UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE ASTRONOMÍA

A STUDY OF GALACTIC STAR FORMATION AND MASSIVE BLACK HOLE GROWTH THROUGH SIMULATIONS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS, MENCIÓN ASTRONOMÍA

FERNANDO FELIPE BECERRA SAAVEDRA

PROFESOR GUÍA:

ANDRÉS ESCALA ASTORQUIZA

MIEMBROS DE LA COMISIÓN:

JORGE CUADRA STIPETICH
DIEGO MARDONES PÉREZ
RICARDO MUÑOZ VIDAL

SANTIAGO DE CHILE
AGOSTO 2012

Estudiamos la formación estelar en escalas galácticas y el crecimiento de agujeros negros a través de simulaciones usando el código de grilla adaptativa de refinamiento Enzo. Nuestro estudio se centra en dos de las más famosas leyes de formación estelar: las ley de Kennicutt-Schmidt y la ley de Silk, ambas relacionan la tasa de formación estelar con propiedades globales de la galaxia. A pesar de que ambas han cuantificado exitosamente esta relación, aún no hay un consenso en el valor exacto de sus pendientes. Nosotros tratamos de clarificar este problema estudiando los factores que influyen en ambas leyes, tales como la densidad de gas superficial, el tiempo orbital y la masa rotacional. Para estudiar el crecimiento de agujeros negros masivos, centramos nuestro trabajo en las relaciones $M_{BH} - M_{bulge}$ y $M_{BH} - \sigma$, las cuales relacionan la masa de un agujero negro con propiedades globales de su galaxia huésped. Tratamos de explicar el origen de ambas relaciones a través de un análisis del Medio Interestelar.

Modelamos galaxias locales con tres componentes: gas, estrellas y materia oscura. Dejamos a los modelos evolucionar por 1 Gyr, y durante la evolución el gas puede formar estrellas, las cuales mueren en forma de supernovas. Al final de las corridas, nuestras simulaciones están caracterizadas por un medio altamente turbulento y compresible, con una Función de Densidad de Probabilidades que puede ser ajustada por una distribución lognormal a altas densidades. Su espectro de potencia de velocidad es bien ajustado por una ley de potencia de pendiente $\sim$ -4 en el espacio k a escalas pequeñas, lo cual es más pronunciado que la turbulencia de Kolmogorov y la de Burger. Este espectro de potencia nos permite deducir una relación del tipo $v_r \sim v_{rot} \left(\frac{\lambda}{R_d}\right)^{-\beta}$, la cual es el nexo necesario entre las propiedades globales de la galaxia y la alimentación del agujero negro masivo central.

Estudiamos la eficiencia de formación estelar en nuestras galaxias, donde investigamos cómo la pendiente de las leyes de Kennicutt-Schmidt y Silk pueden variar dependiendo de cómo definamos las cantidades involucradas en ambas leyes. Las dos leyes son fuertemente dependientes en el criterio ocupado para seleccionar el radio al cual se calculan las densidades superficiales, y el intervalo de tiempo ocupado para medir la tasa de formación estelar. En el primer caso la elección de un radio más grande puede llevar a obtener menores pendientes, mientras que en el segundo caso las pendientes más bajas son obtenidas usando un intervalo de tiempo más grande para promediar las tasas de formación estelar. Ambos efectos pueden cambiar la pendiente de las leyes de formación estelar en el rango entre 0.8 y 2.1.

Nuestras simulaciones también muestran una dependencia en los perfiles iniciales de M_{rot}. Esta relación es más pronunciada al comienzo de las simulaciones, donde la masa rotacional determina el tiempo al cual las galaxias comienzan a formar estrellas, y por lo tanto determina las tasas de formación estelar en etapas tempranas de la evolución.

We study galactic scale star formation and massive black hole growth through simulations using the Adaptive Mesh Refinement code Enzo. We focus our study on two of the most important star formation laws: the Kennicutt-Schmidt and the Silk law, both of them relate star formation rate with global properties of the galaxy. Although both laws have successfully quantified this relation, there is no consensus on the exact value of their slopes. We try to clarify this issue studying the factors that can influence both laws, such as gas surface density, orbital time and rotational mass. To study massive black hole growth, we will center our work in the $M_{BH} - M_{bulge}$ and $M_{BH} - \sigma$ relations, which relate the mass of a black hole with global properties of its host galaxy. We try to explain the origin of both relations from the analysis of the Interstellar Medium.

We model local galaxies with three constituents: gas, stars and dark matter. We let the models evolve for $\sim$1 Gyr, and during the evolution gas is allowed to form stars which also die as supernovae. At the end of the runs, our simulations are characterized by a highly turbulent and compressible medium, with a Probability Density Function that can be fitted by a lognormal distribution at high densities. Its velocity power spectrum is well-fitted by a power law with slope of $\sim$ -4 in k space at small scales, which is steeper than Kolmogorov and Burger turbulence. This power spectrum allows us to deduce a relation of the type $v_r \sim v_{rot} \left(\frac{\lambda}{R_d}\right)^{-\beta}$, which is the necessary link between the global properties of the galaxy and the fueling of central massive black holes.

We study the star formation efficiency in our simulations. where we investigate how the slope of the Kennicutt-Schmidt and the Silk law can vary depending on the way we define the quantities involved in both laws. The two laws are strongly dependent on the criteria used to select the radius at which calculate the surface densities, and the time interval used to measure star formation rates. In the first case the selection of a larger radius can lead to obtain lower slopes, while in the second case the lower slopes are obtained using a longer interval of time to average star formation rates. Both effects can change the slope of the star formation laws in the range 0.8 - 2.1.

Our simulations also show a dependency on the initial M_{rot} profiles. This relation is most pronounced at the beginning of the simulations, where the rotational mass determines the time at which galaxies start to form stars, and hence determine star formation rates at early stages of evolution.

A mis padres

Agradecimientos

Quiero comenzar agradeciendo a mi familia. A mis padres, Fernando y Nilda, y a mis hermanos, Alex y Jaime, que siempre han estado ahí para apoyarme en todo, en los momentos buenos y en los malos. Sin ellos yo no sería nada.

También agradecer a esas personas que desinteresadamente me han ofrecido su cariño y amistad, en especial a María José, quien me ha acompañado (y soportado) este último tiempo. No puedo dejar de mencionar a Felipe Antimán, César Casanova, Felipe Mancilla, Héctor Pérez, Sebastián Román, Kenneth Reese, Javier Sánchez y Héctor Sotomayor, quienes desde tiempos pueriles me han brindado su amistad. Tampoco pueden quedar afuera mis amigas Natalia Cornejo, Lisset Manzano y Natalia Zúñiga, quienes hicieron mucho más ameno mi paso por aquél lúgubre lugar ubicado en Beauchef 850.

Imposible dejar de nombrar a aquellas personas de quienes he aprendido todo lo que sé. A los profesores de Calán, en especial a mi profesor guía Andrés Escala quien me ha enseñado y ayudado en la elaboración de esta tesis. También a Mario Hamuy y Álvaro Núñez quienes me ayudaron en mis postulaciones al doctorado. Yéndome a épocas más tempranas, también quisiera recordar a mis profesores del Instituto Nacional que me enseñaron a enfrentar la vida. Una mención especial se merece Don Luis Elmes, quien con su humanidad nos enseñó a nunca olvidar nuestros sueños.

Mención aparte para la gente de Calán. A todos los alumnos, en especial a mis compañeros de generación y mis compañeros de oficina. A Nicola, Fernando, Jesús y, por supuesto, a Claudia. No puedo dejar de agradecer tampoco a todos los funcionarios, quienes hacen que el departamento pueda vivir. A Don Ricardo, Natalie, Lucho Lalo y Alejandro.

A CONICYT, al Departamento de Astronomía a través del proyecto Basal y al proyecto FONDECYT de iniciación #11090216 por financiar mi Magíster. Al Departamento de Astronomía de la PUC por facilitarme el cluster Geryon, y al Departamento de Ingeniería Matemática de la Universidad de Chile quien a través del Centro de Modelo Matemático también me facilitó el cluster levque. También dar las gracias a SOCHIAS quien financió mi viaje a una escuela de simulaciones numéricas en 2010.

Por último, también agradecer a todos aquellos que no han sido mencionados explícitamente aquí, pero que de una u otra forma han aportado un grano de arena a mi vida.

A todos ellos, mis más sinceros agradecimientos.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Star Formation

Stars are crucial for our Universe. In their interiors is where every heavy element is formed, and therefore where life takes root. To understand the Universe we have to start with the stars. Although they are small relatively to other structures such as galaxies, their influence in the Universe is much more greater than their size. Processes that happen in stars can change their environment on galactic scales (e.g. supernovae). Even though stars have been widely studied, we ignore much of them. In order to model the Universe as a whole, one of the first requirements is to understand the formation and behavior of stars. This requirement is much more important when simulating the formation and evolution of the Universe (e.g. Springel and Hernquist 2003), where we still can not reach the needed resolution to fully develop the formation of stars. Instead we use a star formation recipe, which is basically a predetermined way of creating stars. This recipe is usually based on what we call a star formation law, that relates star formation with global properties of the host galaxy. In the following subsections we will give an outline of the more important theories explaining how gas converts into stars at galactic scales.

1.1.1 The Kennicutt-Schmidt law

One of the first pioneering works studying star formation and developing a star formation law were proposed by Schmidt (1959, 1963). He suggested that the relation between star formation rate volume density (ρ_{SFR}) and gas volume density (ρ_{gas}) was a power-law of index n, with its value ranging from 1 to 2.

$$\rho_{SFR} \propto \rho_{gas}^{n} \tag{1.1}$$

These volume densities are not easily observable, instead we commonly measure surface densities from observations. In order to test Schmidt law with observations, one of the options is to assume a constant gas disk scale height as Kennicutt (1989, 1998) did. Using that assumption, the expression can be rewritten using surface densities instead of volume densities, originating the more common expression (where we have explicitly written n and N to differentiate both indexes)

$$\Sigma_{SFR} \propto \Sigma_{gas}^{N} \tag{1.2}$$

In particular, Kennicutt (1998) measured the index N using Hα, CO and HI observations of disk galaxies and infrared observations of starburst galaxies. To complete the range of gas surface density between disk and starbursts galaxies, he also used the central part of disks as nuclear disks. Since the amount of H_2 can not be directly measured, Kennicutt measured the CO intensity and convert it to H_2 assuming a CO-H_2 conversion factor, which in the case of Kennicutt (1998) is assumed constant for the

whole sample of galaxies. Its value was adopted to be $2.8 \times 10^{28}\mathrm{cm}^{-2}(\mathrm{K\ km\ s}^{-1})^{-1}$ following Bloemen et al. (1986), Kenney (1987). This assumption can be a source of errors in the law as we still do not know how this factor changes with other properties such as redshift or metallicity. The best fit for the data gives a value of $N \sim 1.4$, which is known as the Kennicutt-Schmidt (KS, also known as Schmidt-Kennicutt) relation. It is important to mention that in order to get this value, Kennicutt (1998) only considered the gas enclosed in the region where stars form, not including the rest of the disk. As he took average over entire galaxies for gas surface densities and star formation rates, this relation is also known as *global* KS law.

The physical interpretation proposed in the same study to explain the exponent is that star formation rate is proportional to the product of gas surface density and the inverse of the free-fall time. But at the same time, the free-fall time is related to gas surface density by a power-law $t_{ff} \propto \Sigma_{gas}^{-0.5}$, so that

$$\Sigma_{SFR} \propto \frac{\Sigma_{gas}}{t_{ff}} \propto \Sigma_{gas}^{1.5} \tag{1.3}$$

which results in agreement with the measurements of Kennicutt (1998). This expression tells us that gas is mostly consumed to form stars with a characteristic timescale of a free-fall time, which means that gas collapses into clumps due to gravitational forces.

An alternative version of Equation 1.2 was also studied by Kennicutt (1998) in which star formation rate is correlated with gas surface density and the dynamical time (t_{dyn}) of the system. Analogously to the interpretation of Equation 1.3, the characteristic time scale for the formation of stars is taken to be proportional to a dynamical time. This parameter is commonly taken as the time needed for the gas to do an orbit at the galactic radius, which is also known as the orbital time t_{orb}. This relation was first proposed by Elmegreen (1997), Silk (1997) reason why from now on it will be called Silk law. The analytical expression of the law is given by the following form:

$$\Sigma_{SFR} \propto \frac{\Sigma_{gas}}{t_{dyn}} = \frac{\Sigma_{gas}}{t_{orb}} \tag{1.4}$$

Utilizing exactly the same galaxies previously used to find the exponent in relation 1.2, Kennicutt (1998) found the best fit of slope $N \sim 1$ to the data. Both relations are shown in Figure 1.1. Even though we have treated t_{orb} and t_{dyn} as different quantities conceptually, we have to mention that both time scales are similar in value (Binney and Tremaine 2008)

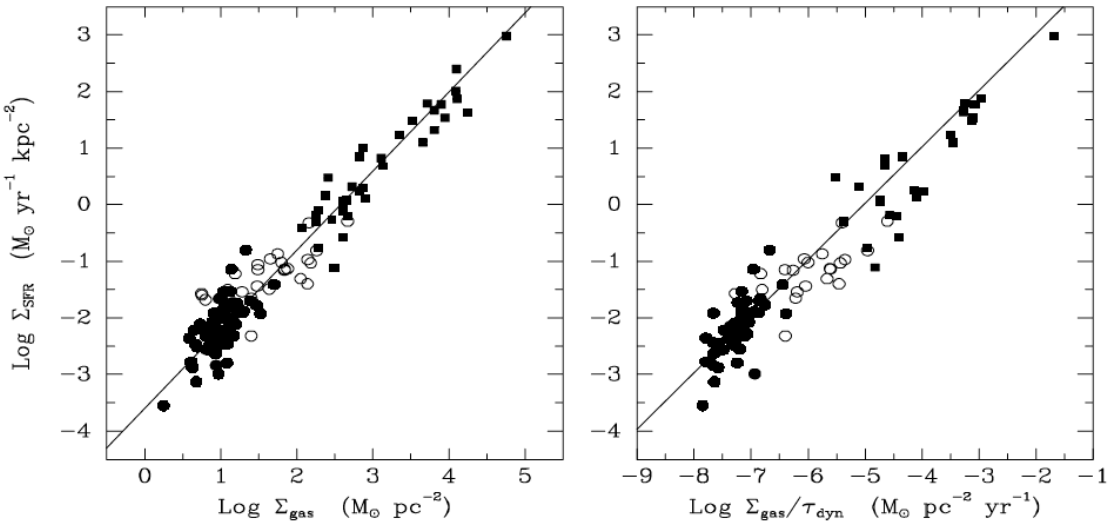


Figure 1.1: Schmidt and Silk from Kennicutt (1998). *Left:* Schmidt law. *Right:* Silk law

Since the seminal study of Kennicutt, numerous studies have been done studying these relations, both observations-based (Martin and Kennicutt 2001, Wong and Blitz 2002, Bouché et al. 2007, Kennicutt et al. 2007, Leroy et al. 2008, Bigiel et al. 2008) and simulations-based (Kravtsov 2003, Li et al. 2005, Tasker and Bryan 2006, 2008, Robertson and Kravtsov 2008, Gnedin and Kravtsov 2010). In general most studies do find that galaxies follow a KS law, but there is no consensus on the exact value of the index N. Almost every value measured is within the range 1-2 originally proposed by Schmidt (1959, 1963), so the focus of present works is to find the factor that produces these variations. In that sense phenomena such as metallicity or a dependence on redshift appears as good candidates to influence the slope of the relation.

