflection angle as pointing towards the source, we can write the deflection angle for the point mass (1.13) as

$$\boxed{\boldsymbol{\alpha}(\boldsymbol{\theta}) \equiv \frac{4GM}{c^2}\ \frac{D_{ds}}{D_s D_d}\ \frac{\boldsymbol{\theta}}{\boldsymbol{\theta}^2}} \quad \text{(Reduced Deflection Angle)} \tag{1.18}$$

Now, we define *Einstein angle* as

$$\boxed{\theta_E = \sqrt{\frac{4GM}{c^2}\ \frac{D_{ds}}{D_s D_d}}} \quad \text{(Einstein Angle).} \tag{1.19}$$

Then, we can rewrite the equation (1.18) as

$$\boldsymbol{\beta} = \boldsymbol{\theta} - {\theta_E}^2\ \frac{\boldsymbol{\theta}}{\boldsymbol{\theta}^2} \tag{1.20}$$

To solve this equation (1.20) we define two scaling factors

$$y = \frac{\boldsymbol{\beta}}{\theta_E}\ ;\ x = \frac{\boldsymbol{\theta}}{\theta_E} \tag{1.21}$$

Then we get

$$\mathbf{y} = \mathbf{x} - \frac{\mathbf{x}}{\mathbf{x}^2}. \tag{1.22}$$

This is a quadratic equation and the solutions are given by,

$$\mathbf{x} = \frac{1}{2}(\mathbf{y} \pm \sqrt{4 + \mathbf{y}^2})\ \frac{\mathbf{y}}{\mathbf{y}}. \tag{1.23}$$

Following information can be drawn from the solution of lens equation:

- Except for the divergence condition $\theta \to 0$ any source at position y has two images.
- The two images are on the opposite sides of the source position when viewed from observer position.
- If the source is exactly behind the lens ($\mathbf{y} = 0$), we see the circular ring called *Einstein ring.*
- The Einstein ring has the angular diameter $2\theta_E$ and it gives the characteristic images separation.

1.1.5 Magnification μ and shear γ

The light rays in gravitational are not bent uniformly, the ones near to the lens are deviated more and the ones that are farther are bent in smaller proportion. This differential deflection gives rise to the distorted and magnified images of the source object.

Let $\mathbf{I}^s(\boldsymbol{\beta})$ be the surface-brightness distribution of the source, then following the conservation of total surface brightness the observed surface-brightness distribution in the lens plane is given by

$$\boldsymbol{I}(\boldsymbol{\theta}) = \mathbf{I}^s\ [\boldsymbol{\beta}(\boldsymbol{\theta})]. \tag{1.24}$$

Now we expand the *true angular position* of the source $\boldsymbol{\beta}$ in terms of *observed angular position* of the source $\boldsymbol{\theta}$ using Taylor expansion around the central observed position $\boldsymbol{\theta_0}$ we get

$$\boldsymbol{\beta}(\boldsymbol{\theta}) = \boldsymbol{\beta}_0 + (\boldsymbol{\theta} - \boldsymbol{\theta}_0)\frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\theta}} \tag{1.25}$$

Here, the term differential of $\boldsymbol{\beta}$ w.r.t. $\boldsymbol{\theta}$ is called *distortion matrix*

$$\boxed{A(\boldsymbol{\theta}) \equiv \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\theta}}} \quad \text{(Distortion Matrix).} \tag{1.26}$$

Now we can write the observed surface brightness $\boldsymbol{I}(\boldsymbol{\theta})$ in terms of distortion matrix as

$$\boxed{\boldsymbol{I}(\boldsymbol{\theta}) = \boldsymbol{I}^s[\beta_0 + A(\boldsymbol{\theta})(\theta - \theta_0)]} \quad \text{(Observed Surface Brightness)} \tag{1.27}$$

In terms of deflection potential ψ *Jacobian matrix* A can be written as

$$A(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\theta}} \tag{1.28}$$

$$= (\delta_{ij} - \frac{\partial^2 \psi(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j})$$

$$= \begin{pmatrix} 1 - \psi, 11 & -\psi_{,12} \\ -\psi, 21 & 1 - \psi_{,22} \ . \end{pmatrix} \tag{1.29}$$

From these deflection potential terms, we define shear components

$$\gamma_1 = \frac{1}{2}(\psi_{,11} + \psi_{,22}) \tag{1.30}$$

$$\gamma_2 = \psi_{,12} \tag{1.31}$$

Here, the two components tensor γ is called *shear* and is given by

$$\boxed{\gamma \equiv \gamma_1 + i\gamma_2 = |\gamma| e^{2i\phi}} \quad \text{(Shear)} \tag{1.32}$$

Here, γ_1 and γ_2 are two components of shear as given in equation (1.30) and ϕ is the phase angle.

The shear has two components γ_1 and γ_1 which can be expressed as

$$\gamma \equiv \gamma_1 + i\gamma_2 = |\gamma| e^{2i\phi}. \tag{1.33}$$

Also, in terms of deflection potential the shear components can be expressed as

$$\gamma_1 = \frac{1}{2}(\psi_{,11} - \psi_{,22}) \tag{1.34}$$

$$\gamma_2 = \psi_{,12}$$

Also, the convergence κ is related to the deflection potential through Poisson equation

$$\nabla^2 \psi(\theta) = 2\kappa(\theta). \tag{1.35}$$

In terms of matrix elements of deflection potential ψ we can write κ as

$$\boxed{\kappa = \frac{1}{2}(\psi_{,11} + \psi_{,22})} \quad \text{(Convergence)} \tag{1.36}$$

Now we can write the distortion matrix A in terms of shear components γ_1 and γ_2 and convergence κ as

$$A(\boldsymbol{\theta}) = \begin{pmatrix} 1-\kappa-\gamma_1 & -\gamma_2 \\ -\gamma_2 & 1-\kappa+\gamma_1 \end{pmatrix} \tag{1.37}$$

If we solve this matrix we get the eigenvalues

$$eig(A) = 1 - \kappa \pm |g|. \tag{1.38}$$

The equation 1.27 has the shape of an ellipse. This means that if a circular galaxy is lensed then the resulting observed image will be an ellipse. The semi-major and semi-minor axes of the ellipse are given by

$$a = \frac{R}{1-\kappa-|g|} \tag{1.39}$$

$$b = \frac{R}{1-\kappa+|g|} \tag{1.40}$$

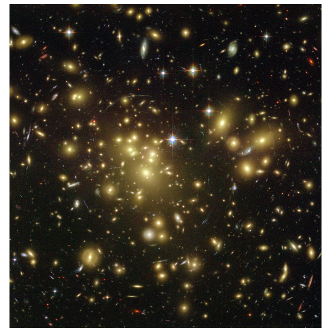
Here, R is the radius of circular source, a is the semi-major axis, b is semi-minor axis, and g is the reduced shear defined as

$$\boxed{g(\boldsymbol{\theta}) \equiv \frac{\gamma(\boldsymbol{\theta})}{1-\kappa(\boldsymbol{\theta})}} \quad \text{(Reduced Shear).} \tag{1.41}$$

The inverse of the determinant of the matrix α gives the magnification tensor. Then we define magnification as

$$\boxed{\mu(\boldsymbol{\theta}) \equiv \frac{1}{det|A|}} \quad \text{(Magnification).} \tag{1.42}$$

(a) Einsteins ring (arc)



(b) Lensing

Figure 1.4: Gravitational Lensing in Abell 1689

1.1.6 Examples of Gravitational Lensing

The gravitational lensing phenomenon has been observed in numerous astrophysical studies. For example here we can see the famous arc of Abell 1689 cluster.

This example shows the gravitational due to one of the most massive galaxy clusters ever known called as “Abell 1689”. These close up images shows Einstein ring (or arc) due to strong lensing (left image) and lensed and magnified galaxies (right image). The cluster galaxies (yellow-white objects) are in Abell cluster and at distance 2.2 billion light-years away. The blue arcs are the distorted images of background galaxies which are again billions of light-years from Abell. The image source is NASA, N. Benitez, et al.

2 Bulge Disk Decomposition

2.1 Bulge Disk Decomposition

We have 201 monochromatic postage stamps of single galaxies without neighbors obtained from Hubble Space Telescope (HST) Deep Field (HDF) [WBD+96]. From these galaxy stamps we want to break each galaxies into bulge and disk parts. To do such a bulge disk decomposition, we use a software called *Galfit* [1] . The **Galfit** is an image analysis algorithm that can model profiles of any given astronomical objects in the given fits image of the data sample. For example, if an fits image contains a galaxy, we can use galfit to fit the bulge and disk components to that galaxy.

