

Early Growth in a Perturbed Universe: Dark Matter Halo Properties in 2LPT and ZA
Simulations

By

Daniel J. Sissom

Dissertation

Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in

PHYSICS

August, 2014

Nashville, TN

Approved:

Date:

Jocelyn K. Holley-Bockelmann, Ph.D.

Andreas A. Berlind, Ph.D.

David A. Weintraub, Ph.D.

Shane M. Hutson, Ph.D.

Robert J. Scherrer, Ph.D.

ACKNOWLEDGMENTS

This is where you thank the people that made your work possible: grant awarding agencies, advisers, your committee, mom and dad, whatever.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	i
LIST OF TABLES	iii
LIST OF FIGURES	iv
I Introduction	1
I.1 The Early Universe	1
I.1.1 The CMB Epoch	1
I.1.1.1 Recombination	1
I.1.1.2 The Cosmic Microwave Background	2
I.1.2 Dark Matter Halo Formation	2
I.1.2.1 Collapse	2
I.1.2.2 Accretion	2
I.1.2.3 Mergers	2
I.1.2.4 Large-scale Structure	2
I.1.3 Halo Properties	2
I.1.3.1 Mass	2
I.1.3.2 Density and Concentration	3
I.1.3.3 Substructure and Environment	3
I.1.4 Baryonic Processes	3
I.1.4.1 The First Stars	4
I.1.4.2 Supermassive Black Holes	4
I.1.4.3 Enrichment and the The Intergalactic Medium	4
I.1.4.4 Reionization	4
I.2 Computational Theory	4
I.2.1 Simulation Initialization	5
I.2.1.1 Initial Conditions and the Surface of Last Scattering	5
I.2.1.2 The Zel'dovich Approximation	5
I.2.1.3 Second-order Lagrangian Perturbation Theory	5
I.2.2 Dark Matter Halos in N-body Simulations	5
I.2.2.1 Spherical Overdensity	5
I.2.2.2 Friends-of-Friends	5
II Numerical Methods	6
II.1 Initialization Code	6

II.1.1	Sampling the Power Spectrum	6
II.1.1.1	Cosmological Parameters	6
II.1.1.2	Sampling	6
II.1.2	Particle Displacement with ZA	6
II.1.3	Particle Displacement with 2LPT	6
II.2	Simulations with GADGET-2	6
II.2.1	GADGET-2	7
II.2.1.1	Force Calculation and Softening	7
II.2.1.2	The Tree Code	7
II.2.1.3	Parallelization	8
II.2.1.4	Metrics	8
II.2.2	Simulations	8
II.3	Halo Finding with ROCKSTAR	8
II.3.1	Halo Identification	8
II.3.2	Halo Properties	9
II.3.2.1	Mass	9
II.3.2.2	Concentration	9
II.3.2.3	Relaxation Parameters	9
II.3.3	Halo Catalogs	9
II.4	CROSSMATCH	10
II.5	Analysis	10
II.5.1	Non-linear Least Squares Fitting	10
II.5.1.1	CurveFit	10
II.5.1.2	Model Functions	10
II.5.1.3	Initial Guess	11
II.5.1.4	Fit Coefficients	11
II.5.2	Halo Properties	11
II.5.2.1	Rockstar Output and Post-processing	11
II.5.2.2	Spherical Overdensity Halo Particles	11
II.5.3	Density Profile Fitting	11
II.5.3.1	Density Profiles	11
II.5.3.2	Fitting	11
II.5.3.3	Mass Profiles	11
II.5.3.4	Characterization of Uncertainty	11
II.5.3.5	Halo Properties	12
II.5.4	Cross-matched Halo Catalog	12
II.5.4.1	Aggregation	12
II.5.4.2	Filtering	12
II.5.5	Halo Comparison	12
II.5.5.1	Match Verification	12
II.5.5.2	Morphology	12
II.5.5.3	Density Profiles	12
II.5.6	Difference Distributions	12
II.5.6.1	Histograms	12
II.5.6.2	Fitting	13

II.5.6.3	Mass Quartiles	13
II.5.7	Redshift Trends	13
II.5.7.1	Mean and Standard Deviation	13
II.5.7.2	Skew	13
II.5.7.3	Kurtosis	13
II.5.8	2-D Histograms	15
II.5.8.1	Binning and Plotting	15
II.5.8.2	Fitting	16
II.5.9	Alternate Difference Distributions	16
II.5.9.1	Equivalent Displacement	16
II.5.9.2	Redshift Trends	16
II.6	Automation	16
III	Exploring Dark Matter Halo Populations in 2LPT and ZA Simulations . . .	17
III.1	Introduction	17
III.2	Numerical Methods	22
III.3	Results	26
III.3.1	Individual halo pairs	27
III.3.2	Difference distributions of halo properties	27
III.3.3	Trends with redshift	28
III.3.4	Trends with halo mass	31
III.3.5	Alternate fractional difference distributions	32
III.4	Discussion	33
III.5	Conclusion	35
IV	Supermassive Black Holes and Their Hosts	42
IV.1	Introduction	42
IV.1.1	Galaxy Properties	42
IV.1.1.1	Color	42
IV.1.1.2	Morphology	43
IV.1.2	Supermassive Black Hole Properties	44
IV.1.3	Correlations	47
IV.1.3.1	The M-Sigma Relation	47
IV.1.3.2	The Fundamental Plane	49
IV.1.3.3	The Green Valley	50
IV.2	Galaxy Evolution	52
IV.2.1	Dark Matter Halos	52
IV.2.2	Galaxy Mergers	53
IV.3	Supermassive Black Hole Growth	54
IV.3.1	Binary Mergers	54
IV.3.1.1	Dynamical Friction and Inspiral	54
IV.3.1.2	The Final Parsec Problem	55
IV.3.1.3	Gravitational Waves and Recoil Kicks	55

IV.3.2	Accretion	59
IV.3.2.1	Bondi-Hoyle-Lyttleton Accretion	59
IV.3.2.2	Disk Accretion and Active Galactic Nuclei	61
IV.4	Conclusion	62
IV.4.1	Correlations	62
IV.4.2	Open Questions	62
V	Conclusion	64
	BIBLIOGRAPHY	65
	Appendices	69
A	BGC2 Python Code	70
B	Density Profile Python Code	72
C	Halo Matching Python Code	80
C.1	Python Code	80
C.2	Submission Script	84

LIST OF TABLES

Table	Page
III.1	Coefficients for linear least squares fits from Figure III.3. 29

LIST OF FIGURES

Figure	Page
II.1 Potential and force softening.	7
II.2 Barnes-Hut oct-tree in two dimensions.	7
II.3 Domain decomposition.	8
II.4 Force parallelism.	9
II.5 Energy conservation for an initially isothermal gas sphere.	10
III.1 Comparison of matched 2LPT and ZA halos	37
III.2 Histograms of ΔM_{vir} and Δc	38
III.3 Statistics as functions of redshift for generalized normal fits	39
III.4 ΔM_{vir} and Δc as a function of $M_{\text{vir,avg}}$	40
III.5 Fractional error distribution statistics as functions of redshift	41
IV.1 The Hubble tuning fork	44
IV.2 Maser orbits fit to a warped disk for NGC4258	46
IV.3 The M- σ relation for galaxies with dynamical measurements	48
IV.4 The fundamental plane for elliptical galaxies	50
IV.5 Distribution of the fraction of galaxies containing AGN	51
IV.6 Rotation curves for 21 Sc galaxies	52
IV.7 Gravitational waveform for a black hole binary merger	57
IV.8 Gravitational wave recoil velocity from black hole mergers	59

CHAPTER I

Introduction

Text goes here. This is where we'll talk about the purpose of the project and the layout of this document.

The structure of this document is as follows: The remainder of this chapter, Chapter I, provides an introduction to the early universe and the processes that lead to galaxy-hosting dark matter halos, as well as the fundamentals of the computational theory for the numerical methods relevant to this discussion. Chapter II examines in more detail the specific numerical methods used for this work, with emphasis on the methodologies of the codes themselves, how they are implemented in the context of the overall simulation and analysis pipeline, and the results obtained at each step. Chapter III is a direct representation of the published paper which (more succinctly) presents an overview of the numerical methods and the main results in this work. Chapter IV is primarily the same material as previously submitted to fulfill the requirements of the Qualifying Exam, and is slightly edited to better suit the tone of this document. Chapter V concludes with a discussion of the results in this work and the greater implications to the overall field.

I.1 The Early Universe

Text goes here.

I.1.1 The CMB Epoch

Text goes here.

I.1.1.1 Recombination

Text goes here.

I.1.1.2 The Cosmic Microwave Background

Text goes here.

I.1.2 Dark Matter Halo Formation

Text goes here.

I.1.2.1 Collapse

Text goes here.

I.1.2.2 Accretion

Text goes here.

I.1.2.3 Mergers

Text goes here.

I.1.2.4 Large-scale Structure

Text goes here.

I.1.3 Halo Properties

Text goes here.

I.1.3.1 Mass

There are a number of ways to define a halo's mass. This becomes significant for mass-sensitive studies, such as the halo mass function (Press & Schechter, 1974; Reed et al., 2007; Heitmann et al., 2006; Lukić et al., 2007), the number density of halos as a function of mass and a key probe of cosmology. For a review, see, e.g., White (2001) and references therein. Additionally, see Voit (2005) and references therein for a more observation-focused discussion.

I.1.3.2 Density and Concentration

The density profile of a DM halo is determined by radially binning the constituent particles into spherical shells, and determining the average density per shell, giving a characteristic $\rho(r)$. The most widely used model for the DM halo density profile is the NFW (Navarro et al., 1996) profile

$$\rho(r) = \frac{\rho_0}{\frac{r}{R_s} \left(1 + \frac{r}{R_s}\right)^2}, \quad (\text{I.1})$$

where ρ_0 is the characteristic density, and the scale radius R_s is the break radius between the inner $\sim r^{-1}$ and outer $\sim r^{-3}$ density profiles.

The halo density profile is quantified by the halo concentration $c \equiv R_{\text{vir}}/R_s$, where R_{vir} is the halo virial radius. Generally, at low redshift, low mass halos are more dense than high mass halos (Navarro et al., 1997a), and concentration decreases with redshift and increases in dense environments (Bullock et al., 2001). Neto et al. (2007) additionally find that concentration decreases with halo mass. Various additional studies have explored concentration's dependence on characteristics of the power spectrum (Eke et al., 2001), cosmological model (Macciò et al., 2008), redshift (Gao et al., 2008; Muñoz-Cuartas et al., 2011), and halo merger and mass accretion histories (Wechsler et al., 2002; Zhao et al., 2003, 2009). For halos at high redshift, Klypin et al. (2011) find that concentration reverses and increases with mass for high mass halos, while Prada et al. (2012) find that concentration's dependence on mass and redshift is more complicated and is better described through $\sigma(M, z)$, the rms fluctuation amplitude of the linear density field.

I.1.3.3 Substructure and Environment

Text goes here.

I.1.4 Baryonic Processes

Early-forming dark matter halos provide an incubator for the baryonic processes that transform the surrounding space and allow galaxies to form. Initial gas accretion can lead to

the formation of the first Pop-III stars (Couchman & Rees, 1986; Tegmark et al., 1997; Abel et al., 2000, 2002), which, upon their death, can collapse into the seeds for supermassive black holes (SMBHs) (Madau & Rees, 2001; Islam et al., 2003; Alvarez et al., 2009; Jeon et al., 2012) or enrich the surrounding medium with metals through supernovae (Heger & Woosley, 2002; Heger et al., 2003). The radiation from these early quasars (Shapiro & Giroux, 1987; Madau et al., 1999; Fan et al., 2001), Pop-III stars (Gnedin & Ostriker, 1997; Venkatesan et al., 2003; Alvarez et al., 2006), and proto-galaxy stellar populations (Bouwens et al., 2012; Kuhlen & Faucher-Giguère, 2012) all play a key role in contributing to the re-ionizing the universe by around $z = 6$ (Barkana & Loeb, 2001). Additionally, halo mergers can drastically increase the temperature of halo gas through shock heating, increasing X-ray luminosity (Sinha & Holley-Bockelmann, 2009), and contribute to the unbinding of gas to form the warm-hot intergalactic medium (Bykov et al., 2008; Sinha & Holley-Bockelmann, 2010; Tanaka et al., 2012).

I.1.4.1 The First Stars

Text goes here.

I.1.4.2 Supermassive Black Holes

Text goes here.

I.1.4.3 Enrichment and the The Intergalactic Medium

Text goes here.

I.1.4.4 Reionization

Text goes here.

I.2 Computational Theory

Text goes here.

I.2.1 Simulation Initialization

Text goes here.

I.2.1.1 Initial Conditions and the Surface of Last Scattering

Text goes here.

I.2.1.2 The Zel'dovich Approximation

Text goes here.

I.2.1.3 Second-order Lagrangian Perturbation Theory

Text goes here.

I.2.2 Dark Matter Halos in N-body Simulations

Text goes here.

I.2.2.1 Spherical Overdensity

Text goes here.

I.2.2.2 Friends-of-Friends

Text goes here.

CHAPTER II

Numerical Methods

Text goes here. Here, we will discuss the computational tools used, their inner workings, and how they are implemented to accomplish their purpose in the pipeline.

II.1 Initialization Code

Text goes here.

II.1.1 Sampling the Power Spectrum

Text goes here.

II.1.1.1 Cosmological Parameters

Text goes here.

II.1.1.2 Sampling

Text goes here.

II.1.2 Particle Displacement with ZA

Text goes here.

II.1.3 Particle Displacement with 2LPT

Text goes here.

II.2 Simulations with GADGET-2

We use the massively parallel TreeSPH cosmological N-body simulation code GADGET-2 for the dark matter simulations presented in this work. In this section, we begin with

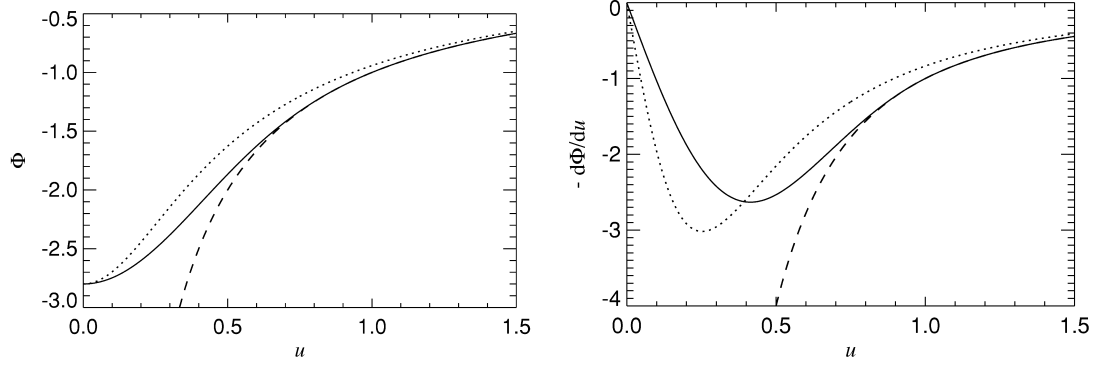


Figure II.1: Potential (*left*) and force (*right*) softening.

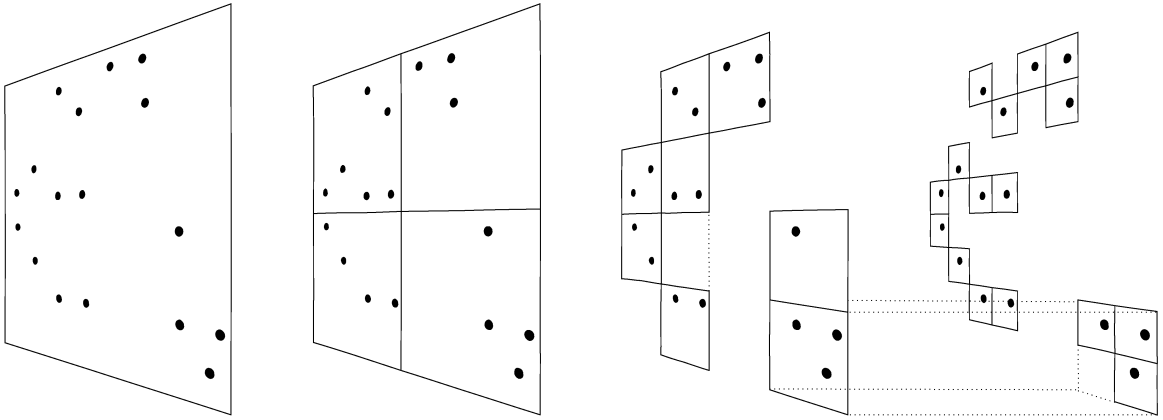


Figure II.2: Barnes-Hut oct-tree in two dimensions.

a discussion of the fundamental concepts presented in the original GADGET code, then proceed to the improvements made in the GADGET-2 code.

II.2.1 GADGET-2

Text goes here.

II.2.1.1 Force Calculation and Softening

Text goes here.

II.2.1.2 The Tree Code

Text goes here.

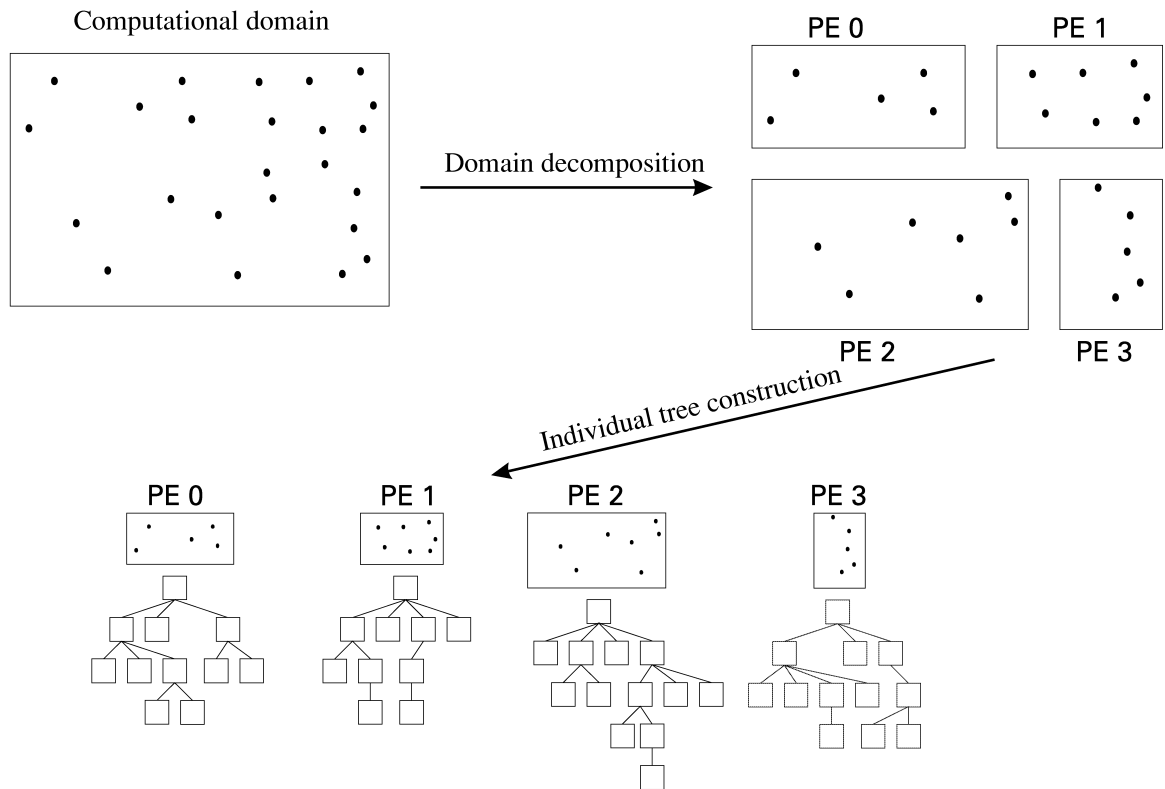


Figure II.3: Domain decomposition.

II.2.1.3 Parallelization

Text goes here.

II.2.1.4 Metrics

Text goes here.

II.2.2 Simulations

Text goes here.

II.3 Halo Finding with ROCKSTAR

Text goes here.

II.3.1 Halo Identification

Text goes here.

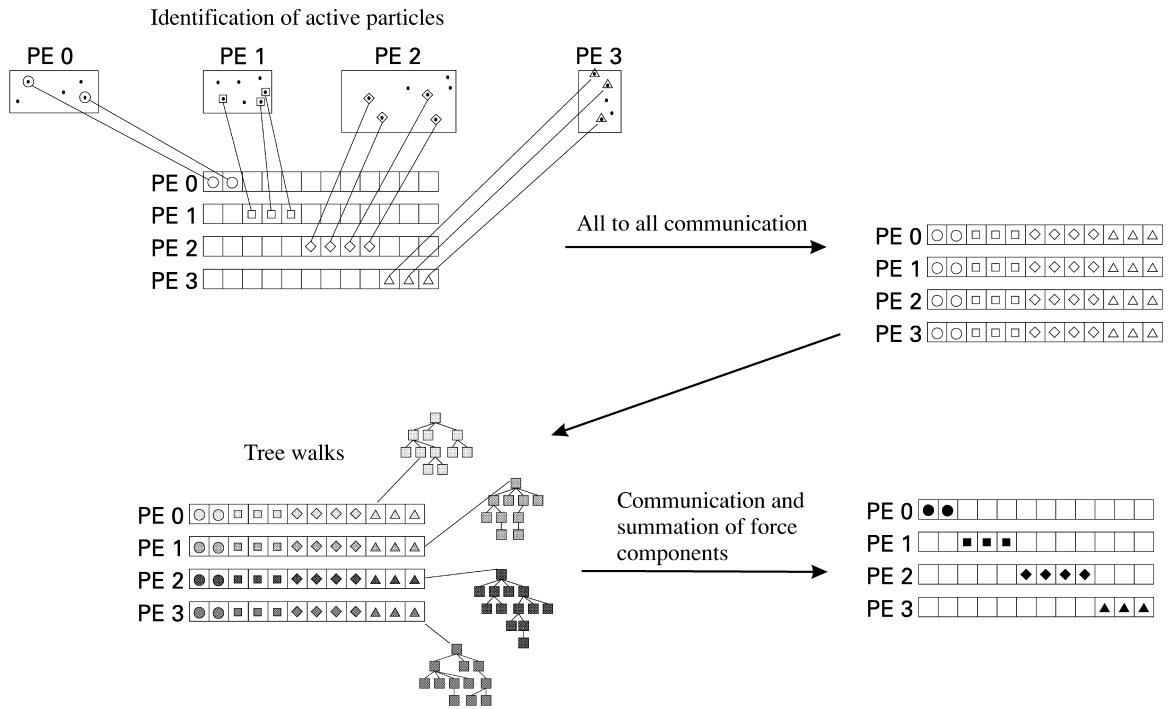


Figure II.4: Force parallelism.

II.3.2 Halo Properties

Text goes here.

II.3.2.1 Mass

Text goes here.

II.3.2.2 Concentration

Text goes here.

II.3.2.3 Relaxation Parameters

Text goes here.

II.3.3 Halo Catalogs

Text goes here.

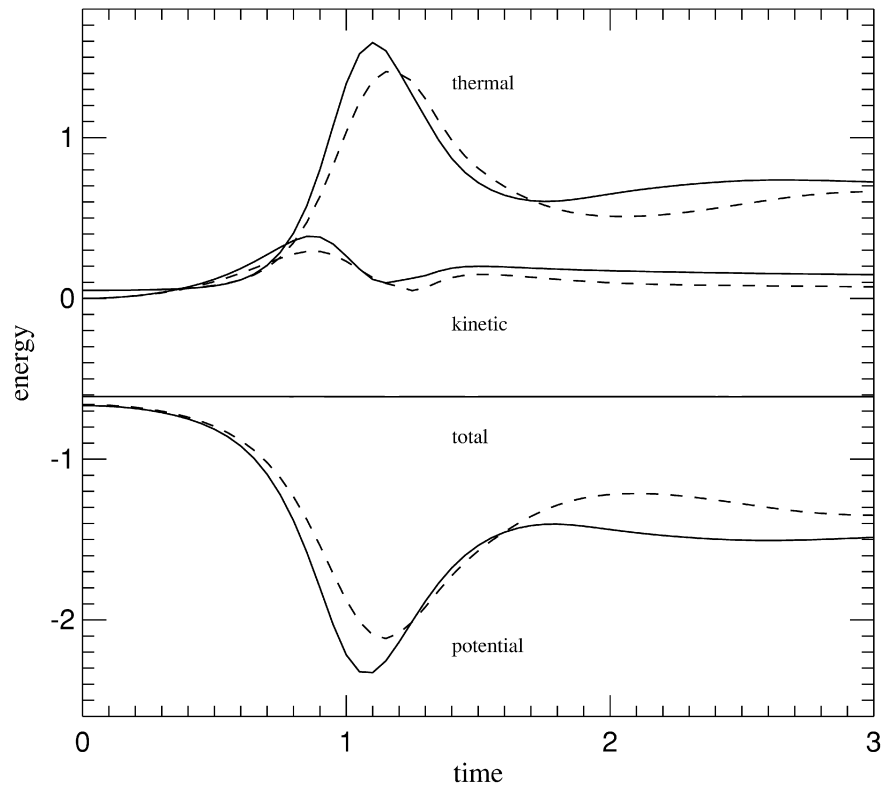


Figure II.5: Energy conservation for an initially isothermal gas sphere.