1.1.2 Paradigms

The Kennicutt-Schmidt law has been very successful explaining the relation between stars and gas at galactic scales, but recent studies contradict one of Kennicutt's findings. Newer studies have shown that star formation is better correlated with molecular gas rather than atomic or total gas (Kennicutt et al. 2007, Bigiel et al. 2008), on the contrary of what Kennicutt (1998) found. A physical explanation is that it is expected that as stars forms in molecular clouds their formation should have a tighter relation with molecular gas. Recently developed instruments and observations have allowed to carry out new studies with sub-kpc resolution to improve the global measurements used by Kennicutt.

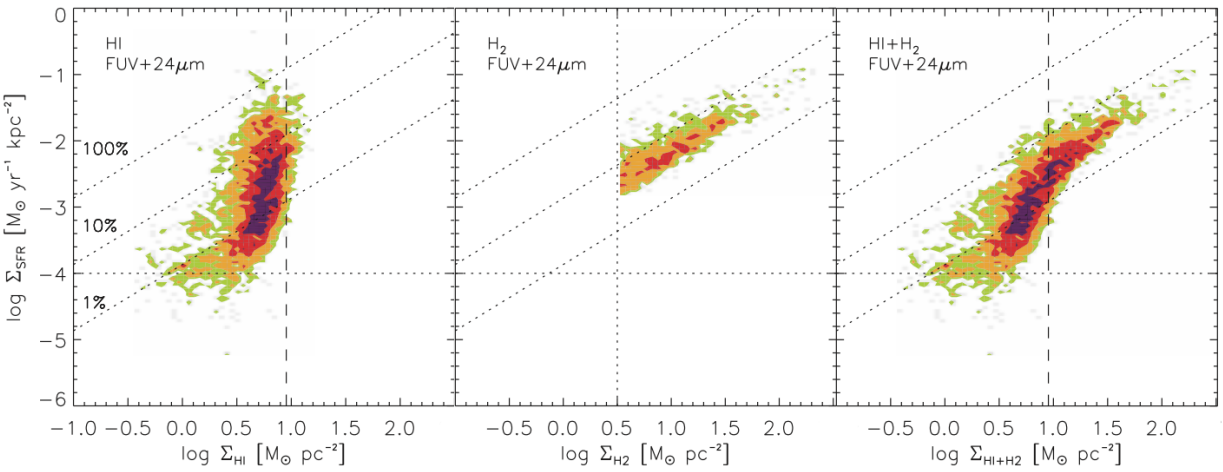
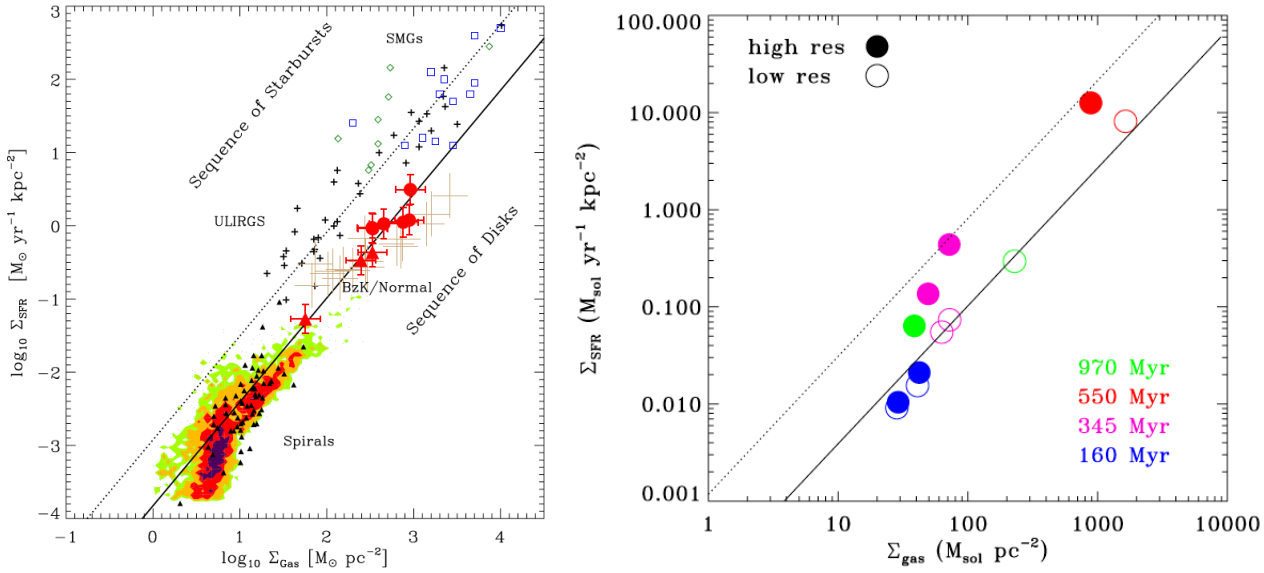


Figure 1.2: Comparison of atomic, molecular and total gas Kennicutt-Schmidt law

Figure 1.2 shows a comparison between the correlation of star formation rate with atomic, molecular and total gas as reported by Bigiel et al. (2008). They found a tighter relation between star formation rate and molecular gas (Figure 1.2 center), with less scatter and a slope ~ 1.0, in contrast to the atomic KS law (Figure 1.2 left) which shows a steeper relation with more scatter. Including both relations they obtain the dependence on total gas (Figure 1.2 right) which has a slope in between the atomic and molecular inclinations. It is worthwhile to note that at $\log \Sigma_{HI+H_2} = 1.0$ there is a break in the relation which translates in a change of slope. This break is originated by a transition between a region dominated by atomic gas to a region dominated by molecular gas, with star formation happening most in the latter region.

More recently, observations of starburst at high redshift have postulated another paradigm. Daddi et al. (2010), Genzel et al. (2010) found that spirals disks at low redshift and starburst galaxies at high redshift follow a Kennicutt-Schmidt law with the same slope but different zero points, in what has been called a "two sequence" law. These results have also been supported through simulations of the Antennae galaxies (Teyssier et al. (2010), Figure 1.3b). One tentative explanation for these discoveries is based on the change of the CO-H_2 factor ($X(\mathrm{CO})$) for galaxies at high redshifts. Unfortunately there is a restriction for this theory: it is only valid in the case of a discontinuous change. On the contrary, if the change is continuous the slope of the KS relation changes from 1.4-1.5 to 1.7-1.9 (Narayanan et al. 2012). Opposed to this explanation, new measurement of the interacting system Arp 158 have shown to follow the double sequence without assuming a different $X(\mathrm{CO})$ conversion factor (Boquien et al. 2011), what would put in doubt the change in the conversion factor as the origin of the two sequence.



(a) Observations from Daddi et al. (2010) (b) Simulations from Teyssier et al. (2010)

Figure 1.3: Two sequence relations

What is most intriguing is that although the two sequences found by Daddi et al. (2010) in KS law, they plotted the same data to study Silk law (Figure 1.4) finding that they follow one unique relation with slope $N \sim 1$ in completely agreement with Kennicutt (1998). This yields to the question of which of the two star formation laws is the intrinsic relation and which one is a by product of the other one. Ultimately, an intrinsic star formation law is what is being sought to include it as a recipe to fully model the formation and evolution of galaxies in the Universe.

1.1.3 Other Star Formation laws

Many other Star Formation laws have been proposed trying to explain the paradigms presented before. A complete summary of the most important ones is presented in Table 1.1. We encourage the reader to go to the given references to look for any specification and detail. Here we will focus on one of them: SFR-M_{rot} law (Escala 2011).

Based on stability criteria, from the dispersion relation for small perturbations (Binney and Tremaine 2008) $\omega^2 = \kappa^2 - 2\pi G\Sigma_{gas}|k| + k^2c_s^2$, Escala and Larson (2008) deduced a characteristic length $\lambda_{rot} =$

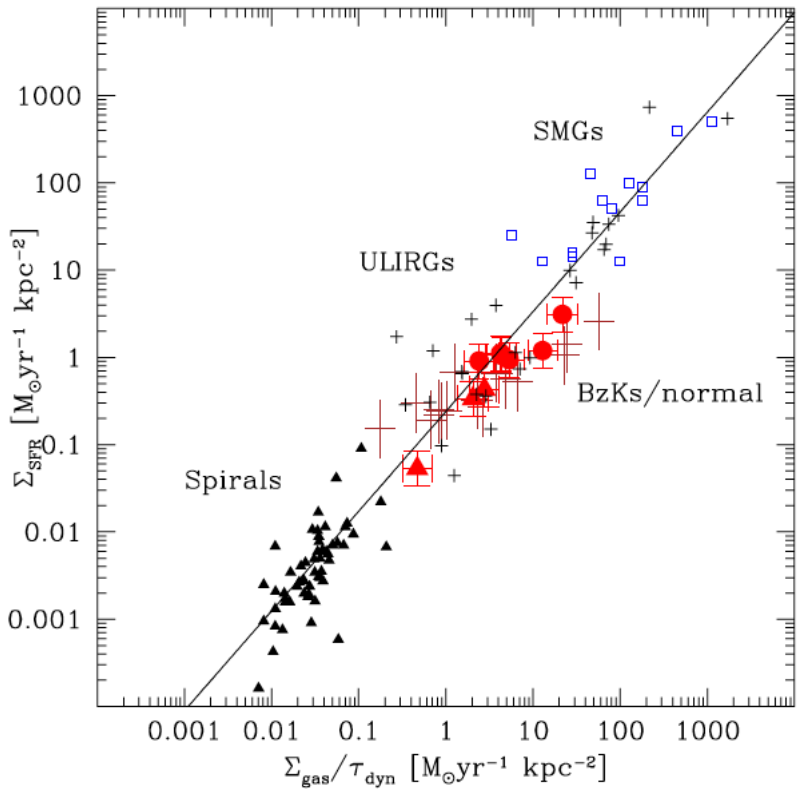


Figure 1.4: Silk law from observations of Daddi et al. (2010)

$4\pi^2 G\Sigma_{gas}/\kappa^2$ beyond which the disk is supported on large scales by rotation. This length has a characteristic mass associated given by the expression (Escala and Larson 2008, Escala 2011)

$$M_{rot} = \frac{4\pi^4 G^2 \Sigma_{gas}^3}{\kappa^4} = 3 \times 10^7 M_\odot \frac{M_{gas}}{10^9 M_\odot} \left(\frac{\eta}{0.2}\right)^2, \tag{1.5}$$

where κ is the epicyclic frequency and $\eta = M_{gas}/M_{dyn}$ is gas fraction.

Later on using observations of disk galaxies and starburst galaxies, Escala (2011) found a relation between star formation rate surface density and M_{rot}. As Equation 1.5 indicates, in order to do that it also is necessary to calculate the gas fraction in galaxies, which is rarely measured in observations. With the available data the best fit curve found by Escala (2011) was

$$\Sigma_{SFR} \propto M_{rot}^{2.3}. \tag{1.6}$$

1.2 Black Hole Growth

It is well-established the existence of a correlation between a central black hole and the global properties of its host galaxy, such as the M_{BH}-M_{bulge} (Marconi and Hunt 2003, Häring and Rix 2004) and the $M_{BH} - \sigma$ relation (Gebhardt et al. 2000, Ferrarese and Merritt 2000). Although they were derived separately, they are not completely independent. Using the Faber-Jackson relation ($R_e \propto \sigma^2 - \sigma^3$, Faber and Jackson (1976)) one of them is derivable from the other one. This also implies that only one of them is a fundamental relation, while the other is a by-product of the first one. In any case, these relations supports a scenario of coevolution of the black hole and its host galaxy.

Theory	Form	References
Disk free-fall time		
fixed scale height	$\Sigma_{SFR} \propto \Sigma_{gas}^{1.5}$	Kennicutt (1998)
variable scale heigh	$\Sigma_{SFR} \propto \frac{\Sigma_{gas}^2}{\sigma_g}\left(1+\frac{\Sigma_\star}{\Sigma_{gas}}\frac{\sigma_g}{\sigma_{\star,z}}\right)^{0.5}$	Elmegreen (1989), Krumholz and McKee (2005)
Orbital timescale	$\Sigma_{SFR} \propto \Sigma_{gas}\tau_{orb}^{-1} = \frac{v(r_{gal})}{2\pi r_{gal}]}$	Silk (1997), Elmegreen (1997)
Cloud-cloud collisions	$\Sigma_{SFR} \propto \Sigma_{gas}\tau_{orb}^{-1}Q_{gas}^{-1}(1-0.7\beta)$	Tan (2000)
Atomic and molecular gas	$\Sigma_{SFR} = f_{H_2}(\Sigma_{gas}, c, Z')\frac{\Sigma_{gas}}{2.6\mathrm{Gyr}} \times \begin{cases} \left(\frac{\Sigma_{gas}}{85M_\odot \mathrm{pc}^{-2}}\right)^{-0.33} & \text{if } \frac{\Sigma_{gas}}{85M_\odot \mathrm{pc}^{-2}} < 1 \\ \left(\frac{\Sigma_{gas}}{85M_\odot \mathrm{pc}^{-2}}\right)^{0.33} & \text{if } \frac{\Sigma_{gas}}{85M_\odot \mathrm{pc}^{-2}} > 1 \end{cases}$	Krumholz et al. (2009)
Rotational mass	$\Sigma_{SFR} \propto M_{rot}^{2.3}$	Escala (2011)

Table 1.1: Star Formation Laws

Several theories have been proposed to explain these relations. Some of them rely upon black hole feedback (Robertson et al. 2006, Di Matteo et al. 2008, Somerville et al. 2008). Others assumes that a fraction of the gas disk is accreted into the center of the galaxy to fuel the black hole (Burkert and Silk 2001, Escala 2006, 2007, Li et al. 2007). In the latter case is required to solve first another question: how to remove or redistribute angular momentum and transport gas to the central region of the galaxy? Numerous studies have been focused on investigate the mechanisms that drive gas to the center, proposing explanations such as gravitational torques due to asymmetries in the disk (Chang et al. 2007), bars (Athanassoula 1992) and the so called "bars-within-bars" (Shlosman et al. 1989, 1990). Although the great variety of possible explanations, there is no an agreement of which one is the dominant one.

In this context, Escala (2006, 2007) suggested that the $M_{BH} - \sigma$ relation can be explained using a simple model of mass transport through a turbulence-dominated α-disk and assuming a relation of the form:

$$v_{turb} \sim v_{rot}\left(\frac{\lambda_{turb}}{R_d}\right)^{-\beta}. \tag{1.7}$$

He derived an expression for the mass of the black hole:

$$M_{BH} \sim \frac{2\sqrt{2}\xi\alpha}{Q}\left(\frac{\lambda_{turb}}{R_d}\right)^{-\beta}\frac{R_e\sigma^2}{G}, \tag{1.8}$$

where $\xi \approx 5.9$, α is from the α-disk model, and Q is the turbulent Toomre's Q parameter. This expression implies the standard $M_{BH} \propto \sigma^4$ (Gebhardt et al. 2000, Ferrarese and Merritt 2000, Tremaine et al. 2002) if the Faber-Jackson relation ($R_e \propto \sigma_e^2$) is assumed. Additionally, in Equation 1.8 we can identify the bulge mass as $M_{vir} = kR_e\sigma^2/G$ recovering at the same time the M_{BH}-M_{bulge} relation (Marconi and Hunt 2003, Häring and Rix 2004).