Here, for the bulge part of the galaxy, we use the de Vaucouleurs profile in Galfit. The de Vaucouleurs profile describes how the surface brightness of a giant elliptical galaxy changes as a function of the radius R from the center of the galaxy.

Let R_e be the radius of an isophote containing the half of the total luminosity for a galaxy, then, for a de Voucouleurs profile, the surface brightness enclosed by the radius R in that galaxy is given by:

$$I(R) = I_e e^{-7.669[(R/R_e)^{1/4}-1]} \tag{2.1}$$

Galfit has this formula as a built-in feature to get the bulge component of the galaxy.

Similarly, to fit the disk profile to a galaxy, we use the profile called *exponential disk profile* in Galfit. This exponential disk profile is a special case of Sersic profile. In the Sersic profile, the total surface brightness enclosed by the radius R around the center of a galaxy is given by

[1] https://users.obs.carnegiescience.edu/peng/work/galfit/galfit.html

$$ln\ I(R) = ln\ I_0 - kR^{1/n} \tag{2.2}$$

where, I_0 is the central surface brightness at $R = 0$ and the parameter n is called *Sersic index* which determines the curviness of the Sersic profile and the parameter k determines the slope of the curve.

For the most of the spiral galaxies and dwarf elliptical galaxies, the Sersic index is close to 1. This case when the Sersic index n is equal to 1 is called exponential disk profile. In exponential disk profile, the surface brightness inside the radius R is given by

$$ln\ I(R) = ln\ I_0 - kR \tag{2.3}$$

In this project, we have 201 base galaxies obtained from the HST survey. Also, we know that some of the galaxies have both bulge and disk components and some of them do not have one of the components. Therefore, when a galaxy has both bulge and disk components, the Galfit gives nice bulge and disk components, however, when the base galaxy itself does not have reasonable bulge and disk components, the Galfit can not give bulge and disk components of that galaxy. In that case, either Galfit fails to give two components or it gives bad parameter. We can see that whether the fitted parameters are good or bad in the log file created from the Galfit. If a parameter is enclosed by asterisk sign then, we can not trust the fitting, and we assume that the given galaxy does not have reasonable bulge-disk components.

Let's say we get two successful bulge and disk components of a base galaxy, then it is easy to choose between bulge and disk images, we choose *devauc profile* as the bulge and *exponential profile* as the disk image. On the other hand, sometimes the Galfit fails to give two clear components and we have to make some assumptions. The failure happens in two ways, either the Galfit fails to run and gives none of the images, or it gives two images but the log parameters are unreliable and we can not trust the output components. In the case of failure of two distinct outputs, instead of two component fitting, we do two separate one component fitting of *devauc profile* and *expdisk profile*. Then we make some criteria to choose bulge and disk components. If the single component fitting of *devauc profile* gives the better fit parameters than single component fitting of *expdisk profile*, we choose the base galaxy as the bulge component and we choose an empty image as the disk component. Similarly, if the single component *expdisk profile* gives better fitting than the single component *devauc* fitting, then we choose the base galaxy as the disk component and we choose an empty image as the bulge component. We should also notice that, while doing the galaxy fitting to our 201 base galaxies, we found that most of the failed galaxies gave better fit for the *devauc profile* than the *expdisk profile* and for the non-distinct bulge disk component cases have far more devauc components than exponential components. To run the Galfit with two components (devauc expdisk) or single component (either devauc or expdisk) we need to use a input parameter text file. The example of input parameter file is given in next page.

```
# Galfit Basics:
# command: galfit INPUT_PARAMETER_FILE
# outputs: a) fit.log, galfit.01
#          b) imgbolock.fits
#
# a) fit.log ==>  appended each time e.g. galfit.01, galfit.02
# b) imgblock.fits ==> it has 4 frames.
#                      0 is empty, 1 is input, 2 is model , 3 is residual.
#
#  command: ds9 -multiframe imgblock.fits
#  output : we can see 4 frames.
#
#
# command: galfit -o3 galfit.01 && rm -r galfit.01
# output : a) subcomps.fits
#
# a) subcomps.fits ==> it has two or more frames.
#                     0 is subcomps.fits, 1 is expdisk, 2 is devauc etc.
#
# INPUT_PARAMETER_FILE for galfit has two components:
# a) CONTROL PARAMETERS:  A-P   (these are compulsory)
# b) OBJECT PARAMETERS: 0-10 Z  (it should be at least one, e.g. devauc)
#
# a) CONTROL PARAMETERS
#        * These are fixed, not initial guesses.
#        * F: The row F is for masking
```

```
#           ic '1 0 %1 0 == ?'  INPUT_GALAXY  > mask.fits
#           If the bad pixel input file is a FITS image, all non-zero valued
#           pixels would be ignored during the fit. The pixel numbers where
#           value is 0 is only fitted.
#        * E: psf fine sampling factor is 2
#        * K: in my case plate scale is 0.03
#
# b) OBJECT PARAMETERS:
#        * These are initial guesses.
#        * Second column 1 means value not-fixed.
#        * Z: 0 fits the model, do not choose 1 while fitting.
#        * Better initial guess makes the simulation faster.
#          e.g. for f606w_gal0.fits  (from ds9 headers)
#          NAXIS1  =  601
#          NAXIS2  =  601
#
#          MAG     =  20.5874
#          RADIUS  =  17.703
#
#          PIXSCALE=  0.03 or 0.06 (first for 0-100,including)
#          MAG0    =  30 or 26.6611 or 26.78212 (first hundred, next f606, f814)
#
#   Main commands : ic '1 0 %1 0 == ?'  INPUT_GALAXY  > mask.fits
#                 rm -r imgblock.fits subcomps.fit ; galfit three_comps.sh
#                 galfit -o3 galfit.01 && rm -r galfit.01
#                 ds9 -cmap a -scale log -multiframe imgblock.fits subcomps.fits &
```

```
#
# ic '1 0 %1 0 == ?'  /Users/poudel/jedisim/simdatabase/galaxies/
#                         f606w_gal107.fits  > mask.fits
# ds9 mask.fits /Users/poudel/jedisim/simdatabase/galaxies/f606w_gal107.fits &

# IMAGE and GALFIT CONTROL PARAMETERS
A) /Users/poudel/jedisim/simdatabase/galaxies_original/f814w_gal301.fits
B) imgblock.fits       # Output data image block
C) none                # Sigma image name (made from data if blank or "none")
D) f814w_psf.fits # Input PSF image and (optional) diffusion kernel
E) 2                   # PSF fine sampling factor relative to data
F) mask.fits           # Bad pixel mask (FITS image or ASCII coord list)
G) none                # File with parameter constraints (ASCII file)
H) 1 601 1 601         # Image region to fit (xmin xmax ymin ymax)
I) 200  200            # Size of the convolution box (x y)
J) 26.78212 # Magnitude photometric zeropoint
K) 0.06 0.06 # Plate scale (dx dy)    [arcsec per pixel]
O) regular             # Display type (regular, curses, both)
P) 0                   # Choose: 0=optimize, 1=model, 2=imgblock, 3=subcomps

# IMAGE and GALFIT OBJECT PARAMETERS
# Component number: 1
# Exponential function (concentration index n = 1)
# This gives disk profile.
0) expdisk            # Object type
1) 300.0 303.0 1 1         # position x, y        [pixel]
```

```
3) 22.1819 1         # total magnitude
4) 14.478 1         #     Rs               [Pixels]
9) 0.5        1       # axis ratio (b/a)
10) 100.0      1      # position angle (PA)  [Degrees: Up=0, Left=90]
Z) 0                  #  Skip this model in output image?  (yes=1, no=0)


# Component number: 2
# deVaucouleur function (concentration index n = 4)
# This gives the bulge profile.
0) devauc             # Object type
1) 300.0 303.0 1 1         # position x, y        [pixel]
3) 22.1819 1         # total magnitude
4) 14.478 1         #     R_e              [Pixels]
9) 0.5        1       # axis ratio (b/a)
10) 100.0       1     # position angle (PA)  [Degrees: Up=0, Left=90]
Z) 0                  #  Skip this model in output image?  (yes=1, no=0)
```

Another point to note is that, to use the Galfit program, we need to use a PSF image to convolve the original galaxy with the given PSF. The PSF image is different for different filter images of a given galaxy. That is, for F814W filter of a base galaxy $galaxy\_f814w\_000.fits$ we use the PSF $psf\_f814w.fits$ and for the galaxy $galaxy\_f606w\_000.fits$ we use the PSF $psf\_f606w.fits$. For all the F814W filter images we use the same $psf\_f814w.fits$ and for all the F606W filter images we use the same $psf\_f066w.fits$ and we have 201 base galaxies ($galaxy\_f814w\_000.fits$ to $galaxy\_f814w\_200.fits$).