II.4 CROSSMATCH

Text goes here.

II.5 Analysis

Text goes here.

II.5.1 Non-linear Least Squares Fitting

Text goes here.

II.5.1.1 CurveFit

Text goes here.

II.5.1.2 Model Functions

Text goes here.

II.5.1.3 Initial Guess

Text goes here.

II.5.1.4 Fit Coefficients

Text goes here.

II.5.2 Halo Properties

Text goes here.

II.5.2.1 Rockstar Output and Post-processing

Text goes here.

II.5.2.2 Spherical Overdensity Halo Particles

Text goes here.

II.5.3 Density Profile Fitting

Text goes here.

II.5.3.1 Density Profiles

Text goes here.

II.5.3.2 Fitting

Text goes here.

II.5.3.3 Mass Profiles

Text goes here.

II.5.3.4 Characterization of Uncertainty

Text goes here.

II.5.3.5 Halo Properties

Text goes here.

II.5.4 Cross-matched Halo Catalog

Text goes here.

II.5.4.1 Aggregation

Text goes here.

II.5.4.2 Filtering

Text goes here.

II.5.5 Halo Comparison

Text goes here.

II.5.5.1 Match Verification

Text goes here.

II.5.5.2 Morphology

Text goes here.

II.5.5.3 Density Profiles

Text goes here.

II.5.6 Difference Distributions

Text goes here.

II.5.6.1 Histograms

Text goes here.

II.5.6.2 Fitting

Text goes here.

II.5.6.3 Mass Quartiles

Text goes here.

II.5.7 Redshift Trends

Text goes here.

II.5.7.1 Mean and Standard Deviation

Text goes here.

II.5.7.2 Skew

Text goes here.

II.5.7.3 Kurtosis

Text goes here.

The standard deviation of a function $f(x_1, x_2, \dots, x_n)$ is, in general, given by

$$s_f = \sqrt{\sum_x \left(\frac{\partial f}{\partial x} \right)^2 s_x^2} \quad (\text{II.1})$$

with summation over all independent variables x . The generalized normal distribution

$$f(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x-\mu|/\alpha)^\beta} \quad (\text{II.2})$$

with mean μ , scale parameter α , and shape parameter β , has excess kurtosis

$$\gamma_2 = \frac{\Gamma(5/\beta)\Gamma(1/\beta)}{\Gamma(3/\beta)^2} - 3. \quad (\text{II.3})$$

The gamma function

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \quad (\text{II.4})$$

has the first derivative

$$\Gamma'(x) = \Gamma(x) \psi_0(x) \quad (\text{II.5})$$

where the digamma function ψ_0 is the derivative of the logarithm of the gamma function and is given by

$$\psi_0(x) = \int_0^{\infty} \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt \quad (\text{II.6})$$

if the real part of x is positive.

We now apply (II.1) to (III.13) to find the standard deviation of the excess kurtosis:

$$s_{\gamma_2} = \sqrt{\left(\frac{d\gamma_2}{d\beta} \right)^2 s_{\beta}^2} \quad (\text{II.7})$$

$$= s_{\beta} \frac{d\gamma_2}{d\beta} \quad (\text{II.8})$$

$$= s_{\beta} \frac{d}{d\beta} \left[\frac{\Gamma(5/\beta) \Gamma(1/\beta)}{\Gamma(3/\beta)^2} - 3 \right]. \quad (\text{II.9})$$

Making the substitution $x = 1/\beta$ and $dx = -1/\beta^2 d\beta$, taking the derivative, and doing a bit

of algebra, we have:

$$s_{\gamma_2} = s_{\beta} \frac{d\gamma_2}{dx} \frac{dx}{d\beta} \quad (\text{II.10})$$

$$= s_{\beta} \left(-\frac{1}{\beta^2} \right) \frac{d}{dx} \left[\frac{\Gamma(5x)\Gamma(x)}{\Gamma(3x)} - 3 \right] \quad (\text{II.11})$$

$$= -s_{\beta} x^2 \left\{ \frac{\Gamma(3x)^2 \frac{d}{dx} [\Gamma(5x)\Gamma(x)] - \Gamma(5x)\Gamma(x) \frac{d}{dx} [\Gamma(3x)^2]}{\Gamma(3x)^4} \right\} \quad (\text{II.12})$$

$$= -s_{\beta} \frac{x^2}{\Gamma(3x)^4} \{ \Gamma(3x)^2 [5\Gamma(5x)\psi_0(5x)\Gamma(x) + \Gamma(5x)\Gamma(x)\psi_0(x)] - \Gamma(5x)\Gamma(x) [6\Gamma(3x)^2\psi_0(3x)] \} \quad (\text{II.13})$$

$$= s_{\beta} \frac{x^2}{\Gamma(3x)^4} \{ 6\Gamma(5x)\Gamma(3x)^2\Gamma(x)\psi_0(3x) - \Gamma(5x)\Gamma(3x)^2\Gamma(x) [5\psi_0(5x) + \psi_0(x)] \} \quad (\text{II.14})$$

$$= s_{\beta} \frac{x^2}{\Gamma(3x)^4} \{ \Gamma(5x)\Gamma(3x)^2\Gamma(x) [6\psi_0(3x) - 5\psi_0(5x) - \psi_0(x)] \} \quad (\text{II.15})$$

$$= s_{\beta} x^2 \frac{\Gamma(5x)\Gamma(x)}{\Gamma(3x)^2} [6\psi_0(3x) - 5\psi_0(5x) - \psi_0(x)]. \quad (\text{II.16})$$

Substituting back in for x and recognizing an occurrence of γ_2 , we have the result

$$s_{\gamma_2} = s_{\beta} \frac{1}{\beta^2} (\gamma_2 + 3) [6\psi_0(3/\beta) - 5\psi_0(5/\beta) - \psi_0(1/\beta)] \quad (\text{II.17})$$

with which we can find the uncertainty in the kurtosis given the value and uncertainty of the shape parameter β .

II.5.8 2-D Histograms

Text goes here.

II.5.8.1 Binning and Plotting

Text goes here.

II.5.8.2 Fitting

Text goes here.

II.5.9 Alternate Difference Distributions

Text goes here.

II.5.9.1 Equivalent Displacement

Text goes here.

II.5.9.2 Redshift Trends

Text goes here.

II.6 Automation

Text goes here.

CHAPTER III

Exploring Dark Matter Halo Populations in 2LPT and ZA Simulations

We study the structure and evolution of dark matter halos from $z = 300$ to $z = 6$ for two cosmological N-body simulation initialization techniques. While the second order Lagrangian perturbation theory (2LPT) and the Zel'dovich approximation (ZA) both produce accurate present day halo mass functions, earlier collapse of dense regions in 2LPT can result in larger mass halos at high redshift. We explore the differences in dark matter halo mass and concentration due to initialization method through three 2LPT and three ZA initialized cosmological simulations. We find that 2LPT induces more rapid halo growth, resulting in more massive halos compared to ZA. This effect is most pronounced for high mass halos and at high redshift, with a fit to the mean normalized difference between 2LPT and ZA halos as a function of redshift of $\mu_{\Delta M_{\text{vir}}} = (7.88 \pm 0.17) \times 10^3 z - (3.07 \pm 0.14) \times 10^{-2}$. Halo concentration is, on average, largely similar between 2LPT and ZA, but retains differences when viewed as a function of halo mass. For both mass and concentration, the difference between typical individual halos can be very large, even for symmetrically distributed quantities, highlighting the shortcomings of ZA-initialized simulations for high- z halo population studies.

III.1 Introduction

The pre-reionization epoch is a time of significant evolution of early structure in the universe. Rare density peaks in the otherwise smooth dark matter (DM) sea lead to the collapse and formation of the first dark matter halos. For example, at $z = 20$, $10^7 M_{\odot}$ halos are $\sim 4\sigma$ peaks, and $10^8 M_{\odot}$ halos, candidates for hosting the first supermassive black hole seeds, are $\sim 5\sigma$ peaks.

These early-forming dark matter halos provide an incubator for the baryonic processes that transform the surrounding space and allow galaxies to form. Initial gas accretion can

lead to the formation of the first Pop-III stars (Couchman & Rees, 1986; Tegmark et al., 1997; Abel et al., 2000, 2002), which, upon their death, can collapse into the seeds for supermassive black holes (SMBHs) (Madau & Rees, 2001; Islam et al., 2003; Alvarez et al., 2009; Jeon et al., 2012) or enrich the surrounding medium with metals through supernovae (Heger & Woosley, 2002; Heger et al., 2003). The radiation from these early quasars (Shapiro & Giroux, 1987; Madau et al., 1999; Fan et al., 2001), Pop-III stars (Gnedin & Ostriker, 1997; Venkatesan et al., 2003; Alvarez et al., 2006), and proto-galaxy stellar populations (Bouwens et al., 2012; Kuhlen & Faucher-Giguère, 2012) all play a key role in contributing to the re-ionizing the universe by around $z = 6$ (Barkana & Loeb, 2001). Additionally, halo mergers can drastically increase the temperature of halo gas through shock heating, increasing X-ray luminosity (Sinha & Holley-Bockelmann, 2009), and contribute to the unbinding of gas to form the warm-hot intergalactic medium (Bykov et al., 2008; Sinha & Holley-Bockelmann, 2010; Tanaka et al., 2012).

While a number of parameters are required to fully characterize a DM halo, a first-order description can be obtained from its mass and density profile. There are a number of ways to define a halo’s mass. This becomes significant for mass-sensitive studies, such as the halo mass function (Press & Schechter, 1974; Reed et al., 2007; Heitmann et al., 2006; Lukić et al., 2007), the number density of halos as a function of mass and a key probe of cosmology. For a review, see, e.g., White (2001) and references therein. Additionally, see Voit (2005) and references therein for a more observation-focused discussion.

From a simulation standpoint, the two most common ways to obtain halo mass are to define either spherical overdensity halos or friends-of-friends (FOF) halos. The spherical overdensity method identifies regions above a certain density threshold, either with respect to the critical density $\rho_c = 3H^2/8\pi G$ or the background density $\rho_b = \Omega_m \rho_c$, where Ω_m is the matter density of the universe. The mass is then the mass enclosed in a sphere of some radius with mean density $\Delta\rho_c$, where Δ commonly ranges from ~ 100 to ~ 500 . Alternatively, the FOF method finds particle neighbors and neighbors of neighbors defined

to be within some separation distance (Einasto et al., 1984; Davis et al., 1985). Halo mass, then, is simply the sum of the masses of the constituent particles.

The density profile of a DM halo is determined by radially binning the constituent particles into spherical shells, and determining the average density per shell, giving a characteristic $\rho(r)$. The most widely used model for the DM halo density profile is the NFW (Navarro et al., 1996) profile

$$\rho(r) = \frac{\rho_0}{\frac{r}{R_s} \left(1 + \frac{r}{R_s}\right)^2}, \quad (\text{III.1})$$

where ρ_0 is the characteristic density, and the scale radius R_s is the break radius between the inner $\sim r^{-1}$ and outer $\sim r^{-3}$ density profiles.

The halo density profile is quantified by the halo concentration $c \equiv R_{\text{vir}}/R_s$, where R_{vir} is the halo virial radius. Generally, at low redshift, low mass halos are more dense than high mass halos (Navarro et al., 1997a), and concentration decreases with redshift and increases in dense environments (Bullock et al., 2001). Neto et al. (2007) additionally find that concentration decreases with halo mass. Various additional studies have explored concentration's dependence on characteristics of the power spectrum (Eke et al., 2001), cosmological model (Macciò et al., 2008), redshift (Gao et al., 2008; Muñoz-Cuartas et al., 2011), and halo merger and mass accretion histories (Wechsler et al., 2002; Zhao et al., 2003, 2009). For halos at high redshift, Klypin et al. (2011) find that concentration reverses and increases with mass for high mass halos, while Prada et al. (2012) find that concentration's dependence on mass and redshift is more complicated and is better described through $\sigma(M, z)$, the rms fluctuation amplitude of the linear density field.

The subtle $\mathcal{O}(10^{-5})$ density perturbations in place at the CMB epoch are vulnerable to numerical noise and intractable to simulate directly. Instead, a displacement field is applied to the particles to evolve them semi-analytically, nudging them from their initial positions to an approximation of where they should be at a more reasonable starting redshift for the numerical simulation. Starting at a later redshift saves computation time as well as avoiding

interpolation systematics and round-off errors (Lukić et al., 2007).

The Zel’dovich approximation (Zel’dovich, 1970) and 2nd-order Lagrangian Perturbation Theory (Buchert, 1994; Buchert et al., 1994; Bouchet et al., 1995; Scoccimarro, 1998) are the two canonical frameworks for the initial particle displacement involved in generating simulation initial conditions. Zel’dovich approximation (ZA, hereafter) initial conditions (Klypin & Shandarin, 1983; Efstathiou et al., 1985) displace initial particle positions and velocities via a linear field, while 2nd-order Linear Perturbation Theory (2LPT, hereafter) initial conditions (Scoccimarro, 1998; Sirko, 2005; Jenkins, 2010) add a second-order correction term to the expansion of the displacement field.

Following Jenkins (2010), we briefly outline the second-order Lagrangian perturbation theory and compare it to the Zel’dovich approximation. In 2LPT, a displacement field $\Psi(q)$ is applied to the initial positions q to yield the Eulerian final comoving positions

$$x = q + \Psi. \quad (\text{III.2})$$

The displacement field is given in terms of two potentials $\phi^{(1)}$ and $\phi^{(2)}$ by

$$x = q - D_1 \nabla_q \phi^{(1)} + D_2 \nabla_q \phi^{(2)}, \quad (\text{III.3})$$

with linear growth factor D_1 and second-order growth factor $D_2 \approx -3D_1^2/7$. The subscripts q refer to partial derivatives with respect to the Lagrangian coordinates q . Likewise, the comoving velocities are given, to second order, by

$$v = -D_1 f_1 H \nabla_q \phi^{(1)} + D_2 f_2 H \nabla_q \phi^{(2)}, \quad (\text{III.4})$$

with Hubble constant H and $f_i = d \ln D_i / d \ln a$, with expansion factor a . The relations $f_1 \approx \Omega^{5/9}$ and $f_2 \approx 2\Omega^{6/11}$, with matter density Ω , apply for flat models with a non-zero cosmological constant (Bouchet et al., 1995). The f_1 , f_2 , and D_2 approximations here are

very accurate for most actual Λ CDM initial conditions, as Ω is close to unity at high starting redshift (Jenkins, 2010). We may derive $\phi^{(1)}$ and $\phi^{(2)}$ by solving a pair of Poisson equations

$$\nabla_q^{(1)}(q) = \delta^{(1)}(q), \quad (\text{III.5})$$

with linear overdensity $\delta^{(1)}(q)$, and

$$\nabla_q^{(2)}(q) = \delta^{(2)}(q). \quad (\text{III.6})$$

The second order overdensity $\delta^{(2)}(q)$ is related to the linear overdensity field by

$$\delta^{(2)}(q) = \sum_{i>j} \left\{ \phi_{,ii}^{(1)}(q) \phi_{,jj}^{(1)}(q) - \left[\phi_{,ij}^{(1)}(q) \right]^2 \right\}, \quad (\text{III.7})$$

where $\phi_{,ij} \equiv \partial^2 \phi / \partial q_i \partial q_j$. For initial conditions from the Zel’dovich approximation, or first-order Lagrangian initial conditions, the $\phi^{(2)}$ terms of Equations III.3 and III.4 are ignored.

Cosmological simulations that follow the initial collapse of dark matter density peaks into virialized halos often neglect to consider the nuances of initialization method. Non-linear decaying modes, or transients, will be damped as $1/a$ in ZA. In 2LPT, however, transients are damped more quickly as $1/a^2$. It should be expected, then, that structure in 2LPT will be accurate after fewer e -folding times than in ZA (Scoccimarro, 1998; Crocce et al., 2006; Jenkins, 2010). The practical result is that high- σ DM density peaks at high redshift are suppressed in ZA compared with 2LPT for a given starting redshift (Crocce et al., 2006).

While differences in ensemble halo properties, such as the halo mass function, between simulation initialization methods are mostly washed away by $z = 0$ (Scoccimarro, 1998), trends at earlier redshifts are less studied (Lukić et al., 2007). In this paper, we explore the effects of ZA and 2LPT on the evolution of halo populations at high redshift. It is thought that 2LPT allows initial DM overdensities to get a “head start” compared with ZA, allowing

earlier structure formation, more rapid evolution, and larger possible high-mass halos for a given redshift. We explore this possibility by comparing halo properties in (otherwise identical) simulations initialized with ZA and 2LPT.

We discuss the simulations, halo finding, and analysis methods in Section III.2, results in Section III.3, implications, caveats, and future work in Section III.4, and finally summarize our results and conclude in Section IV.4.

III.2 Numerical Methods

We use the Nbody tree/SPH code GADGET-2 (Springel et al., 2001; Springel, 2005) to evolve six dark matter-only cosmological volumes from $z_{\text{start}} = 300$ to $z = 6$ in a Λ CDM universe. Each simulation is initialized using WMAP-5 (Komatsu et al., 2009) parameters. For each of the three simulation pairs, we directly compare 2LPT and ZA by identically sampling the CMB transfer function and displacing the initial particle positions to the same starting redshift using 2LPT and ZA. The three sets of simulations differ only by the initial phase sampling random seed. Each volume contains 512^3 particles in a $10 h^{-1}$ Mpc box. Full simulation details are discussed in Holley-Bockelmann et al. (2012).

One facet often overlooked when setting up an N-body simulation is an appropriate starting redshift, determined by box size and resolution (Lukić et al., 2007). Initialization with 2LPT allows for a later starting redshift compared with an equivalent ZA-initialized simulation. However, many ZA simulations do not take this into account, starting from too late of an initial redshift (Crocce et al., 2006; Jenkins, 2010). In order to characterize an appropriate starting redshift, the relation between the initial rms particle displacement and mean particle separation must be considered. The initial rms displacement Δ_{rms} is given by

$$\Delta_{\text{rms}}^2 = \frac{4\pi}{3} \int_{k_f}^{k_{\text{Ny}}} P(k, z_{\text{start}}) dk, \quad (\text{III.8})$$

where $k_f = 2\pi/L_{\text{box}}$ is the fundamental mode, L_{box} is the simulation box size, $k_{\text{Ny}} = \frac{1}{2}Nk_f$ is the Nyquist frequency of an N^3 simulation, and $P(k, z_{\text{start}})$ is the power spectrum at

starting redshift z_{start} . In order to avoid the “orbit crossings” that reduce the accuracy of the initial conditions, Δ_{rms} must be some factor smaller than the mean particle separation $\Delta_p = L_{\text{box}}/N$ (Holley-Bockelmann et al., 2012). For example, making orbit crossing a $\sim 10\sigma$ event imposes $\Delta_{\text{rms}}/\Delta_p = 0.1$. However, for small-volume, high-resolution simulations, this quickly leads to impractical starting redshifts. Continuing our example, satisfying $\Delta_{\text{rms}}/\Delta_p \sim 0.1$ for a $10h^{-1}$ Mpc, 512^3 simulation suggests $z_{\text{start}} \approx 799$. Starting at such a high redshift places such a simulation well into the regime of introducing errors from numerical noise caused by roundoff errors dominating the smooth potential. A more relaxed requirement of $\Delta_{\text{rms}}/\Delta_p = 0.25$ yields $z_{\text{start}} = 300$, which we adopt for this work.

For each of our six simulations, we use the 6-D phase space halo finder code ROCKSTAR (Behroozi et al., 2013) to identify spherical overdensity halos at each timestep. ROCKSTAR follows an adaptive hierarchical refinement of friends-of-friends halos in 6-D phase space, allowing determination of halo properties such as halo mass, position, virial radius, internal energy, and number of subhalos. ROCKSTAR tracks halos down to a threshold of around 20 particles, but we use a more conservative 100 particle threshold for our analysis. We use all particles found within the virial radius to define our halos and their properties.

We identify matching halos based on the highest fraction of matching particles contained in each at any given timestep. We remove subhalo matches (i.e. a halo must not be contained within another halo) and halo pairs with fewer than 100 particles in either 2LPT or ZA. We are left with approximately 60,000 total halo pairs for our three boxes at $z = 6$. With halo catalogues matched between simulations, we can compare properties of individual corresponding halos. To mitigate the effects of cosmic variance on our small volumes, we “stack” the three simulation boxes for each initialization method, and combine the halos from each into one larger sample in our analysis.

Halo concentration is derived from ROCKSTAR’s output for R_s and R_{vir} . Here, R_{vir} is the virial radius as defined by Bryan & Norman (1998). Figure III.1 makes evident the difficulty in fitting density profiles and obtaining concentration measurements for typical

realistic halos. Large substructure, as displayed by the ZA halo, can disrupt the radial symmetry of the halo and cause significant deviations in the density profile. Centering can also be an issue in these cases. Due to these complications, there are a number of approaches for finding halo concentrations (Prada et al., 2012), but for consistency, we use the values derived from ROCKSTAR’s fitting for our concentration measurements.

At each simulation snapshot, we measure and compare a number of parameters for halos in both 2LPT and ZA simulations. For each quantity q , we create histograms of Δq , the normalized difference in q between halos in the 2LPT and ZA simulations, defined as

$$\Delta q = \frac{q_{2\text{LPT}} - q_{\text{ZA}}}{q_{\text{avg}}}, \quad (\text{III.9})$$

where $q_{\text{avg}} = \frac{1}{2}(q_{2\text{LPT}} + q_{\text{ZA}})$. For each of these, we fit the Δq histograms with a generalized normal distribution (Nadarajah, 2005) with the probability density function

$$f(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x-\mu|/\alpha)^\beta}, \quad (\text{III.10})$$

where μ is the mean, α is the scale parameter, β is the shape parameter, and Γ is the gamma function

$$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx. \quad (\text{III.11})$$

The shape parameter β is restricted to $\beta \geq 1$. This allows the distribution to potentially vary from a Laplace distribution ($\beta = 1$) to a uniform distribution ($\beta = \infty$) and includes the normal distribution ($\beta = 2$). The distribution has variance

$$\sigma^2 = \frac{\alpha^2 \Gamma(3/\beta)}{\Gamma(1/\beta)} \quad (\text{III.12})$$

and excess kurtosis

$$\gamma_2 = \frac{\Gamma(5/\beta)\Gamma(1/\beta)}{\Gamma(3/\beta)^2} - 3. \quad (\text{III.13})$$

The distribution is symmetric, and thus has no skewness by definition. As such, the values for skew presented below are measured directly from the data.

As our fitting distributions are symmetrical and skew must therefore be measured directly from the data, in order to derive uncertainties for skew, we measure the skew of the distributions for each of our three simulation boxes individually as well as for the single stacked data set. Uncertainty in skew is then simply the standard deviation of the mean of the skew of the three individual boxes.

Determining the uncertainty in the kurtosis is slightly more involved, as kurtosis is determined by a transformation of the generalized normal distribution's shape parameter β according to Equation III.13. Following the standard procedure for propagation of uncertainty, we calculate the standard deviation of the kurtosis as

$$s_{\gamma_2} = \sqrt{\left(\frac{d\gamma_2}{d\beta}\right)^2 s_{\beta}^2} \quad (\text{III.14})$$

$$= s_{\beta} \frac{d}{d\beta} \left[\frac{\Gamma(5/\beta)\Gamma(1/\beta)}{\Gamma(3/\beta)^2} - 3 \right]. \quad (\text{III.15})$$

The derivative of the gamma function is

$$\Gamma'(x) = \Gamma(x)\psi_0(x), \quad (\text{III.16})$$

where the digamma function ψ_0 is the derivative of the logarithm of the gamma function and is given by

$$\psi_0(x) = \int_0^{\infty} \left(\frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt \quad (\text{III.17})$$

if the real part of x is positive. Now, taking the derivative of γ_2 and doing a bit of algebra gives us

$$s_{\gamma_2} = s_{\beta} \frac{1}{\beta^2} (\gamma_2 + 3) [6\psi_0(3/\beta) - 5\psi_0(5/\beta) - \psi_0(1/\beta)], \quad (\text{III.18})$$

with which we can find the uncertainty in the kurtosis given the value and uncertainty of

the shape parameter β estimated from the least squares fit routine.