1.3 Objectives

Our aim is to study the factors that influence the Kennicutt-Schmidt and Silk laws. Among these factors we found the radius used to calculate surface densities, and the method used to calculate star formation rates (SFR). We will see how the slope changes depending on the values chosen. Additionally, we will investigate how M_{rot} affects these relations and the scattering it can introduce in them.

Our work will be carried out using numerical simulations of local galaxies. Our study will be centered in a total of nine simulations which will be performed in an Adaptive Mesh Refinement (AMR) code.

The initial parameters of these simulations will be chosen according to our aim.

We will also take advantage of the simulations to study the validity of Equation 1.7 in the models we will perform. In order to do that, we will examine the properties of the Interstellar Medium (ISM) at the end of our simulations, and we will verify that a relation of this kind is obtained.

1.4 Thesis structure

We will introduce the code used in §2, detailing the structure, the algorithm and the way it works. In §3 we will describe the characteristics of our models, emphasizing the physical processes involved in them and how we choose their configuration. The analysis of the runs will be presented and discussed in §4. Finally, we will present the conclusions of our work in §5.

Chapter 2

The code

The two most common approaches to do numerical simulations in astrophysics are Adaptive Mesh Refinement (AMR, Berger and Oliger (1984), Berger and Colella (1989)) and Smoothed Particle Hydrodynamics (SPH, Gingold and Monaghan (1977), Lucy (1977), for a review see Monaghan (1992), and Springel (2010) for a more recent one) The former is based in an Eulerian hydrodynamics method for solving equations, while the latter uses a Lagrangian one. Both of them have advantages and disadvantages. On one side, the advantages of SPH are: (1) resolution not limited to grid spacing Δx; and (2) solutions are translational and rotational invariant. On the other side, AMR also has advantages over SPH methods: (1) eliminates Poisson noise representing fluids as a continuum, not discrete particles; (2) low density cells are computed as accurately as high density cells at the same cost; (3) integral conservation laws are straightforward to implement for mass, momentum, energy and magnetic flux which are numerically conservative to machine roundoff; and (4) due to hydrodynamics solvers, shocks are captured in 1-2 cells with correct entropy generation and non-oscillatory shocks.

In this work we are focusing in how gas evolves to study star formation on galactic scales. This process also includes supernovae explosions as stellar feedback which generates steep shocks of gas. The best technique to resolve this kind of shocks is AMR. In particular we will use ENZO[1] code (Bryan and Norman 1997, Norman and Bryan 1999, O'Shea et al. 2004), which is one among other implementations of AMR.

2.1 Enzo

Enzo is an Eulerian numerical method based on the structured AMR algorithm by Berger and Colella (1989). It was first developed for 3D hydrodynamic cosmological simulations and it is written in a mixture of C++ and Fortran. The former is used for high-level functions and data structures, while the latter is used for computationally intensive low-level functions. It uses the Message-Passing Interface (MPI)[2] to implement communication between processors and Hierarchy Data Format 5 (HDF5)[3] to write data and restart files in a platform independent format.

Among the routines implemented in this code we can find gravity solver, hydrodynamic solvers, N-body solver and different routines to model physical processes such as species solver, heating, cooling, star formation and feedback, and radiative transfer. Figure 2.1 shows one example of the loop followed by Enzo to run a cosmological simulation. The code starts calculating the expansion factor for a fixed time. The next step is to solve the gravitational field in the simulation. In third place the hydrodynamical equations are evolved, which besides gas dynamics also includes solving species, calculating heating and cooling rates, forming stars and radiative transfer. After that, N-body solver is used to evolve particles,

[1] http://enzo-project.org
[2] http://www.mcs.anl.gov/mpi/
[3] http://www.hdfgroup.org/HDF5/

which can be dark matter, stars, black holes or sink particles. After the whole process, the loop enters a new timestep and it starts again.

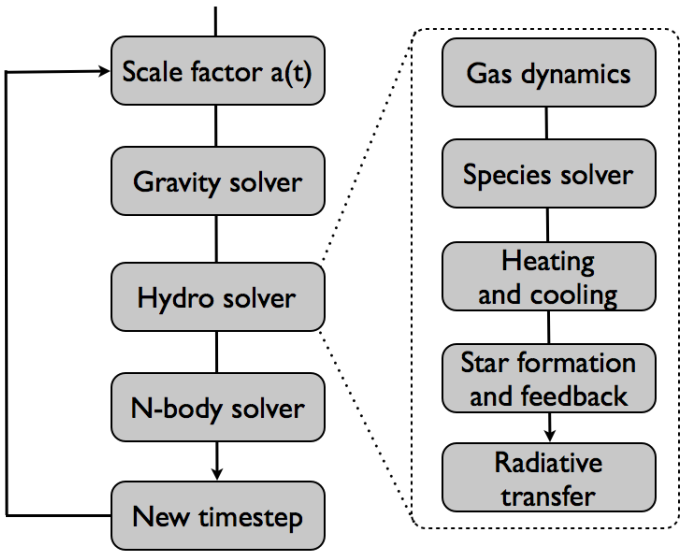


Figure 2.1: Main loop of Hydrodynamic Cosmology Code

2.2 Adaptive Mesh Refinement

The primary goal of the Structured Adaptive Mesh Refinement developed by Berger and Colella (1989) is to focus the computational effort in regions where it is most useful. Given that hydro solvers have already a great resolution by themselves, improve them will not increase accuracy significantly. So what this method really do is to select the regions where a better resolution is needed and apply the solvers in them. In this way what we are looking for is to minimize the cost for a fixed accuracy.

2.2.1 Grids

In order to discretized the hydrodynamical equations, the entire domain D is split in quadrilateral meshes which are logically connected between them. Each mesh belongs to a certain level which represent how refined the grid is. In the case of level 0, the sum of all grids at this level is the entire domain D, that also means that in level 0 grids has to align with each other.

$$D = G_0 = \bigcup_k G_{0,k} \tag{2.1}$$

If we generalize Equation 2.1 for any level l, then we get

$$G_l = \bigcup_k G_{l,k} \tag{2.2}$$

In general there can be overlapping grids at the same level, so that $G_{l,j} \cap G_{l,k} \neq 0$, for $j \neq k$, but discrete solution should not depend on how grids are decomposed. Grids on a level l are called sibling grids. One level above, on level $l-1$, the grids are parent grids while one level below are child grids. The greater the level, the finer the grid is.

Grids at different levels in the grid hierarchy have to fulfill two requirements to be “properly nested"

(i) a fine grid starts and ends at the corner of a parent grid

(ii) A level l grid has to be completely contained on at least one level $l-1$ cell.

Besides space, grids are also refined in time by the same mesh refinement ratio $R = \Delta x_{l-1}/\Delta x_l$:

$$\frac{\Delta t_l}{\Delta x_l} = \frac{\Delta t_{l-1}}{\Delta x_{l-1}} = \ldots = \frac{\Delta t_0}{\Delta x_0} \tag{2.3}$$

This means that in fine grids more timesteps are taken, increasing accuracy in both space and time. It does not mean that the finest timestep will be imposed globally, but each level has its own timestep in order to get the accuracy needed.

A point $(x, y) \in D$ may be contained in several grids. The solution $u(x, y)$ is taken from the finest grid containing that point, which is also the most accurate. In case there are more than one fine grids containing the point, the solution can be taken from any grid, since the solution on the intersection of overlapping fine grids will be identical.

Figure 2.2 shows an example of an AMR grid hierarchy for the case of $R = 2$, and 3 levels of refinement.

2.2.2 Integration algorithm

AMR assumes there is a basic, underlying, conservative, explicit finite difference scheme of the form:

$$u_{i,j}^{n+1} = u_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(F_{i+1/2,j} - F_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(G_{i,j+1/2} - G_{i,j-1/2}\right) \tag{2.4}$$

The values $u_{i,j}$ are cell-centered values of the grid with coordinates (i, j) and we have used the notation $u(t^n) = u^n$. This scheme is applied on every cell at every level without exceptions, but there are two cases in which this result should be modified

(i) the cell is overlaid by a finer level grid

(ii) the cell abuts a fine grid interface but is not itself covered by any fine grid.

In case (i) the grid value of the coarse grid at level $l-1$ is simply replaced by the average of the fine grid values at level l. Both grid values are calculated using Equation 2.4, but as the fine grid has a better resolution in time and space, the result of the scheme is more accurate. Then the coarse grid value is simply discarded. Taking as an example the grey square in Figure 2.3 and assuming a refinement radio r the expression of the replacement is

$$u_{i,j}^{coarse} \leftarrow r^{-\alpha} \sum_{k} \sum_{m} u_{k,m}^{fine}. \tag{2.5}$$

In case (ii) we also have to ensure the scheme to be conservative in the boundary between a fine grid and a coarse grid. That means that the fluxes into the fine grid across the boundary must equal the flux out of the coarse cell. Taking as an example cell (i, j) (yellow square) in Figure 2.3, we have to correct the flux in the following way

$$\begin{aligned} u_{i,j}(t + \Delta t_{coarse}) = u_{i,j}(t) &- \frac{\Delta t_{coarse}}{\Delta x}\left[F_{i+1/2,j}(t) - \frac{1}{r^{\alpha}} \sum_{q=0}^{r-1} \sum_{p=0}^{r-1} F_{k+1/2,m+p}(t + q\Delta t_{fine})\right] \\ &- \frac{\Delta t_{coarse}}{\Delta y}\left[G_{i,j+1/2}(t) - F_{i,j-1/2}(t)\right] \end{aligned} \tag{2.6}$$

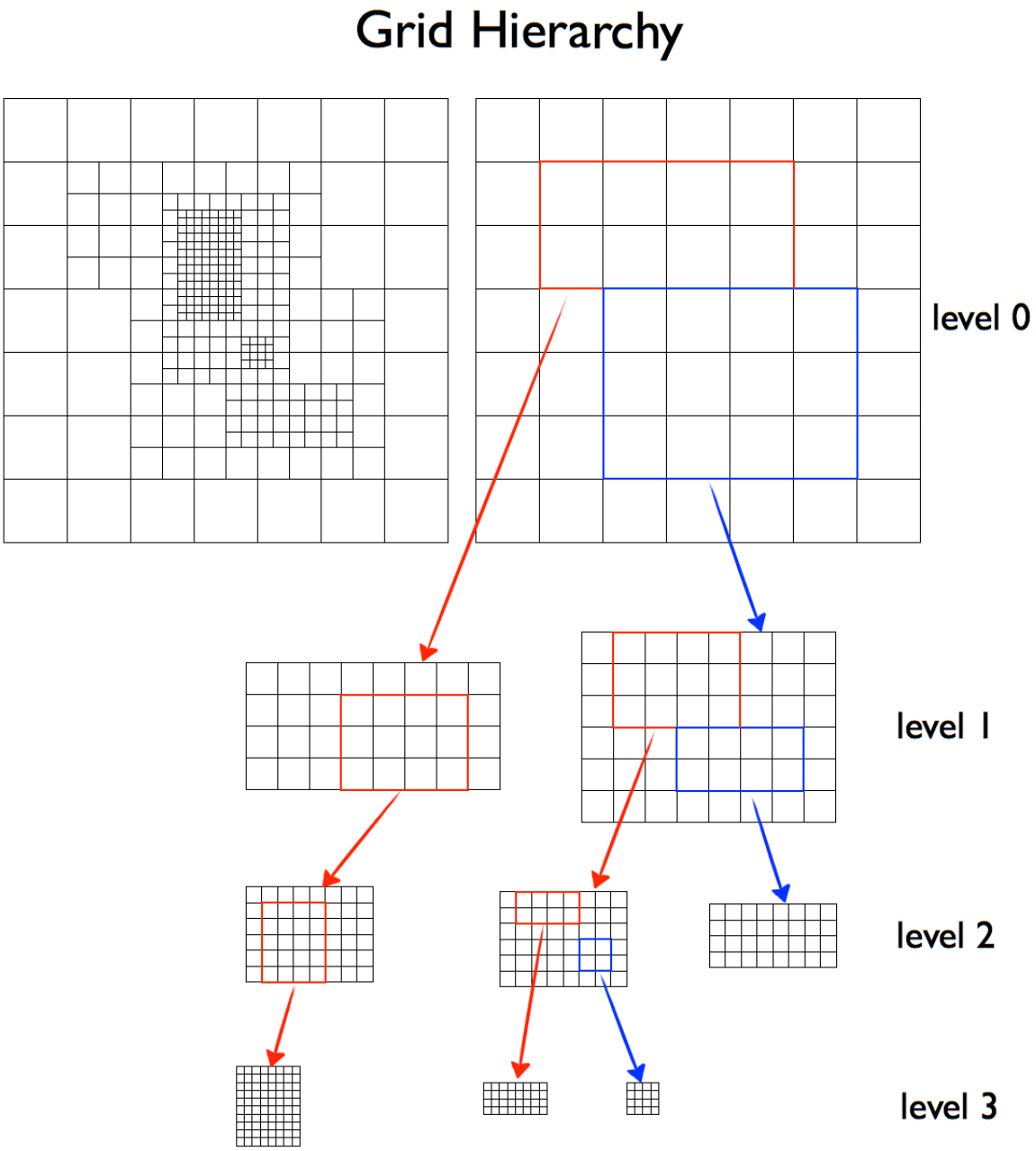


Figure 2.2: Illustration of an AMR grid hierarchy

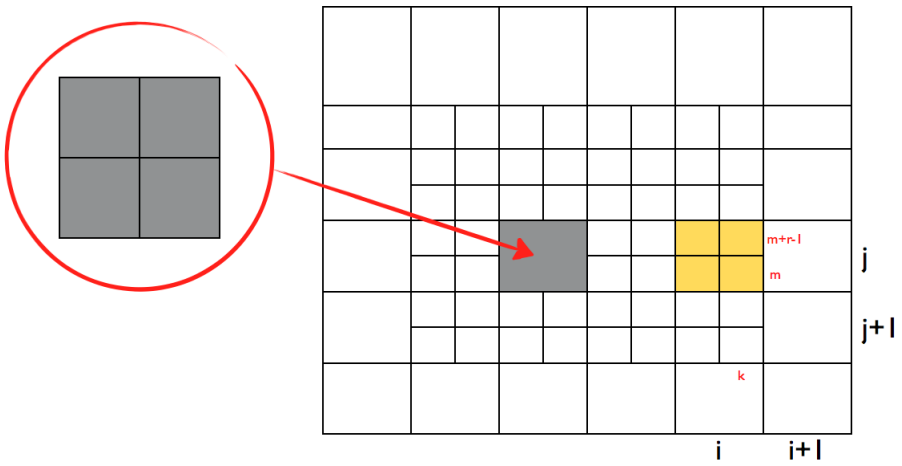


Figure 2.3: Finer grid and boundary corrections

2.2.3 Boundary conditions

To complete the discussion we have to refer to boundary conditions. The integration scheme showed in Equation 2.4 needs initial values to start working, these values are provided by the user which consists on information from outside the problem domain.

For a grid at level l boundary values are copied from adjacent cells at the same level l where they are available, otherwise the AMR algorithm uses bilinear interpolation from coarser level solution values to compute boundary values. If needed, the algorithm can also interpolate linearly in time. This can be summarized in three steps.