Here, in our project we use the F814W filter images taken from the HST ACS Wide Field Channel Camera. To create the relevant PSF, we use an on-line tool called *STScI TinyTim Web Application* [2].

The TinyTim web application needs some parameters to create a PSF. For this project we chose the following parameters:

Table 2.1: Tiny Tim Parameters

Camera	ACS - Wide Field Channel
Chip	1
Pixel Position	301 301
Filter	F814w
Spectrumtype	Blackbody
Spectrumvalue	6000
PSF diameter	5.0 arcsec
Focus	0.0

[2] http://www.stsci.edu/hst/observatory/focus/TinyTim

3 Generating Point Spread Function (PSF) Files

3.1 Creating PSF Using SED file

In case of galaxy fitting software **Galfit**, we created the PSF needed using an on-line tool called **TinyTim**. The PSF was specially designed for HST ACS Wide Field Camera observations. Here, we again want to create a general purpose PSF using a flat SED. SED stands for "Spectral Energy Distribution", which is simply a table of flux and wavelength. In our galaxy simulation program **Jedisim**, we use the PSF created by a program called **PhoSim**. PhoSim is a photon simulator application which uses Monte Carlo codes to calculate the physics of the atmosphere and the telescope & camera optics. [3]

To run **PhoSim**, we need the following inputs:

- a SED file
- an instance catalog file
- a background file

3.1.1 SED File For PhoSim

Here, we are using flat SED file that comes with the package and is called *phosim/data/SEDs/sed_flat.txt* The plot of wavelength versus flux of flat SED is given below:

Here, the SED has flux values for wavelengths 300 to 1200 nanometers. We are particularly interested only for the wavelength range of LSST r band filter. Looking at the file $phosim/data/lsst/filter\_2.txt$ and choosing only the range of the filter for transmission $>= 5\%$ we get the range of 531 nm to 696 nm. From now on the range 531 nm to 696 nm will be called the broadband range. We split this broadband into 21 equal parts and call each part a narrowband. For example,

[3] $https://bitbucket.org/phosim/phosim\_release/wiki/Home$

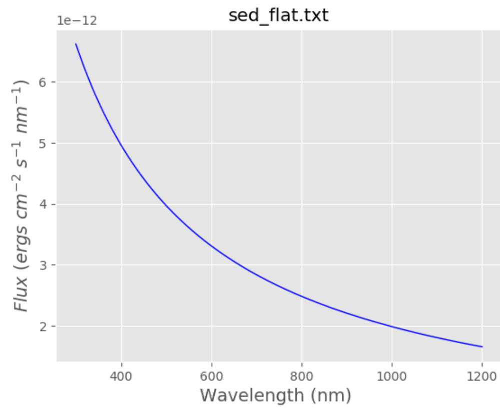


Figure 3.1: Flat SED

narrowband0 is from wavelength 5310 Angstrom to 5388 Angstrom. From these wavelengths we create 21 narrowband SED files and one broadband SED file.

3.1.2 Background File for PhoSim

We have got SED file from SEDs directory recommended by PhoSim, now we need to choose the background. For the background we chose a simple background file as shown below:

```
zenith_v 1000.0

raydensity 0.0

pixelsize 1.5

saturation 0

blooming 0

chargesharing 0
```

Here we the zenith_v is the default parameter provided by the software. We have chosen raydensity as 0.0, this is also the default setting. We have used the pixelsize of 1.5 and removed saturation by using saturation parameters value set to 0. We also avoided blooming and set the parameter to 0. Similarly, the effect of charge sharing between ions is also removed.

3.1.3 Instance Catalog For PhoSim

The third component needed is "instance catalog file". The important parameters in an instance catalog are SIM_SEED, SIM_VISTIME, and object. The output PSF file will depend on the initial random seed to the PhoSim software. For the simplicity, we choose this seed number to be 1000. Similarly we choose the simulation time to be 5 minutes (equal to 300 seconds) as the SIM_VISTIME parameter. The "object" parameter consist of multiple fields. For example, we have used:

```
object 0 0.0 0.0 24 ../../../Research/psf_creation_phosim/scripts/
narrowband_seds/narrowband0.sed 0 0 0 0 0 0 star none none


These fields corresponds to following settings:
object ID RA DEC MAG_NORM SED_NAME REDSHIFT GAMMA1 GAMMA2 KAPPA DELTA_RA DELTA_DEC


The non-trivial components are magnitude of the star, name of the SED file used an
```

[4] An example of instance catalog file can be found in PhoSim software installation directory *phosim/examples/small_catalog*. An example of the instance catalog we used is given below:

```
Unrefracted_RA_deg 0
Unrefracted_Dec_deg 0
Unrefracted_Azimuth 0
```

[4] https://bitbucket.org/phosim/phosim_release/wiki/Instance%20Catalog

```
Unrefracted_Altitude 89

Slalib_date 1994/7/19/0.298822999997

Opsim_rotskypos 0

Opsim_rottelpos 0

Opsim_moondec -90

Opsim_moonra 180

Opsim_expmjd 49552.3

Opsim_moonalt -90

Opsim_sunalt -90

Opsim_filter 2

Opsim_dist2moon 180.0

Opsim_moonphase 10.0

Opsim_obshistid 99999999

Opsim_rawseeing 0.65

SIM_SEED  1000

SIM_MINSOURCE 1

SIM_TELCONFIG 0

SIM_CAMCONFIG 1

SIM_VISTIME 300

SIM_NSNAP 1

object 0 0.0 0.0 24 ../../../Research/psf_creation_phosim/scripts/

narrowband_seds/narrowband0.sed 0 0 0 0 0 0 star none none
```

3.1.4 PSF Generation

Once we have all three parts: a SED file, an instance catalog, and a background file, then we can use the program PhoSim to generate PSF files. The software PhoSim gives us multiple outputs, out of which what we are interested in is only the electron image named as $lsst\_e\_99999999\_f2\_R22\_S11\_E000.fits.gz$. For a given SED, we unzip this file and take as the PSF file. 1 narrowband SED files, so this means we will get one broadband PSF and 21 narrowband PSFs. For example, the first PSF file is called "psf0_unnormalized.fits" and so on. After this, we normalize all the 21 PSF files, i.e., make the total sum of the pixels of each fits image to zero. For example the file "psf0.fits" have total sum of pixels equal to 1 and same holds for "psf1.fits" to "psf20.fits".

4 Galaxy Simulation

In this section we want to generate the realistic simulation for the upcoming ground based large telescope survey called Large Synoptic Survey Telescope (LSST). We have past data from the space based survey called “Hubble Space Telescope” (HST) using HST ACS Wide Field Camera. The HST data was cleaned and single galaxies were isolated. In previous chapter 2, we created the bulge and disk components of each of these base galaxies and we will be using these bulge and disk components instead of the original clean galaxies from the Hubble images.

4.0.1 Creation of Scaled Bulge, Disk, and Monochromatic Images

We have total 201 number of HST images, so, after bulge disk decomposition, we have 201 bulge images and 201 disk images. We have selected the cutoff redshift of 0.2. This means, we have 201 bulge.fits files for redshift 0.2 and 201 disk.fits files for redshift 0.2. Here, we want to simulate the galaxies at different redshift, let’s say at redshift of 1.5. Then, we need to create 201 scaled_bulge.fits and 201 scaled_disk.fits with appropriate bulge to disk ratio. We also create 201 scaled_bulge_disk.fits for monochromatic study. We keep the “sum of flux of 201 bulge files” same as “sum of 201 scaled_bulge files” and “sum of flux of 201 disk files” same as “sum of flux of 201 scaled_disk files” but the ratio of any particular bulge and disk file is different. For example, the ratio of “flux of bulge_100.fits / flux of disk_100.fits” is different than the ratio of “flux of scaled_bulge_100.fits / flux of scaled_disk_100.fits”.