In addition to distributions of Δq , we also consider distributions of

$$\Delta'q = \frac{q_{2\text{LPT}} - q_{\text{ZA}}}{q_{\text{ZA}}} \quad (\text{III.19})$$

in order to better quantify the fraction of halos differing by a given amount between 2LPT and ZA simulations. This distribution is inherently non-symmetrical, and is only defined for $\Delta'q \geq -1$ for positive quantities like mass and concentration. In order to consider halo pairs that differ by a certain amount in either direction (e.g. pairs that differ by 10%, whether larger in 2LPT or ZA), a relation for equivalent displacement is required. Rearranging Equation III.19 yields

$$q_{2\text{LPT}} = (\Delta'q + 1)q_{\text{ZA}}, \quad (\text{III.20})$$

and making the substitution $x = \Delta'q + 1$ gives us

$$q_{2\text{LPT}} = xq_{\text{ZA}}. \quad (\text{III.21})$$

For a given x_1 , we want an x_2 such that $x_2 = 1/x_1$. Substituting now for x_1 and x_2 and rearranging gives us

$$\Delta'q_2 = \frac{1}{\Delta'q_1 + 1} - 1, \quad (\text{III.22})$$

the value for which a halo pair with a larger q in ZA would differ by the same factor as a halo pair with a larger q in 2LPT where $\Delta'q = \Delta'q_1$.

III.3 Results

With our catalog of matched dark matter halos, we directly compare differences in halo properties arising from initialization with 2LPT vs ZA. We consider halos on a pair-by-pair basis as well as the entire sample as a whole. Overall, we find 2LPT halos undergo more growth at a given redshift than their ZA counterparts.

III.3.1 Individual halo pairs

We compare large scale morphologies, density profiles, and various other halo properties for halo pairs on an individual halo-by-halo basis for several of the most massive halos. Morphologies appear similar for most halos, indicating good halo matches between simulations. However, many pairs display differences in central morphology, such as the number and separation of central density peaks. We interpret these cases to be examples of differences in merger epochs, in which case one halo may still be undergoing a major merger, while its companion is in a more relaxed post-merger state. We give an example of one such pair at $z = 6$ in Figure III.1. The top two rows show density projections of the nuclear regions for a large 2LPT and matching ZA halo (first and second rows, respectively). We find the ZA halo to contain two distinct density peaks with a separation of ~ 10 kpc, while the 2LPT halo displays only a single core. On the third and fourth rows, we plot the density profiles of the same two halos (2LPT and ZA, respectively). Here, with nearly identical virial radii, it is readily seen that the 2LPT halo is more concentrated than the ZA halo.

III.3.2 Difference distributions of halo properties

For the halo population as a whole, we consider distributions of halo virial mass M_{vir} and concentration c . We plot histograms of ΔM_{vir} and Δc in the left and right columns, respectively, of Figure III.2 for three representative timesteps at redshifts of $z = 14.7$, $z = 10.3$, and $z = 6.0$. For each panel, the blue histogram features the entire halo sample, and the smaller gray-filled green histogram displays only the top 25% most massive halos, ordered by 2LPT mass. Fits to the primary histograms are overplotted as red dashed curves.

Throughout the simulation, we find a tendency for 2LPT halos to be more massive. At $z = 15$, the mean of the ΔM_{vir} distribution is $(9.3 \pm 1.2) \times 10^{-2}$. The mean is consistently positive (heavier 2LPT halos) and is most displaced from zero at high redshift. The peak of the distribution gradually moves closer to zero as we progress in redshift. We find the least difference between paired halos for the final snapshot at $z = 6$, with $\mu_{\Delta M_{\text{vir}}} =$

$$(1.79 \pm 0.31) \times 10^{-2}.$$

The higher-order moments of the ΔM_{vir} distribution are of interest as well, as we find significant deviation from a Gaussian distribution. As we use the symmetrical generalized normal distribution as our fit function, the skewness of the data is unable to be measured from the fit itself. However, a qualitative deviation from symmetry can be readily observed. By $z = 6$, we end up with a rather symmetrical distribution, with both sides of the histogram equally well described by our fit. However, at higher redshift, we note a marked increase in skewness and deviation from this symmetry. As redshift increases, we observe an increasing difference between the fit curve and the bins to the left of the histogram peak.

We find the distributions to be much closer to a Laplace distribution than a Gaussian, with shape parameter consistently sitting at or very close to $\beta = 1$. Compared to a Gaussian distribution, the larger excess kurtosis implies a narrower central peak and heavier outlying tails. Our fit constrains $\beta \geq 1$, so the kurtosis of the data itself could potentially be higher than the fit implies.

We find no overall preference for more concentrated 2LPT or ZA halos. In contrast to the ΔM_{vir} histograms, Δc shows very little deviation from symmetry about zero. Throughout the simulation, we find the distributions to have a mean close to zero and negligible skew. The widths of the distributions are much wider than those for ΔM_{vir} , with an order of magnitude difference by $z = 6$. As with mass, concentration histograms are sharply peaked with heavy tails, implying a tendency for halo pairs to move towards the extremes of either very similar or very discrepant concentrations.

III.3.3 Trends with redshift

In Figure III.3, we more quantitatively assess the evolution of our various trends hinted at in Figure III.2. Here, we plot the mean, root mean square (RMS), standard deviation, skew, and kurtosis for ΔM_{vir} and Δc as functions of redshift. Uncertainty in the mean is estimated directly by the least squares fitting routine.

Table III.1: Coefficients for linear least squares fits from Figure III.3.

	ΔM_{vir}	Δc
A	$(7.88 \pm 0.17) \times 10^{-3}$	$(3.62 \pm 0.95) \times 10^{-3}$
B	$(-3.07 \pm 0.14) \times 10^{-2}$	$(-2.34 \pm 0.84) \times 10^{-2}$

The mean for ΔM_{vir} is positive and highest at high redshift, trending toward zero by the end of the simulation. Distributions for Δc retain means close to and consistent with zero. Standard deviation decreases slightly for both ΔM_{vir} and Δc . From $z = 15$ to $z = 6$, standard deviation falls from $(9.0 \pm 1.5) \times 10^{-2}$ to $(6.08 \pm 0.31) \times 10^{-2}$ for ΔM_{vir} and from 0.73 ± 0.11 to 0.551 ± 0.026 for Δc .

We find least square linear fits for both mean ΔM_{vir} vs z and mean Δc vs z . Coefficients for slope A and y-intercept B for the fit equation $\mu = Az + B$ are given in Table III.1 for both cases. We find a significant trend for ΔM_{vir} , with a slope $\sim 46\sigma$ from zero. Conversely, the slope for Δc is much smaller and, considering the larger spread of the underlying distributions, can be considered negligible. For ΔM_{vir} , the y-intercept coefficient B likely has little meaning in terms of the actual behavior at $z = 0$, as we expect the trend to level out at later redshift.

We do note, however, that the mean can be deceiving as an indicator of total difference between halo populations, especially when it is close to zero as with concentration. It should be noted that while the mean can indicate a lack of average difference between the whole sample of 2LPT and ZA halos, there can still be very large discrepancies between many individually paired halos. We visualize this by plotting the RMS of ΔM_{vir} and Δc , which is plotted as a green dotted line. Unlike the mean, standard deviation, and kurtosis, which are measured from fits to the histograms, RMS is measured directly from the data and is not dependent on fitting. The large RMS values are indicative of how much overall difference can arise between 2LPT and ZA halos, even though the differences may average to zero when considering the entire population. The RMS for both ΔM_{vir} and Δc starts highest at high redshift—0.19 for ΔM_{vir} and 0.57 for Δc at $z = 15$ —and steadily decreases

throughout the simulation, reaching minimums of 0.11 for ΔM_{vir} and 0.45 for Δc by $z = 6$.

Additionally, it is of interest to consider the percentage of halo pairs that are “wrong” at some given time, regardless of whether the quantity is higher in 2LPT or ZA. For example, if we count halos outside a slit of $\varepsilon = 10\%$ around $\Delta q = 0$, we find that by $z = 6$, 14.6% of halo pairs still have substantially mismatched masses, and 74.3% have mismatched concentrations. It is evident that a substantial percentage of halo pairs can have markedly different growth histories, even when there is little or no offset in the ensemble halo population average.

Kurtosis is consistently large for both mass and concentration, with a slight increasing trend throughout the simulation for concentration. It reaches maximum values of 17.5 ± 2.4 at redshift 10 for ΔM_{vir} and 15.4 ± 1.0 at the end of the simulation at redshift 6 for Δc . Skew is positive for much of the simulation for mass, but is much smaller for concentration. We find average skews of 0.39 ± 0.29 for ΔM_{vir} and 0.045 ± 0.028 for Δc . These higher moment deviations from Gaussianity hint at the non-linear dynamics at play in halo formation.

The narrow peak and heavy tails of the distribution may indicate a fair amount of sensitivity to initial differences in halo properties, in that halo pairs that start out within a certain range of the mean are more likely move closer to the mean, while pairs that are initially discrepant will diverge even further in their characteristics. This is indicative of the non-linear gravitational influence present during halo evolution, and is further supported by a kurtosis that increases with time.

The skew at high redshift for ΔM_{vir} may give another hint at the non-linear halo formation process. Runaway halo growth causes more massive halos to favor faster mass accretion and growth. The positively skewed distributions show a picture of 2LPT halo growth in which initial differences in mass are amplified the most readily in the earliest forming and most massive halos, again indicating the extra kick-start to halo growth provided by 2LPT initialization. While the slight decrease in skew with redshift may be counter-intuitive to this notion, it is likely that the large number of newly formed halos begin to mask the signal

from the smaller number of large halos displaying this effect.

III.3.4 Trends with halo mass

We consider ΔM_{vir} and Δc as a function of average halo mass $M_{\text{vir,avg}} = (M_{\text{vir,2LPT}} + M_{\text{vir,ZA}})/2$ in Figure III.4. The data is binned on a 2-D grid with a logarithmic color map for three representative timesteps. A linear fit to the data is overplotted in red, and a dotted blue line is provided at $\Delta M_{\text{vir}} = 0$ and $\Delta c = 0$ to guide the eye.

We find that ΔM_{vir} tends to increase with increasing $M_{\text{vir,avg}}$, a trend that is, again, most pronounced at high redshift. 2LPT halos are consistently more massive than their ZA counterparts, with the difference increasing with average halo mass. While less massive halo pairs have a larger spread in the difference in 2LPT and ZA mass, more massive halo pairs are consistently heavier in 2LPT than in ZA. At redshift 15, the least squares fit to the data produces the fit equation $\Delta M_{\text{vir}} = 5.6 \times 10^{-2} M_{\text{vir,avg}} - 0.33$. The slope of the fit line trends towards zero as we progress in redshift, with little average mass dependence and a fit of $\Delta M_{\text{vir}} = 6.4 \times 10^{-3} M_{\text{vir,avg}} - 2.5 \times 10^{-2}$ by $z = 6$.

We find a small trend for more massive halo pairs to be more concentrated in ZA, but this trend is weaker than for ΔM_{vir} . The fit equations for $z = 15$ and $z = 6$ are $\Delta c = -5.3 \times 10^{-2} M_{\text{vir,avg}} - 0.45$ and $\Delta c = -9.3 \times 10^{-3} M_{\text{vir,avg}} - 8.9 \times 10^{-2}$, respectively. The negative slope might be expected, as halo concentration is expected to decrease with increasing mass, at least at later redshift (Neto et al., 2007), and we find high mass halos to be more massive in 2LPT than in ZA. However, the dependence of concentration on mass and redshift at high redshift is more complicated (Klypin et al., 2011; Prada et al., 2012). The data have a larger variance than ΔM_{vir} , and fits have an overall shallower slope. Mass dependence all but disappears by $z = 6$. To reconcile these trends with the symmetrical concentration distributions of Figure III.2, we note that the trends in mass may be hidden by integration across the entire mass range and still result in overall Δc distributions symmetric about zero.

III.3.5 Alternate fractional difference distributions

In Figure III.5, we plot, as functions of redshift, statistics derived from the alternate fractional difference distributions $\Delta' M_{\text{vir}}$ and $\Delta' c$ (see Equation III.19). In the left column, we plot the $\Delta' q$ of the peak of the distribution along with the $\Delta' q$ where various percentages of the halo pairs fall at or above $\Delta' q$.

As the $\Delta' q$ value of peak of the distribution is the location of the mode, it represents the most typical halo pair. While concentration differences remain close to zero throughout the simulation, mass difference peak moves from a $\Delta' M_{\text{vir}}$ of 8.7×10^{-2} at $z = 15$ to 2.9×10^{-2} at $z = 6$. The 1% of halo pairs with the largest excess 2LPT mass are at least 1.97 times ZA mass at $z = 15$ and 1.45 times ZA mass at $z = 6$. For concentration, the 1% most 2LPT concentrated halo pairs differ by at least a factor of 6.00 at $z = 15$ and 3.73 at $z = 6$.

In the right column of Figure III.5, we plot the fraction of halos f_h that fall outside various $\Delta' q$ values. The solid lines represent halo pairs that have $\Delta' q$ greater than or equal to the listed values, i.e., the fraction of halo pairs where the 2LPT halo has a virial mass or concentration that is at least 1.1, 1.5, 2.0, or 5.0 times that of its corresponding ZA halo. The dashed lines additionally count halos with $\Delta' q$ at or below the corresponding equivalent displacement (see Equation III.22) and represent the fraction of halo pairs where one halo has a virial mass or concentration at least 1.1, 1.5, 2.0, or 5.0 times that of its companion, regardless of whether the 2LPT or ZA value is higher.

We find that 50% of halo pairs are at least 10% more massive in 2LPT at $z = 15$. By $z = 6$, this has fallen to 10%. Furthermore, 0.81% are at least twice as massive in 2LPT at $z = 15$, and by $z = 6$, this has only reduced to 0.26%. Halos in 2LPT are at least twice as concentrated as their ZA counterparts for at least 12% of the halo population at $z = 15$ and at least 7.8% of the population by $z = 6$. Halo pairs that are at least 5 times as concentrated in 2LPT make up 1.3% at $z = 15$ and 0.26% at $z = 6$.

When we additionally consider halo pairs that are less than or equal to the equivalent displacement, i.e. pairs where either the 2LPT or ZA halos has the higher mass or concen-

tration, we include an even larger percentage of the population. We find 54% of the halo pairs differ in mass by at least 10% at $z = 15$, with 16% differing by $z = 6$. Halos that are at least twice as massive in either 2LPT or ZA account for 1.1% at $z = 15$ and 0.46% at $z = 6$. Halos that are at least twice as concentrated in either 2LPT or ZA account for 25% at $z = 15$ and 15% at $z = 6$.

III.4 Discussion

As we evolve our DM halo population from our initial redshift to $z = 6$, we find that simulation initialization with 2LPT can have a significant effect on halo population compared to initialization with ZA. The second order displacement boost of 2LPT provides a head start on the initial collapse and formation of DM halos. This head start manifests itself further along in a halo's evolution as more rapid growth and earlier mergers. 2LPT halos are, on average, more massive than their ZA counterparts, with a maximum mean ΔM_{vir} of $(9.3 \pm 1.2) \times 10^{-2}$ at $z = 15$. The larger mass for 2LPT halos is more pronounced for higher mass pairs, while 2LPT halo concentration is larger on the small mass end. Both mass and concentration differences trend towards symmetry about zero as halos evolve in time, with the smallest difference observed at the end of the simulations at $z = 6$, with a mean ΔM_{vir} of $(1.79 \pm 0.31) \times 10^{-2}$. Casual extrapolation of our observed trends with redshift to today would indicate that, barring structure like massive clusters that form at high redshift, 2LPT and ZA would produce very similar halo populations by $z = 0$. However, the larger differences at high redshift should not be ignored.

The earlier formation times and larger masses of halos seen in 2LPT-initialized simulations could have significant implications with respect to early halo life during the Dark Ages. Earlier forming, larger halos affect the formation of Pop-III stars, and cause SMBHs to grow more rapidly during their infancy (Holley-Bockelmann et al., 2012). The epoch of peak star formation may also be shifted earlier. This could additionally affect the contribution of SMBHs and early star populations to re-ionization. Larger halos may also influence

studies of the high- z halo mass function, abundance matching, gas dynamics, AGN, clustering, and large scale structure formation.

In these discussions, it is important to note that it is wrong to assume that the ZA halo properties are the “correct” halo properties, even in a statistical sense. While halo mass suggests the most obvious shortcoming of ZA simulations, even properties such as concentration—that show little difference on average between 2LPT and ZA—can have large discrepancies on an individual halo basis. Failure to consider uncertainties in halo properties for high z halos in ZA simulations can lead to catastrophic errors.

We note a few caveats with our simulations and analysis. We did not exclude substructure when determining the properties of a halo, and although this would not change the broad conclusions herein, care must be taken when comparing to works which remove subhalo particles in determining halo mass and concentration. Halo matching is not perfect, as it is based on one snapshot at a time, and may miss count halos due to merger activity and differences in merger epochs. However, we believe this effect to be minor. While we compared ROCKSTAR’s output with our own fitting routines and found them to be in good agreement, ROCKSTAR does not provide goodness of fit parameters for its NFW profile fitting and R_s measurements. It also may be debated whether it makes sense to even consider concentration of halos at high redshift which are not necessarily fully virialized.

ROCKSTAR does not provide goodness-of-fit parameters for its internal density profile measurements used to derive concentration, so error estimates for concentration values of individual halos are unknown. Additionally, proper density profile fitting is non-trivial, as the non-linear interactions of numerical simulations rarely result in simple spherical halos that can be well described using spherical bins.

We use a simulation box size of only $(10 \text{ Mpc})^3$. This is too small to effectively capture very large outlier density peaks. We would, however, expect these large uncaptured peaks to most affected by 2LPT initialization, so the effects presented here may even be dramatically underestimated. Additionally, a larger particle number would allow us to consider smaller

mass halos than we were able to here, and to better resolve all existing structure. A higher starting redshift could probe the regime where 2LPT initialization contributes the most. It would also be of interest to evolve our halo population all the way to $z = 0$. The addition of baryons in a fully hydrodynamical simulation could also affect halo properties. These points may be address in future studies.

III.5 Conclusion

We analyzed three 2LPT and ZA simulation pairs and tracked the spherical overdensity dark matter halos therein with the 6-D phase space halo finder code ROCKSTAR to compare the effect of initialization technique on properties of particle–matched dark matter halos. This approach allowed us to directly compare matching halos between simulations and isolate the effect of using 2LPT over ZA. In summary, we found the following:

- 2LPT halos get a head start in the formation process and grow faster than their ZA counterparts. Companion halos in 2LPT and ZA simulations may have offset merger epochs and differing nuclear morphologies.
- 2LPT halos are, on average, more massive than ZA halos. At $z = 15$, the mean of the ΔM_{vir} distribution is $(9.3 \pm 1.2) \times 10^{-2}$, and 50% of 2LPT halos are at least 10% more massive than their ZA companions. By $z = 6$, the mean ΔM_{vir} is $(1.79 \pm 0.31) \times 10^{-2}$, and 10% of 2LPT halos are at least 10% more massive.
- This preference for more massive 2LPT halos is dependent on redshift, with the effect most pronounced at high z . This trend is best fit by $\Delta M_{\text{vir}} = (7.88 \pm 0.17) \times 10^{-3} z - (3.07 \pm 0.14) \times 10^{-2}$.
- Earlier collapse of the largest initial density peaks causes the tendency for more massive 2LPT halos to be most pronounced for the most massive halos, especially at high z . We find a trend of $\Delta M_{\text{vir}} = 5.6 \times 10^{-2} M_{\text{vir,avg}} - 0.33$ for $z = 15$. By $z = 6$, this has flattened to $\Delta M_{\text{vir}} = 6.4 \times 10^{-3} M_{\text{vir,avg}} - 2.5 \times 10^{-2}$.

- Halo concentration, on average, is similar for 2LPT and ZA halos. However, even by the end of the dark ages, the width of the Δc distribution— $\sigma_{\Delta c} = 0.551 \pm 0.026$ at $z = 6$ —is large and indicative of a significant percentage of halos with drastically mismatched concentrations, despite the symmetrical distribution of Δc . At $z = 15$, 25% of halo pairs have at least a factor of 2 concentration difference, with this falling to 15% by $z = 6$.
- There is a slight trend for ZA halos to be more concentrated than 2LPT halos at high mass. We find $\Delta c = -5.3 \times 10^{-2} M_{\text{vir,avg}} - 0.45$ at $z = 15$ and $\Delta c = -9.3 \times 10^{-3} M_{\text{vir,avg}} - 8.9 \times 10^{-2}$ at $z = 6$. This is not visible in the symmetrical Δc distributions, as the trends are roughly centered about zero and are washed away when integrated across the entire mass range.

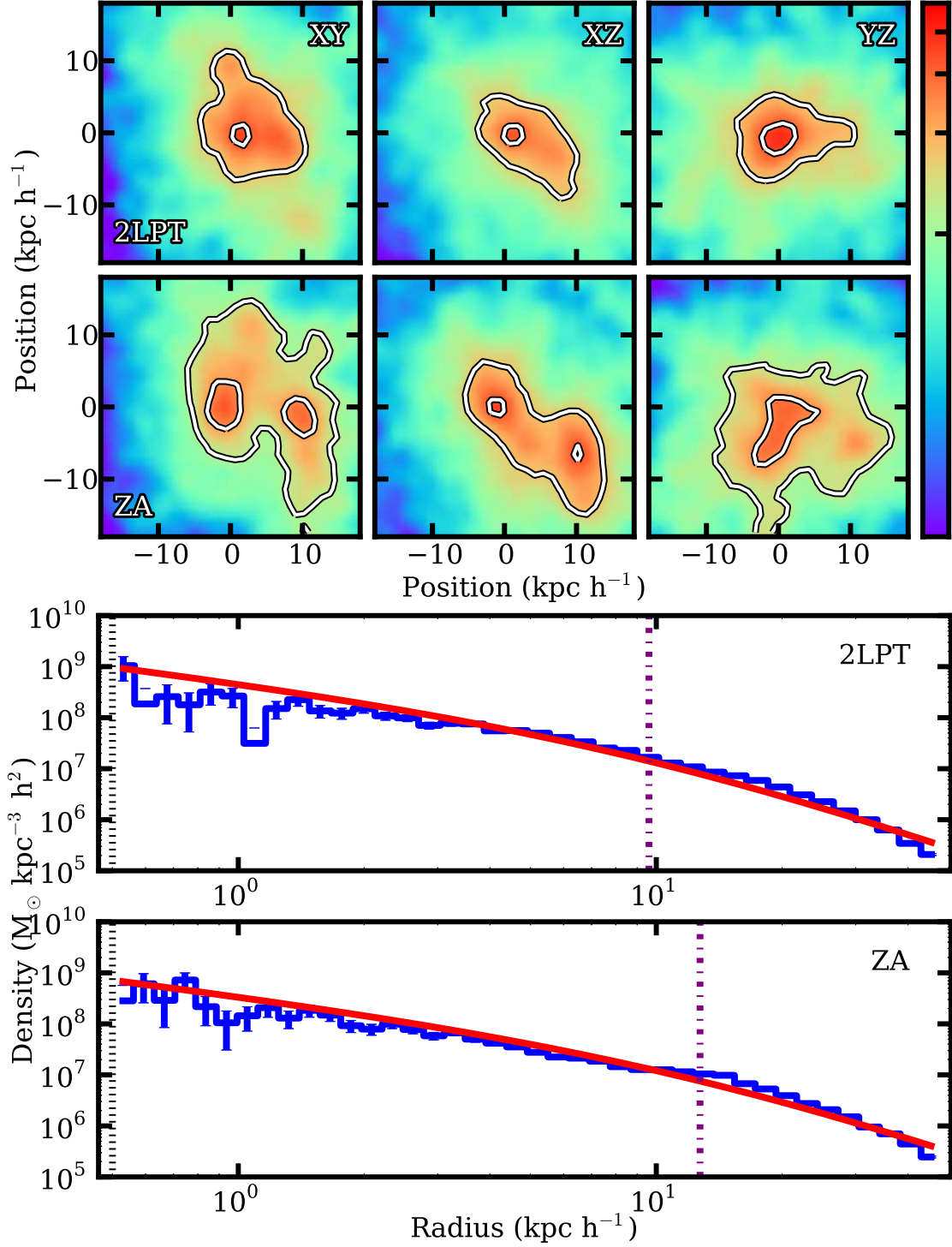


Figure III.1: *Top two rows:* Density projections for two matching halos at $z = 6$. The first and second row are 2LPT and ZA, respectively. The halos appear to be either undergoing or have recently undergone a major merger. The 2LPT halo appears to be more relaxed and further along in the merger process, while the ZA halo lags behind, still displaying two distinct cores. The halos have masses of $5.95 \times 10^9 M_\odot$ for 2LPT and $5.85 \times 10^9 M_\odot$ for ZA. *Bottom two rows:* Density profiles for the same two halos as above. NFW profiles are fit to logarithmic radial bins of particle position and are overplotted as red curves. The purple dot-dash lines mark the scale radii. The black dotted lines mark the resolution limit of the simulations.