(i) find solutions values from parent grids at level $l-1$ on a slightly larger rectangular piece enclosing the border cells

(ii) interpolate linearly for the border values

(iii) if there are neighboring grid at level l that could supply some values, overwrite the linearly interpolated values from step (ii)

2.2.4 Creating grid hierarchy

Finally the grid hierarchy is built using an error estimation for each grid at each level. This allows to select which grids need to be refined in order to get smaller errors. Refining grids implies that the grid hierarchy has to be re-created from finest to coarsest grids. When a grid at level l is modified, then all finer grids are changed as well, but the coarser grids may remain the same.

In that process only one level can be created at a time, although many levels can be changed during the regriding operation. Let's see an example to explain it: suppose there is a base level l_{base} where grids will stay fixed and finer levels go from level $l_{base}+1$ to l_{finest}. The error estimation is first applied to the grid at level l_{finest} and if there are points where it is too high, then this cell is flagged for refinement and a different level $l_{finest}+1$ grid is created. Then we estimate the error on the grids at level $l_{finest}-1$ and apply the same procedure. In case there are flagged points then a new level l_{finest} grid is created, making sure that if there are any level $l_{finest}+1$ grid, they are properly contained in the level l_{finest}

grids. This method is applied until we reach level l_{base} and estimate error there.

2.3 Implementation

The AMR grid patches are the primary data structures in Enzo. Each one of this patches is an object which contains field variables and particles data. They are organized in a distributed hierarchy following two different methods: a tree data hierarchy and a level-based array of linked lists. An example of this distributed hierarchy in the case of two processors and six grids is illustrated in Figure 2.4

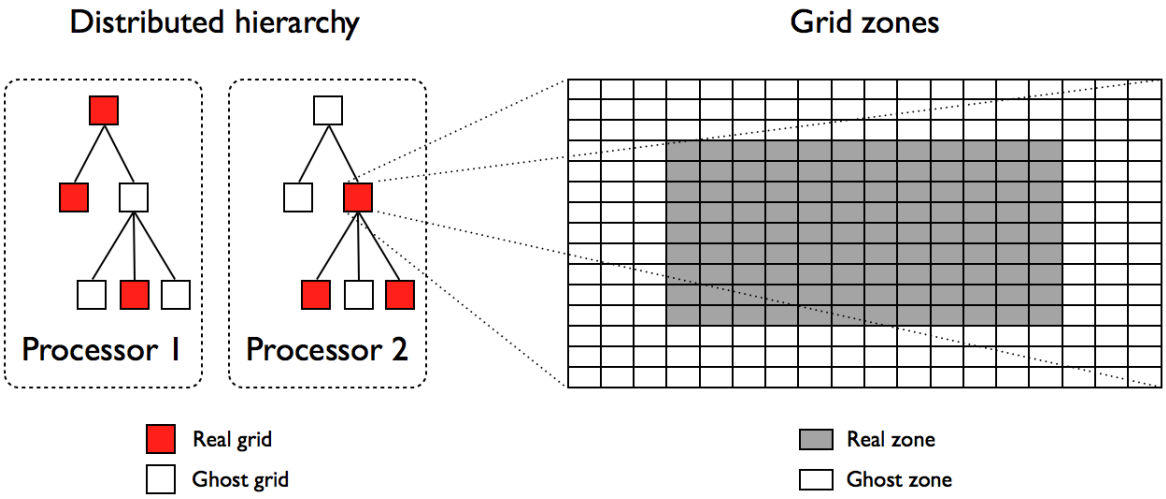


Figure 2.4: Example of distribute hierarchy using two processors and six grids

Each processor contains the whole distributed AMR hierarchy, but not all processor contain all grid data. Storing all grids in each processor would be significantly expensive in terms of memory, so it is much better to store some of them as real grids and the other ones as ghost grids. A grid is a real grid in a processor if its data is allocated on that processor, and a ghost grid if its data is not. A grid is a real grid in only one processor and a ghost grid in all others. Each data field within a grid is an array of zones in 1, 2 or 3 dimensions. There are two types of zones: real and ghost zones. Ghost zones store temporarily neighboring grid values to update real zones data fields when required. As seen in Figure 2.4 ghost zones are surrounding real zones and are 3 zones deep.

Communication between processors can only be done in grid of the same level because different levels are computed in a serial way, which means that the next level can only be evolved once the previous one has finished. Using ghost grids and linked lists, real grids can communicate and exchange information between processors to update their data fields.

2.3.1 Algorithm

The basic Enzo algorithm can be written in pseudocode as Figure 2.5 shows. The main procedure is the `EvolveLevel` routine, which receives the current level and time as arguments. Once this routine has

begun, the first step is to call the procedure `SetBoundaryValues` to calculate the boundary values for each grid. It can be done in two different ways: interpolating from a parent grid or obtaining the values from neighboring grids as seen in §2.2.3. After that, Enzo enters a while cycle in which there is a recursive call to `EvolveLevel` so that all refined levels are advanced as well.

```
EvolveLevel(level)
begin
     SetBoundaryValues
     while (Time < ParentTime)
     begin
          dt = ComputeTimeStep(level)
          PrepareDensityField(dt)
          SolveHydroEquations(dt)
          Time += dt
          SetBoundaryValues
          EvolveLevel(level+1, dt)
          FluxCorrection
          Projection
          RebuildHierarchy(level+1)
     end
end
```

Figure 2.5: Enzo AMR algorithm

Inside the while cycle the procedure `ComputeTimeStep` calculates the timestep of the current grid. As seen in Equation 2.3, in a linear system the timestep should be calculated using $\Delta t_{l+1} = \Delta t_l / R$ where Δt_{l+1} is the current timestep, Δt_l is the timestep of the coarse grid and R is the refinement factor. Usually systems are nonlinear, in this case timestep can be computed choosing the minimum between the following

a)

$$\Delta t_{hydro} = \min \left(\kappa_{hydro} \frac{a \Delta x}{c_s + |v_x|} \right)_L$$ where κ_{hydro} is the Courant Safety Factor

b)

$$\Delta t_{dm} = \min \left(\kappa_{dm} \frac{a \Delta x}{v_{dm,x}} \right)_L$$ where κ_{dm} is the Particle Courant Safety Factor

c)

$$\Delta t = f_{exp} \left(\frac{a}{\dot{a}} \right)$$ where f_{exp} is the Maximum Expansion Factor

d)

$$\Delta t_{accel} = \min \left(\sqrt{\frac{\Delta x}{\vec{g}}} \right)_L$$

Once the timestep is calculated, the procedure `PrepareDensityField` solves the Poisson equation and `SolveHydroEquations` evolves hydrodynamical equations. Following Figure 2.5 Enzo increases the current time by one timestep and calls `SetBoundaryValues` and `EvolveLevel` recursively. In that way, all finer grids are also evolved. The best resolution of the finer grids allows to compute more precise values which are used to correct values from coarse grids calling procedures `FluxCorrection` and `Projection`, methods described in §2.2.2. Finally, all grids in the level given as an argument are rebuilt using `RebuildHierarchy` to reflect the changing solution (§2.2.4). This procedure is in charge of flag

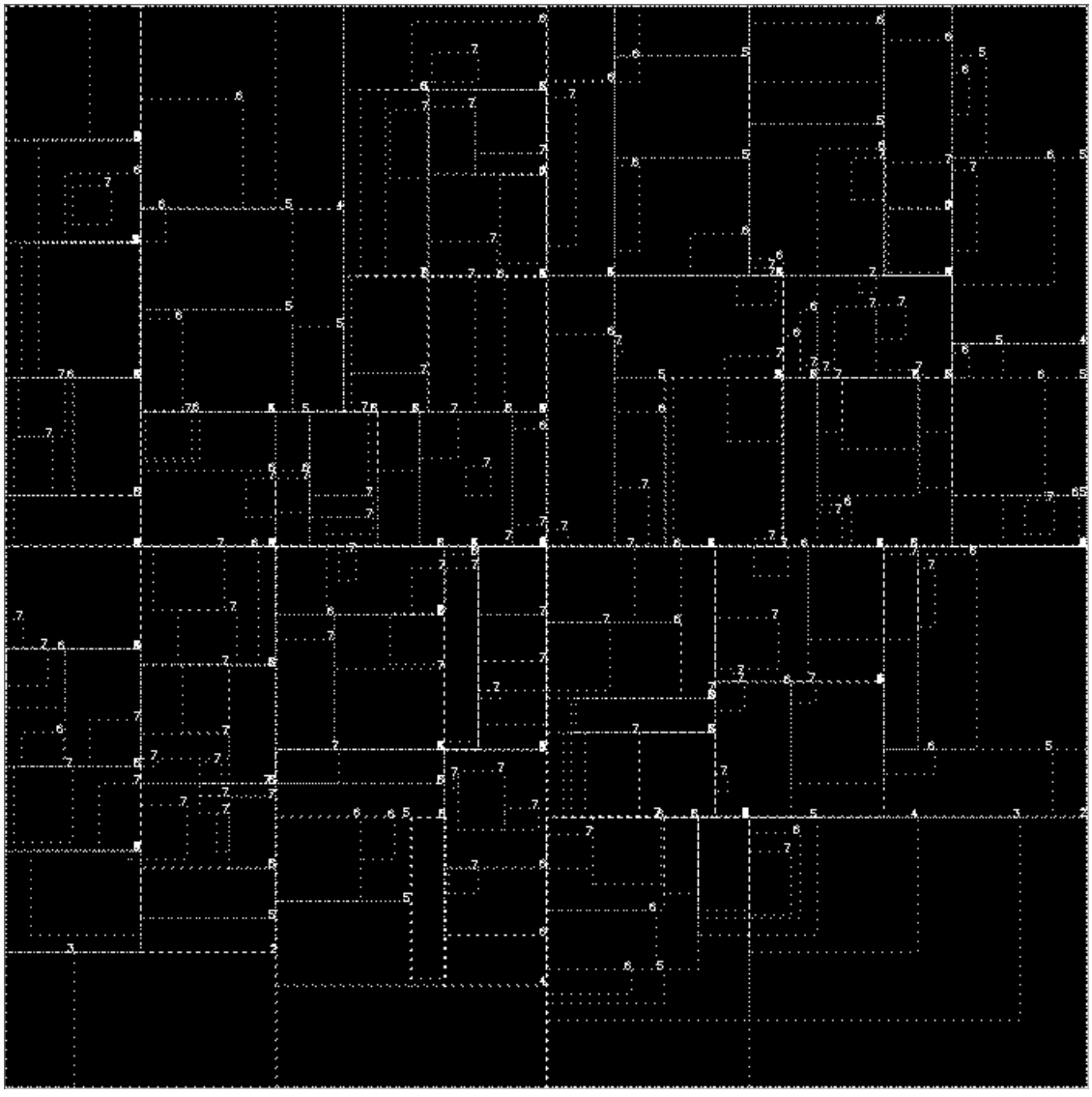


Figure 2.6: Grids on a simulation using baryon mass as refinement criteria

cells for refinement following criteria such as baryon density, baryon mass, particles mass, Jeans length and cooling time. Using these criteria the grids will not be uniform as showed in Figure 2.2. On the contrary, we will get a non linear hierarchy as shown in Figure 2.6 where there is an example of the grids obtained using baryon mass as refinement criteria and a maximum refinement level of 7. The small numbers indicate the level at which the cell belongs. As a last step, `RebuildHierarchy` also can create new grids interpolating values from their parent grids or copying values from old grids, which are deleted.

The algorithm described above applies to any level in the grid hierarchy. An important feature of this algorithm is the recursive call of it, which allows that all levels with finer subgrids are evolved as well. In that way grids are evolved from coarse to fine using their individually determined timesteps. This temporal integration scheme is known as "W-cycle" as Figure 2.7 shows. This order of integration allows us to achieve second order accuracy on time on the entire hierarchy.

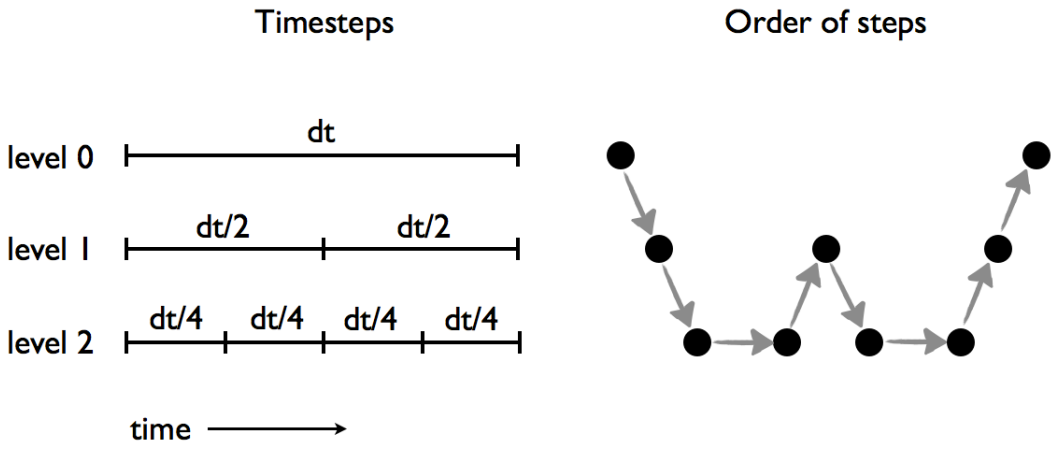


Figure 2.7: The “W" cycle

2.4 Gravity solver

2.4.1 Self-Gravity

ENZO also includes an algorithm to resolve the acceleration that the baryonic mass will have due to itself. This behavior is completely defined by Equation 2.7 (the Poisson equation), which says that any baryonic mass density distribution will generate a gravitational potential that will affect itself. So for this case we first compute the potential field ϕ from the baryonic density ρ.

$$\nabla^2\phi = 4\pi G\rho \tag{2.7}$$

In order to do that the mass density has to be completely defined everywhere. This is relatively easy for grids because they have an attribute which stores the density value, but for particles is a little more complicated. In case there is a particle, we have to distribute its mass among the eight nearest cells. For that, we advance the particle position by half step and we build an imaginary cube around it as Figure 2.8 shows. We then calculate the fraction of the cube that is superposed with each cell and we deposit that same fraction of the mass of the particle on the surrounding cells. This method is called Cloud-in-Cell.

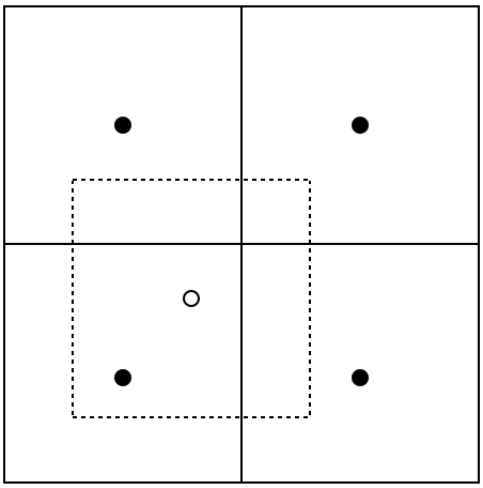


Figure 2.8: Cloud-in-Cell (CIC)

Root grid

To solve Poisson equation in root grid, Enzo applies a Fast Fourier Transform obtaining

$$\tilde{\phi}(k) = G(k)\tilde{\rho}(k) \tag{2.8}$$

and then it computes acceleration from the gravitational potential to update the velocities.