For bulge and disk we have two different spectral energy distribution (SED) files. Each SED file has first column as wavelength and 12 more columns with flux at galaxy age 1Gyr to 12Gyr. In these original SED files, the wavelength separation is 5 Angstrom, but we need the wavelength separation of 1 Angstrom. So, I interpolate the bulge and disk SED files to get 1 Angstrom separation. Then we have bulge and disk SED files with wavelength range of 1000 Angstrom to 12,000 Angstrom with bin width of 1 Angstrom.

The SED file for bulge is called “ssp_pf.cat” and for disk is called “exp9_pf.cat”. For example, the first few lines of SED file for bulge looks like this:

```
#         lambda        flux[0]        flux[1]        flux[2]        flux[3]
         1000    2.21285e-07   2.190075e-07    1.67015e-07    1.41375e-07   1.6
         1005  2.3901664e-07  2.3592124e-07  1.7978445e-07  1.5221227e-07  1.84
         1015  2.3995361e-07  2.3961105e-07  1.8290807e-07  1.5483825e-07  1.91
         1025   1.502035e-07  1.5788075e-07  1.2088081e-07  1.0232831e-07    1.
         1035  2.1904999e-07  2.2166077e-07  1.6932341e-07   1.433061e-07   1.7
```

As we can see the wavelength separation of two consecutive rows is 5 Angstrom. But, we need the wavelength separation of 1 Angstrom, so we interpolate these SED files and create two new files called “ssp_pf_interpolated_z1.5.csv” and “exp9_pf_interpolated_z1.5.csv”. The first few lines of interpolated file for bulge looks like this:

```
# wavelength     flux_z                         flux12
1000           1.6701500000000e-07 3.2000000000000e-07
1001           1.6882096813517e-07 3.2623930383731e-07
1002           1.7114507534542e-07 3.3295580963054e-07
1003           1.7384314348808e-07 3.3994876350512e-07
1004           1.7677099442049e-07 3.4701741158646e-07
1005           1.7978445000000e-07 3.5396100000000e-07
1006           1.8273933208394e-07 3.6057877487116e-07
1007           1.8549146252966e-07 3.6666998232538e-07
1008           1.8789666319448e-07 3.7203386848806e-07
```

```
Note: Here flux_z is for redshift z = 1.5
```

Now, we have interpolated bulge and disk SED files and bulge and disk fits files. From these, we can create scaled_bulge, scaled_disk, and scaled_bulge_disk files. For this, we need to find the bulge_factor (bf) and disk_factor (df). Then, then we can have scaled galaxies:

$$scaled\_bulge = bf * bulge.fits \tag{4.1}$$

$$scaled\_disk = df * disk.fits \tag{4.2}$$

To find bulge and disk factors, first we find fraction of bulge ratio and fraction of disk ratio as follows:

$$f_{ratioB} = \frac{\int_{\lambda 0}^{\lambda 20} f_{bz}(\lambda) d\lambda}{\int_{\lambda hst0}^{\lambda_{hst20}} f_{bzcut}(\lambda) d\lambda} \tag{4.3}$$

$$f_{ratioD} = \frac{\int_{\lambda 0}^{\lambda 20} f_{dz}(\lambda) d\lambda}{\int_{\lambda hst0}^{\lambda_{hst20}} f_{dzcut}(\lambda) d\lambda} \tag{4.4}$$

Here, f_{bz} is the flux column from the SED file according the redshift z for the bulge and f_{bzcut} is the flux column for cutout galaxy. Here, we have used the galaxy cutout redshift as $z_{cutout} = 0.2$. Similarly we have the flux columns for disk galaxies.

The wavelengths λ_0 and λ_{20} are the LSST R-band filter blue and red wavelengths. This range is 5520Å to 6910Å (Refer to: $https://www.lsst.org/about/camera/features$). We divide these wavelengths by a factor $(1 + z)$ to get the range 2208 to 2764 for the redshift of 1.5.

Similarly, for the HST the wavelengths are $\lambda_{hst0} = 7077.5$Å and $\lambda_{hst20} = 9588.5$Å after dividing by $1 + z = 1.2$ we get $\lambda_{hst0} = 5897.9$Å and

$\lambda_{hst0} = 7990.4\text{Å}$. We can get more details about HST ACS/WFC filter at the website $http:$
$//www.stsci.edu/hst/acs/documents/handbooks/current/c05\_imaging2.html$.

Then, we get bulge factor and disk factor using the formula:

$$bf = \frac{F_b + F_d}{F_b * f_{ratioB} + F_d * f_{ratioD}} * f_{ratioB} \tag{4.5}$$

$$bd = \frac{F_b + F_d}{F_b * f_{ratioB} + F_d * f_{ratioD}} * f_{ratioD} \tag{4.6}$$

$$\tag{4.7}$$

where, F_b is the flux of a bulge file (for example, a fitsfile, $simdatabase/bulge\_f8/f814w\_bulge0.fits$) and F_d is the flux of a disk file (for example, a fitsfile $simdatabase/disk\_f8/f814w\_disk0.fits$). For 201 bulge and disk files we have 201 bulge and disk factors.

After we get these bulge and disk factors we simply multiply them by the bulge.fits and disk.fits to get scaled_bulge.fits and scaled_disk.fits.

4.0.2 PSF Creation for Bulge, Disk, and Monochromatic Images

From the PhoSim Software we have created 21 narrowband PSF files. Now, we will use them to create PSF for scaled bulge, disk, and monochromatic images. For that we need to have the weights for bulge and disk images. The weights for bulge and disk narrowbands are calculated from the interpolated SED file using following equations:

$$b0 = \frac{\int_{\lambda 0}^{\lambda 1} f_b(\lambda) d\lambda}{\int_{\lambda 0}^{\lambda_{20}} f_b(\lambda) d\lambda} \tag{4.8}$$

$$d0 = \frac{\int_{\lambda 0}^{\lambda 1} f_d(\lambda) d\lambda}{\int_{\lambda 0}^{\lambda_{20}} f_d(\lambda) d\lambda} \tag{4.9}$$

Similarly, we get bulge and disk weights for other narrowbands such as b1,b2,...,b20 and d1,d2,...,d20. Here, the wavelength λ is the wavelength for the LSST survey R-band filter (lsst.org) and have values blue side as 5520 Angstrom and red side as 6910 Angstrom. Then we divide these numbers by $1 + z$ with given redshift, for example $z = 1.5$ we get 2208A to 2764A. Then, we break these range into 21 parts and make them integer. These parts looks like this: [2208, 2234, 2261, 2287, 2314, 2340, 2367, 2393, 2420, 2446, 2473, 2499, 2526, 2552, 2579, 2605, 2632, 2658, 2685, 2711, 2738, 2764]. Here, the first narrowband is 2208 Angstrom to 2234 Angstrom. We choose the values between these wavelengths from the file "ssp_pf_interpolated_z1.5.csv" and get the bulge weight b0. Similarly, from the wavelengths 2208 to 2234 from the file "exp9_pf_interpolated_z1.5.csv" we get the disk weight d0.

The values of weights for bulge and disk for 21 narrowbands for redshift $z = 1.5$ looks like following:

```
bulge_wt disk_wt
0.01753 0.05077
0.02135 0.05241
0.02002 0.05121
0.02420 0.05202
0.01903 0.04349
0.01839 0.04748
0.01610 0.04425
0.01784 0.04623
0.02842 0.04627
0.03415 0.04855
0.02118 0.04483
```

```
0.02211 0.04696

0.02725 0.04371

0.05117 0.04767

0.04450 0.04629

0.04585 0.04710

0.14001 0.04963

0.13688 0.05040

0.13214 0.04779

0.09094 0.04873

0.07093 0.04422
```

After getting these 21 bulge and disk weights, we get the scaled PSF as given below:

$$p_b = \frac{b0 * p0 + b1 * p1 + ... + b20 * p20}{b0 + b1 + ... + b20} \quad \text{(psf for bulge)} \tag{4.10}$$

$$p_d = \frac{d0 * p0 + d1 * p1 + ... + d20 * p20}{d0 + d1 + ... + d20} \quad \text{(psf for disk)} \tag{4.11}$$

Here, p_b, p_d,and p_m are PSF for bulge, disk, and monochromatic respectively. Also the quantities $b0, b1, ..., b20$ and $d0, d1, ..., d20$ are bulge and disk weights for 21 narrowbands as described above.