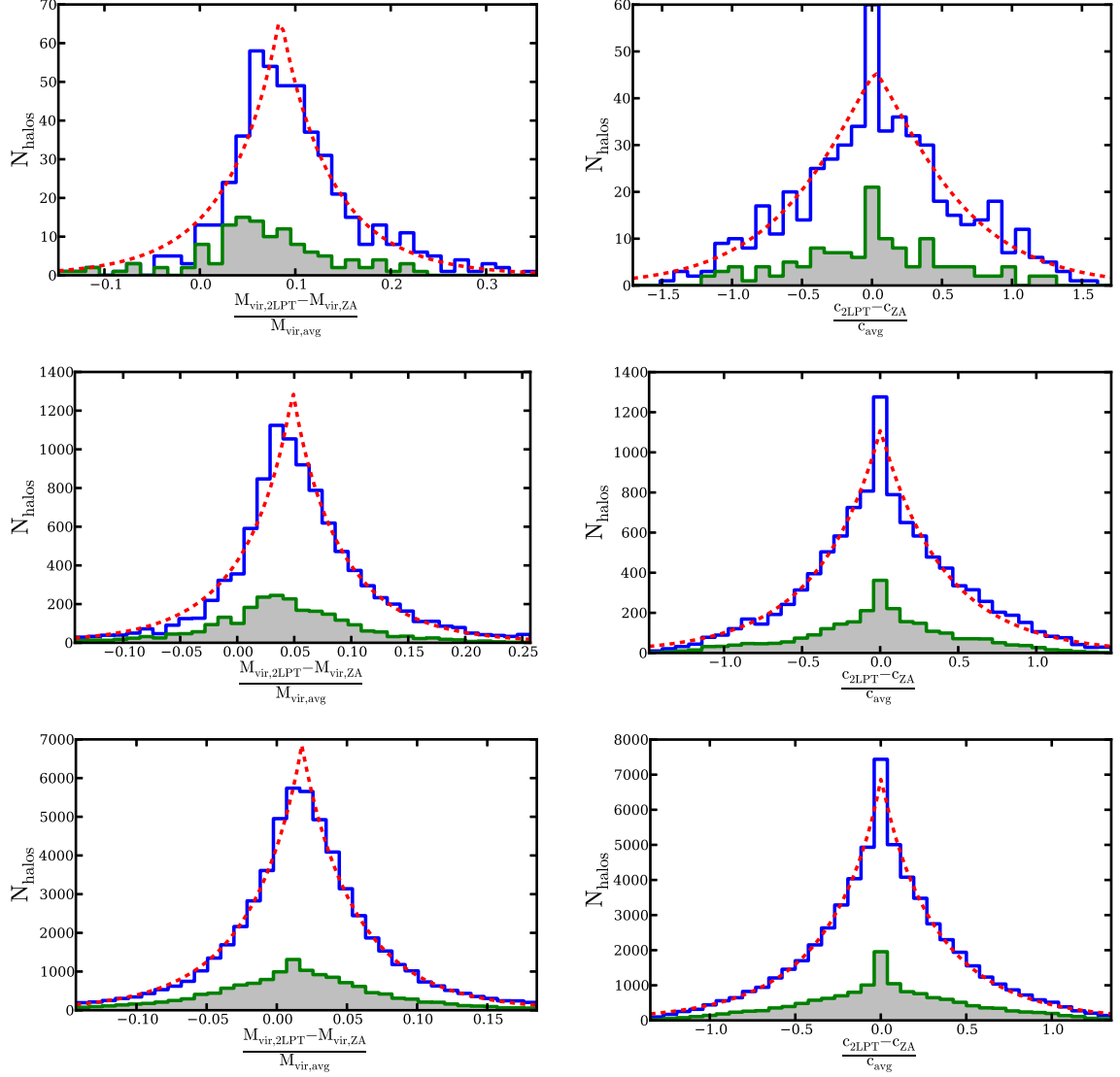


Figure III.2: Histograms of ΔM_{vir} (left column) and Δc (right column) for snapshots at $z = 14.7$, $z = 10.3$, and $z = 6.0$ (top, middle, and bottom panels, respectively). The small gray-filled histograms count only the top 25% most massive halos. The main histograms are fit with a generalized normal distribution with parameters for mean, scale, and shape, overplotted as the red dashed line (see Equation III.10). The distributions for ΔM_{vir} have positive means and heavier 2LPT halos, with the most pronounced difference at high redshift. The distributions shown here have means of $(8.4 \pm 1.8) \times 10^{-2}$, $(4.87 \pm 0.87) \times 10^{-2}$, and $(1.79 \pm 0.31) \times 10^{-2}$, respectively. The skew of the distribution is also the most positive at high redshift, and shifts toward symmetry by $z = 6$. The Δc distributions remain symmetric about zero and have negligible skew. The means are consistent with zero, at $(2.6 \pm 2.7) \times 10^{-2}$, $(0.2 \pm 2.6) \times 10^{-2}$, and $(0.3 \pm 1.1) \times 10^{-2}$, respectively. Both distributions have excess kurtosis consistently larger than that of a standard Gaussian distribution, with a sharp peak and heavy tails.

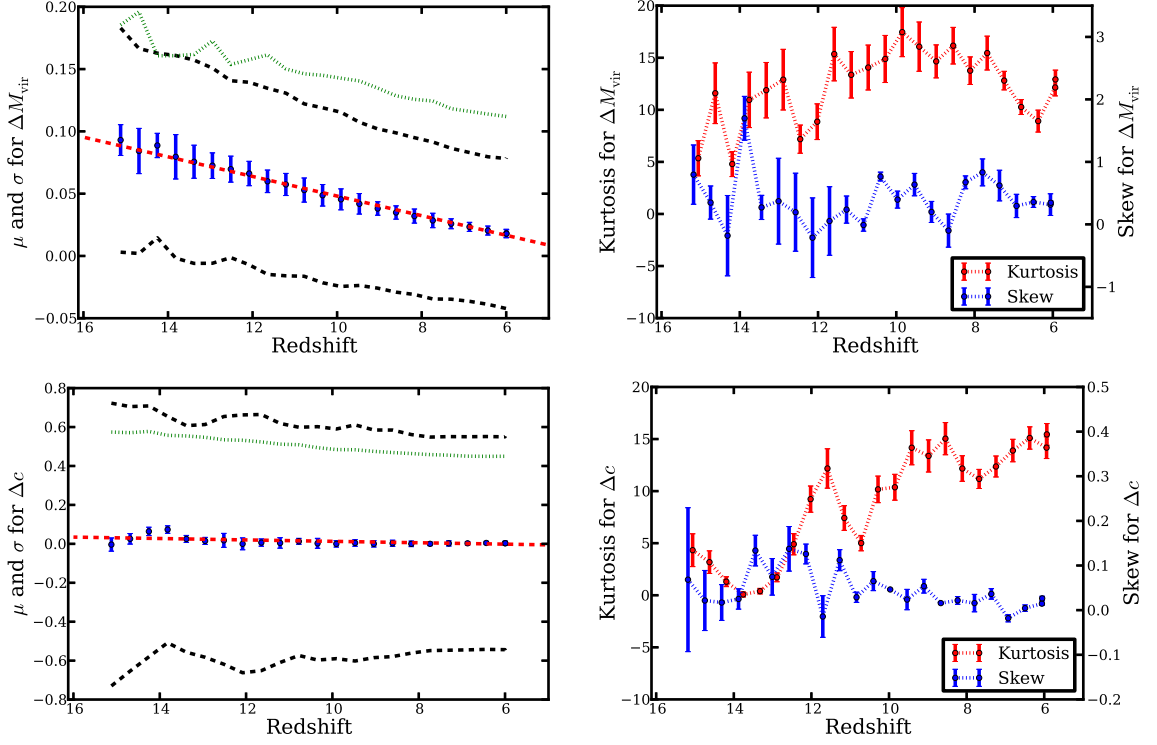


Figure III.3: Mean, standard deviation, and RMS (*left column*) and skew and excess kurtosis (*right column*) as functions of redshift for ΔM_{vir} (*top row*) and Δc (*bottom row*). In the left column, μ is plotted as blue points, and $\mu \pm \sigma$ is plotted as the black dashed line, and RMS values are plotted as a green dotted line. The red dashed line is a linear fit to the mean. We find a significant trend for μ for ΔM_{vir} to be more positive at higher redshift and gradually shift toward zero as the simulation progresses, with a fit function of $\mu_{\Delta M_{\text{vir}}} = (7.88 \pm 0.17) \times 10^{-3}z - (3.07 \pm 0.14) \times 10^{-2}$. The mean for Δc , however, remains at or very near zero for most of the simulation and is fit by $\mu_{\Delta c} = (3.62 \pm 0.95) \times 10^{-3}z - (2.34 \pm 0.84) \times 10^{-2}$. The ΔM_{vir} and Δc distributions narrow over time, with a slight decrease in σ . In the right column, we plot skew (blue line) and excess kurtosis (red line). Skew is positive for much of the simulation for ΔM_{vir} , but is much smaller for Δc . Kurtosis is large (much more peaked than Gaussian) for both ΔM_{vir} and Δc throughout much of the simulation, and especially at later redshift.

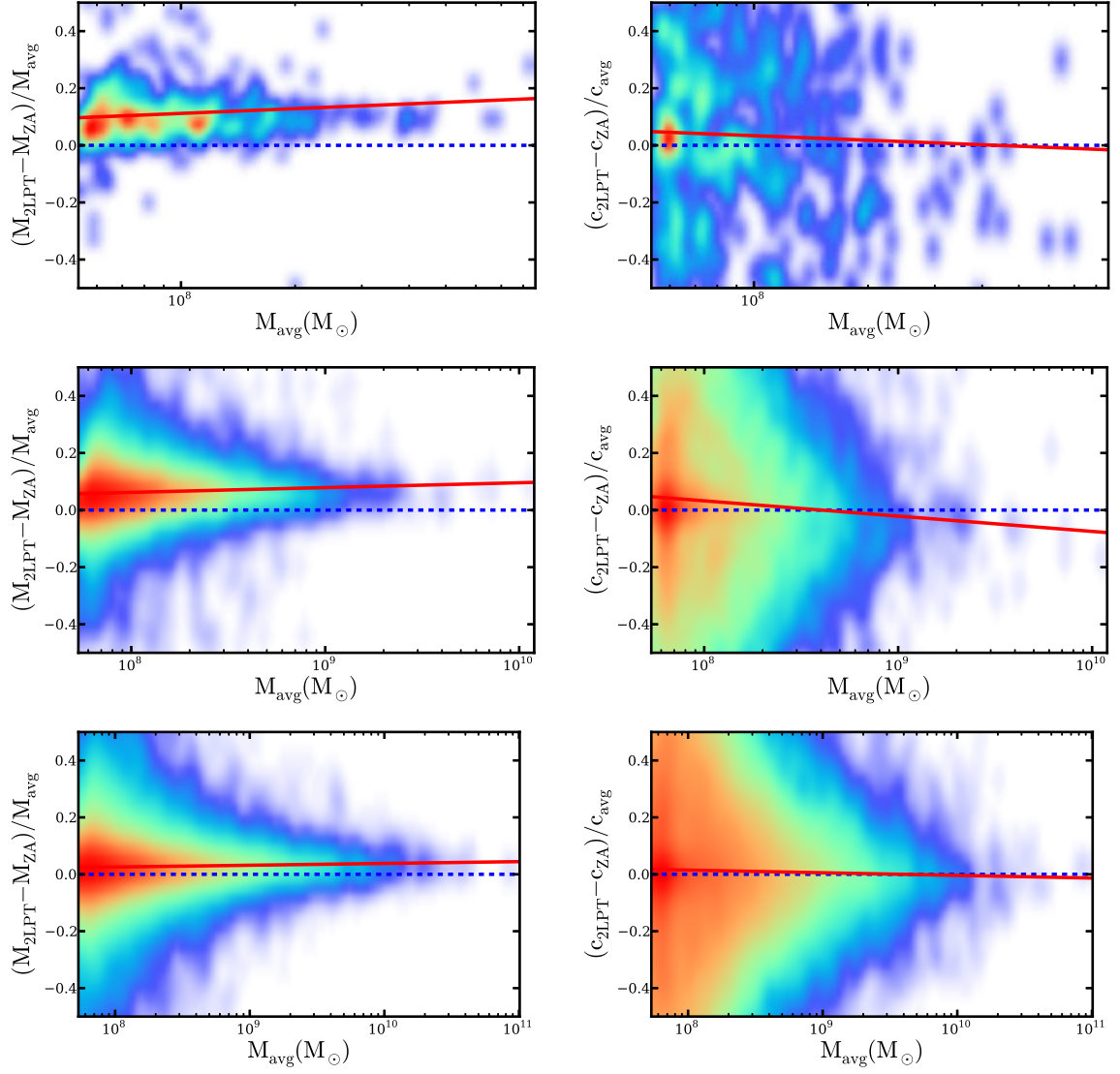


Figure III.4: ΔM_{vir} (left column) and Δc (right column) as functions of $M_{\text{vir,avg}}$. Halos are counted in 2-D rectangular bins and smoothed with a Gaussian kernel with a logarithmic color scale. The red line is the least-squares best fit to the data. The blue dashed line at zero is provided to guide the eye. The three rows again correspond to snapshots at $z = 14.7$, $z = 10.3$, and $z = 6.0$. We again see the overall offset for positive ΔM_{vir} as before, and additionally find that more massive halo pairs are more likely to have even larger ΔM_{vir} , especially at high redshift. Fit equations for the left column panels are $\Delta M_{\text{vir}} = 5.6 \times 10^{-2} M_{\text{vir,avg}} - 0.33$, $\Delta M_{\text{vir}} = 1.7 \times 10^{-2} M_{\text{vir,avg}} - 7.3 \times 10^{-2}$, and $\Delta M_{\text{vir}} = 6.4 \times 10^{-3} M_{\text{vir,avg}} - 2.5 \times 10^{-2}$, respectively. Concentration show a small but opposite trend for more massive halos to be more concentrated in ZA than in 2LPT. The right column panels have fit equations $\Delta c = -5.3 \times 10^{-2} M_{\text{vir,avg}} + 0.46$, $\Delta c = -4.5 \times 10^{-2} M_{\text{vir,avg}} + 0.46$, and $\Delta c = -9.3 \times 10^{-3} M_{\text{vir,avg}} + 8.9 \times 10^{-2}$, respectively.

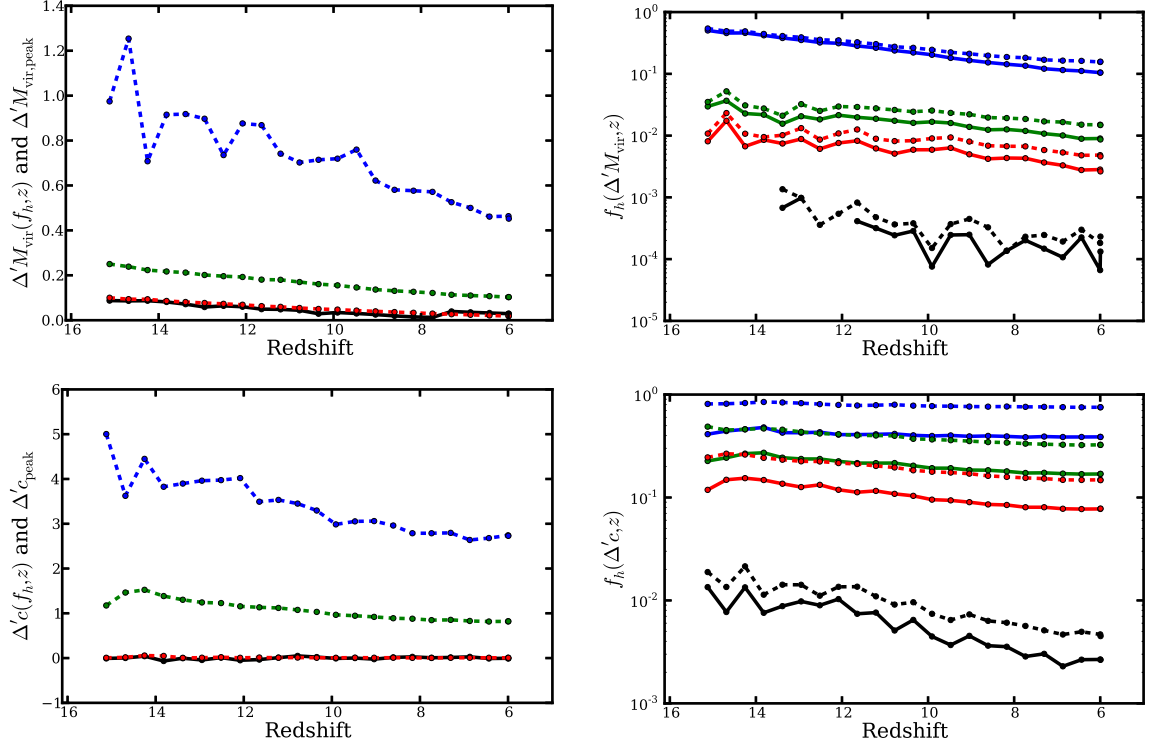


Figure III.5: Fractional error distributions statistics for $\Delta'M_{\text{vir}}$ (top row) and $\Delta'c$ (bottom row) as functions of redshift. *Left column:* The $\Delta'q$ of the peak of the distribution (black line), and the $\Delta'q$ where 50% (red dashed line), 10% (green dashed line), and 1% (blue dashed line) of the halos fall at or above $\Delta'q$. As with distributions of ΔM_{vir} , $\Delta'M_{\text{vir}}$ has the largest positive displacement at high redshift and steadily decreases throughout the simulation. Additionally, $\Delta'c$ maintains a peak near zero and has a spread much larger than that of $\Delta'M_{\text{vir}}$. *Right column:* The fraction of halos with $\Delta'q$ greater than 0.10 (solid blue line), 0.50 (solid green line), 1.00 (solid red line), and 4.00 (solid black line). The dashed lines additionally count halo pairs with $\Delta'q$ lower than the corresponding equivalent displacements of -0.09, -0.33, -0.50, and -0.80, respectively (see Equation III.22). We find that 50% of 2LPT halos are at least 10% more massive than their ZA companions at $z = 15$, reducing to 10% by $z = 6$. Halos in 2LPT are at least twice as concentrated for 12% of halos at $z = 15$ and 7.8% of halos at $z = 6$.

CHAPTER IV

Supermassive Black Holes and Their Hosts

IV.1 Introduction

The study of the evolution of galaxies and the growth of the supermassive black holes at their cores go hand in hand. Although the typical length scales for the two can vary by many orders of magnitude, they seem inexorably linked. Observational correlations between galaxy and supermassive black hole properties hint at an underlying co-evolution driven by shared mechanisms.

IV.1.1 Galaxy Properties

How do we describe a galaxy? Being extended, resolvable objects, galaxies provide a unique wealth of observable characteristics not obtainable from point sources such as stars. While many characteristics can be deduced about point sources, the actual observations themselves come down to measuring position on the sky and measuring flux as a function of frequency and time. From this information, all that we know about stars and other point sources, such as temperature, age, size, and composition, can be inferred. However, for extended objects like galaxies, we are given more to work with.

IV.1.1.1 Color

A galaxy's color is determined by its stellar component. While a galaxy in itself may be resolvable, for all but the most nearby of galaxies, individual stars are not. What we see when looking at a particular small section of a galaxy is the averaged-together light from stars in that section.

Broadly, bluer late-type spirals have a $u - r$ color of around $1.3 - 2.0$, while redder early-type galaxies have a $u - r$ color of around $2.3 - 2.7$. The color of a galaxy can be a good indicator for its age and evolutionary stage. Star formation processes generally tend

to produce many smaller, cooler, redder stars and fewer larger, hotter, bluer stars. These small, cool stars are much longer-lived than their massive counterparts, while the large, warm stars are much brighter. After star formation turns off, the short-lived blue stars begin to die off, and the galaxy becomes redder, as more of the fraction of total light comes from the red end of the population.

IV.1.1.2 Morphology

The extended nature of galaxies allows us to observe their morphology. The classification scheme originally devised by Hubble (1926) places galaxies into the four broad categories: elliptical, spiral, lenticular, and irregular. Elliptical galaxies tend to be larger, redder, have less gas, and dominated by more radial orbits. Spiral galaxies tend to be smaller, bluer, have more gas, and have more of a disk component. Spirals can have a number of arms, a central bulge, and a central bar. Lenticular galaxies are middle-of-the-road galaxies, with both a strong central bulge like an elliptical, and an extended disk like a spiral, however without spiral arms. Irregular galaxies tend to defy this simple classification scheme, and can be found in any number of configurations.

Figure IV.1 is a cartoon of the classification scheme. To the left of the diagram are elliptical galaxies. The subcategories are an indication of the shape of the galaxy, with the most spherical on the left and progressing to more flattened shapes to the right. On the right of the diagram are spiral galaxies. These are broken into two branches, based on whether or not the galaxy contains a central bar. Moving from right to left, the spiral arms of the galaxies become more tightly wound, and the central bulges become more dominant. At the center of the diagram where the spiral fork meets the elliptical line, lie lenticular galaxies. Irregular galaxies are, as the name would imply, irregular and do not fall on the diagram.

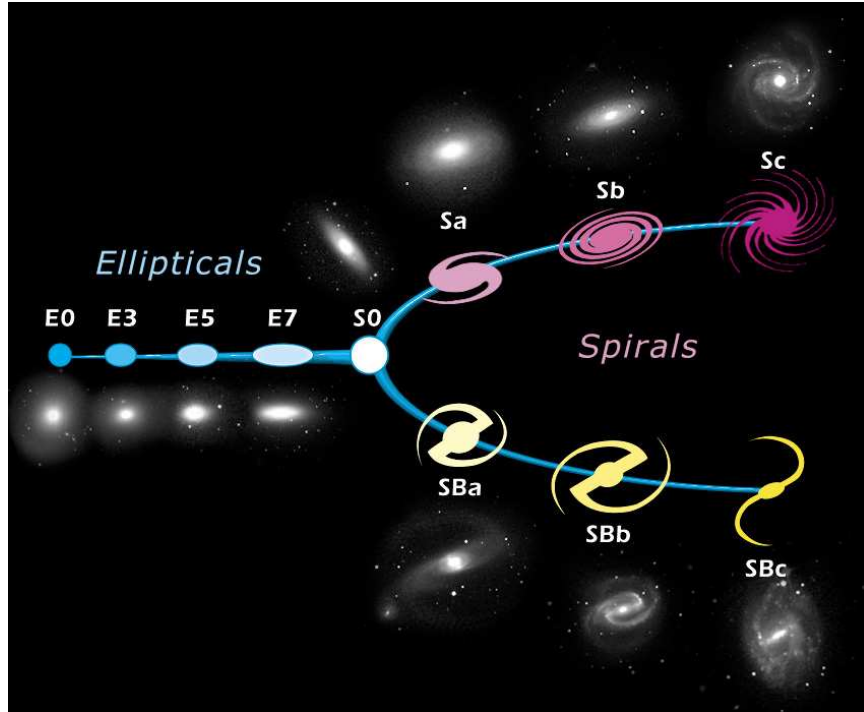


Figure IV.1: The Hubble tuning fork. On the left of the diagram are elliptical galaxies. E0 galaxies are the most spherical, while E7 are the most flattened or elongated. S0 are lenticular galaxies. The top branch on the right are spiral galaxies with no bar, while the bottom right branch are spiral galaxies with a bar. Both progress from tightly wound spiral arms and large bulges to loosely wound spiral arms and small to no bulges, going from Sa to Sc or SBa to SBc.

IV.1.2 Supermassive Black Hole Properties

A non-merging black hole, much like an elementary particle, can be described simply by its mass, charge, and spin. Its effect on its local spacetime, infalling matter, and surrounding environment all come back to these three parameters. However, determination of these parameters and the study of how black holes interact with their surroundings can be quite involved.

Black holes are, by their very nature, black, and difficult to observe. We cannot see light emitted directly from a black hole as we would a star, since a black hole is defined as an object massive and compact enough to not allow light within its event horizon to escape. We are forced, therefore, to employ other methods of measuring black holes.

Thus far, the majority of progress in the measurement of black hole properties has been in measuring mass. There are a number of ways to measure the mass of a black hole. Here, we will briefly discuss masers, stellar dynamics, gas dynamics, and reverberation mapping as methods of measuring a supermassive black hole's mass.

Astrophysical masers are sources of stimulated spectral line emission in the microwave band formed in regions of high-density gas comprised of molecules such as hydroxyl, formaldehyde, and water (Lo, 2005). Since the emission frequencies of these sources are very well constrained, high-accuracy Doppler shifts can be determined. These Doppler shifts can then be used to determine velocities for the masers, and thus how much mass is enclosed by their orbits. If these masers lie very close to the supermassive black hole (SMBH) in the center of their galaxy, the enclosed mass can be constrained to be primarily that of the SMBH.