Subgrids

In subgrids Enzo does not solve Poisson equation directly, instead it first interpolates the gravitational potential from parent grids and assign this value to each subgrid. Enzo then solves Poisson equation in each of them using multigrid method. This method has a few disadvantages such as propagation of errors from coarse levels to fine levels and inconsistency of potential gradient across boundaries. Both of them can be improved, in the first case we can ameliorate it using six ghost zones in the case of gravity, and in the second case we can copy boundary conditions for potential from sibling grids and resolve Poisson equation.

2.4.2 External Gravity

In the case of external gravity the user can define the potential field directly in the initialization of the problem, that means that there is no need to solve Poisson equation and we can derived acceleration directly from potential. So in case we want to put a single point source of gravity, then we have to specify the acceleration field due to that source which will be simply

$$a(r) = \frac{GM}{r^3} \tag{2.9}$$

2.5 Hydrodynamical solver

One of the most important features in Enzo is the numerical solver for hydrodynamics equations. The complete set of hydrodynamic equations are

$$\frac{\partial \rho}{\partial t} + \vec{v} \cdot \nabla \rho = -\rho \nabla \cdot \vec{v} \tag{2.10}$$

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla)\vec{v} = -\frac{1}{\rho}\nabla p - \nabla \phi \tag{2.11}$$

$$\frac{\partial E}{\partial t} + \vec{v} \cdot \nabla E = -\frac{1}{\rho}\nabla \cdot (p\vec{v}) - \vec{v} \cdot \nabla \phi + \Gamma - \Lambda \tag{2.12}$$

$$E = e + \frac{1}{2}\vec{v}^2 \tag{2.13}$$

Here the dependent variables are mass volume density ρ, velocity $\vec{v}$ and the total energy density E. Equation 2.10 is known as continuity equation and expresses conservation of mass. Equations of motion are represented by Equation 2.11 which describe conservation of momenta. The remaining equation (Equation 2.12) is conservation of total energy, which is the sum of internal and kinetic energy as shown in Equation 2.13. Here we have added a source term Γ and a sink term Λ to account for energy input and output respectively, such as supernovae explosions or radiative cooling.

To close this set of equations another two are necessary. The first one relates pressure with mass density and internal energy density. This is also known as Equation of State (EOS) and commonly is assumed to be the ideal gas EOS:

$$e = \frac{p}{(\gamma - 1)\rho} \tag{2.14}$$

The second equation required is the self-gravity equation, also known as Poisson equation (Equation 2.15) which relates mass density with gravitation potential ϕ.

$$\nabla^2 \phi = 4\pi G(\rho_{total} - \rho_0) \tag{2.15}$$

Enzo has a wide variety of hydrodynamical solvers implemented: a direct-Eulerian version of the Piecewise-Parabolic Method (PPM, Colella and Woodward 1984); a cartesian, 3D version of ZEUS (Stone and Norman 1992); a Runge Kutta second-order based on Monotone Upstream-centered Schemes for Conservation Laws (MUSCL); and a similar solver to the previous one but including Magneto Hydrodynamics (MHD) from Dedner et al. (2002) (Wang and Abel 2009). For simplicity we will only discuss ZEUS which is the solver used in our simulations.

2.5.1 ZEUS

ZEUS is a three-dimensional Eulerian hydrocode that solves the fluid equations using the method of finite-differences with a time-explicit, multistep solution procedure. Among its characteristics are its robustness, its simplicity and its speed. The code is completely described in Stone and Norman (1992), but here we will summarize the most important points and briefly discuss how they are implemented in Enzo.

Figure 2.9 shows the flow chart for the code. Once the problem is completely initialized, the first step is to compute the gravitational potential from the mass density solving Poisson equation (Equation 2.15). After that, the code evolves the hydrodynamic equations entering the source and the transport step, which will be described in §2.5.2 and §2.5.3 respectively. Then the code is able to go to the next grid, compute its timestep and solve the set of equations in it. The final data outputs and graphics will be

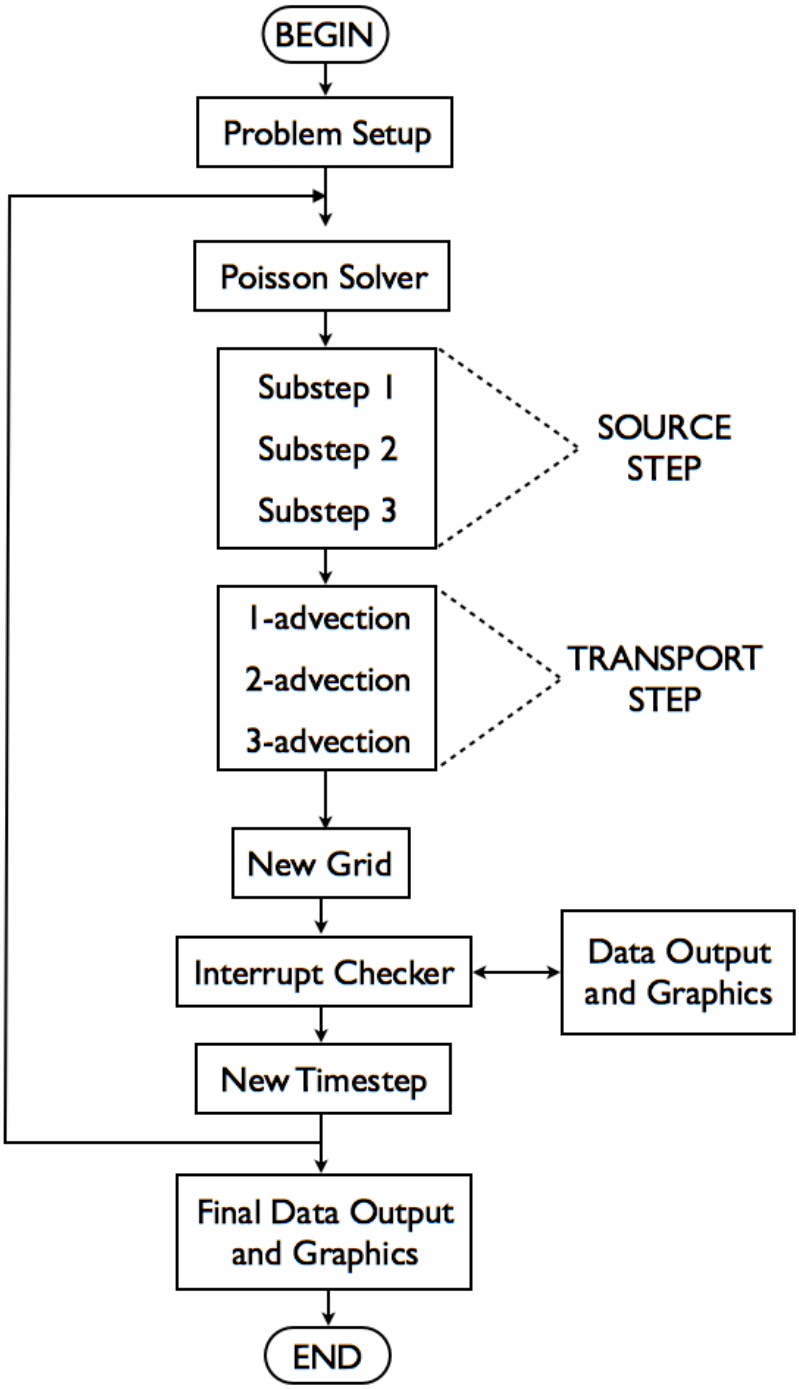


Figure 2.9: ZEUS flow chart

written once the code had finished all grids, but intermediate data outputs and graphics can be written anytime during the execution of the code.

2.5.2 ZEUS Source Step

This step consists basically in solving Equations 2.11 and 2.12 using finite-difference approximations. In order to do that the code has to compute the pressure p before it starts the Source step, so it obtains p from Equation 2.14. A second requirement is to compute gravitational potential ϕ, which ZEUS can get using Equation 2.15. With both quantities p and ϕ calculated, the Source step can now begin.

The Source step is subdivided in three substeps. In their description we will use the following notation: the value computed before Source step begins will have a superscript n. Then, the superscripts $n+a$, $n+b$ and $n+c$ will correspond to the values computed after the Substep 1, Substep 2 and Substep 3, respectively.

Substep 1

The main objective is to calculate the new velocity of the fluid, which has to be updated adding pressure gradients and gravitational forces. So first ZEUS does the correction by pressure

$$v_j^{n+a} = v_j^n - \frac{\Delta t}{\Delta x_j}\frac{p_j^n - p_{j-1}^n}{(\rho_j^n + \rho_{j-1}^n)/2} \tag{2.16}$$

and then it corrects by acceleration (gravitational force).

Substep 2

In this substep a new term q is included to account for viscous stresses and dissipation in the form of an artificial viscosity. Similar to what kinematic viscosity does in real fluids, this term smooth discontinuities which may appear in the flow, where the finite-difference equations break down (numerical errors). Including this term leads to a new correction in velocity and energy density given by

$$v_j^{n+b} = v_j^{n+a} - \frac{\Delta t}{\Delta x_j}\frac{q_j^{n+a} - q_{j-1}^{n+a}}{(\rho_j^n + \rho_{j-1}^n)/2} \tag{2.17}$$

$$e_j^{n+b} = e_j^n - \frac{\Delta t}{\Delta x_j}\frac{q_j^{n+a}(v_j - v_{j+1})}{\rho_j^n} \tag{2.18}$$

Here in Equations 2.17 and 2.18 the artificial viscosity can be chosen in two different ways. The first option is to assume a quadratic viscosity term as formulated by Vonneumann and Richtmyer (1950):

$$q_j^{quad} = \begin{cases} Q_{AV}\rho_j(v_{j+1} - v_j)^2 & \text{if } (v_{j+1} - v_j) < 0 \\ 0 & \text{otherwise} \end{cases}$$

or we can choose a linear viscosity coefficient as defined by Lapidus (1967), Norman and Winkler (1986):

$$q_j^{lin} = C_1\rho C_a\Delta v = C_1\rho\sqrt{\frac{\gamma p}{\rho}}(v_{j+1} - v_j)$$

where C_a is the adiabatic speed of sound.

Substep 3

Finally, a compressional heating term is also added to improve energy conservation. To do that we define $p^{n+1/2} = (p^n + p^{n+1})/2$ and write the finite-difference version of Equation 2.12 in the following way

$$e^{n+1} = e^n - \Delta t p^{n+1/2} \nabla \cdot \vec{v} \tag{2.19}$$

which can be rewritten using the ideal gas equation of state (Equation 2.14)

$$e_j^{n+c} = e_j^{n+b} \left(\frac{1 - \frac{\Delta t}{2}(\gamma - 1)(\nabla \cdot \vec{v})_j}{1 + \frac{\Delta t}{2}(\gamma - 1)(\nabla \cdot \vec{v})_j} \right) \tag{2.20}$$

2.5.3 ZEUS Transport Step

The following step is to solve equations of motion in their integral forms which is basically to compute fluxes in a conservative way. The equations can be obtained applying the divergence theorem to Equation 2.10 and are given by the following expressions:

$$\frac{d}{dt} \int_V \rho dV = \int_{dV} \rho \vec{v} \cdot d\vec{S} \tag{2.21}$$

$$\frac{d}{dt} \int_V \rho \vec{v} dV = \int_{dV} \rho \vec{v}^2 \cdot d\vec{S} \tag{2.22}$$

$$\frac{d}{dt} \int_V e dV = \int_{dV} e \vec{v} \cdot d\vec{S} \tag{2.23}$$

The Transport step is done in one direction at a time, that means that it starts with x-direction, continue with y-direction and finish with z-direction in the 3 dimensional case. So let's take for example Equation 2.21. If we try to solve it using finite-differences in the x-direction we obtain

$$\rho_j^{n+d} = \rho_j^n - \frac{\Delta t}{\Delta x}(v_{j+1/2}^{n+c} \rho_{j+1/2}^* - v_{j-1/2}^{n+c} \rho_{j-1/2}^*) \tag{2.24}$$

where the superscript $n + d$ indicates that it is the value after the Transport step, similar to the notation used in §2.5.2

To compute the new value of the density, we first need values for mass density and velocity in the boundaries of the grid. The problem is that while the velocities are face-centered, the mass density is grid-centered. One way of obtaining the value of mass density at the boundaries is via interpolation from its centered values. The interpolated values are represented by the asterisks in Equation 2.24. Here we use the second-order-accurate van Leer method (van Leer 1977), which basically consists in a piecewise linear interpolation of the zone-centered value w given by:

$$w_j^* = \begin{cases} w_{j-1} + (\Delta x_{j-1} - (v_j - vg_j)\Delta t)\,(dw_{j-1}/2) & \text{if } v_j - vg_j > 0 \\ w_j - (\Delta x_j + (v_j - vg_j)\Delta t)\,(dw_j/2) & \text{if } v_j - vg_j < 0 \end{cases}$$

where dw_i are the monotonized van Leer slopes computed from the harmonic average:

$$dw_j = \begin{cases} \frac{2(\Delta w_{j-1/2} \Delta w_{j+1/2})}{\Delta w_{j-1/2} + \Delta w_{j+1/2}} & \text{if } \Delta w_{j+1/2} \Delta w_{j-1/2} > 0 \\ 0 & \text{otherwise} \end{cases}$$

2.6 Particles

As seen in §2.4 particles can contribute to the density field and influence gravitationally the surrounding material. Besides this characteristic, particles are also evolved following N-body dynamics, that means that each particle also has position, velocity and acceleration. The calculation of the acceleration of the particle is just the opposite to the process of distribution of mass described in §2.4. This time, the acceleration is interpolated from the values of the eight nearest cells as illustrated in Figure 2.8.

With the value of the acceleration we are able to update its position and velocity using the so called "drift-kick-drift" leapfrog algorithm.

$$x^{n+1/2} = x^n + \frac{\Delta t}{2} v^n \tag{2.25a}$$

$$v^{n+1} = v^n + \Delta t a^{n+1/2} \tag{2.25b}$$

$$x^{n+1} = x^{n+1/2} + \frac{\Delta t}{2} v^{n+1} \tag{2.25c}$$

Here we first evolve the particle position by half-step maintaining constant the velocity, which is called a "drift". Then we advanced particle velocity one step keeping position constant (a "kick") and finally we again advance position by half-step with a constant velocity (another "drift"). This order of integration allows us to reach second order accuracy. Finally these particles attributes are stored in the locally most-refined grid. As a side note, it is no necessary to store previous timesteps which reduces significantly the memory used.

Chapter 3

The simulations

The galaxies are simulated in a box of $666h^{-1}$ kpc with periodic boundary conditions. The size of the parent grid is 128^3 and we proceed down to additional 7 subgrids of refinement. This level of refinement allows us to reach a resolution of $\sim$40pc which is a reasonable resolution to resolve star formation and feedback (Ceverino and Klypin 2009). The criteria used for refinement is of two kinds and both of them have to be fulfill: refinement by baryon mass if the density of the cell is four times the average density, and refinement by Jeans length to ensure that it is at least resolved by 4 cells to prevent artificial fragmentation (Truelove et al. 1997).

We also add radiative gas cooling through a file where we specified the values of cooling rates for each temperature following the curves of Sarazin and White (1987) down to $T = 10^4$K and Rosen and Bregman (1995) down to $T = 300$K. As pointed out by Ceverino and Klypin (2009) this range of temperatures will let us resolve the Interstellar Medium (ISM) to observe a multi-phase medium.

Finally, the simulations are evolved in comoving coordinates in a ΛCDM universe, where we have adopted the values $\Omega_m = 0.3$, $\Omega_\Lambda = 0.7$, and $H_0 = 67$ km s^{-1} Mpc^{-1}.