Now we have PSF files for bulge and disk cases but we do not have the PSF file for the monochromatic case. To create the PSF file for the monochromatic case we need to get bulge factor and disk factor. First we define the flux ratio quantity f_r as follows:

$$f_r = \frac{\sum(\frac{F_{sb}}{F_{sd}})}{Ngals} \tag{4.12}$$

Here, F_{sb} is the flux of the given scaled bulge, F_{sd} is the flux of the given scaled disk and $Ngals$ is number of galaxies. Here number of galaxies $Ngals$ is equal to 201.

Then, we calculate disk part and bulge part of f_r as following:

$$f_{rb} = \frac{f_r}{1 + f_r} \tag{4.13}$$

$$f_{rd} = \frac{1}{1 + f_r} \tag{4.14}$$

For example, for redshift $z = 1.5$ I got the values: $f_r = 0.0022$, $f_{rb} = 0.0021$, and $f_{rd} = 0.99785444$.

After getting bulge factor and disk factor, we get the PSF for monochromatic case using following formula:

$$p_m = f_{rd}\ p_d + f_{rb}\ p_b \quad \text{(psf for monochromatic)} \tag{4.15}$$

4.0.3 Jedisim Simulations

Jedisim is a computer program that simulates realistic LSST images from HST images using various physics parameters. It was initially developed by Dr. Ian Dell'Antonio of Brown University (2014) and after that heavily expanded and maintained by Bhishan Poudel of Ohio University (2014-2021) with the help of Prof. Dr. Douglas Clowe. This program takes in scaled bulge and scaled disk galaxies and finally will create chromatic and monochromatic lsst images. It will also create 90 degree rotated images for the chromatic and monochromatic lsst images. The **Jedisim** program itself consist of various sub programs, which I will describe briefly below.

4.0.3.1 Create the Catalogs for Jedisim

We use the subprogram **jedicatalog** to create the three catalog files need by Jedisim. The files are **catalog.txt**, **convolvedlist.txt**, and **distortedlist.txt**. The **catalog.txt** file contains various important quantities of a galaxy. Each row of catalog.txt file contains following parameters: galaxy_name, center_x, center_y, angle, redshift, pixscale, old_magnitude, old_radius, new_magnitude, new_radius, stamp_name, distorted_file_name. We will need these parameters to transform the galaxies.

In the jedicatalog program we specify a galaxy by six parameters magnitude, radius, image, redshift,position, and angle.

Magnitude In this simulation we have chosen the simulated galaxies magnitudes within range $22 \leq M \leq 28$. The galaxies are distributed with the power law,

$$P(M + dM) \propto 10^{BM} \tag{4.16}$$

where M is the magnitude and $B = 0.33 \ln 10$ is an empirical constant (refer to [**?**]). The magnitude zero-point is taken to be 30 throughout the simulations by convention

Radius The simulated galaxies have the magnitudes between 22 and 28. For each magnitudes, we have a radius database for r50 radii. We choose a radius randomly from that radius bin for the given magnitude.

Image The postage stamp image is chosen randomly from the list of r50 radii such that the chosen r50 radius is larger than the radius of original galaxy. This makes sure that images are always sized down and no information is artificially created by scaling.

Redshift In this simulation we have chosen the fixed redshift of 1.5. However, we can choose random redshifts for each magnitude bin from magnitude 22 to 28 is we opt to vary the redshifts of galaxies. The redshift database was obtained from ZCOSMOS database.

Position The position of center of the postage stamps are chosen randomly from the range [301,40,660] . This range is taken to ensure that all 600 by 600 postage stamps lie completely within the range [0,40,960]. In later simulation step, we will trim the border by 480 pixels so as to ensure uniform distribution of galaxies with typical edge effects.

Angle We chose the angle of orientation of a galaxy randomly between 0 to 360 degrees. We should note that the orientation of galaxies has three degree of freedom, but since we are dealing with 2D projections of galaxies, we can only make the orientation random in one degree of freedom.

The **convolvedlist.txt** contains the names of files to be written after we convolve a galaxy with a PSF. A typical row of convolvedlist appears like this $jedisim\_out/out0/convolved/convolved\_band\_0.fits$. When we convolve a large fitsfile with a PSF, due to the memory restrictions of computer instead of creating single large convolved file we create 6 convolved bands and later combine them into a single large convolved galaxy.

The **distortedlist.txt** contains the names of the galaxies that will be after we distort them using Singular Isothermal Profile Lens. A typical row of convolvedlist appears like this $jedisim\_out/out0/distorted\_0/distorted\_0.fits$. There are 12,420 rows and the last row is $jedisim\_out/out0/distorted\_0/distorted\_12419.fits$.

The program **jedicatalog** will create

4.0.3.2 Transform the Galaxies

We transform the scaled bulge, scaled disk, and scaled bulge_disk files using **jeditransform**. This sub routine reads in the catalog file and various transforming physics parameters from that file and then transforms the galaxies. This program reads 201 bulge (or disk) galaxies and create 12,420 HST stamps.

4.0.3.3 Distort the Galaxies

Here, we use the Singular Isothermal Sphere (SIS) profile to lens the galaxies. We have chosen fixed position of the lens to be (6144,6144) and taken dispersion velocity $\sigma_v = 1000km/s$. In the singular isothermal profile the density is calculated as

$$\rho(r, \sigma_v) = \frac{\sigma_v^2}{2\pi G r^2} \tag{4.17}$$

where, G is the gravitational constant and r is radius in pixels. Since the total mass inside radius r diverges as r reaches to infinity, the SIS model is non-physical. However, when the profile is finitely bounded, it constitutes a possible physical distribution and can be used as a lens.

We may also use the Navarro-Frenk-White (NFW) profile which does not suffer from the divergence problem. The NFW profile is given by:

$$\rho(r, \rho_0, R_s) = \frac{\rho_0}{\frac{r}{R_s}(1 + \frac{r}{R_s})^2} \tag{4.18}$$

where, $\rho_0, and\ R_s$ are the parameters dependent on the halo we use.

4.0.3.4 Convolve the Galaxies with the PSF

After we distort the galaxies, we convolve the big HST image with the PSF. For the bulge components we convolve the big HST galaxy with scaled bulge PSF and for the disk components we convolve the big HST image with scaled disk PSF.

4.0.3.5 Rescale the Galaxies from HST to LSST

Until now, we have been dealing with the HST images and HST PSF images. Now, we scale down the pixels of HST to LSST using a routine **jedirescale**. After rescaling we go to the PIXSCALE 0.2 of LSST from the PIXSCALE of 0.06 of HST.

For bulge components, the output of jedirescale gives us lsst_bulge file and similarly disk components gives us lsst_disk fits files. Also, for the bulge_disk files we get lsst_monochromatic unnoised file.

4.0.3.6 Create Monochromatic LSST Image

From the **jedirescale** program if we feed the bulge_disk images as the input fits-files, we will get the lsst_bulge_disk fits-file as the output. We add the Poisson noise of mean noise 10 pixels to get the LSST monochromatic image. This is one of the main output of the **Jedisim** program.

4.0.3.7 Create Chromatic LSST Image

From the **jedirescale** program we get lsst_bulge and lsst_disk images. We combine them and add the Poisson noise of mean noise 10 pixels to get the LSST chromatic image. This is one of the main output of the **Jedisim** program.

4.0.3.8 Rotated galaxies output from Jedisim

If we run the **Jedisim** program for the normal case we will get two main outputs, namely, lsst.fits and lsst_mono.fits. But, the galaxies are randomly orientated in the universe and we may also want the 90 degree rotated versions of the galaxies. For, this purpose, the program Jedisim, will also gives us 90 degree rotated versions of the output files named as lsst90.fits and lsst_mono90.fits. So, in the end of one run of Jedisim we will get four important output files, two for non-rotated galaxies and two for rotated galaxies.

5 Star Creation

When we run jedisim, we get chromatic and monochromatic versions of simulated galaxies. These galaxy images (lsst.fits, lsst90.fits, lsst_mono.fits and lsst_mono90.fits) are already convolved with the PSF images and we had also added some fixed sky noise of 10 (obtained from simulation configuration file). However, we have not inject the stellar contamination and WCS addition to these galaxies yet. In this section, we will discuss the process of creating simulated stars and combining them with the galaxy images. The DMSTACK and OBS_FILE pipeline demands that the input images should have stars associated with them and have World Coordinate System (WCS) values attached to the input fitsfiles.