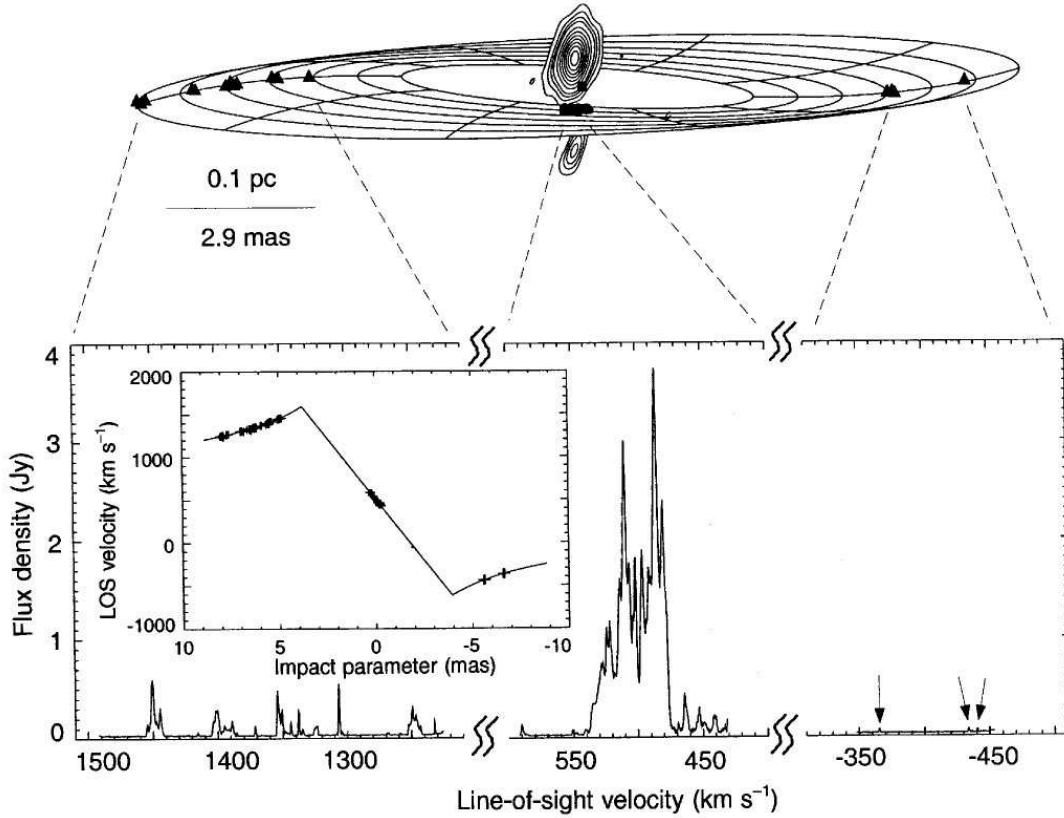


Figure IV.2: Maser orbits fit to a warped disk for NGC4258. Masers can also be useful for distance determinations. Here, the positions and velocities of water masers are able to be fit to a warped disk model surrounding a supermassive black hole. This allows the interpolation of physical radii away from the black hole, giving us both the black hole mass and an standard ruler to allow precise determination of the distance to NGC4258. (Herrnstein et al., 1999)

Stellar dynamics and gas dynamics both probe light coming from matter near the black hole. The width of broadened spectral lines from either the stars or gas can be used to determine a velocity dispersion for the matter local to the SMBH. This velocity dispersion, therefore, can then be used to determine the potential through which the matter is traveling, and thus the mass of the black hole.

A special case of stellar dynamics for which the orbits of the constituent stars can be resolved—namely, for the case of our own Milky Way—adds another dimension to our knowledge of the stellar orbits. Over time, we can observe the proper motion on the sky for these orbits. Combining these measurements with Doppler measurements for radial

velocity yields full orbital solutions. Then, it simply requires Kepler’s laws to determine the mass of the SMBH.

Reverberation mapping can be thought of as “echo-mapping” the gas disk around a SMBH. Continuum emission very near the black hole travels outward and stimulates broad line emission in surrounding gas. Any changes in the continuum emission will take time to propagate to the broad line region, since the speed of light is finite. By measuring the timing difference in the change in continuum emission and change in stimulated broad line emission, the physical distance from the SMBH to the broad line region can be inferred. With this radius, and the velocity of the gas in the broad line region measured by the width of the broadened lines, a black hole mass can be determined (Blandford & McKee, 1982).

IV.1.3 Correlations

Correlations between varying properties of galaxies and black holes can provide much deeper insight into the dynamics that shape the evolution of both. Of particular interest here are the fundamental plane of elliptical galaxies, the $M - \sigma$ relation, and the green valley-AGN relation.

IV.1.3.1 The M-Sigma Relation

If we consider the all the observable properties of a galaxy and compare them to the mass of its SMBH, the tightest correlation can be found with the velocity dispersion σ of the galaxy’s bulge. Such a tight correlation is surprising, as the sphere of influence of a typical SMBH does not extend much past order a few pc, while bulges exist on scales of a kpc or greater. In essence, the supermassive black hole and the outer edges of the bulge shouldn’t “feel” each other. Nevertheless, the correlation is indeed there, suggesting some mechanism that influences—or is influenced by—both of them. Gültekin et al. (2009) use a sample of 49 M_{BH} measurements and 19 upper limits to measure this correlation, and find $\log(M_{BH}/M_{\odot}) = \alpha + \beta \log(\sigma/200 \text{ km s}^{-1})$ with $(\alpha, \beta, \epsilon_0) = (8.12 \pm 0.08 M_{\odot}, 4.24 \pm 0.41 M_{\odot}, 0.44 \pm 0.06 M_{\odot})$ for all galaxies and $(\alpha, \beta, \epsilon_0) = (8.23 \pm 0.08 M_{\odot}, 3.96 \pm 0.42 M_{\odot}, 0.31 \pm$

$0.06M_{\odot})$ for ellipticals, where ε_0 is the intrinsic scatter in the relation.

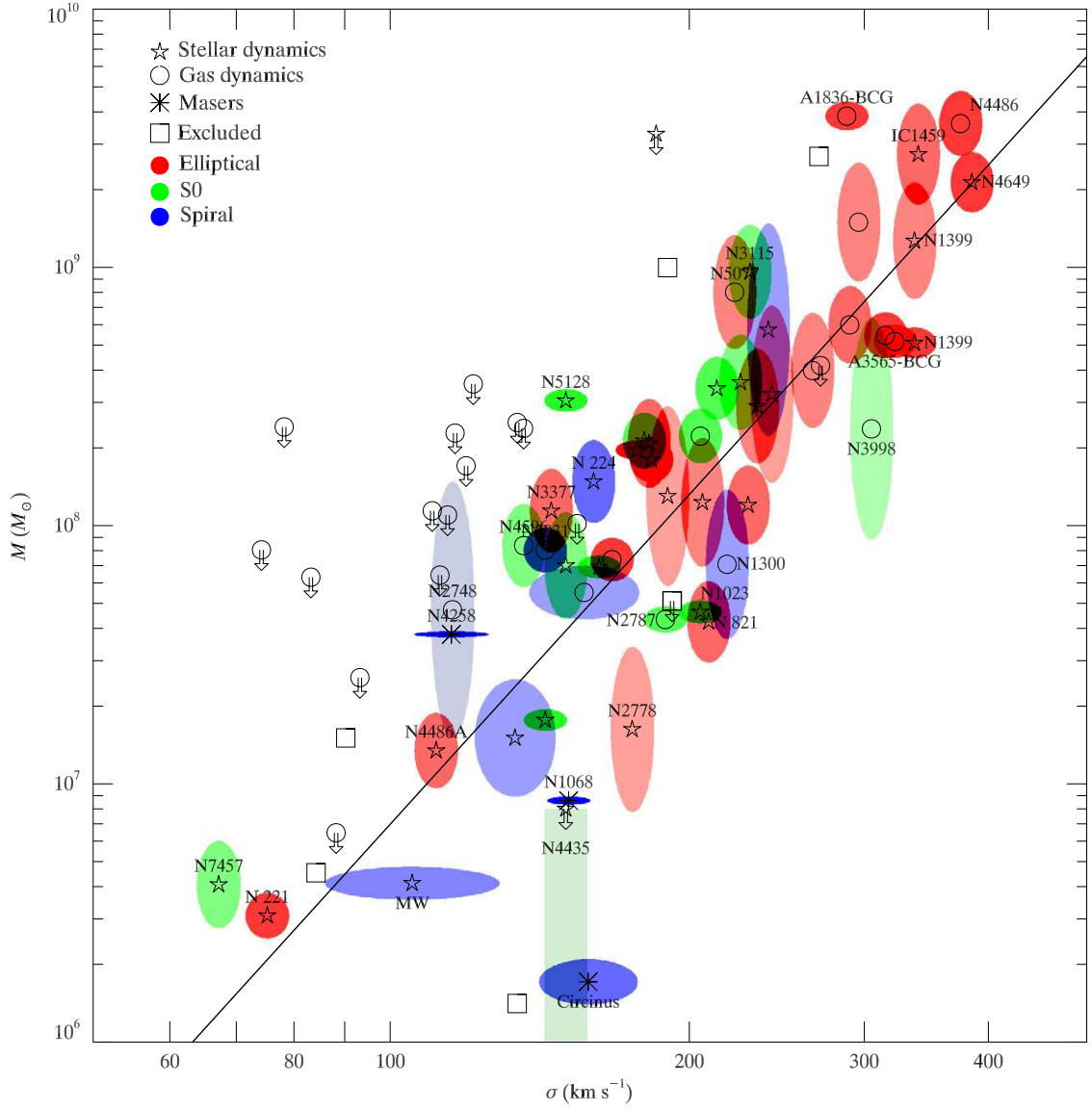


Figure IV.3: The M - σ relation for galaxies with dynamical measurements. Black hole mass is plotted vs velocity dispersion of its host spheroid. The symbols represent the method by which the black hole mass was measured: pentagrams for stellar dynamics, circles for gas dynamics, and asterisks for masers. Upper limits are given by arrows. Error ellipses are colored by galaxy type, with red for ellipticals galaxies, green for lenticular galaxies, and blue for spiral galaxies. The saturation of the color is inversely proportional to the area of the ellipse. For this sample, the best fit relation is $M_{BH} = 10^{8.12} M_{\odot} (\sigma/200 \text{ km s}^{-1})^{4.24}$. Galaxies not included in this fit are labeled as squares. (Gültekin et al., 2009)

IV.1.3.2 The Fundamental Plane

While not a direct correlation with the properties of supermassive black holes, the fundamental plane of elliptical galaxies offers insight into the characteristics of their hosts. The fundamental plane is a three-parameter correlation between properties of elliptical galaxies: velocity dispersion, effective radius, and surface brightness. This correlation (Figure IV.4) between these three parameters is tighter than the combination of any two alone (Djorgovski & Davis, 1987). The fit for this correlation can be given as $\log R_e = 0.36(\langle I \rangle_e / \mu_B) + 1.4 \log \sigma_0$, where R_e is the effective radius in kpc, $\langle I \rangle_e$ is the mean surface brightness interior to R_e in units of μ_B , and σ_0 is the velocity dispersion in km s^{-1} (Binney & Merrifield, 1998).

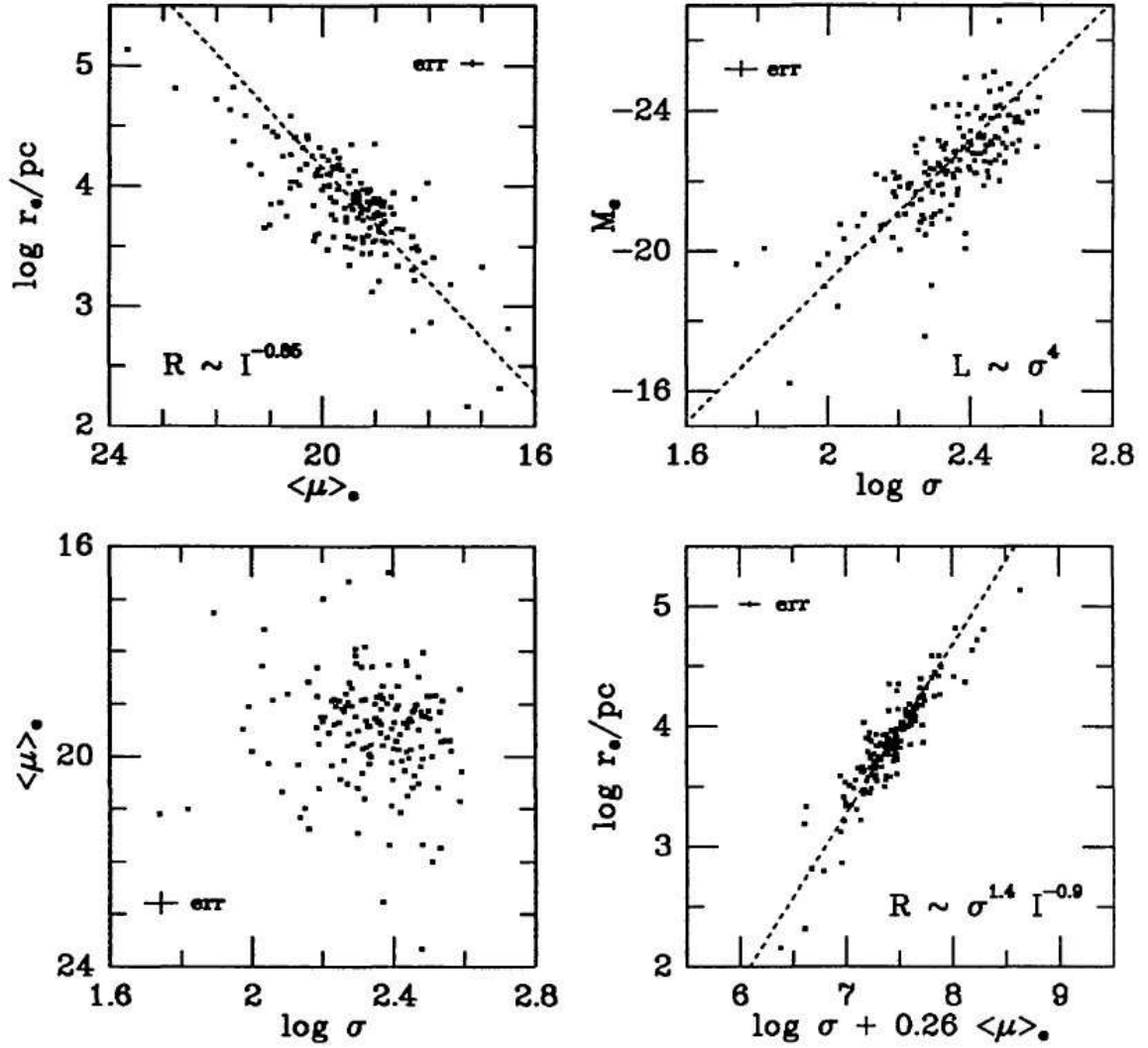


Figure IV.4: The fundamental plane for elliptical galaxies. *Top panels:* The top panels show the one-parameter scaling relations, with the relation between radius and mean surface brightness on the left and the relation between luminosity and velocity dispersion (the Faber-Jackson relation) on the right. *Bottom left:* The relation between the surface brightness and velocity dispersion. This is an almost face-on view of the fundamental plane. *Bottom right:* The relation between the effective radius and the combination of surface brightness and velocity dispersion. This is the edge-on view of the fundamental plane. (Kormendy & Djorgovski, 1989)

IV.1.3.3 The Green Valley

When considering both the color and stellar mass of a galaxies, a correlation emerges where many galaxies lie in either the “blue cloud” of bluer, lower mass galaxies, or the “red

sequence” of redder, generally higher mass galaxies. The area between these two is known as the “green valley” and, while not as populated as the blue cloud or red sequence, holds special interest when active galactic nuclei (AGN) are considered. AGN are very luminous regions at the centers of some galaxies. Schawinski et al. (2010) show that galaxies falling on the green valley are much more likely to host AGN than galaxies on the blue cloud or red sequence, hinting at an underlying link between the evolution of galaxies, and the activity at their centers.

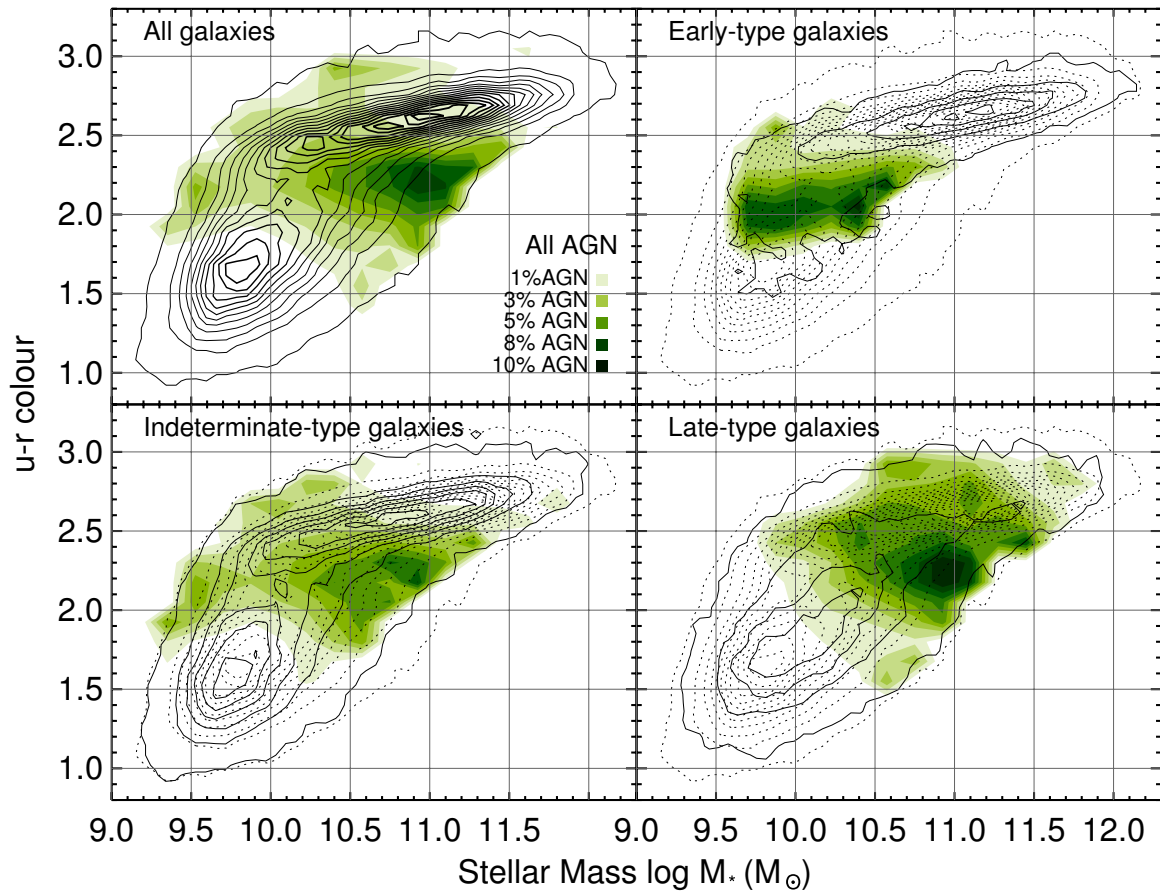


Figure IV.5: Distribution of the fraction of galaxies containing AGN. Galaxy color in $u-r$ is plotted vs stellar mass. The contours are the galaxy population for all galaxies (top-left), early-type galaxies (top right), intermediate-type galaxies (bottom left), and late type galaxies (bottom right). For the three sub-samples, dotted contours represent the full sample for comparison. The green shaded contours represent the fraction of galaxies in that subsample that contain active galactic nuclei. It can be clearly seen that the AGN fraction is highest for galaxies falling within the green valley. (Schawinski et al., 2010)

IV.2 Galaxy Evolution

IV.2.1 Dark Matter Halos

Every galaxy resides inside a dark matter halo. Often about an order of magnitude larger in both radius and mass than the baryonic component, dark matter halos dominate the large-scale behavior of galaxies. Dark matter is matter that is thought to interact very weakly or not at all with light and ordinary matter, except gravitationally. Evidence for dark matter comes from a number of sources, including the relatively flat rotational velocity curve of galaxies, the velocity dispersion of galaxies, gravitational lensing measurements, galaxy clustering, and the offset between the gas and dominant mass measured in the Bullet cluster. Here we will briefly discuss the evidence from flat rotation curves.

If there were no dark matter component and only the baryonic components (i.e. stars and gas) contributed to the galactic potential, we would expect the rotational velocity of galaxies to fall off with radius. However, observations show that the rotation curve remains relatively flat (Rubin et al., 1980). Figure IV.6 shows several observed rotation curves.

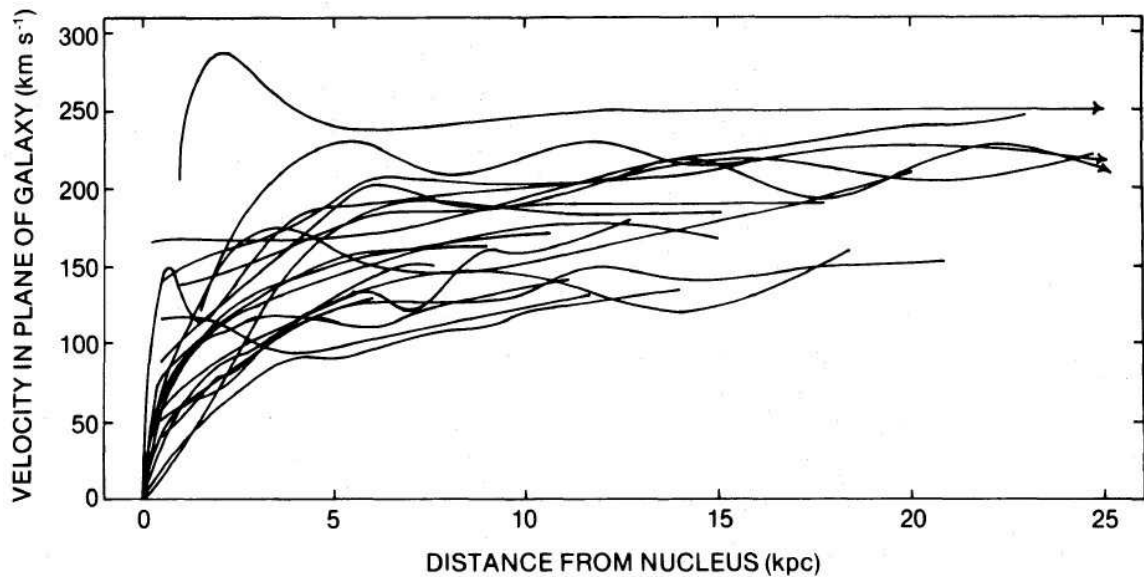


Figure IV.6: Rotation curves for 21 Sc galaxies. It is readily identifiable that the rotation curves do not fall off as would be expected for galaxies without a dark matter component. (Rubin et al., 1980)

Navarro et al. (1997b) found that dark matter halos generally follow the same density

profile, regardless of mass. This universal dark matter density profile can be given as

$$\rho(r) \propto \frac{1}{(r/a)(1+r/a)^2}, \quad (\text{IV.1})$$

where a is the radius where the profile transitions from an r^{-1} power law to an r^{-3} power law.

IV.2.2 Galaxy Mergers

Galaxy mergers are the fundamental mechanism by which galaxies grow and evolve. Collisions between galaxies trigger processes that can alter nearly all the properties of the galaxies. Naturally, mergers increase the mass of galaxies. Starting from small perturbations in the early universe, gravity slowly pulls matter together to form larger and larger clumps. These clumps of gas and dark matter eventually form stars, beginning what we think of as typical galaxies, and over time, these galaxies merge together into larger and larger galaxies.

Mergers affect many other properties of galaxies as well. Mergers distort the shapes of galaxies, causing long tidal tails to form and the entire morphology to appear irregular. The disk structures of spiral galaxies that form from the settling of the rotational component are distorted and “puffed up” into components with ever increasing bulge-like properties.

Mergers can trigger wide-scale starburst events, where a large portion of gas goes into the formation of stars. Much of the gas component of the galaxy can subsequently be blown out by the winds from the supernovae of short-lived O and B stars. This shuts off star formation, and as the stellar population is no longer replenished with new high-mass stars, the galaxy becomes progressively redder as large stars die.

The general trend is for mergers to move galaxies from the right side of the Hubble tuning fork towards the left, turning blue, gas rich spirals into red, gas poor ellipticals. This process is aided by the AGN feedback also triggered during galaxy mergers, as we discuss in the following section.

IV.3 Supermassive Black Hole Growth

Supermassive black holes grow by two primary mechanisms, binary mergers and gas accretion. Through a combination of these, black holes can grow to as large as $\sim 10^9\text{--}10^{10} M_\odot$ by $z = 0$.

IV.3.1 Binary Mergers

When two galaxies merge, the supermassive black holes at their hearts begin a process that will eventually lead to their coalescence. There are generally thought to be three stages to this journey. First, the black holes sink towards the center of the merged galaxy through mass segregation and dynamical friction until they form a bound orbit with each other. Then, the black holes tighten their orbit through three-body scattering of nearby stars. Finally, as the black holes become close enough together for general relativistic effects to come into play, gravitational waves are emitted and radiate away the remaining orbital energy until the binary coalesces.