3.1 Galaxy

We model our galaxies as a three-component system which includes gas, stars and dark matter, similar to Wang et al. (2010). The treatment for each component is different. In the case of gas we model it using grids, while in the case of stars and dark matter they are modeled as external potentials which are fixed in time. That means that we do not add star particles or dark matter particles at the beginning of the simulations, but that does not exclude us to allow creation of particles once the simulations are evolved.

3.1.1 Gas

We initialize the gas using an exponential profile in the radial direction in cylindrical coordinates combined with a sech^2 profile in the vertical direction:

$$\rho_{gas}(R, z) = \rho_0 \exp(-R/R_0)\mathrm{sech}^2\left(\frac{z}{2z_0}\right), \tag{3.1}$$

where R_0 is the disk scale-length, z_0 is the disk scale-height and ρ_0 is the central volume density which can be expressed as a function of z_0, ρ_0 and the total gas mass M_{gas}

$$\rho_0 = \frac{M_{gas}}{8\pi z_0 R_0^2} \tag{3.2}$$

Integrating Equation 3.1 in the vertical direction from $-z_0$ to $+z_0$ gives the gas surface density of the disk:

$$\Sigma_{gas}(R) = 4\rho_0 z_0 \exp(-R/R_0) = \Sigma_0 \exp(-R/R_0). \tag{3.3}$$

For the whole set of simulations we will use $R_0 = 35$ kpc and $z_0 = 0.4$ kpc.

3.1.2 Dark Matter

The dark matter component will be modeled as an external time-independent gravitational field which will be fixed through the evolution of the galaxies. As explained in §2.4.2, this approach only affects gas dynamics changing its velocity due to the acceleration field produced by the density profile. It allows us to focus the study in the evolution of gas. Here we will use the popular Navarro-Frenk-White profile (NFW, Navarro et al. 1997) for dark matter density

$$\rho_{DM}(r) = \frac{\rho_{crit}\delta_c}{(r/r_s)(1+r/r_s)^2}, \tag{3.4}$$

where $r_s = r_{200}/c$ is a characteristic radius, $\rho_{crit} = 3H^2/8\pi G$ is the critical density (H is the current value of Hubble's constant), c is the concentration parameter, and δ_c is given by:

$$\delta_c = \frac{200}{3}\frac{c^3}{[\ln(1+c) - c/(1+c)]}. \tag{3.5}$$

Integrating Equation 3.4 in volume we obtain the dark matter mass profile:

$$M_{DM}(r) = \frac{M_{200}}{f(c)}\left[\ln(1+x) - \frac{x}{1+x}\right], \tag{3.6}$$

where $x = rc/r_{200}$ and $f(c)$ is defined in the following way:

$$f(c) = \ln(1+c) - \frac{c}{1+c} \tag{3.7}$$

We will adopt a value $c = 12$ for the concentration parameter and $M_{200} = 10^{12} M_\odot$ for the mass enclosed by the virial radius.

3.1.3 Stars

Finally in the case of the star component we use a Miyamoto-Nagai (Miyamoto and Nagai 1975) profile, where the potential and the density are given by (Miyamoto and Nagai 1975, Binney and Tremaine 2008):

$$\phi(R,z)_{stars} = \frac{-GM_{star}}{\sqrt{R^2 + \left(a + \sqrt{z^2+b^2}\right)^2}} \tag{3.8}$$

$$\rho_{stars} = \left(\frac{b^2 M_{star}}{4\pi}\right)\frac{aR^2 + (a+3\sqrt{z^2+b^2})(a+\sqrt{z^2+b^2})^2}{[R^2 + (a+\sqrt{z^2+b^2})^2]^{5/2}(z^2+b^2)^{3/2}} \tag{3.9}$$

The potential of Equation 3.8 is an intermediate case between the Plummer potential (Plummer 1911) which is recovered in the case when $a = 0$, and the Kuzmin potential (also known as Toomre's model 1, Kuzmin 1956, Toomre 1963) which is recovered in the case when $b = 0$. Its main characteristic is to model a disk and a bulge depending on the values chosen for a and b, which allows to reproduce the effect that stars have upon gas.

3.2 Star formation

Although we do not initialize the galaxy with star particles, we do let star particles be created through the simulation. As mentioned in §3.1 and §3.1.3 we will focus our study on the evolution of the gas and how it creates stars. This will only depend on how the stars influence gravitationally the surrounding gas and not on how many stars are at the beginning of the simulation, which is well modeled by the approach described previously.

In order to create a new star particle we follow Cen and Ostriker (1992) algorithm with a density restriction added. For a cell to enter into the Cen & Ostriker algorithm its density has to be greater than a density threshold:

$$\rho_{cell} > \rho_{thres} \tag{3.10}$$

If this requirement is satisfied then the cell has to fulfill three physical criteria: the gas has to be contracting, the time it takes to cool has to be less that the time it takes to collapse, and it has to be gravitationally unstable. These criteria are represented by Equations 3.11a, 3.11b and 3.11c respectively.

$$\vec{\nabla} \cdot \vec{v} < 0 \tag{3.11a}$$

$$t_{cool} < t_{dyn} \equiv \sqrt{\frac{3\pi}{32 G \rho_{tot}}} \tag{3.11b}$$

$$m_{cell} > m_{Jeans} \tag{3.11c}$$

Once a cell has passed all the requirements, a new star is created and its mass is calculated as a function of the efficiency, the gas density and the cell-width

$$m_\star = \epsilon \rho_{gas} \Delta x^3 \tag{3.12}$$

It is useful to notice that there is a substantial difference between Equation 3.12 and the one proposed by Cen and Ostriker (1992). Here we do not consider the efficiency per dynamical time, which means that there is no delay in star formation. It implies that the cell does not wait a time t_{dyn} to turn gas into stars. We adopt this way to prevent that during the time it takes to create the star (a dynamical time) the density keeps growing, which will result in a star with a greater mass.

Additionally, we also consider a numerical restriction to ensure low cost of time and memory. A star particle will only be created if its mass given by Equation 3.12 is greater than a minimum mass set to $10^5 M_\odot$. In that way we prevent having big amounts of particles which will slow considerably the simulations performed.

3.2.1 Stellar feedback

We also add stellar feedback due to supernovae explosions which is treated as an injection of energy on the surrounding gas. Here we do feedback as if the star particles are created over a long period of time, so the particle actually loses mass over time in an exponentially decaying way. To calculate star mass over time, we use the integral form of Cen and Ostriker (1992):

$$m_{stars}(t) = m_\star \int_{t_{SF}}^{t} \frac{t - t_{SF}}{\tau^2} \exp\left[\frac{-(t - t_{SF})}{\tau}\right] dt \tag{3.13}$$

We then inject an energy equivalent to a supernova of 10^{51} ergs for every 55 $M_\odot$ of star formed. We do this increasing the momentum of the surrounding gas directly due to supernovae explosions. This kind of feedback has been postulated as one of the most important processes in self-regulation of star formation.

3.3 The models

We performed a total of nine simulations summarized in Table 3.1, which differ between each other by any of the parameters M_{gas}, M_{star}, M_{200}, a and/or b. We started with our fiducial simulation which we tagged as "F". The choice of the values for each parameter of our fiducial simulation is based on the Milky Way properties. We then subdivide the rest of the simulations in two groups: Group A and Group B. From now on, when we refer to "Group A" it will include the simulations labeled as Group A in the table plus our fiducial simulation F. The same will be applied when referring to "Group B".

	Name	M_{gas} $10^9[M_\odot]$	M_{stars} $10^{10}[M_\odot]$	M_{200} $10^{11}[M_\odot]$	a [kpc]	b [kpc]
Fiducial	F	10	4	10	3.5	0.2
Group A	AI	10	10	10	3.5	0.2
	AII	10	4	1	3.5	0.2
	AIII	10	1	10	3.5	0.2
Group B	BI	8	2.537	6	3.428	0.496
	BII	7	1.148	5	2.267	0.036
	BIII	6	1.795	3	3.056	2.974
	BIV	5	1.455	2	0.000	8.454
	BV	4	2.388	1	3.603	11.030

Table 3.1: Galaxy parameters

3.3.1 Group A

The main characteristic that defines this group is that M_{gas} and Σ_{gas} are the same in every simulation. What we are trying to do here is to start always with the same gas mass profile while varying dark matter and stars profiles. In order to do that, we fixed $M_{gas} = 1 \times 10^{10} M_\odot$ and we vary M_{star} and M_{200}. So for example we have three simulations with $M_{200} = 1 \times 10^{12} M_\odot$ with values of 1×10^{10}, 4×10^{10} and $10 \times 10^{10} M_\odot$ for M_{star} which will allow us to study the influence of the stellar profile in the star formation. Similarly we also have two simulations with the same initial star mass $M_{star} = 4 \times 10^{10} M_\odot$ but with different values for M_{200} ($1 \times 10^{11} M_\odot$ and $10 \times 10^{11} M_\odot$). It is also important to notice that we have not changed the parameters a and b of the Miyamoto-Nagai profile for stars (Equation 3.8) which means that the shape of the stars profile has not varied.

Figure 3.1 shows the gas surface density profile at the beginning of our fiducial simulation F, which at the same time is equal for each galaxy of the Group A. Varying M_{200} and M_{star} also implies that the dynamical properties will change among them, including dynamical time, rotation curves and rotational mass. The different M_{rot} profiles for this group at $t = 0$ are shown in Figure 3.2 where simulations F, AI, AII, AIII are represented by the blue, dark cyan, black, and red lines respectively.

If Σ_{gas} has the same profile for all Group A simulations at the beginning, the Kennicutt-Schmidt law tells us that we should expect very similar star formation rates for the whole group. We suppose that we will not obtain exactly the same SFR, but little variations between them which will be interpreted as scattering in the KS relation. We will study this dispersion as a result of changing secondary parameters such as the dynamical time or the rotational mass.

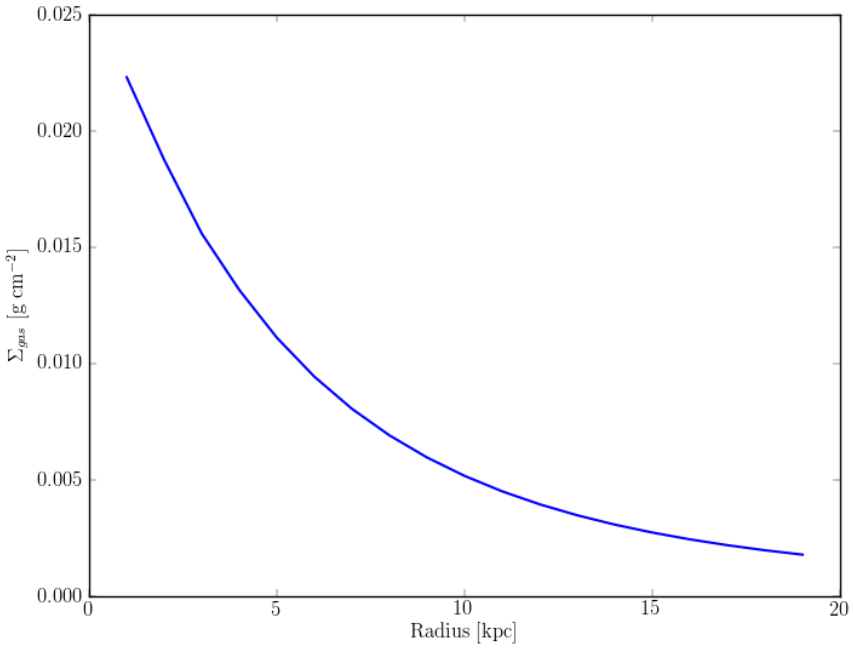


Figure 3.1: Initial gas surface density profile for Group A

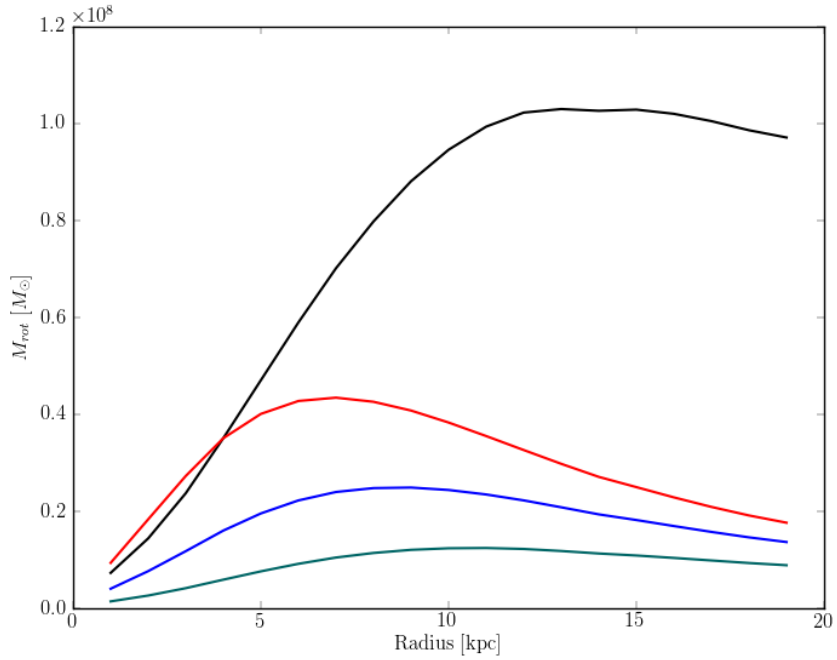


Figure 3.2: Initial M_{rot} profiles for Group A. Colors are as follow: blue for model F, dark cyan for model AI, black for model AII and red for model AIII

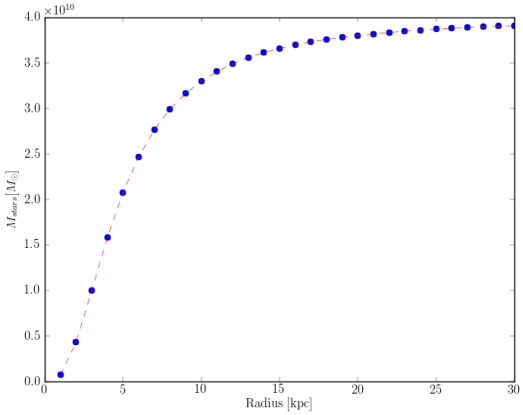
3.3.2 Group B

For Group B we try something similar to the previous group. In this case we will keep constant the initial rotational mass M_{rot} profile, while changing the dynamical time and the gas surface density. So in this case all parameters in Group B simulations are calculated in order to obtain a similar rotational mass profile. From Equation 1.5 we can see that we do not have a direct way of controlling the value of M_{rot}. So what we do here is to take our fiducial simulation F, calculate its rotational mass profile and take it as a reference for the other simulations. We then vary M_{gas} from $1 \times 10^9 M_\odot$ to $1 \times 10^{10} M_\odot$, and M_{200} from $1 \times 10^{11} M_\odot$ to $1 \times 10^{12} M_\odot$ in intervals of $10^9 M_\odot$ and $10^{11} M_\odot$ respectively. Then for each pair of values we found a "theoretical" profile for M_{star} in order to get a similar rotational mass profile to that of reference. We then fit a "real" star mass profile following Equations 3.8 and 3.9 to the theoretical ones, obtaining the parameters M_{star}, a and b. Figure 3.3 shows the theoretical profile expected as blue points and our best least-squares fit keeping a Miyamoto-Nagai profile for stars as a red dashed line.