5.0.1 Create text file for star positions

Here, from section 3.1.4 we have 21 PSF files for 21 narrow-bands of whole flux region so as to study the chromatic effects of PSF. We also have "weighted PSF files" for bulge, disk and, monochromatic cases (psfb.fits, psfd.fits and, pdfm.fits). The jedisim simulation output files have a shape of NAXIS1 = 3398 and NAXIS2 = 3398 but these images are created after re-scaling HST images with LSST pixscale (i.e. we change HST pixscale of 0.06 to LSST pixscale of 0.2). The HST simulated images have the size of NAXIS1 = 12288 and NAXIS2 = 12288. So we need to create star.fitsfiles with initial shape of 12288 and later we will convolve it with the respective PSF to get the desired shape. While creating the star files, we want to exclude the stars at the border of the images, so we choose some offset to deal with this. In this simulation we chose an offset of 10%, this means we have steller objects only in the region between *offset* to *0.1 * 12288* and there are no stars between 0 to offset and so on. To use the same stars for all the simulation for the verification, we keep the fixed seed of random number generator ($SEED = 100$) and randomly generate *n_star* number of stars using **log-uniform** distribution. Here we choose to create 300 stars with values between 10^3 and 10^7 from the log-uniform distribution. The "star.txt" files will have n_star (= 300) rows with two columns x and y as the position of stars to be created with.

5.0.2 Create star.fits file from text file

Here we have star.txt file with positions and value for the star. This text file has 3 columns: x,y and star_value and number of rows is equal to number of stars. To create fitsfile from text file, we use the script "a02_cr_star_nstar_low_high.py". This script first creates an empty fitsfile with all values 0.0 with shape $NAXIS1 = 12288$ and $NAXIS2 = 12288$. Then, we read the position of star x,y from text file and put the value of that position there. For 300 stars, we have only 300 non-zero values and all others values are zero. Here, We also should note that the star.fits files do not depend on redshift and we can use the same star.fitsfiles for different redshift simulations.

5.0.3 Convolve and re-scale star.fits with psf bdm files

We have single star.fits file that does not depend on the redshift. We also have three weighted PSF files for bulge, disk and, monochromatic cases, viz. (psfb.fits, psfd.fits and, psfm.fits) which depend on the choice of redshift. We convolve the star.fits files with these three different PSF files. The resultant files have the shape of 12288 but the final output of jedisim has the shape of 3398, so we re-scale the convolved image with LSST pixscale of 0.2 and get the shape 3398 so that we can add these files to jedisim outputs.

5.0.4 Add stars to jedisim outputs

The jedisim simulation gives the simulation images such as: *jout_z1.5_ngals10k_000_099/lsst_000.fits*, *jout_z1.5_ngals10k_000_099/lsst90_000.fits*, *jout_z1.5_ngals10k_000_099/lsst_mono_000.fits*,

jout_z1.5_ngals10k_000_099/lsst_mono90_000.fits and we also have star files such as:

star_redshift_nstar_low_high/starb_z1.5_300_e3_e7.fits,

star_redshift_nstar_low_high/starb_z1.5_300_e3_e7.fits,

star_redshift_nstar_low_high/starb_z1.5_300_e3_e7.fits.

Here, in star files, 300 means that there are 300 stars and e3, e7 means that these stars have value between 10^3 to 10^7 drawn randomly from log-uniform distribution (not log-normal distribution).

To run the DMSTACK pipeline, we need to add these stars to chromatic and monochromatic files obtained from Jedisim simulations. For the 0 degree and 90 degree rotated versions of monochromatic files (lsst_mono.fits, lsst_mono90.fits), we combine the **starm.fits** file to these files.

For chromatic cases (lsst.fits and lsst90.fits) we add both **starb.fits** and **stard.fits** files to them. In the end, we also need to add the fake world co-ordinates system (WCS) to these files. In summary the input file for DMSTACK looks like this:

```
wcs_star_z1.5_300_e3_e7_lsst_000.fits =
jout_z1.5_ngals10k_000_099/lsst_000.fits
+ stars_z_nstar_low_high/starb_z1.5_300_e3_e7.fits
+ stars_z_nstar_low_high/stard_z1.5_300_e3_e7.fits
+ WCS

wcs_star_z1.5_300_e3_e7_lsst90_000.fits =
jout_z1.5_ngals10k_000_099/lsst90_000.fits
+ stars_z_nstar_low_high/starb_z1.5_300_e3_e7.fits
+ stars_z_nstar_low_high/stard_z1.5_300_e3_e7.fits
```

```
+ WCS

wcs_star_z1.5_300_e3_e7_lsst_mono_000.fits =
jout_z1.5_ngals10k_000_099/lsst_mono_000.fits
+ stars_z_nstar_low_high/starm_z1.5_300_e3_e7.fits
+ WCS

wcs_star_z1.5_300_e3_e7_lsst_mono90_000.fits =
jout_z1.5_ngals10k_000_099/lsst_mono90_000.fits
+ stars_z_nstar_low_high/starm_z1.5_300_e3_e7.fits
+ WCS
```

6 DMSTACK ANALYSIS

6.1 Introduction to LSST Pipeline

Using jedisim we get the simulated galaxy cluster images. Here we use 1500 jedisim simulations with redshift $z = 1.5$ and number of galaxies in each field $ngals = 10,000$. To extract the physical quantities from these simulated fields we use LSST Science Pipeline which we call **dmstack**. Using dmstack we get various flags such as PSF was used or not and so on and many other astrophysical quantities. The dmstack gives us centroid positions, moment estimates, flux calculations and many other important quantities.

In this project we use dmstack version 13.0 using python 2.7 and miniconda environment. We installed LSST software package **lsst-distrib** and package manager **eups**. To run the dmstack we also need to install other dependent package obs_file. We should note that for the compatibility issue we need to use the specific version of dmstack and obs_file given in above link.

The detail installation instruction is given below:

```
#=================== Part 1: conda env lsst  ===============
# First create python conda evironment "lsst" using instructions from
# https://pipelines.lsst.io/v/13-0/install/conda.html
# make sure you have python version 2.7
conda config --add channels http://conda.lsst.codes/stack/0.13.0
conda create --name lsst python=2
source activate lsst
conda install lsst-distrib
source eups-setups.sh
setup lsst_distrib
```

```
#============== Part 2: download obs_file github repo ============
# 1. The dmstack needs obs_file project. So download the following
# github repository to your desktop
mkdir -p ~/Softwares
cd ~/Softwares

# Then download the repo
# "https://github.com/SimonKrughoff/obs_file/tree/tickets/DM-6924"

# Make sure you download the specific version given above,
# not the latest repository.

#=========== Part 3: Testing the Installation ============
mkdir -p ~/test_dmstack
cd ~/test_dmstack
wget https://github.com/bhishanpdl/DMstack_obsfile_Clusters/blob/master/
example/trial00_good_fits.zip?raw=true
unzip trial00_good_fits.zip\?raw\=true # this gives trial00_good.fits
ls
# we must have trial00_good.fits file
# (note that this file is obtained after adding fake WCS
# and stars to jedisim output file trail00.fits)

# Setup lsst environment
source activate lsst && source eups-setups.sh && setup lsst_distrib
```

```
# Setup obs_file
cd ~/Softwares/obs_file && setup -k -r . && scons && cd - && ls

# Create input and output directories
rm -rf input output; mkdir input output

# Provide the mapper
mkdir input; echo "lsst.obs.file.FileMapper" > input/_mapper

#  Ingest the data
ingestImages.py input/ trial00.fits --mode link

# Process the data
# processCcd.py --help
# We can optionally have configuration file called "processCcdConfig.py"
#in the working directory to adjust dmstack parameters.

echo 'config.charImage.repair.cosmicray.nCrPixelMax=1000000' > processCcdConfig.py
processCcd.py input/ --id filename=trial00.fits --config isr.noise=50
--output output --configfile processCcdConfig.py --clobber-config

# Look at the output file (src.fits)
# The dmstack gives output in the form of fitstable (not csv file)
# We can use tools like fv.app to open the table.
# We can also use python package astropy to create csv from this fits table.
```

```
cp output/src/*/src.fits src.fit
/Users/poudel/Applications/fv/fv.app/Contents/MacOS/fv src.fit
```

6.2 Preparation of Data for dmstack

The dmstack needs specific requirements for the data to be processed. For example, the input fits image needs to have WCS and must have PSF stars in its galaxy field image. Without additional stars added to the jedisim output, the dmstack pipeline simply fails and does not provide any outputs. Here, in our jedisim simulation, we have not added any WCS physics to the galaxy images. For the usage of our simulation images with dmstack, we add same fake WCS values and some stars to the simulation.

```python
from astropy import wcs
w = wcs.WCS(naxis=2)
w.wcs.crpix = [1800.0, 1800.0]
w.wcs.crval = [0.1, 0.1]
w.wcs.cdelt = np.array([-5.55555555555556E-05,5.55555555555556E-05])
w.wcs.ctype = ["RA---TAN", "DEC--TAN"]
```

After we add these WCS values to our simulations. We also need to add the stars to our image. We already have created the stars in the section 5. We combine the WCS and stars to the output of jedisim simulation and run it through dmstack.