IV.3.1.1 Dynamical Friction and Inspiral

During the majority of the inspiral process, the black holes do not “feel” each other’s gravitational pull. Instead, interactions with the galaxy itself push the holes together.

As it travels through a galaxy, a black hole—or any massive body—is slowed by the surrounding field of matter. Gravitational attraction pulls surrounding matter toward the black hole. However, as the black hole is moving with respect to the local medium, the attracted particles will tend to fall behind the black hole. This creates a wake of overdensity that gravitationally attracts the black hole from behind and slows its velocity. Chandrasekhar (1943) develops this notion of dynamical friction for the motion of a star through a sea of other stars. If the distribution of velocities of the surrounding particles is Maxwellian, the acceleration on the black hole can be written as

$$\frac{d\mathbf{v}_M}{dt} = -\frac{4\pi G^2 M \rho \ln \Lambda}{v_M^3} \left[\text{erf}(X) - \frac{2X}{\sqrt{\pi}} e^{-X^2} \right] \mathbf{v}_M, \quad (\text{IV.2})$$

where v_M is the velocity of the black hole, M is its mass, ρ is the density of surrounding matter, erf is the error function, $\ln \Lambda$ is the Coulomb logarithm, and $X \equiv v_M/(\sqrt{2}\sigma)$ where σ is the velocity dispersion of the surrounding medium (Binney & Tremaine, 1988). As the black hole is slowed by dynamical friction, it loses angular momentum and sinks towards the center of the galaxy's potential well.

IV.3.1.2 The Final Parsec Problem

Dynamical friction and mass segregation can only take us so far. Once the black holes are close enough together, they form a bound binary orbit. This generally occurs for separations of around a few to tens of parsecs. This presents a problem, however, since the orbit needs to shrink to around 10^{-2} – 10^{-3} pc in order for gravitational wave emission to remove energy from the orbit in a significant amount. The orbit can be tightened with three-body scattering of stars that wander through the orbit of the binary, however, in the spherical galaxies where mergers often take place, there is a depletion of stars with orbits that intersect the binary. Khan et al. (2011), however, show that the non-spherical, triaxial potential typical of post-merger galaxy remnants can efficiently funnel stars through the orbit of the black hole binary with sufficient intensity to tighten the binary orbit to the gravitational wave regime.

IV.3.1.3 Gravitational Waves and Recoil Kicks

Once the black hole binary separation reaches the point where strong field general relativistic effects come into play, we no longer require external influences to nudge the black holes together. In the final plunge toward coalescence, the black hole binary sheds energy through emission of gravitational radiation. As energy is radiated away, the binary tightens its orbit until the two black holes merge into one. Following this coalescence, the resultant black hole undergoes a “ringdown” phase, in which the distorted space time settles back down into a black hole that can again be simply described by mass, charge, and spin.

The emission of gravitational waves has two interesting consequences. First, the radiation from two merging supermassive black holes is extremely loud, and can potentially

provide an observational signature of the process for gravitational wave observatories. Second, the gravitational waves carry linear momentum, leading to a recoil “kick” imparted to the black hole merger remnant.

Recent advances in numerical relativity simulations have provided a much deeper insight into the black hole binary merger process than has been previously available. Waveforms produced from these simulations (Figure IV.7) can be used to predict what gravitational wave observatories such as LIGO and LISA would expect to observe for signals originating from merging supermassive black hole binaries. Having these waveforms as templates for comparison to data can greatly increase the signal to noise ratio for these detectors, potentially allowing the gravitational wave events to be seen among the sea of noise. These waveforms produced from simulations of the last few orbits of inspiral through the merger and ringdown can be combined with waveforms suggested from post-Newtonian approximations for the longer duration inspiral to provide a complete extended signal to match against.

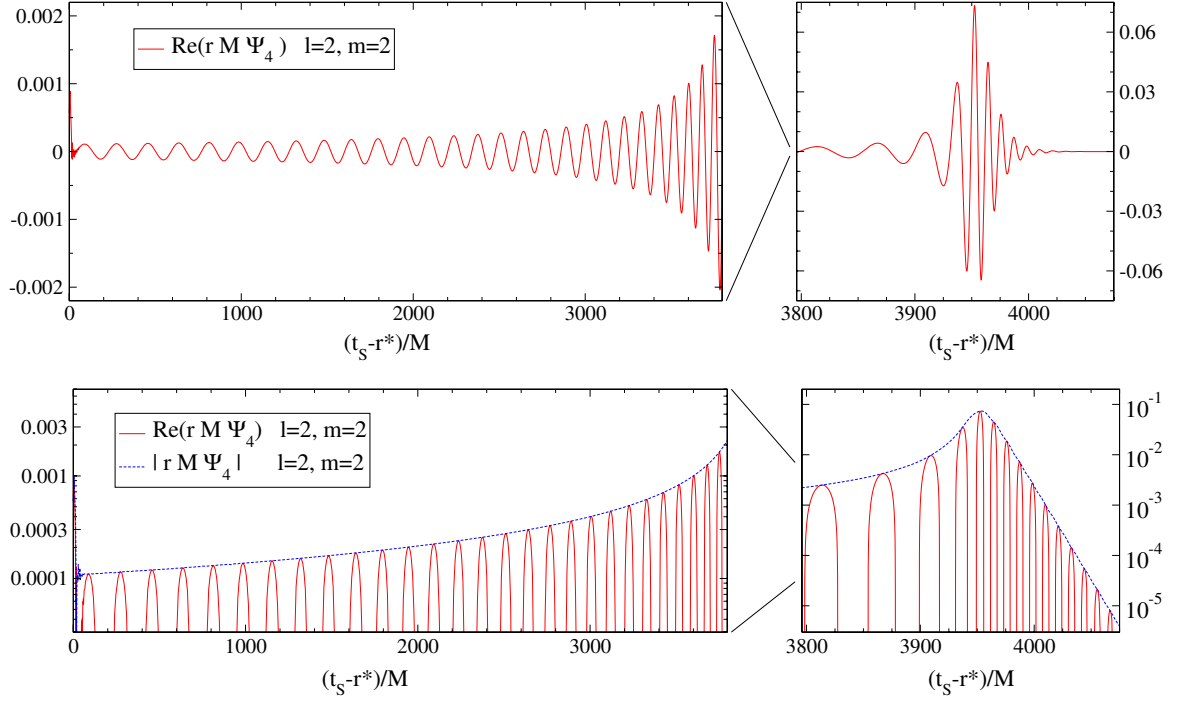


Figure IV.7: Gravitational waveform for an equal-mass, non-spinning black hole binary merger. This is the final waveform, extrapolated to infinity, from the numerical relativity simulation of Scheel et al. (2009). The waveform is shown on the top panel with a linear y-axis and on the bottom panel with a logarithmic y-axis. The left panels are the earlier stages of inspiral, and the right panels show the merger and ringdown stages.

For asymmetric mergers, gravitational radiation is emitted anisotropically. This causes a recoil kick, in which the gravitational waves impart a net velocity to the final black hole with respect to the original center of mass. The magnitude and direction of this kick are dependent on the mass ratio of the binary and the spins of the two black holes—in all, a 7-dimensional parameter space. This large parameter space has been largely explored with numerical relativistic simulations, and analytic equations can be fit to the data to predict the recoil from a given merger configuration. Holley-Bockelmann et al. (2008), give these equations as

$$\mathbf{v}_{kick} = (1 + e) \left[\hat{\mathbf{x}}(v_m + v_{\perp} \cos \xi) + \hat{\mathbf{y}}v_{\perp} \sin \xi + \hat{\mathbf{z}}v_{\parallel} \right], \quad (\text{IV.3})$$

where

$$v_m = A \frac{q^2(1-q)}{(1+q)^5} \left[1 + B \frac{q}{(1+q)^2} \right], \quad (\text{IV.4})$$

$$v_\perp = H \frac{q^2}{(1+q)^5} \left(\alpha_2^\parallel - q \alpha_1^\parallel \right), \quad (\text{IV.5})$$

$$v_\parallel = K \cos(\Theta - \Theta_0) \frac{q^2}{(1+q)^5} \left(\alpha_2^\perp - q \alpha_1^\perp \right). \quad (\text{IV.6})$$

Here, the fitting constants are $A = 1.2 \times 10^4 \text{ km s}^{-1}$, $B = -0.93$, $H = (7.3 \pm 0.3) \times 10^3 \text{ km s}^{-1}$, and $K = (6.0 \pm 0.1) \times 10^4 \text{ km s}^{-1}$. The \hat{z} unit vector is in the direction of the orbital angular momentum, and \perp and \parallel refer to components perpendicular and parallel to \hat{z} , respectively. The fitting parameters are the eccentricity e , the mass ratio $q \equiv M_2/M_1$, and the reduced spin parameters $\alpha_i \equiv S_i/M_i^2$ where S is the spin angular momentum. The orientation of the merger is given by the angles Θ , Θ_0 , and ξ (Holley-Bockelmann et al., 2008).

Slices through this parameter space are shown in Figure IV.8. For certain configurations of the merger, the recoil velocity can be very high. Very asymmetric mergers can produce recoils as high as $\sim 4000 \text{ km s}^{-1}$. These large recoils can be enough for the black hole to escape the potential well of its host galaxy and be ejected. Even less extreme recoil kicks can affect the evolution of black holes, as the kicked black hole can oscillate about its host's center, potentially changing its local gas environment and accretion rate.

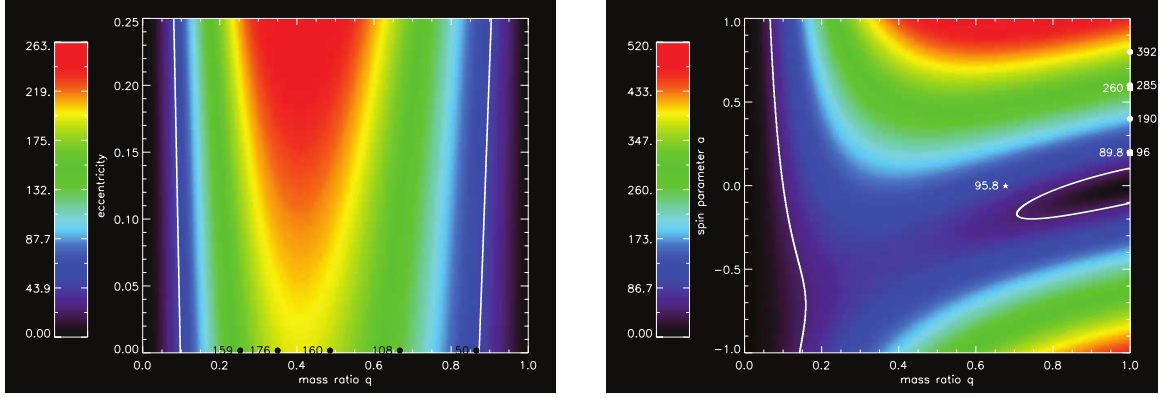


Figure IV.8: *Left:* Gravitational wave recoil velocity from a merger of nonspinning black holes as a function of eccentricity and mass ratio. Data from numerical relativity simulations (González et al., 2007) are overlaid along the zero eccentricity line. The overlaid white contours are the escape velocity of a typical globular cluster, 50 km s^{-1} . *Right:* Gravitational wave recoil kick velocity as a function of spin parameter and mass ratio for a merger of spinning black holes on a circular orbit with spins perpendicular to the orbital plane of the binary and anti-aligned with each other. Again, the 50 km s^{-1} escape velocity of a globular cluster is overlaid as white contours. Results from numerical relativity simulations are over-plotted: squares for Koppitz et al. (2007), circles for Herrmann et al. (2007), and star for Brüggmann et al. (2004). (Holley-Bockelmann et al., 2008)

IV.3.2 Accretion

Although mergers play an important role in the evolution of supermassive black holes, gas accretion can often dominate in terms of mass growth. Gas can fall into a black hole in a number of ways. Here, we will discuss accretion onto a moving black hole, spherical accretion onto a stationary black hole, and disk accretion onto a stationary black hole.

IV.3.2.1 Bondi-Hoyle-Lyttleton Accretion

Let us first consider a massive object, in this case our black hole, moving through a uniform density gas medium. Just as in the case of dynamical friction, particles close enough to the black hole will feel a gravitational attraction, causing them to move toward the black hole. As they move closer, the black hole is also moving through the medium, causing the gas particles to focus behind the black hole. As the particle stream reaches the wake directly

behind the black hole, it collides with opposing streams, causing the angular momentum to go to zero. If these particles are bound, they will proceed to fall onto the black hole. Hoyle & Lyttleton (1939) derive an impact parameter for which particles will be accreted,

$$\sigma < \sigma_{HL} = \frac{2GM}{v_\infty^2}, \quad (\text{IV.7})$$

and a mass accretion from the wake column at a rate of

$$\dot{M}_{HL} = \pi \sigma_{HL}^2 v_\infty \rho_\infty = \frac{4\pi G^2 M^2 \rho_\infty}{v_\infty^3}, \quad (\text{IV.8})$$

where v_∞ and ρ_∞ are the velocity and density far away from the black hole, respectively. Expanding upon this analysis, Bondi & Hoyle (1944) suggest that the accretion rate should rather be

$$\dot{M}_{BH} = \frac{2\alpha \pi G^2 M^2 \rho_\infty}{v_\infty^3}, \quad (\text{IV.9})$$

where α is a constant between 1 and 2, with a typical value of around 1.25.

For an accretor at rest in an isotropic gas medium, one would expect accretion to be a spherical process. Bondi (1952) considers this configuration, and finds the accretion rate for this “temperature-limited” case to be

$$\dot{M}_{Bondi} = \frac{2\pi G^2 M^2 \rho_\infty}{c_{s,\infty}^3}, \quad (\text{IV.10})$$

where $c_{s,\infty}$ is the speed of sound far away from the black hole.

Extrapolating between this result and the “velocity-limited” case of Equation IV.9 suggests (Bondi, 1952)

$$\dot{M}_{BH} = \frac{2\pi G^2 M^2 \rho_\infty}{(c_{s,\infty}^2 + v_\infty^2)^{3/2}} \quad (\text{IV.11})$$

as an order of magnitude estimate of the more general case of accretion. Numerical simulations (Shima et al., 1985) suggest an additional factor of 2 is needed for better agreement

with simulation results, giving us a generally applicable formula for the accretion rate,

$$\dot{M}_{BH} = \frac{4\pi G^2 M^2 \rho_\infty}{(c_{s,\infty}^2 + v_\infty^2)^{3/2}}. \quad (\text{IV.12})$$

IV.3.2.2 Disk Accretion and Active Galactic Nuclei

Active galactic nuclei play a fundamental role in the evolution of both supermassive black holes and their host galaxies. As gas falls in to a black hole in the center of a galaxy, its angular momentum forces it into an accretion disk. As matter moves towards the SMBH, it transfers its gravitational potential energy to thermal energy. For accretion disks around supermassive black holes, this can cause the disk to emit large amounts of electromagnetic radiation (Lin & Papaloizou, 1996).

This emitted radiation is important in a number of ways. Most critical to the SMBH itself is the radiation pressure exerted on infalling matter. This radiation pressure sets an upper limit on the rate of accretion, as there is a point where the force from emitted radiation balances the force of gravity for infalling gas (Rybicki & Lightman, 1979). This limit, known as the Eddington limit, is given by

$$L_{Edd} = 4\pi G M c m_H / \sigma_T = 1.25 \times 10^{38} \text{erg s}^{-1} (M/M_\odot), \quad (\text{IV.13})$$

where c is the speed of light, m_H is the mass of hydrogen, and σ_T is the Thompson cross section.

The radiation given off by the accretion disk affects galactic properties as well. Powerful AGN can strip away gas from the center of the galaxy, halting star formation. This can quickly change a galaxy from a blue, gaseous, star forming galaxy into one that is red, dry, and dead.

IV.4 Conclusion

We have seen that galaxies and the supermassive black holes at their centers both have their most dramatic periods of evolution around the same time. Galaxy mergers grow both the galaxy and the SMBH. Galaxies grow and become more elliptical as mergers bring in additional mass on orbits that can disrupt their gaseous disks. These mergers also bring in counterpart supermassive black holes that fall toward the center of the galaxy and merge with the central SMBH, while also triggering accretion events and AGN feedback that pump energy back into the galaxy, shutting off star formation.

IV.4.1 Correlations

In light of these shared growth mechanisms, the correlations mentioned in Section IV.1 begin to move from a purely observational coincidence to a natural result of co-evolution. The $M-\sigma$ relation is a natural byproduct of the simultaneous growth of supermassive black holes and their galaxies during merger events. The mass of the SMBH increases due to the merging of binary companions and increased levels of accretion, while the host mass, and thus velocity dispersion, increases due to the infalling galaxy itself. Likewise, the overabundance of AGN in galaxies lying in the green valley is the consequence of simultaneous change. Mergers both trigger highly luminous AGN feedback and cause an inexorable shift from the blue cloud, through the green valley, to the red sequence. Even the increase in scatter of the $M-\sigma$ relation at low masses can be explained by the galaxies having lower mass, and therefore being more likely to allow a gravitational wave recoil kicked black hole of a given velocity to escape.

IV.4.2 Open Questions

In the end, there remain a number of open questions. How can very large supermassive black holes form so early? What is dark matter actually made of? How do galaxies retain their black holes if merger recoils can kick them with velocities greater than the escape velocity of the galaxy? Over what range are our correlations truly valid? These are just

some of the questions that are currently being investigated, and promise to provide a rich field of study for years to come.

CHAPTER V

Conclusion

BIBLIOGRAPHY

- Abel, T., Bryan, G. L., & Norman, M. L. 2000, *Astrophys. J.*, 540, 39
- . 2002, *Science*, 295, 93
- Alvarez, M. A., Bromm, V., & Shapiro, P. R. 2006, *Astrophys. J.*, 639, 621
- Alvarez, M. A., Wise, J. H., & Abel, T. 2009, *Astrophys. J., Lett.*, 701, L133
- Barkana, R., & Loeb, A. 2001, *Phys. Rep.*, 349, 125
- Behroozi, P. S., Wechsler, R. H., & Wu, H.-Y. 2013, *Astrophys. J.*, 762, 109
- Binney, J., & Merrifield, M. 1998, *Galactic Astronomy*, Princeton Series in Astrophysics (Princeton University Press)
- Binney, J., & Tremaine, S. 1988, *Galactic Dynamics*, Princeton Series in Astrophysics (Princeton University Press)
- Blandford, R. D., & McKee, C. F. 1982, *Astrophys. J.*, 255, 419
- Bondi, H. 1952, *Mon. Not. R. Astron. Soc.*, 112, 195
- Bondi, H., & Hoyle, F. 1944, *Mon. Not. R. Astron. Soc.*, 104, 273
- Bouchet, F. R., Colombi, S., Hivon, E., & Juszkievicz, R. 1995, *Astron. Astrophys.*, 296, 575
- Bouwens, R. J., Illingworth, G. D., Oesch, P. A., et al. 2012, *Astrophys. J., Lett.*, 752, L5
- Brüggemann, B., Tichy, W., & Jansen, N. 2004, *Phys. Rev. Lett.*, 92, 211101
- Bryan, G. L., & Norman, M. L. 1998, *Astrophys. J.*, 495, 80
- Buchert, T. 1994, *Mon. Not. R. Astron. Soc.*, 267, 811
- Buchert, T., Melott, A. L., & Weiss, A. G. 1994, *Astron. Astrophys.*, 288, 349
- Bullock, J. S., Kolatt, T. S., Sigad, Y., et al. 2001, *Mon. Not. R. Astron. Soc.*, 321, 559
- Bykov, A. M., Paerels, F. B. S., & Petrosian, V. 2008, *Space Sci. Rev.*, 134, 141
- Chandrasekhar, S. 1943, *Astrophys. J.*, 97, 255
- Couchman, H. M. P., & Rees, M. J. 1986, *Mon. Not. R. Astron. Soc.*, 221, 53
- Crocce, M., Pueblas, S., & Scoccimarro, R. 2006, *Mon. Not. R. Astron. Soc.*, 373, 369
- Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, *Astrophys. J.*, 292, 371

- Djorgovski, S., & Davis, M. 1987, *Astrophys. J.*, 313, 59
- Efstathiou, G., Davis, M., White, S. D. M., & Frenk, C. S. 1985, *Astrophys. J., Suppl. Ser.*, 57, 241
- Einasto, J., Klypin, A. A., Saar, E., & Shandarin, S. F. 1984, *Mon. Not. R. Astron. Soc.*, 206, 529
- Eke, V. R., Navarro, J. F., & Steinmetz, M. 2001, *Astrophys. J.*, 554, 114
- Fan, X., Narayanan, V. K., Lupton, R. H., et al. 2001, *Astron. J.*, 122, 2833
- Gao, L., Navarro, J. F., Cole, S., et al. 2008, *Mon. Not. R. Astron. Soc.*, 387, 536
- Gnedin, N. Y., & Ostriker, J. P. 1997, *Astrophys. J.*, 486, 581
- González, J. A., Sperhake, U., Brüggmann, B., Hannam, M., & Husa, S. 2007, *Phys. Rev. Lett.*, 98, 091101
- Gültekin, K., Richstone, D. O., Gebhardt, K., et al. 2009, *The Astrophysical Journal*, 698, 198
- Heger, A., Fryer, C. L., Woosley, S. E., Langer, N., & Hartmann, D. H. 2003, *Astrophys. J.*, 591, 288
- Heger, A., & Woosley, S. E. 2002, *Astrophys. J.*, 567, 532
- Heitmann, K., Lukić, Z., Habib, S., & Ricker, P. M. 2006, *Astrophys. J., Lett.*, 642, L85
- Herrmann, F., Hinder, I., Shoemaker, D., Laguna, P., & Matzner, R. A. 2007, *Astrophys. J.*, 661, 430
- Herrnstein, J. R., Moran, J. M., Greenhill, L. J., et al. 1999, *Nature*, 400, 539
- Holley-Bockelmann, K., Gültekin, K., Shoemaker, D., & Yunes, N. 2008, *Astrophys. J.*, 686, 829
- Holley-Bockelmann, K., Wise, J. H., & Sinha, M. 2012, *Astrophys. J., Lett.*, 761, L8
- Hoyle, F., & Lyttleton, R. A. 1939, *Proceedings of the Cambridge Philosophical Society*, 35, 405
- Hubble, E. P. 1926, *Astrophys. J.*, 64, 321
- Islam, R. R., Taylor, J. E., & Silk, J. 2003, *Mon. Not. R. Astron. Soc.*, 340, 647
- Jenkins, A. 2010, *Mon. Not. R. Astron. Soc.*, 403, 1859
- Jeon, M., Pawlik, A. H., Greif, T. H., et al. 2012, *Astrophys. J.*, 754, 34
- Khan, F. M., Just, A., & Merritt, D. 2011, *Astrophys. J.*, 732, 89

- Klypin, A. A., & Shandarin, S. F. 1983, *Mon. Not. R. Astron. Soc.*, 204, 891
- Klypin, A. A., Trujillo-Gomez, S., & Primack, J. 2011, *Astrophys. J.*, 740, 102
- Komatsu, E., Dunkley, J., Nolte, M. R., et al. 2009, *Astrophys. J., Suppl. Ser.*, 180, 330
- Koppitz, M., Pollney, D., Reisswig, C., et al. 2007, *Phys. Rev. Lett.*, 99, 041102
- Kormendy, J., & Djorgovski, S. 1989, *Ann. Rev. Astron. Astrophys.*, 27, 235
- Kuhlen, M., & Faucher-Giguère, C.-A. 2012, *Mon. Not. R. Astron. Soc.*, 423, 862
- Lin, D., & Papaloizou, J. 1996, *Annual Review of Astronomy and Astrophysics*, 34, 703
- Lo, K. Y. 2005, *Ann. Rev. Astron. Astrophys.*, 43, 625
- Lukić, Z., Heitmann, K., Habib, S., Bashinsky, S., & Ricker, P. M. 2007, *Astrophys. J.*, 671, 1160
- Macciò, A. V., Dutton, A. A., & van den Bosch, F. C. 2008, *Mon. Not. R. Astron. Soc.*, 391, 1940
- Madau, P., Haardt, F., & Rees, M. J. 1999, *Astrophys. J.*, 514, 648
- Madau, P., & Rees, M. J. 2001, *Astrophys. J., Lett.*, 551, L27
- Muñoz-Cuartas, J. C., Macciò, A. V., Gottlöber, S., & Dutton, A. A. 2011, *Mon. Not. R. Astron. Soc.*, 411, 584
- Nadarajah, S. 2005, *Journal of Applied Statistics*, 32, 685
- Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, *Astrophys. J.*, 462, 563
- . 1997a, *Astrophys. J.*, 490, 493
- . 1997b, *Astrophys. J.*, 490, 493
- Neto, A. F., Gao, L., Bett, P., et al. 2007, *Mon. Not. R. Astron. Soc.*, 381, 1450
- Prada, F., Klypin, A. A., Cuesta, A. J., Betancort-Rijo, J. E., & Primack, J. 2012, *Mon. Not. R. Astron. Soc.*, 423, 3018
- Press, W. H., & Schechter, P. 1974, *Astrophys. J.*, 187, 425
- Reed, D. S., Bower, R., Frenk, C. S., Jenkins, A., & Theuns, T. 2007, *Mon. Not. R. Astron. Soc.*, 374, 2
- Rubin, V. C., Ford, W. K. J., & Thonnard, N. 1980, *Astrophys. J.*, 238, 471
- Rybicki, G. B., & Lightman, A. P. 1979, *Radiative processes in astrophysics* / George B. Rybicki, Alan P. Lightman (Wiley, New York :), xv, 382 p. :