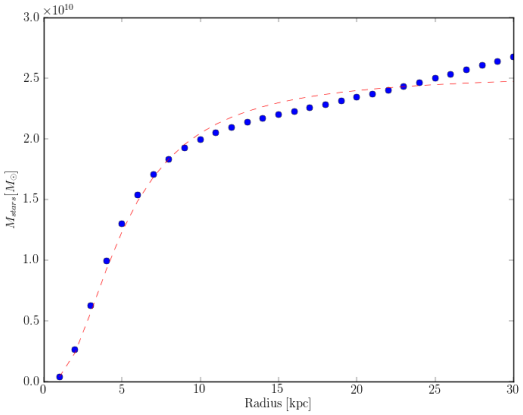
From our fits it is worthwhile to note a few things. First, of the total of star mass-dark matter mass pairs we obtained from our fit, we only selected five to plot. This is mainly because of two reasons: these simulations had the lowest chi-square values at the same time that they cover a wide range in the initial gas mass, from $M_{gas} = 4 \times 10^9 M_\odot$ to $M_{gas} = 9 \times 10^9 M_\odot$ (see Table 3.1). A second point is that in Figure 3.3a we recover the original rotational mass profile for our fiducial simulation F using our fit code, which indicates at least that our code is working well.

As we pointed out before, we do not expect to have exactly the same rotational mass profiles for the simulations in this group, on the contrary they will be different but we should minimize these differences. In Figure 3.4 we plot all the M_{rot} profiles for Group B. They agree quite well in general with differences no more than 20% for a given radius. We also show in Figure 3.5 the gas surface density profiles which will obviously be different because of the values chosen for the initial M_{gas}. In both figures the blue, yellow, cyan, magenta, green and orange lines represent simulations F, BI, BII, BIII, BIV and BV respectively.

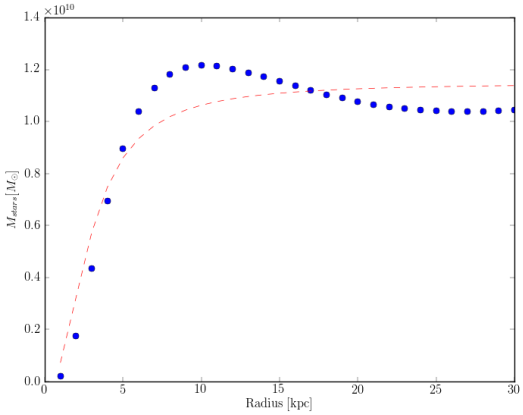
As a final statement we want to highlight that we will do a similar search for second parameters as the one proposed in §3.3.1, but this time centered on M_{rot} law. As we keep M_{rot} constant, Σ_{gas} and t_{dyn} will change. That means that in this opportunity gas surface density and the dynamical time will be the second parameters. Comparing both groups will lead us to the conclusion on the existence of a fundamental law and the factors that alter it.



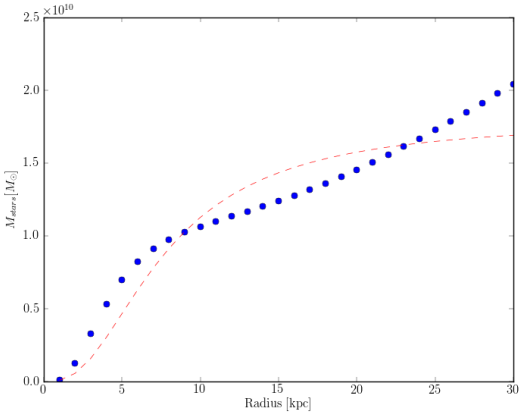
(a) Model F



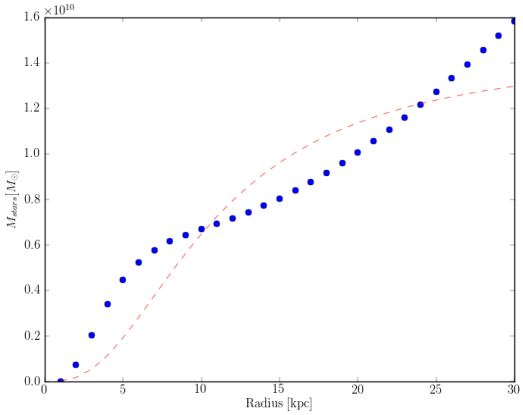
(b) Model BI



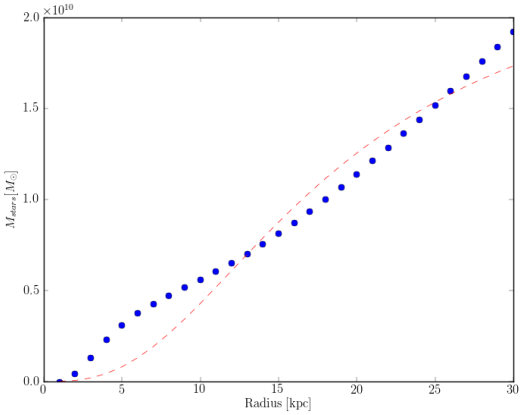
(c) Model BII



(d) Model BIII



(e) Model BIV



(f) Model BV

Figure 3.3: Least-square fit for star mass profiles. Blue dot is the profile needed, and red dashed line is our best fit.

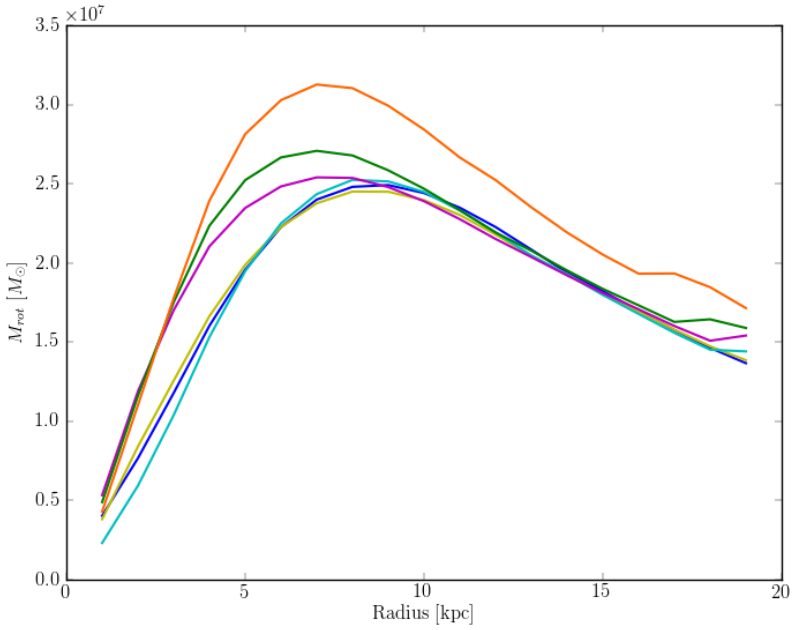


Figure 3.4: Initial M_{rot} profiles for Group B galaxies. Colors are as follow: blue is simulation F, yellow BI, cyan BII, magenta BIII, green BIV and orange BV

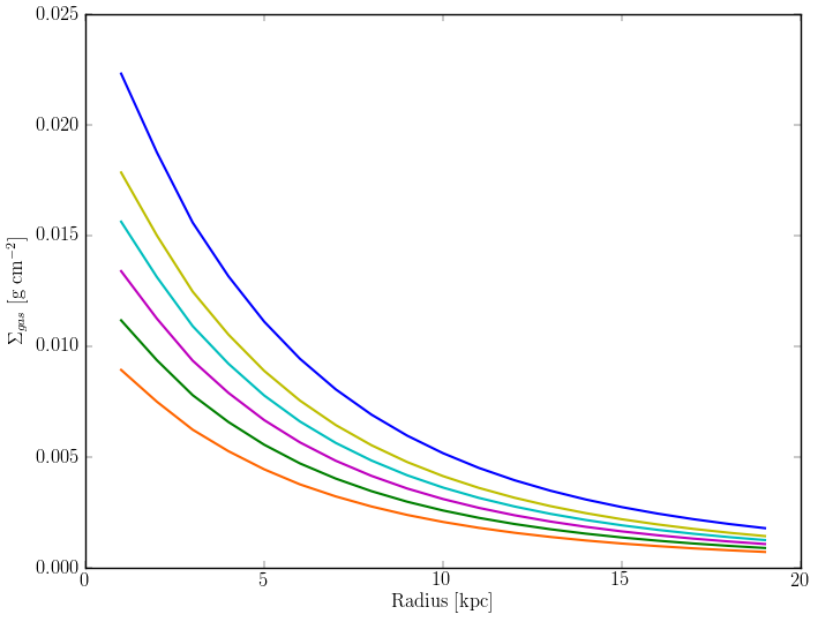


Figure 3.5: Initial gas surface density profiles for Group B. Colors are as in Figure 3.4

Chapter 4

Results and analysis

Before we discuss the results, we will show the evolution of each galaxy. We let them evolve for a total time of $\sim$ 1Gyr. In general all simulations follow the same evolution pattern. First they start fragmenting in the central part of the disk (because of the shorter t_{dyn}) to soon start fragmenting in the outer parts. This is expected because the Toomre Q parameter (Toomre 1964) is below unity in all our simulations as we will see later in §4.2.1. Another interesting feature is the effect of stellar feedback. The influence of supernovae can be better appreciated in Figure 4.1 which shows an edge-on view of simulation BIV at t=640 Myr, where the gas ejected forms gas filaments that propagates in the vertical direction of the disk.

From Figure 4.2 to 4.10 we present a series of images representing face-on snapshots of the simulations. From left to right the columns correspond to times t $\sim$100, 200, 400, and 800 Myr. The upper row shows density slices in a scale from 1×10^{-28} g cm^{-3}to 1×10^{-20} g cm^{-3}and the lower row shows the position of particles representing recently formed stars. Despite of the similar pattern, a few differences can be seen at times t=100 and 200 Myr where we can appreciate that the evolution of some galaxies such as AI is slower than the rest. For the same reason, as some galaxies start to fragment earlier, the stars are also created early. Seeing Figure 4.3 it is obvious that it has not yet created stars by time t=200 Myr, while others such as Figure 4.5 have started long time ago. The same effect from stellar feedback of Figure 4.1 can be easily seen in Figure 4.8 and 4.9 at t =400 Myr. From this point of view we see regions of low densities. These "dark bubbles" are supernovae which have exploded injecting energy to the surrounding material and provoking the ejection of the gas.

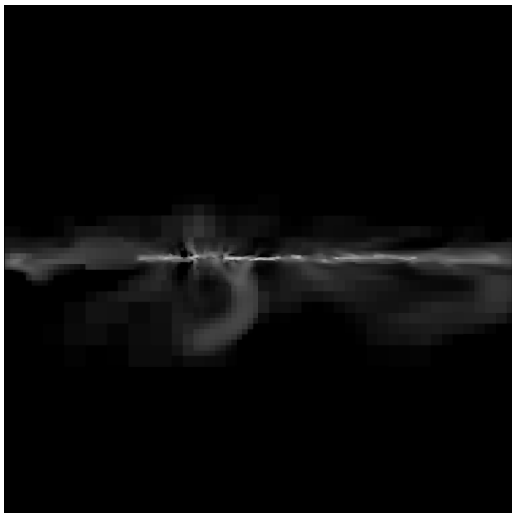


Figure 4.1: Edge-on view of simulation BIV at t=640 Myr

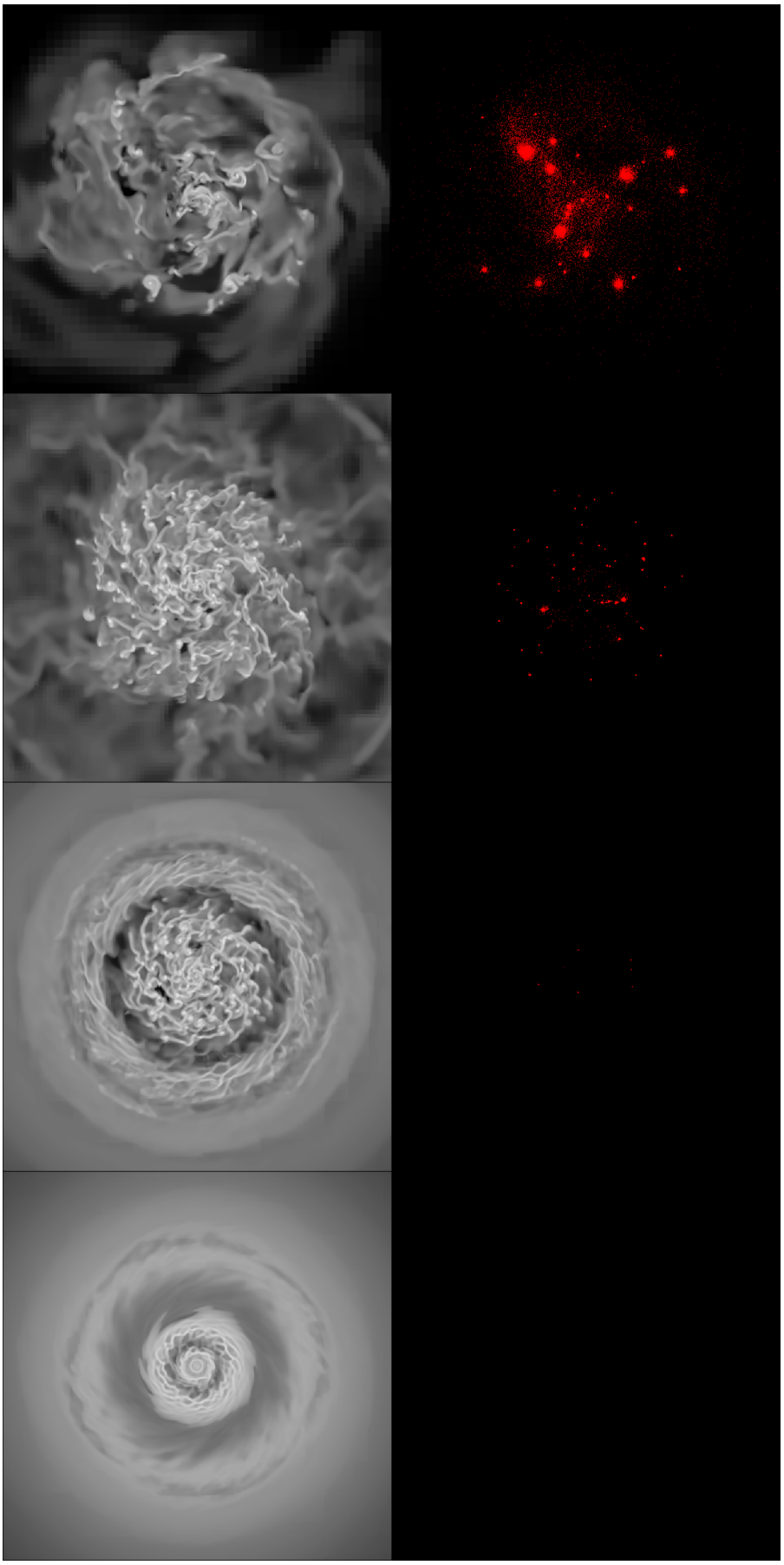


Figure 4.2: Evolution of model F. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