6.3 Cleaning of dmstack outputs

The dmstack gives us files like *src_lsst_z*1.5_000.*csv* which have 90 flags (such as calib_psfCandidate) and 77 parameters (such as deblen_nChild, ext_shapeHSM_HsmShapeRegauss_e1, base_SdssCentroid_x and so on). We select the objects based on these flags and parameters.

Filtering of dmstack objects

```
# choose only not-PSF candidates
calib_psfCandidate == 0.0

# choose objects that do not have any children
deblend_nChild == 0.0

# remove unphysical objects (ellipticity less than 2.0)
ellip = sqrt(e1**2 + e2**2)
ellip < 2.0

#select only few columns after filtering:
cols_select = ['base_SdssCentroid_x', 'base_SdssCentroid_y',
                'base_SdssCentroid_xSigma','base_SdssCentroid_ySigma',
                'ext_shapeHSM_HsmShapeRegauss_e1',
                'ext_shapeHSM_HsmShapeRegauss_e2',
                'base_SdssShape_flux']
data = data[cols_select]

# remove all NaN values
data = data.dropna()

# exclude strong lens objects <=154 distance
# The shape of lsst.fits file is 3998,3998 and center is 1699,1699.
```

```
df['x_center'] = 1699
df['y_center'] = 1699
df['distance'] = ( (df['x[0]'] - df['x_center'])**2 +
                   (df['x[1]'] - df['y_center'])**2 )**0.5
df = df[df.distance > 154]
```

6.3.1 Merge 4 catalogs into one catalog using IMCAT

From jedisim and dmstack, for the simulation number 000 we get four files, viz, lsst_000.fits, lsst90_000.fits,lsst_mono_000.fits, and lsst_mono90_000.fits. We parse the dmstack cleaned columns "ext_shapeHSM_HsmShapeRegauss_e1" and "ext_shapeHSM_HsmShapeRegauss_e1" as two vector "g" in IMCAT. Then we use **cleanct** value 50 to do the selection. After that we combine four files using **mergecats** command with value 5.

```
# create new columns and cleaning (four files)
lc -C -n fN -n id -N '1 2 x' -N '1 2 errx' -N '1 2 g'
-n ellip -n flux -n radius
< "${M9T}".txt  |  lc +all 'mag = %flux log10 -2.5 *'
  |  cleancat 10  |  lc +all -r 'mag' > "${M9C}".cat



# merge 4 catalogs
mergecats 5 "${MC}".cat "${M9C}".cat "${LC}".cat "${L9C}".cat >
${catalogs}/merge.cat &&



lc -b +all
```

```
'x = %x[0][0] %x[1][0] + %x[2][0] + %x[3][0]
+ 4 / %x[0][1] %x[1][1] + %x[2][1] + %x[3][1] + 4 / 2 vector'
'gm  = %g[0][0] %g[1][0] + 2 / %g[0][1] %g[1][1] + 2 / 2 vector'
'gc  = %g[2][0] %g[3][0] + 2 / %g[2][1] %g[3][1] + 2 / 2 vector'
'gmd = %g[0][0] %g[1][0] - 2 / %g[0][1] %g[1][1] - 2 / 2 vector'
'gcd = %g[2][0] %g[3][0] - 2 / %g[2][1] %g[3][1] - 2 / 2 vector'
< ${catalogs}/merge.cat > ${final}/final_${i}.cat

# Notes
x   =
gm  = (g00 g10 + g01 g11) / 2
gc  = (g20 g30 + g21 g31) / 2
gmd = (g00 g10 - g01 g11) / 2
gcd = (g20 g30 - g21 g31) / 2
```

6.3.2 Reduced Shear for monochromatic and chromatic

Here, we have the components of reduces shear as 2-vector and we create new terms for squared of reduced shear, squared of reduced shear for monochromatic and squaredof reduced shear for chromatic cases.

```
df['g_sq']      = df['g[0][0]'] **2 + df['g[0][1]'] **2
df['gmd_sq']    = df['gmd[0]']  **2 + df['gmd[1]']  **2
df['gcd_sq']    = df['gcd[0]']  **2 + df['gcd[1]']  **2

df['gm_sq']     = df['gm[0]']**2 + df['gm[1]']**2
df['gc_sq']     = df['gc[0]']**2 + df['gc[1]']**2
```

We plot histogram plot of squared of reduced shear for monochromatic and chromatic cases. The distribution looks similar, but we have a problem, we see a bump near 0.2.

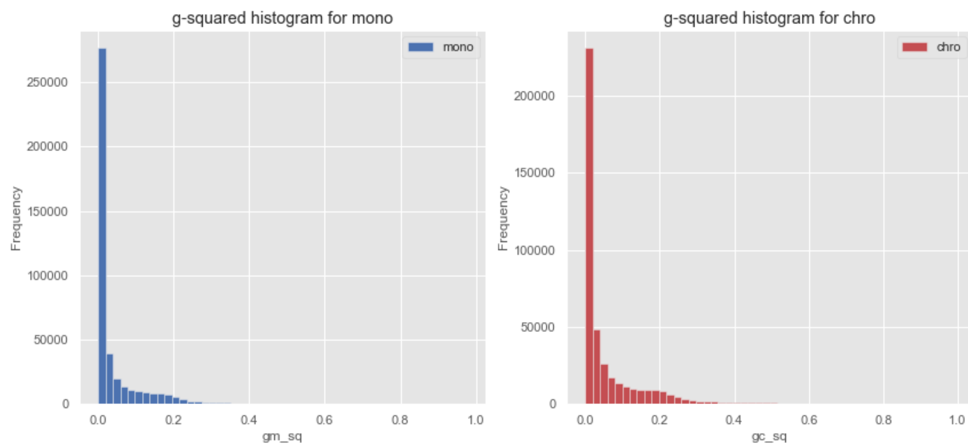


Figure 6.1: Squares of reduced shears

To investigate the flaw around bump, we look at the contour plot of gsq versus gmdsq.