- Schawinski, K., Urry, C. M., Virani, S., et al. 2010, *The Astrophysical Journal*, 711, 284
- Scheel, M. A., Boyle, M., Chu, T., et al. 2009, *Phys. Rev. D*, 79, 024003
- Scoccimarro, R. 1998, *Mon. Not. R. Astron. Soc.*, 299, 1097
- Shapiro, P. R., & Giroux, M. L. 1987, *Astrophys. J., Lett.*, 321, L107
- Shima, E., Matsuda, T., Takeda, H., & Sawada, K. 1985, *Mon. Not. R. Astron. Soc.*, 217, 367
- Sinha, M., & Holley-Bockelmann, K. 2009, *Mon. Not. R. Astron. Soc.*, 397, 190
- . 2010, *Mon. Not. R. Astron. Soc.*, 405, L31
- Sirko, E. 2005, *Astrophys. J.*, 634, 728
- Springel, V. 2005, *Mon. Not. R. Astron. Soc.*, 364, 1105
- Springel, V., Yoshida, N., & White, S. D. M. 2001, *New Astron.*, 6, 79
- Tanaka, T., Perna, R., & Haiman, Z. 2012, *Mon. Not. R. Astron. Soc.*, 425, 2974
- Tegmark, M., Silk, J., Rees, M. J., et al. 1997, *Astrophys. J.*, 474, 1
- Venkatesan, A., Tumlinson, J., & Shull, J. M. 2003, *Astrophys. J.*, 584, 621
- Voit, G. M. 2005, *Reviews of Modern Physics*, 77, 207
- Wechsler, R. H., Bullock, J. S., Primack, J. R., Kravtsov, A. V., & Dekel, A. 2002, *Astrophys. J.*, 568, 52
- White, M. 2001, *Astron. Astrophys.*, 367, 27
- Zel'dovich, Y. B. 1970, *Astron. Astrophys.*, 5, 84
- Zhao, D. H., Jing, Y. P., Mo, H. J., & Börner, G. 2009, *Astrophys. J.*, 707, 354
- Zhao, D. H., Mo, H. J., Jing, Y. P., & Börner, G. 2003, *Mon. Not. R. Astron. Soc.*, 339, 12

Appendices

Appendix A

BGC2 Python Code

```
1  #!/usr/bin/env python
2
3  import sys
4  import struct
5
6  def read_bgc2(filename):
7      offset = 4
8      groupoffset = 8
9      particleoffset = 8
10
11     headersize = 1024
12     groupsize = 4*8 + 10*4
13     particlesize = 1*8 + 6*4
14
15     headerformat = '=Q16q19d'
16     groupformat = '=2q2Q10f'
17     particleformat = '=q6f'
18
19     print "Reading"+filename+"..."
20     fd = open(filename, 'rb')
21     bin_string = fd.read()
22     fd.close()
23     print "Finished reading file."
24     bin_string = bin_string[offset:]
25
26     # Header stuff
27     header_bin = bin_string[:headersize]
28     header_pad = headersize - 36*8
29     header = list(struct.unpack(headerformat, header_bin[:-header_pad]))
30
31     # Group stuff
32     ngroups = header[8]
33     print 'ngroups=%d' % ngroups
34     groupstart = headersize + groupoffset
35     groupend = groupstart + ngroups*groupsizesize
36     group_bin = bin_string[groupstart:groupend]
37     group = []
38     for i in range(ngroups):
39         group.append(list(struct.unpack(groupformat, group_bin[i*groupsizesize:(i+1)*groupsizesize])))
40
41     # Particle stuff
42     particlestart = headersize + groupoffset + ngroups*groupsizesize + particleoffset
43     particle_bin = bin_string[particlestart:]
44     particle = []
45     p_start = 0
46     for i in range(ngroups):
47         npart = group[i][2]
48         particle.append([])
49         for j in range(npart):
50             particle[i].append(list(struct.unpack(particleformat, particle_bin[p_start:p_start+particlesize])))
51             p_start += particlesize
52             p_start += particleoffset
53
54     print "Finished parsing bgc2 file"
55     return header, group, particle
56
57
58 def main():
59     header, group, particle = read_bgc2(sys.argv[1])
60
61     print 'Header contents:'
62     for value in header:
63         print value
64     print
65
66     print 'Group[0] contents:'
67     for value in group[0]:
68         print value
69     print
70
71     print 'Particles in group[0]:'
72     for part in particle[0]:
73         print part
74     print
75
76     print 'Group[1] contents:'
77     for value in group[1]:
78         print value
79     print
80
81     print 'Particles in group[1]:'
82     for part in particle[1]:
```

```
83     print part
84
85
86
87
88
89 if __name__ == '__main__':
90     main()
```

Appendix B

Density Profile Python Code

```
1  #!/usr/bin/env python
2
3  import sys
4  import bgc2
5  import numpy as np
6  import matplotlib.pyplot as plt
7  from matplotlib.ticker import MultipleLocator
8  from scipy.optimize import curve_fit
9  from scipy.stats import chisquare
10
11  #read_mode = 'ascii2'
12  read_mode = 'bgc2'
13
14  if read_mode == 'bgc2':
15      use_bgc2 = True
16      use_all = False
17      individual_masses = False
18      halo_id = 146289
19      nbins = 50
20      nfit = 500
21      ooms = 3.0
22      mass_scale = 1.0
23      common_mass = 5.33423e5
24      dist_scale = 1.0e3
25      #res_limit = 0.488
26      #res_limit = 4.0
27      res_limit = 0.5
28      #res_limit = 10.0
29      draw_frac = 0.1
30      tick_base_major = 100.0
31      tick_base_minor = 10.0
32      find_com = False
33  elif read_mode == 'ascii':
34      use_bgc2 = False
35      use_all = True
36      individual_masses = True
37      halo_id = 0
38      nbins = 100
39      nfit = 500
40      ooms = 5.0
41      mass_scale = 1.0e12
42      dist_scale = 200.0
43      res_limit = 1.0e-2
44      draw_frac = 2.0e-4
45      tick_base_major = 80.0
46      tick_base_minor = 20.0
47      find_com = True
48  elif read_mode == 'ascii2':
49      use_bgc2 = False
50      use_all = True
51      individual_masses = True
52      halo_id = 0
53      nbins = 100
54      nfit = 500
55      ooms = 3.5
56      mass_scale = 1.0e10
57      dist_scale = 1.0
58      #res_limit = 3.0e-1
59      res_limit = 1.0
60      draw_frac = 1.0e-2
61      tick_base_major = 200.0
62      tick_base_minor = 40.0
63      find_com = True
64  else:
65      sys.exit(98712)
66
67  #outfile = 'asciitest_halo_properties.txt'
68  outfile = 'density_profile_halos.dat'
69  comfile = 'center_of_mass.txt'
70
71  make_plot = False
72  #make_plot = True
73  draw_density = True
74  #plot_base = 'asciitest_density_profile.fig.'
75  plot_base = 'figure_'
76  plot_ext = '.eps'
77  dist_units = 'kpc'
78  xlabel_proj = [r'XUPositionU(%sUh$^{-1}$)' % (dist_units), r'XUPositionU(%sUh$^{-1}$)' % (dist_units), r'YU
79  PositionU(%sUh$^{-1}$)' % (dist_units)]
80  ylabel_proj = [r'YUPositionU(%sUh$^{-1}$)' % (dist_units), r'ZUPositionU(%sUh$^{-1}$)' % (dist_units), r'ZU
81  PositionU(%sUh$^{-1}$)' % (dist_units)]
82  xlabel_prof = r'RadiusU(%sUh$^{-1}$)' % (dist_units)
```



```

81 ylabel_prof = r'Density(M$_{\odot}$%s$^{-3}$h$^{2}$)' % (dist_units)
82 npixels = 50
83
84 #common_mass = 1.0e-7
85 #common_mass = 1.0e5
86 mass_col = 0
87 pos_cols = (1,2,3)
88 vel_cols = (4,5,6)
89 halo_id_col = 0
90
91 grav_const = 4.3e-6 # kpc M_sol^-1 (km/s)^2
92
93
94 def read_files(files):
95     data = 0
96     for file in files:
97         print 'Reading file %s...' % (file)
98         if data == 0:
99             data = np.genfromtxt(file, comments='#')
100         else:
101             data = np.append(data, np.genfromtxt(file, comments='#'), axis=0)
102     print 'Finished reading files.'
103     return data
104
105
106 def my_chisq(ydata, ymod, deg=2, sd=None):
107     """
108     Returns the reduced chi-square error statistic for an arbitrary model,
109     chisq/nu, where nu is the number of degrees of freedom. If individual
110     standard deviations (array sd) are supplied, then the chi-square error
111     statistic is computed as the sum of squared errors divided by the standard
112     deviations. See http://en.wikipedia.org/wiki/Goodness\_of\_fit for reference.
113     ydata, ymod, sd assumed to be Numpy arrays. deg integer.
114
115     Usage:
116     >>> chisq=redchisq(ydata, ymod, n, sd)
117     where
118     ydata: data
119     ymod: model evaluated at the same x points as ydata
120     n: number of free parameters in the model
121     sd: uncertainties in ydata
122
123     Rodrigo Nemmen
124     http://goo.gl/8S10o
125     """
126     # Chi-square statistic
127     if sd==None:
128         chisq=np.sum((ydata-ymod)**2)
129     else:
130         chisq=np.sum( ((ydata-ymod)/sd)**2 )
131
132     # Number of degrees of freedom assuming 2 free parameters
133     nu=ydata.size-1-deg
134     return chisq/nu
135
136
137
138 def calc_m_enclosed(mass, pos):
139     r = np.sqrt(pos[:,0]**2 + pos[:,1]**2 + pos[:,2]**2)
140     r = np.sort(r)
141     first_good_bin = 0
142     for i in range(len(r)):
143         if r[i] > res_limit:
144             first_good_bin = i
145             break
146     print 'r1=', r[first_good_bin-1]
147     print 'r2=', r[first_good_bin]
148     print 'r3=', r[first_good_bin+1]
149     m_extra = mass[0] * first_good_bin
150     r = r[first_good_bin:]
151     #m_enclosed = np.zeros(len(r))
152     #for i in range(len(r)):
153     #    m_enclosed[i] = mass[0] * (i + 1.0)
154     m_enclosed = (np.arange(len(r)) + 1.0) * mass[0] + m_extra
155     return r, m_enclosed
156
157
158 def calc_density_profile(mass, pos):
159     r = np.sqrt(pos[:,0]**2 + pos[:,1]**2 + pos[:,2]**2)
160     max_r = r.max()
161     #min_r = max_r / 10**ooms
162     min_r = res_limit
163     log_range = np.log10(max_r) - np.log10(min_r)
164
165     #global nbins
166     local_nbins = float(nbins + 1)
167     #nbins = len(r) / 1000
168     while True:
169         bins = np.arange(local_nbins)
170         bins = max_r * 10.0**(log_range * bins / (local_nbins-1.0) - log_range)
171         bin_mass, r_bins = np.histogram(r, bins, weights=mass)
172         if (bin_mass == 0.0).any():

```

```

173     local_nbins -= 1
174     continue
175     else:
176         break
177
178     #print 'Binning particles using bin edges of \n', r_bins
179
180     rho = bin_mass / (sphere_vol(r_bins[1:]) - sphere_vol(r_bins[:-1]))
181
182     N_bin, blah = np.histogram(r, bins)
183     rho_err = poisson_error(N_bin) * rho
184
185     return r_bins, rho, rho_err
186
187
188 def logbin(pos):
189     r = np.sqrt(pos[:,0]**2 + pos[:,1]**2 + pos[:,2]**2)
190     max_r = r.max()
191     min_r = max_r / 10**ooms
192     log_range = np.log10(max_r) - np.log10(min_r)
193
194     global nbins
195     nbins = float(nbins + 1)
196     bins = np.arange(nbins)
197     bins = max_r * 10.0**(log_range * bins / (nbins-1.0) - log_range)
198
199     hist, bin_edges = np.histogram(r, bins)
200     #print 'Binning particles using bin edges of \n', bin_edges
201     return hist, bin_edges
202
203
204 def poisson_error(N):
205     err = np.sqrt(N) / N
206     return err
207
208
209 def sphere_vol(r):
210     volume = (4.0 / 3.0) * np.pi * r**3
211     return volume
212
213
214 def get_rho_0(R_s, R_vir):
215     H = 70.0e-3 # km s^-1 kpc^-1
216     G = 4.3e-6 # kpc M_sol^-1 (km/s)^2
217     rho_crit = 3.0 * H**2 / (8.0 * np.pi * G)
218
219     v = 178
220     c = R_vir / R_s
221     g = 1.0 / (np.log(1.0+c) - c/(1.0+c))
222     delta_char = v * c**3 * g / 3.0
223
224     return rho_crit * delta_char
225
226
227 def nfw_fit_rho0(r, R_s, rho_0):
228     if R_s >= 1.0:
229         return (R_s - 1.0) * np.exp(r) + rho_0 / ((r / R_s) * (1.0 + r / R_s)**2)
230     return rho_0 / ((r / R_s) * (1.0 + r / R_s)**2)
231
232
233 def nfw_fit_rho0_log(r, R_s, rho_0):
234     r = 10.0**r
235     R_s = 10.0**R_s
236     rho_0 = 10.0**rho_0
237     profile = rho_0 / ((r / R_s) * (1.0 + r / R_s)**2)
238     return np.log10(profile)
239
240
241 def nfw_def_rho0(R_vir):
242     def _nfw_def_rho0(r, R_s):
243         rho_0 = get_rho_0(R_s, R_vir)
244         return rho_0 / ((r / R_s) * (1.0 + r / R_s)**2)
245     return _nfw_def_rho0
246
247
248 def nfw_databin_rho0(rho_0):
249     def _nfw_databin_rho0(r, R_s):
250         return rho_0 / ((r / R_s) * (1.0 + r / R_s)**2)
251     return _nfw_databin_rho0
252
253
254 def dm_profile_fit_rho0_log(r, R_s, rho_0, alpha):
255     r = 10.0**r
256     R_s = 10.0**R_s
257     rho_0 = 10.0**rho_0
258     alpha = 10.0**alpha
259     profile = rho_0 / ((r / R_s) * (1.0 + r / R_s)**alpha)
260     return np.log10(profile)
261
262
263 def dm_profile_fit_rho0(r, R_s, rho_0, alpha):
264     return rho_0 / ((r / R_s) * (1.0 + r / R_s)**alpha)

```

```

265
266
267 def dm_profile_def_rho0(R_vir):
268     def _dm_profile_def_rho0(r, R_s, alpha):
269         rho_0 = get_rho_0(R_s, R_vir)
270         return rho_0 / ((r / R_s) * (1.0 + r / R_s)**alpha)
271     return _dm_profile_def_rho0
272
273
274 def dm_profile_databin_rho0(rho_0):
275     def _dm_profile_databin_rho0(r, R_s, alpha):
276         return rho_0 / ((r / R_s) * (1.0 + r / R_s)**alpha)
277     return _dm_profile_databin_rho0
278
279
280 def nfw_cdf(r, R_s, rho_0):
281     r = 10.0**r
282     R_s = 10.0**R_s
283     rho_0 = 10.0**rho_0
284     profile = rho_0 * R_s * (np.log(1.0 + r / R_s) - 1.0 / (1.0 + r / R_s))
285     return np.log10(profile)
286
287
288 def nfw_cdf_nolog(r, R_s, rho_0):
289     profile = rho_0 * R_s * (np.log(1.0 + r / R_s) - 1.0 / (1.0 + r / R_s))
290     return profile
291
292
293 def mass_profile(s, c):
294     g = 1.0 / (np.log(1.0 + c) - c / (1.0 + c))
295     return g * (np.log(1.0 + c * s) - c * s / (1.0 + c * s))
296
297
298 def fit_mass_profile(s, m_enclosed, err=None, R_vir=None):
299     #for i in range(len(s)):
300     #    if s[i] > res_limit:
301     #        first_good_bin = i
302     #        break
303     first_good_bin = 0
304
305     #popt, pcov = curve_fit(nfw_cdf, np.log10(r), np.log10(m_outside), sigma=np.log10(err))
306     # popt, pcov = curve_fit(nfw_cdf, np.log10(r), np.log10(m_outside))
307     # popt = 10.0**popt
308     # pcov = 10.0**pcov
309     popt, pcov = curve_fit(mass_profile, s, m_enclosed)
310
311     print 'fit_params=', popt
312     print 'covariance=', pcov
313     nfw_r = np.linspace(s[0], s[-1], nfit)
314     nfw_fit = mass_profile(nfw_r, popt[0])
315     chi2_fit = mass_profile(s, popt[0])
316
317     chi2 = chisquare(np.log10(m_enclosed[first_good_bin:]), np.log10(chi2_fit[first_good_bin:]))
318     chi2_nolog = chisquare(m_enclosed[first_good_bin:], chi2_fit[first_good_bin:])
319     print 'chi_square=', chi2
320     print 'chi_square_nolog=', chi2_nolog
321     return nfw_r, nfw_fit, popt, pcov, chi2[0]
322
323
324 def fit_profile(r, rho, err=None, R_vir=None):
325     first_good_bin = 0
326     # for i in range(len(r)):
327     #     if r[i] > res_limit:
328     #         rho_0_databin = rho[i]
329     #         first_good_bin = i
330     #         break
331     # print 'first_good_bin =', first_good_bin
332
333     #----- choose one fitting type -----#
334     #popt, pcov = curve_fit(nfw_fit_rho0, r, rho, sigma=err)
335     #popt, pcov = curve_fit(nfw_def_rho0(R_vir), r, rho, p0=[10.0], sigma=err)
336     #popt, pcov = curve_fit(nfw_databin_rho0(rho_0_databin), r, rho, sigma=err)
337     blah = 3
338     if blah == 0:
339         for i in range(100):
340             a = 2.0 * np.random.random() * 0.1 * r.max()
341             b = 2.0 * np.random.random() * 10.0
342             c = 2.0 * np.random.random() * 2.0
343             try:
344                 popt, pcov = curve_fit(dm_profile_fit_rho0, r, rho, p0=[a,b,c], sigma=err)
345             except RuntimeError:
346                 continue
347             if (popt[0] < r.max()) and (popt[2] >= 0.0):
348                 break
349             elif i >= 99:
350                 print 'no good fit found for this halo...'
351                 return None, None, None, None, None
352     elif blah == 1:
353         #a = r.max() / 100.0
354         a = 0.001
355         b = rho[first_good_bin]
356         c = 0.001

```

```

357     #popt, pcov = curve_fit(dm_profile_fit_rho0, r, rho, sigma=err)
358     print '-----'
359     print 'rho_0_before=', b
360     #try:
361     #popt, pcov = curve_fit(dm_profile_fit_rho0, r, rho, p0=[a,b,c], sigma=err, maxfev=1, xtol=100.0)
362     popt, pcov = curve_fit(dm_profile_fit_rho0, r, rho, p0=[a,b,c], sigma=err, xtol=1.0e-1)
363     #except RuntimeError:
364     #    print 'just checking for now...'
365     print 'rho_0_after=', popt[1]
366     #sys.exit()
367     elif blah == 2:
368     #popt, pcov = curve_fit(dm_profile_fit_rho0_log, np.log10(r), np.log10(rho), sigma=np.log10(err))
369     popt, pcov = curve_fit(nfw_fit_rho0_log, np.log10(r), np.log10(rho), sigma=np.log10(err))
370     popt = 10.0**popt
371     pcov = 10.0**pcov
372     elif blah == 3:
373     popt, pcov = curve_fit(nfw_fit_rho0, r, rho, sigma=err, p0 = [0.1, 1.0])
374
375     #popt, pcov = curve_fit(dm_profile_def_rho0(R_vir), r, rho, sigma=err)
376     #popt, pcov = curve_fit(dm_profile_databin_rho0(rho_0_databin), r, rho, sigma=err)
377     #-----#
378
379     print 'fit_params=', popt
380     print 'covariance=', pcov
381
382     nfw_r = np.linspace(r[0], r[-1], nfit)
383     #----- choose one fitting type -----#
384     nfw_fit = nfw_fit_rho0(nfw_r, popt[0], popt[1])
385     #nfw_fit = nfw_def_rho0(R_vir)(nfw_r, popt[0])
386     #nfw_fit = nfw_databin_rho0(rho_0_databin)(nfw_r, popt[0])
387     #nfw_fit = dm_profile_fit_rho0(nfw_r, popt[0], popt[1], popt[2])
388     #nfw_fit = dm_profile_def_rho0(R_vir)(nfw_r, popt[0], popt[1])
389     #nfw_fit = dm_profile_databin_rho0(rho_0_databin)(nfw_r, popt[0], popt[1])
390     #-----#
391     #----- choose one fitting type -----#
392     chi2_fit = nfw_fit_rho0(r, popt[0], popt[1])
393     #chi2_fit = nfw_def_rho0(R_vir)(r, popt[0])
394     #chi2_fit = nfw_databin_rho0(rho_0_databin)(r, popt[0])
395     #chi2_fit = dm_profile_fit_rho0(r, popt[0], popt[1], popt[2])
396     #chi2_fit = dm_profile_def_rho0(R_vir)(r, popt[0], popt[1])
397     #chi2_fit = dm_profile_databin_rho0(rho_0_databin)(r, popt[0], popt[1])
398     #-----#
399
400     #chi2 = my_chisq(rho, chi2_fit, 2, err)
401     chi2 = chisquare(rho, chi2_fit)
402     print 'chi_square=', chi2
403     chi2 = chi2[0]
404
405     return nfw_r, nfw_fit, popt, pcov, chi2
406
407
408 def draw_projection(fig, place, plot_lim, x, y):
409     ax = plt.subplot(2,3,place+1, aspect='equal')
410     im = ax.plot(x, y, linestyle='', marker='.', markersize=1, markeredgcolor='blue')
411     ax.set_xlabel(xlabel_proj[place])
412     ax.set_ylabel(ylabel_proj[place])
413     ax.set_xlim(-plot_lim, plot_lim)
414     ax.set_ylim(-plot_lim, plot_lim)
415     ax.xaxis.set_major_locator(MultipleLocator(tick_base_major))
416     ax.xaxis.set_minor_locator(MultipleLocator(tick_base_minor))
417     ax.yaxis.set_major_locator(MultipleLocator(tick_base_major))
418     ax.yaxis.set_minor_locator(MultipleLocator(tick_base_minor))
419     return fig
420
421
422 def draw_density_projection(fig, place, plot_lim, x, y):
423     limits = [[-plot_lim, plot_lim], [-plot_lim, plot_lim]]
424     ax = plt.subplot(2,3,place+1, aspect='equal')
425     #ax.set_xlim(-plot_lim, plot_lim)
426     #ax.set_ylim(-plot_lim, plot_lim)
427     #im = ax.plot(x, y, linestyle='', marker='.', markersize=1, markeredgcolor='blue')
428     z, xedges, yedges = np.histogram2d(x, y, bins = npixels, range = limits)
429     #z = np.log10(z)
430     im = ax.imshow(z.T, extent=(-plot_lim, plot_lim, -plot_lim, plot_lim), interpolation='gaussian', origin='lower')
431
432     ax.locator_params(nbins=6)
433     ax.set_xlabel(xlabel_proj[place])
434     ax.set_ylabel(ylabel_proj[place])
435     # ax.xaxis.set_major_locator(MultipleLocator(tick_base_major))
436     # ax.yaxis.set_major_locator(MultipleLocator(tick_base_major))
437     # ax.xaxis.set_minor_locator(MultipleLocator(tick_base_minor))
438     # ax.yaxis.set_minor_locator(MultipleLocator(tick_base_minor))
439     return fig
440
441 def draw_density_profile(fig, r, rho, err=None):
442     ax = plt.subplot(2,1,2)
443     im = ax.loglog(r, rho, linestyle='steps-mid-')
444     line1 = ax.axvline(res_limit, color='black', linestyle=':')
445     #ax.set_xlim(r_bins[0], r_bins[-1])
446     ax.set_xlim(r[0] - (r[1]-r[0]), r[-1] + (r[-1]-r[-2]))
447     ax.set_xlabel(xlabel_prof)