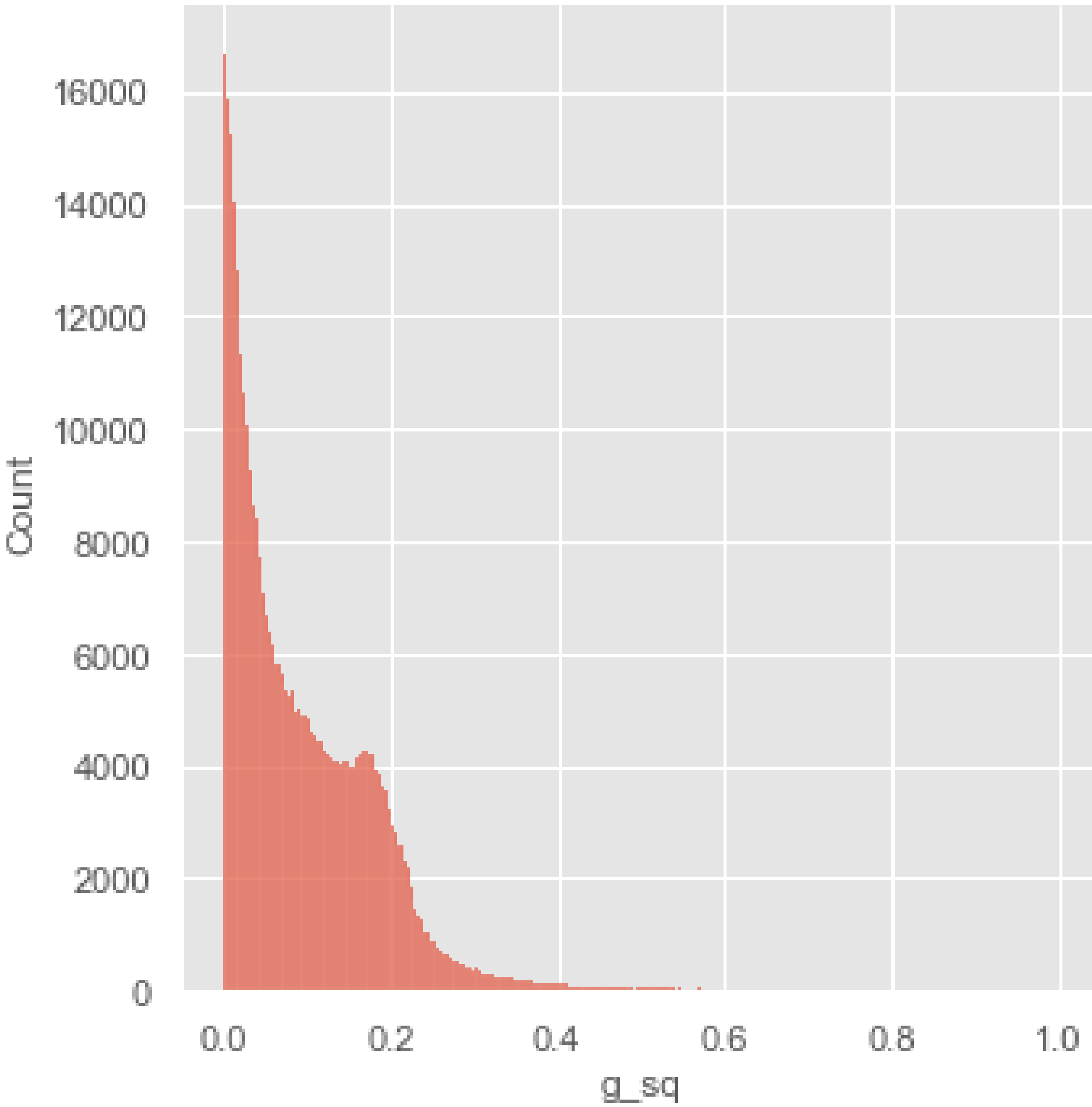


Figure 6.2: Squares of reduced shears with bump zoomed

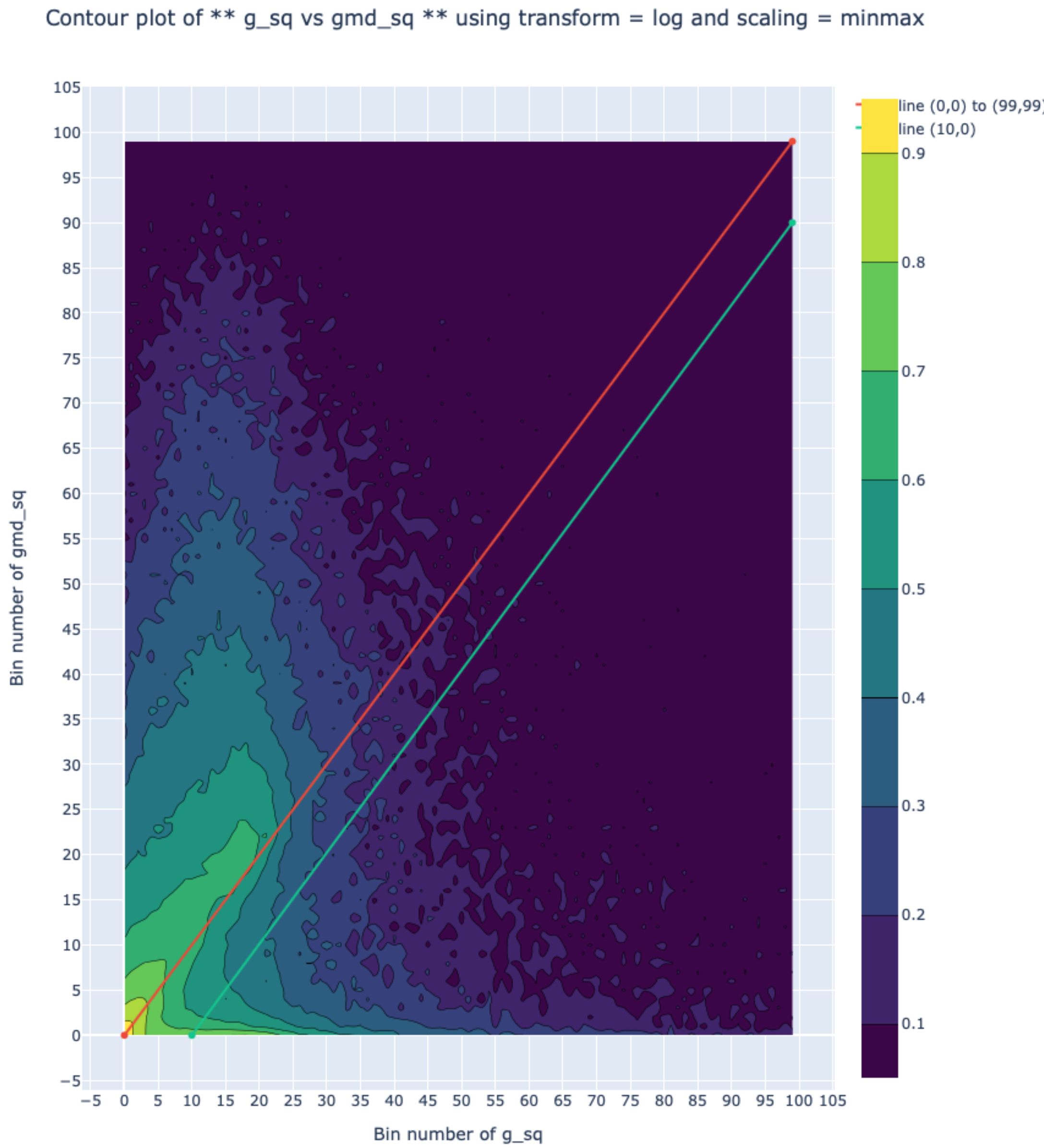


Figure 6.3: Square of Reduced Shear versus Square of Difference

7 DMSTACK ANALYSIS

References

[BS01] M. Bartelmann and P. Schneider. Weak gravitational lensing. *Physics Reports*, 340:291–472, January 2001. arXiv:astro-ph/9912508, doi:10.1016/S0370-1573(00)00082-X

[DED20] F. W. Dyson, A. S. Eddington, and C. Davidson. A determination of the deflection of light by the sun's gravitational field, from observations made at the total eclipse of may 29, 1919. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 220(571-581):291–333, 1920. URL: http://rsta.royalsocietypublishing.org/content/220/571-581/291, arXiv:http://rsta.royalsocietypublishing.org/content/220/571-581/291.full.pdf, doi:10.1098/rsta.1920.0009

[Har03] James B. Hartle. *Gravity: An Introduction to Einsteins General Relativity*. CUMMINGS, 2003. URL: http://www.ebook.de/de/product/3250263/james_b_hartle_gravity_an_introduction_to_einstein_s_general_relativity.html

[Sch07] Peter Schneider. *Extragalactic Astronomy and Cosmology: An Introduction*. Springer, 2007.

[WBD+96] Robert E. Williams, Brett Blacker, Mark Dickinson, W. Van Dyke Dixon, Henry C. Ferguson, Andrew S. Fruchter, Mauro Giavalisco, Ronald L. Gilliland , Inge Heyer, Rocio Katsanis, Zolt Levay, Ray A. Lucas, Douglas B. McElroy, Larry Petro, Marc Postman, Hans-Martin Adorf, and Richard Hook. The Hubble Deep Field: Observations, Data Reduction, and Galaxy Photometry. , 112:1335, Oct 1996. arXiv:astro-ph/9607174, doi:10.1086/118105

Appendix: An Appendix

A.1 A Section in the Appendix