```

```

448 ax.set_ylabel(ylabel_prof)
449 if err != None:
450     err_bars = ax.errorbar(r, rho, yerr=err, linestyle='None')
451 return fig, ax
452
453
454 def draw_nfw_profile(fig, ax, r, rho, R_s=None):
455     ax.loglog(r, rho, linestyle='-', color='red')
456     if R_s != None:
457         line = ax.axvline(R_s, color='purple', linestyle='-.')
458     return fig
459
460
461 def calc_kinetic_energy(mass, vel):
462     vsq = vel[:,0]**2 + vel[:,1]**2 + vel[:,2]**2
463     energy = 0.5 * np.sum(mass*vsq)
464     return energy
465
466
467 def calc_potential_energy(mass, pos):
468     local_sqrt = np.sqrt
469     partial_sum = 0.0
470     for i in range(len(mass)):
471         for j in range(len(mass)):
472             if j != i:
473                 r_diff = local_sqrt((pos[i,0] - pos[j,0])**2 + (pos[i,1] - pos[j,1])**2 + (pos[i,2] - pos[j,2])**2)
474                 partial_sum = partial_sum - mass[i]*mass[j]/r_diff
475     energy = partial_sum * grav_const / 2.0
476     return energy
477
478
479 def calc_angular_momentum(mass, pos, vel):
480     ang_mom_x = np.sum(mass * (pos[:,1] * vel[:,2] - pos[:,2] * vel[:,1]))
481     ang_mom_y = np.sum(mass * (pos[:,2] * vel[:,0] - pos[:,0] * vel[:,2]))
482     ang_mom_z = np.sum(mass * (pos[:,1] * vel[:,2] - pos[:,2] * vel[:,1]))
483     ang_mom = np.sqrt(ang_mom_x**2 + ang_mom_y**2 + ang_mom_z**2)
484     return ang_mom
485
486
487 def main():
488     with open(outfile, 'w') as fd:
489         #fd.write('#halo_mass concentration R_vir R_s +- err rho_0 +- err alpha +- err chi_square\n')
490         fd.write('#halo_id'+'#halo_mass'+'#x_pos'+'#y_pos'+'#z_pos'+'#c_u'+'#err'+'#R_vir'+'#R_s'+'#err'+'#rho_0'+'#err'+'#alpha'+'#err'+'#chi_square'+'#nbins'+'#N_part\n')
491     # with open(comfile, 'w') as fd:
492     #     fd.write('#id mass dx dy dz\n')
493
494     # if use_bgc2 == True:
495     #     header, halos, particles = bgc2.read_bgc2(sys.argv[1])
496     #     for i in range(len(halos)):
497     #         if halos[i][halo_id_col] == halo_id:
498     #             index = i
499     #     halo_particles = np.asarray(particles[index])
500     #     pos = halo_particles[:,pos_cols[0]:pos_cols[0]+3] * dist_scale
501     #     r_vir = halos[index][4] * dist_scale
502     # else:
503     #     # Read in particle files
504     #     data = read_files(sys.argv[1:])
505     #     # Select particles with a given halo ID and convert positions from Mpc to kpc
506     #     if use_all == False:
507     #         halo_particles = data[np.where(data[:,halo_id_col] == halo_id)]
508     #     if use_all == True:
509     #         halo_particles = data
510     #     del data
511     #     pos = halo_particles[:,pos_cols[0]:pos_cols[0]+3] * dist_scale
512     #     r_vir = 241.48
513     #     #r_vir = pos.max()
514
515     for input_file in sys.argv[1:]:
516         if use_bgc2 == True:
517             #header, halos, particles = bgc2.read_bgc2(sys.argv[1])
518             header, halos, particles = bgc2.read_bgc2(input_file)
519             halos = np.asarray(halos)
520             indices = np.argsort(halos[:,2]) # sort by number of particles
521             indices = indices[::-1] # start with the biggest
522         else:
523             data = read_files([input_file])
524             # Select particles with a given halo ID and convert positions from Mpc to kpc
525             if use_all == False:
526                 particles = [data[np.where(data[:,halo_id_col] == halo_id)]]
527             if use_all == True:
528                 particles = [data]
529             del data
530
531
532     itteration = 0
533     #for index in range(len(halos)):
534     #for index in range(1):
535     #for index in indices[:10]:
536     for index in indices:
537         if ((len(particles[index]) >= 100) and (halos[index][1] == -1)):

```

```

538     print '-----',
539
540     halo_particles = np.asarray(particles[index])
541     pos = halo_particles[:,pos_cols[0]:pos_cols[0]+3] * dist_scale
542     vel = halo_particles[:,vel_cols[0]:vel_cols[0]+3]
543
544     if use_bgc2 == True:
545         halo_id = halos[index][0]
546         r_vir = halos[index][4] * dist_scale
547         halo_mass = halos[index][5]
548         halo_pos = np.array([halos[index][6] * dist_scale, halos[index][7] * dist_scale, halos[index][8] *
dist_scale])
549         halo_vel = np.array([halos[index][9], halos[index][10], halos[index][11]])
550     else:
551         r_vir = 241.48
552         halo_id = 0
553         #halo_mass = mass[0] * len(halo_particles)
554         halo_pos = np.array([0.0, 0.0, 0.0])
555         halo_vel = np.array([0.0, 0.0, 0.0])
556
557
558     if individual_masses == True:
559         mass = halo_particles[:,mass_col] * mass_scale
560     else:
561         mass = np.ones(halo_particles.shape[0]) * common_mass * mass_scale
562
563     if use_bgc2 == False:
564         halo_mass = mass[0] * len(halo_particles) #fix placement of this for ascii test
565
566     print 'Using%dparticles_in_halo%d.' % (halo_particles.shape[0], halo_id)
567
568     # Find center of mass
569     if find_com == True:
570         mass_tot = mass.sum()
571         m_pos = mass.reshape(mass.shape[0],1) * pos
572         com = m_pos.sum(axis=0) / mass_tot
573         pos = pos - com
574         print 'Center_of_mass=%(%g,%g,%g)' % (com[0], com[1], com[2])
575     else:
576         pos = pos - halo_pos
577         vel = vel - halo_vel
578
579     #with open(comfile, 'a') as fd:
580     # fd.write("%d %g %g %g %g\n" % (halo_id, halo_mass, halo_pos[0] - com[0], halo_pos[1] - com[1],
halo_pos[2] - com[2]))
581
582     # Bin halo particles into logarithmic shells and compute density
583     r_bins, rho, rho_err = calc_density_profile(mass, pos)
584
585     if len(r_bins) < 5:
586         print 'Too few bins. Skipping this halo.'
587         with open(outfile, 'a') as fd:
588             fd.write("%8d%16.12g%14.10g%14.10g%14.10g%14d+-%14d%14d%14d+-%14d%14d+-%14d%14d+-%14d%14d\n" % (halo_id, halo_mass, halo_pos[0], halo_pos[1], halo_pos[2], -9999, -9999, -9999, -9999, -9999, -9999, -9999, -9999, -9999, -9999, len(halo_particles)))
589         continue
590
591     # hist, r_bins = logbin(pos)
592     # err = poisson_error(hist)
593     # rho = mass * hist / (sphere_vol(r_bins[1:]) - sphere_vol(r_bins[:-1]))
594     # rho_err = err * rho
595     mid_bins = 10.0**(0.5 * (np.log10(r_bins[1:]) + np.log10(r_bins[:-1])))
596     print 'nbins=', len(mid_bins)
597
598     # Don't pass NaN's to fitting routine
599     rho_err_nonan = np.copy(rho_err)
600     nan_check = np.isnan(rho_err_nonan)
601     for i in range(len(rho_err_nonan)):
602         #if (nan_check[i] == True):
603             rho[i] = 1.0e-10
604         if (mid_bins[i] < res_limit) or (nan_check[i] == True):
605             rho_err_nonan[i] = 1.0e10
606
607
608     # r, m_enclosed = calc_m_enclosed(mass, pos)
609
610     # Fit an NFW profile to the data
611     try:
612         nfw_r, nfw_fit, popt, pcov, chisq = fit_profile(mid_bins / r_vir, rho / rho.max(), err = rho_err_nonan /
rho.max(), R_vir = 1.0)
613         #nfw_r, nfw_fit, popt, pcov, chisq = fit_mass_profile(r / r_vir, m_enclosed / halo_mass)
614         nfw_r = nfw_r * r_vir
615         nfw_fit = nfw_fit * rho.max()
616         #nfw_fit = nfw_fit * halo_mass
617         scale_radius = popt[0] * r_vir
618         scale_radius_err = pcov[0,0] * r_vir
619         rho_0 = popt[1] * rho.max()
620         rho_0_err = pcov[1,1] * rho.max()
621         concentration = r_vir / scale_radius
622         concentration_err = concentration * scale_radius_err / scale_radius
623
624     # Print parameters

```


Appendix C

Halo Matching Python Code

C.1 Python Code

```
1  #!/usr/bin/env python
2
3  import sys
4  import getopt
5  import numpy as np
6
7
8  def main():
9      # read and parse command line arguments
10     opts, args = get_args(sys.argv[1:])
11     output_file, match_file, densprof_files, parents_files, ascii_files = parse_args(opts, args)
12
13     # read in headers as lists and data as numpy arrays
14     match_header, match_data = read_files(match_file, header_line = 3)
15     densprof_header1, densprof_data1 = read_files(densprof_files[0], header_line = 0)
16     densprof_header2, densprof_data2 = read_files(densprof_files[1], header_line = 0)
17     parents_header1, parents_data1 = read_files(parents_files[0], header_line = 0)
18     parents_header2, parents_data2 = read_files(parents_files[1], header_line = 0)
19     ascii_header1, ascii_data1 = read_files(ascii_files[(len(ascii_files)/2)], header_line = 0)
20     ascii_header2, ascii_data2 = read_files(ascii_files[(len(ascii_files)/2):], header_line = 0)
21     print 'Finished reading files.'
22
23     # filter matches, remove duplicate halo matches, and reorder match columns
24     print 'Filtering match data...'
25     match_data = filter_matches(match_data)
26     if filter_duplicate_matches:
27         match_data = filter_dups(match_data, unique_col = match_id1_col)
28         match_data = filter_dups(match_data, unique_col = match_id2_col)
29     if reorder_match_columns:
30         match_header, match_data = reorder_match_cols(match_header, match_data)
31
32     # calculate number of subhalos and add column to parents data and headers
33     print 'Finding number of subhalos...'
34     parents_header1.append('N_subs')
35     parents_header2.append('N_subs')
36     parents_data1 = count_subs(parents_data1)
37     parents_data2 = count_subs(parents_data2)
38
39     # create header
40     print 'Making header...'
41     header = make_header(match_header, densprof_header1, densprof_header2, \
42                          parents_header1, parents_header2, ascii_header1, ascii_header2)
43
44     # match halos
45     print 'Matching halos...'
46     halos = match_halos(match_data, [densprof_data1, densprof_data2, \
47                                     parents_data1, parents_data2, ascii_data1, ascii_data2])
48
49     # filter based on given criteria and sort
50     print 'Filtering halo data...'
51     if filter_halo_properties:
52         halos = filter_halos(halos)
53     if sort_col != None:
54         sort_mask = halos[:, sort_col].argsort()
55         sort_mask = sort_mask[::-1]
56         halos = halos[sort_mask]
57
58     # output matched table
59     print 'Writing results...'
60     write_results(output_file, header, halos)
61
62     print 'Finished.'
63
64
65 def get_args(arglist):
66     try:
67         opts, args = getopt.gnu_getopt(arglist, shortopts, longopts)
68     except getopt.GetoptError:
69         print "Invalid option(s)."
70         print help_string
71         sys.exit(2)
72     if opts == []:
73         print 'No options given.'
74         print help_string
75         sys.exit(2)
76     return opts, args
77
78
79 def parse_args(opts, args):
80     densproffiles = None
```



```

81 parentsfiles = None
82 asciifiles = None
83 use_ascii = False
84 for opt in opts:
85     if (opt[0] == '-h') or (opt[0] == '--help') or (opts == None):
86         print help_string
87         sys.exit(0)
88     if (opt[0] == '-o') or (opt[0] == '--outfile'):
89         outfile = opt[1]
90     if (opt[0] == '-m') or (opt[0] == '--match'):
91         matchfile = opt[1]
92     if (opt[0] == '-d') or (opt[0] == '--density'):
93         densproffiles = create_append(densproffiles, opt[1])
94     if (opt[0] == '-p') or (opt[0] == '--parents'):
95         parentsfiles = create_append(parentsfiles, opt[1])
96     if (opt[0] == '-a'):
97         use_ascii = True
98 if use_ascii:
99     if len(args) % 2 != 0:
100         print 'Must have an even number of ascii files!'
101         sys.exit(3)
102     for arg in args:
103         asciifiles = create_append(asciifiles, arg)
104 return outfile, matchfile, densproffiles, parentsfiles, asciifiles
105
106
107 def create_append(lst, value):
108     if lst == None:
109         lst = [value]
110     else:
111         lst.append(value)
112     return lst
113
114
115 def read_files(files, header_line = None, comment_char = '#'):
116     header = None
117     data = None
118     if type(files) == str:
119         files = [files]
120
121     if header_line != None:
122         with open(files[0], 'r') as fd:
123             for line in range(header_line):
124                 fd.readline()
125             header = fd.readline()
126         if header[0] != comment_char:
127             print "Header must start with a '%s'" % comment_char
128             sys.exit(4)
129         header = header[1:]
130         header = header.split()
131
132     for file in files:
133         print 'Reading file %s...' % (file)
134         if data == None:
135             data = np.genfromtxt(file, comments='#')
136         else:
137             data = np.append(data, np.genfromtxt(file, comments='#'), axis=0)
138
139     if header_line == None:
140         return data
141     else:
142         return header, data
143
144
145 def filter_matches(halos):
146     if filter_bad_matches:
147         halos = halos[halos[:, match_id1_col] != -1]
148         halos = halos[halos[:, match_id2_col] != -1]
149     if (min_npart != 0) and (min_npart != None):
150         halos = halos[halos[:, match_npart1_col] >= min_npart]
151         halos = halos[halos[:, match_npart2_col] >= min_npart]
152     if (minperc_ncommon != 0) and (minperc_ncommon != None):
153         halos = halos[halos[:, match_ncommon_col] / halos[:, match_npart1_col] >= minperc_ncommon]
154         halos = halos[halos[:, match_ncommon_col] / halos[:, match_npart2_col] >= minperc_ncommon]
155     return halos
156
157
158 def filter_dups(halos, unique_col = 0):
159     ncommon = halos[:, match_ncommon_col]
160     n1 = halos[:, match_npart1_col]
161     n2 = halos[:, match_npart2_col]
162     rank = ncommon**2 / (n1 * n2) - np.abs(n1 - n2) / (n1 + n2)
163
164     sort_mask = np.argsort(rank)
165     halos = halos[sort_mask]
166
167     unique, mask = np.unique(halos[:, unique_col], return_index=True)
168     halos = halos[mask]
169     return halos
170
171
172 def reorder_match_cols(match_header, match_data):

```

```

173 global match_id1_col
174 global match_id2_col
175 global match_hnum1_col
176 global match_hnum2_col
177 global match_npart1_col
178 global match_npart2_col
179 global match_ncommon_col
180
181 order = [match_id1_col, match_id2_col, \
182          match_hnum1_col, match_hnum2_col, \
183          match_npart1_col, match_npart2_col, \
184          match_ncommon_col]
185 match_header = [match_header[index] for index in order]
186 match_data = match_data[:, order]
187
188 match_id1_col = 0
189 match_id2_col = 1
190 match_hnum1_col = 2
191 match_hnum2_col = 3
192 match_npart1_col = 4
193 match_npart2_col = 5
194 match_ncommon_col = 6
195
196 return match_header, match_data
197
198
199 def count_subs(halos):
200     id = halos[:, id_col]
201     parents = halos[:, parents_col]
202     parents = parents[parents != -1]
203     nsubs = (id[:, np.newaxis] == parents).sum(axis = 1)
204     halos = np.column_stack((halos, nsubs))
205     return halos
206
207
208 def make_header(match, densprof1, densprof2, parents1, parents2, ascii1, ascii2):
209     # zeroeth line just lists column number
210     total_len = len(match + densprof1 + densprof2 + parents1 + parents2 + ascii1 + ascii2)
211     header_line0 = [str(i) for i in range(total_len)]
212     header_line0 = 'uu'.join(header_line0)
213     header_line0 = '#' + header_line0
214
215     # first line denotes which file columns are from
216     match_repeat = len(match) - 4
217     densprof_repeat = len(densprof1 + densprof2) - 4
218     parents_repeat = len(parents1 + parents2) - 4
219     ascii_repeat = len(ascii1 + ascii2) - 4
220
221     match_part = 'uu'.join(['|---', 'cross', 'match'] + ['----'] * match_repeat + ['---|'])
222     densprof_part = 'uu'.join(['|---', 'density', 'profile'] + ['----'] * densprof_repeat + ['---|'])
223     parents_part = 'uu'.join(['|---', 'rockstar', 'parents'] + ['----'] * parents_repeat + ['---|'])
224     ascii_part = 'uu'.join(['|---', 'rockstar', 'ascii'] + ['----'] * ascii_repeat + ['---|'])
225
226     header_line1 = 'uu'.join([match_part, densprof_part, parents_part, ascii_part])
227     header_line1 = '#' + header_line1
228
229     # second line labels 2lpt and za columns
230     tot_len = len(match + densprof1 + densprof2 + parents1 + parents2 + ascii1 + ascii2)
231     header_line2 = ['2lpt' if i % 2 == 0 else 'za' if i % 2 == 1 else 'blah' for i in range(tot_len - 1)]
232     header_line2.insert(len(match) - 1, 'matched')
233     header_line2 = 'uu'.join(header_line2)
234     header_line2 = '#' + header_line2
235
236     # third line pulls labels from original file headers
237     match_part = match
238     densprof_part = interweave(densprof1, densprof2)
239     parents_part = interweave(parents1, parents2)
240     ascii_part = interweave(ascii1, ascii2)
241
242     header_line3 = match_part + densprof_part + parents_part + ascii_part
243     header_line3 = 'uu'.join(header_line3)
244     header_line3 = '#' + header_line3
245
246     header = [header_line0, header_line1, header_line2, header_line3]
247     return header
248
249
250 def interweave(list1, list2):
251     newlist = list1 + list2
252     newlist[::2] = list1
253     newlist[1::2] = list2
254     return newlist
255
256
257 def interweave_np_2d(array1, array2):
258     newarray = np.empty((len(array1), len(array1[0]) + len(array2[0])))
259     newarray[:, ::2] = array1
260     newarray[:, 1::2] = array2
261     return newarray
262
263
264 def match_halos(matches, arrays):

```

```

265     halos = matches.copy()
266     for i, array in enumerate(arrays):
267         if array != None:
268             match_id_col = i % 2
269             halos = sort_stack(halos, array, match_id_col)
270
271     # interweave columns so that matching 2lpt/za columns are adjacent
272     tmp_halos = halos
273     halos = np.empty((len(tmp_halos), len(tmp_halos[0])))
274     halos[:, :len(matches[0])] = matches
275     startcol = len(matches[0])
276     for i in range(0, len(arrays), 2):
277         colrange1 = len(arrays[i][0])
278         colrange2 = len(arrays[i+1][0])
279         endcol = startcol + colrange1 + colrange2
280
281         cols1 = tmp_halos[:, startcol:startcol+colrange1]
282         cols2 = tmp_halos[:, startcol+colrange1:startcol+colrange1+colrange2]
283
284         halos[:, startcol:endcol] = interweave_np_2d(cols1, cols2)
285         startcol = endcol
286     return halos
287
288
289 def sort_stack(halos, array, match_id_col):
290     # add empty columns to halos to later fill with halo data
291     rows = len(halos)
292     origcols = len(halos[0])
293     newcols = len(array[0])
294     empty = np.empty((rows, newcols))
295     empty[:] = np.nan
296     halos = np.column_stack((halos, empty))
297
298     # remove halos from array with no matches
299     match_id = halos[:, match_id_col]
300     array_id = array[:, id_col]
301     array_mask = np.in1d(array_id, match_id)
302     array = array[array_mask]
303
304     # create mask so we only add lines for halos in array
305     array_id = array[:, id_col]
306     halo_mask = np.in1d(match_id, array_id)
307     masked_halos = halos[halo_mask]
308
309     # create masks to sort by halo id
310     match_id_sort_mask = np.argsort(masked_halos[:, match_id_col])
311     sorted_masked_halos = masked_halos[match_id_sort_mask]
312
313     # sort array by halo id and copy to empty columns of view of halos
314     array_id_sort_mask = np.argsort(array[:, id_col])
315     sorted_masked_halos[:, origcols:] = array[array_id_sort_mask]
316
317     # 'unmask' - put data back in original halos
318     masked_halos[match_id_sort_mask] = sorted_masked_halos
319     halos[halo_mask] = masked_halos
320
321     return halos
322
323
324 def filter_halos(halos):
325     #todo
326     return halos
327
328
329 def write_results(output_file, header, halos):
330     format = get_format(halos[0])
331     with open(output_file, 'w') as fd:
332         for line in header:
333             fd.write(line + '\n')
334     np.savetxt(fd, halos, fmt=format)
335
336
337 def get_format(line):
338     format = ['%d' if col in int_cols else '%1.14g' for col in range(len(line))]
339     format = '_'.join(format)
340     return format
341
342
343 help_string = '''
344 Available options are:
345 ----h, --help
346 ----v, --verbose
347 ----o<outfile>, --outfile<outfile>
348 ----m<matchlist>, --match<matchlist>
349 ----d<densityprofile_file>, --density<densityprofile_file>
350 ----p<parents_file>, --parents<parents_file>
351 ----a<ascii_files>, --ascii<ascii_files>_must_be_last)
352 '''
353 shortopts = "hvo:m:d:p:a"
354 longopts = ["help", "verbose", "outfile=", "matchfile=", "density=", "parents=", "ascii"]
355
356 int_cols = []

```

```

357 lt_vals = []
358
359 gt_cols = []
360 gt_vals = []
361
362 eq_cols = []
363 eq_vals = []
364
365 ne_cols = []
366 ne_vals = []
367
368 #int_cols = [0, 1, 2, 3, 4, 5, 6, 7, 8]
369 int_cols = []
370
371 match_id2_col      = 1
372 match_npart2_col   = 2
373 match_id1_col      = 4
374 match_npart1_col   = 5
375 match_ncommon_col  = 6
376 match_hnum1_col    = 3
377 match_hnum2_col    = 0
378
379 id_col             = 0      # col of each input file
380 sort_col           = 47     # col of final table - use None to turn off sorting
381 parents_col        = -1
382
383 filter_bad_matches = True
384 filter_duplicate_matches = False
385 reorder_match_columns = True
386 filter_halo_properties = False
387 min_npart          = 20     # use 0 or None to use all size halos
388 minperc_ncommon    = 0.05  # a fraction, use 0 or None to use any match percent
389
390
391 if __name__ == '__main__':
392     main()

```

C.2 Submission Script

```

1  #!/usr/bin/env bash
2  #PBS -M djsissom@gmail.com
3  #PBS -m bae
4  #PBS -l nodes=1:ppn=1
5  #PBS -l pmem=40000mb
6  #PBS -l mem=4000mb
7  #PBS -l walltime=1:00:00
8  #PBS -o out.log
9  #PBS -j oe
10
11 minsnap=0
12 maxsnap=61
13
14 minbox=1
15 maxbox=3
16
17 # Change to working directory
18 echo $PBS_NODEFILE
19 cd $PBS_0_WORKDIR
20
21 for ((i=$minbox; i<=$maxbox; i++)); do
22
23     for ((snap=$minsnap; snap<=$maxsnap; snap++)); do
24
25         if [ $snap -lt 10 ]; then
26             j=00$snap
27         elif [ $snap -lt 100 ]; then
28             j=0$snap
29         fi
30
31         base_dir="/projects/simulations/rockstar/box${i}"
32         crossmatch_dir=${base_dir}/crossmatch/snap${j}
33         snap_dir_2lpt=${base_dir}/2lpt/snap${j}
34         snap_dir_za=${base_dir}/za/snap${j}
35         logfile=${crossmatch_dir}/match_halos.log
36
37         echo "Starting box${i} snap${j}..."
38
39         {
40             #mpiexec -verbose -n 1 \
41             ./match.py -o ${crossmatch_dir}/halos.dat \
42                 -m ${crossmatch_dir}/crossmatch_000.txt \
43                 -d ${snap_dir_2lpt}/halos/density_profile_halos.dat \
44                 -d ${snap_dir_za}/halos/density_profile_halos.dat \
45                 -p ${snap_dir_2lpt}/halos/out_0.list.parents \
46                 -p ${snap_dir_za}/halos/out_0.list.parents \
47                 -a \
48                 ${snap_dir_2lpt}/halos/halos_0.*.ascii \
49                 ${snap_dir_za}/halos/halos_0.*.ascii \
50                 > ${logfile} 2>&1
51
52             echo 'Aligning columns...' >> ${logfile} 2>&1

```

```

53     column -t ${crossmatch_dir}/halos.dat > ${crossmatch_dir}/tmp156546.dat 2>> ${logfile}
54     mv ${crossmatch_dir}/tmp156546.dat ${crossmatch_dir}/halos.dat 2>> ${logfile}
55     echo 'Finished.' >> ${logfile} 2>&1
56     echo "Finished_box${i}_snap${j}"
57 }
58 #} &
59
60 done
61
62 done
63
64 wait
65
66 # - end of script

```