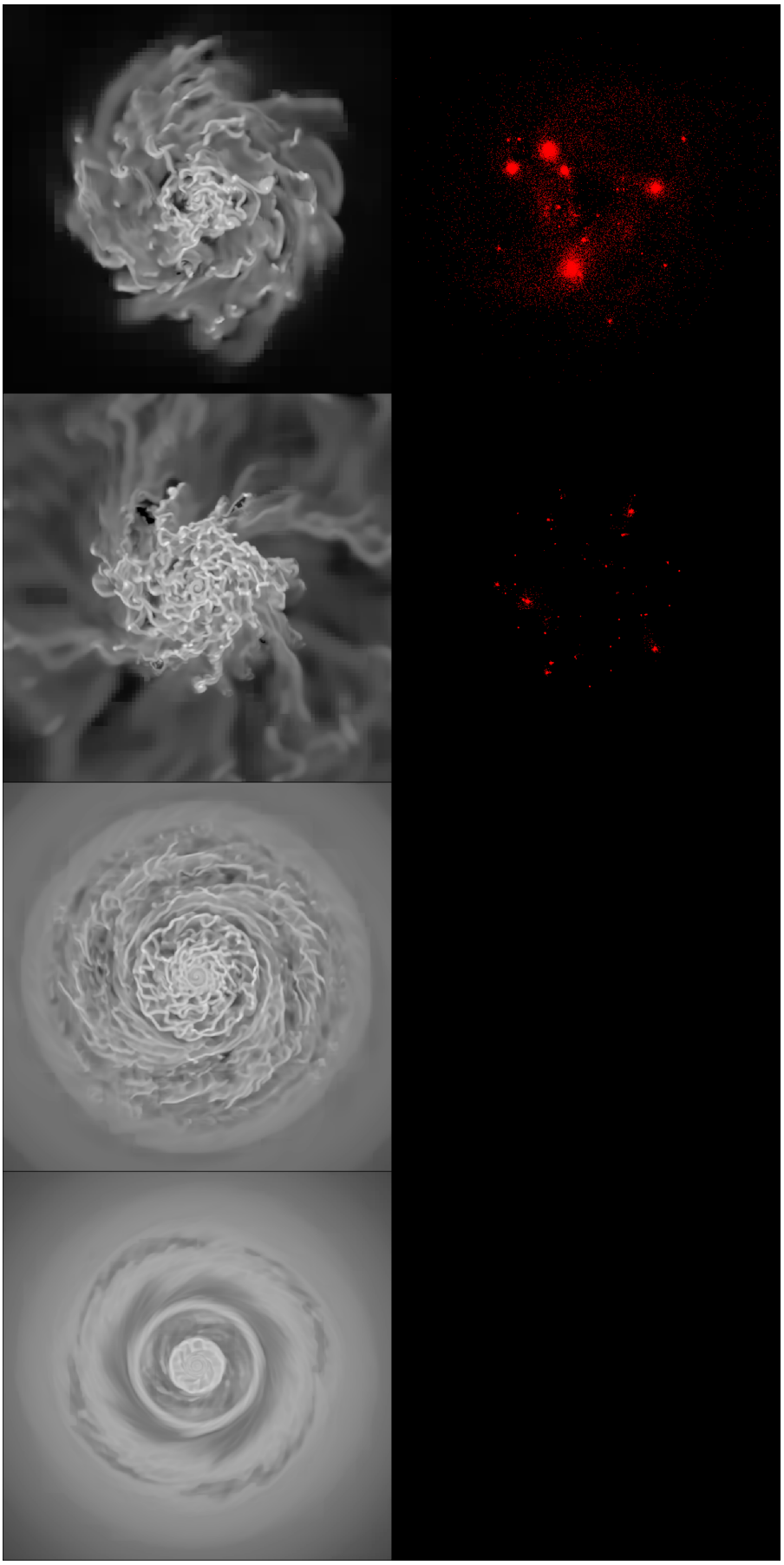


Figure 4.3: Evolution of model AI. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

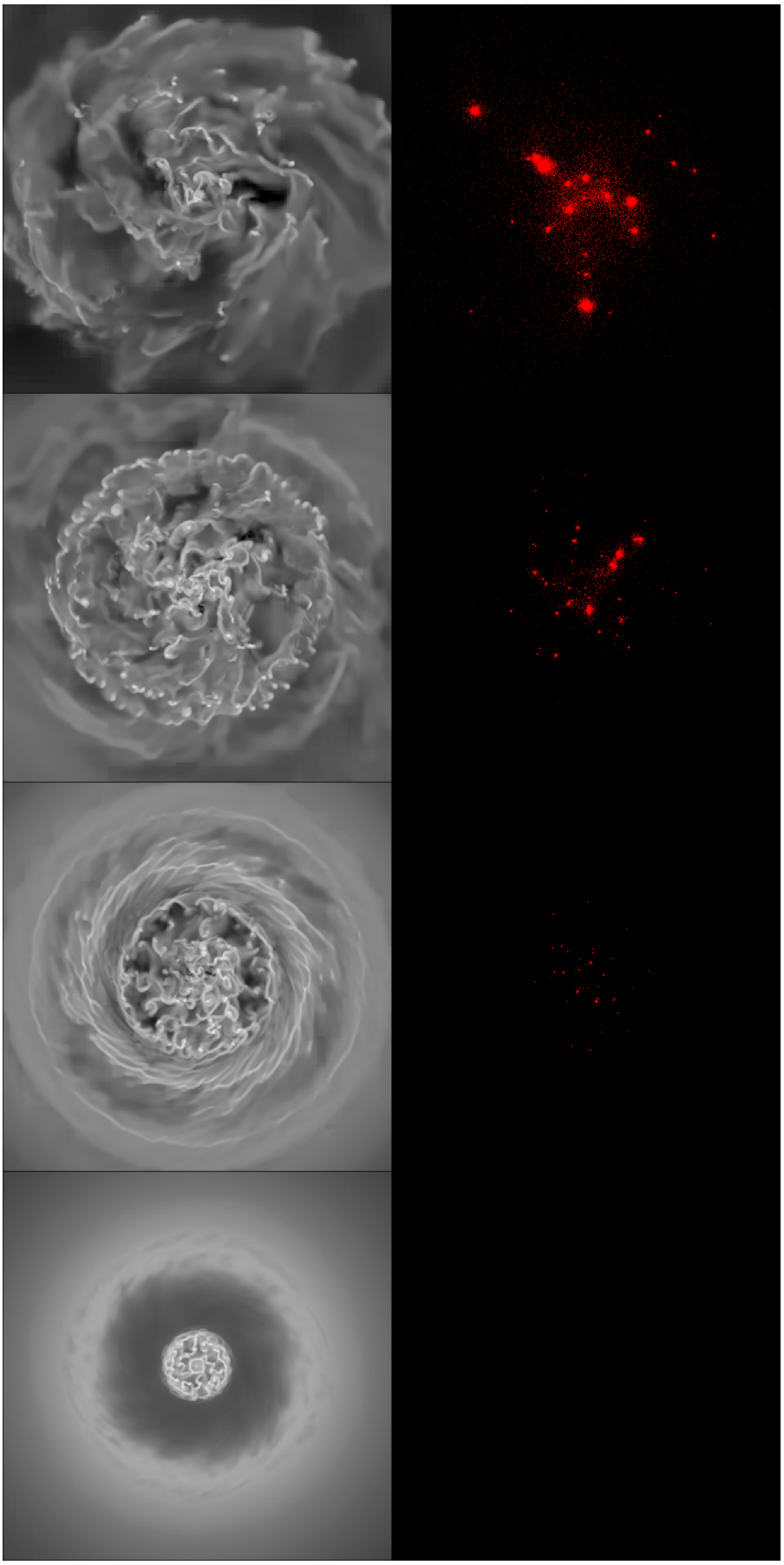


Figure 4.4: Evolution of model AII. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

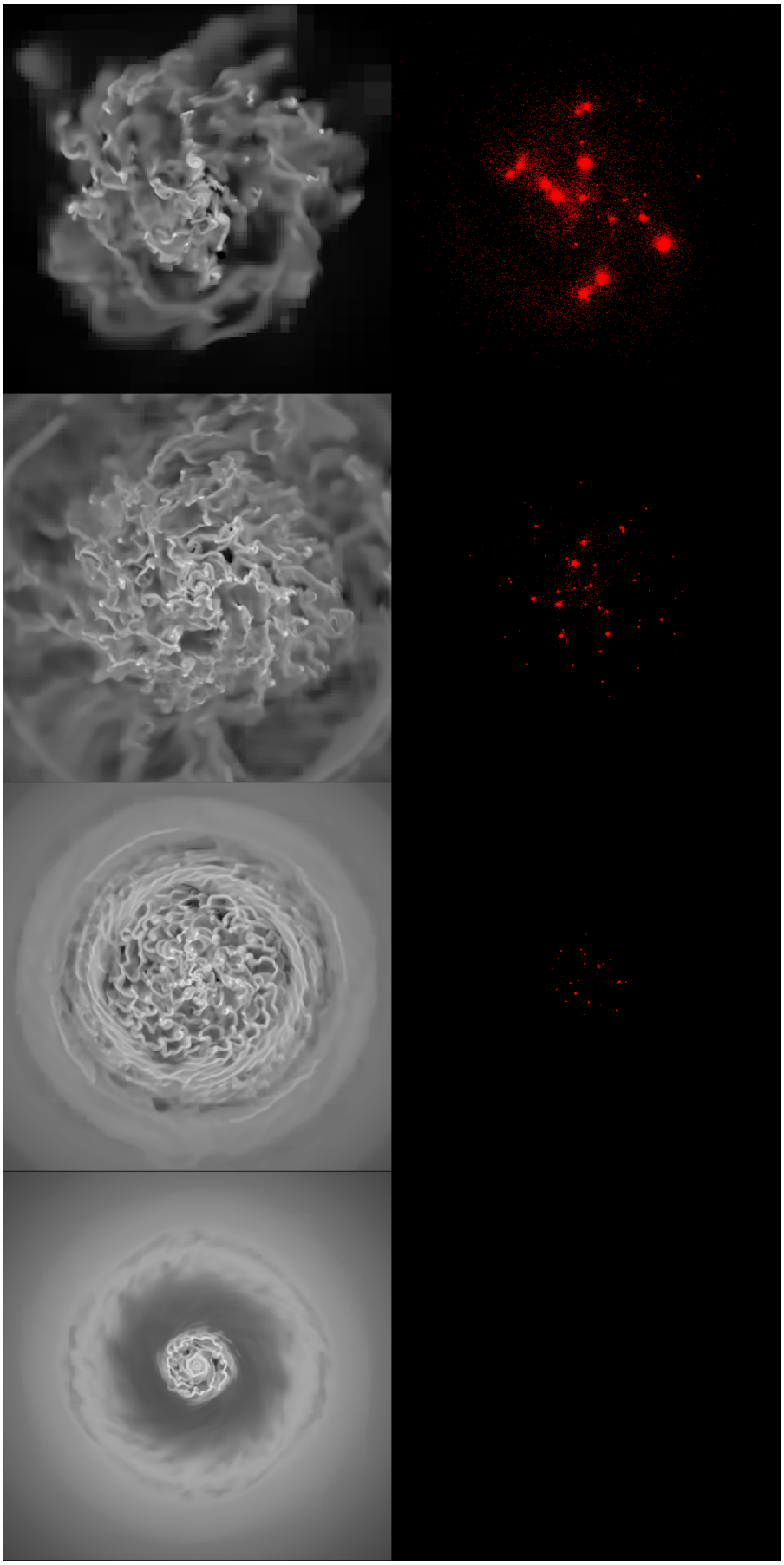


Figure 4.5: Evolution of model AIII. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

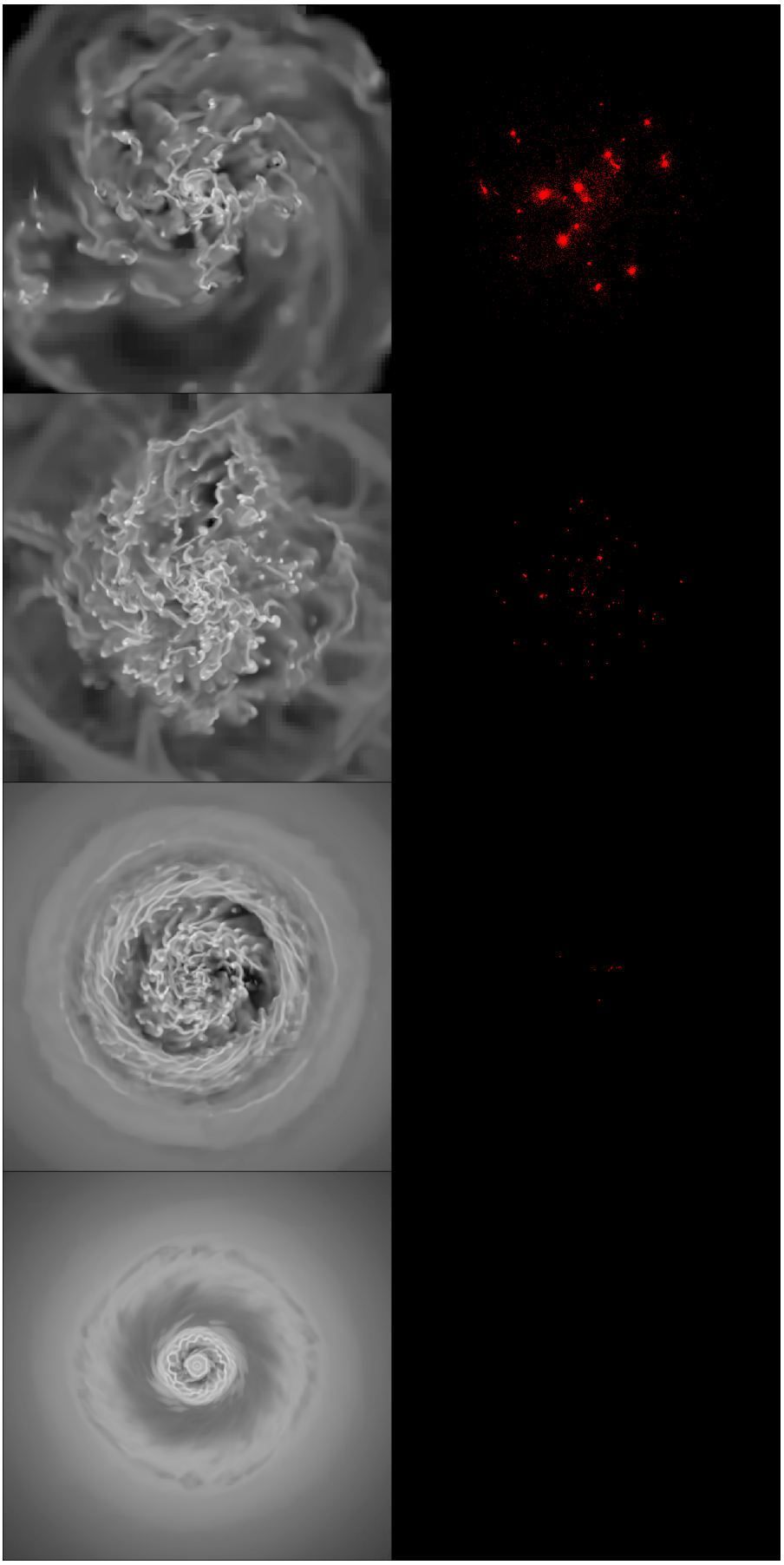


Figure 4.6: Evolution of model BI. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

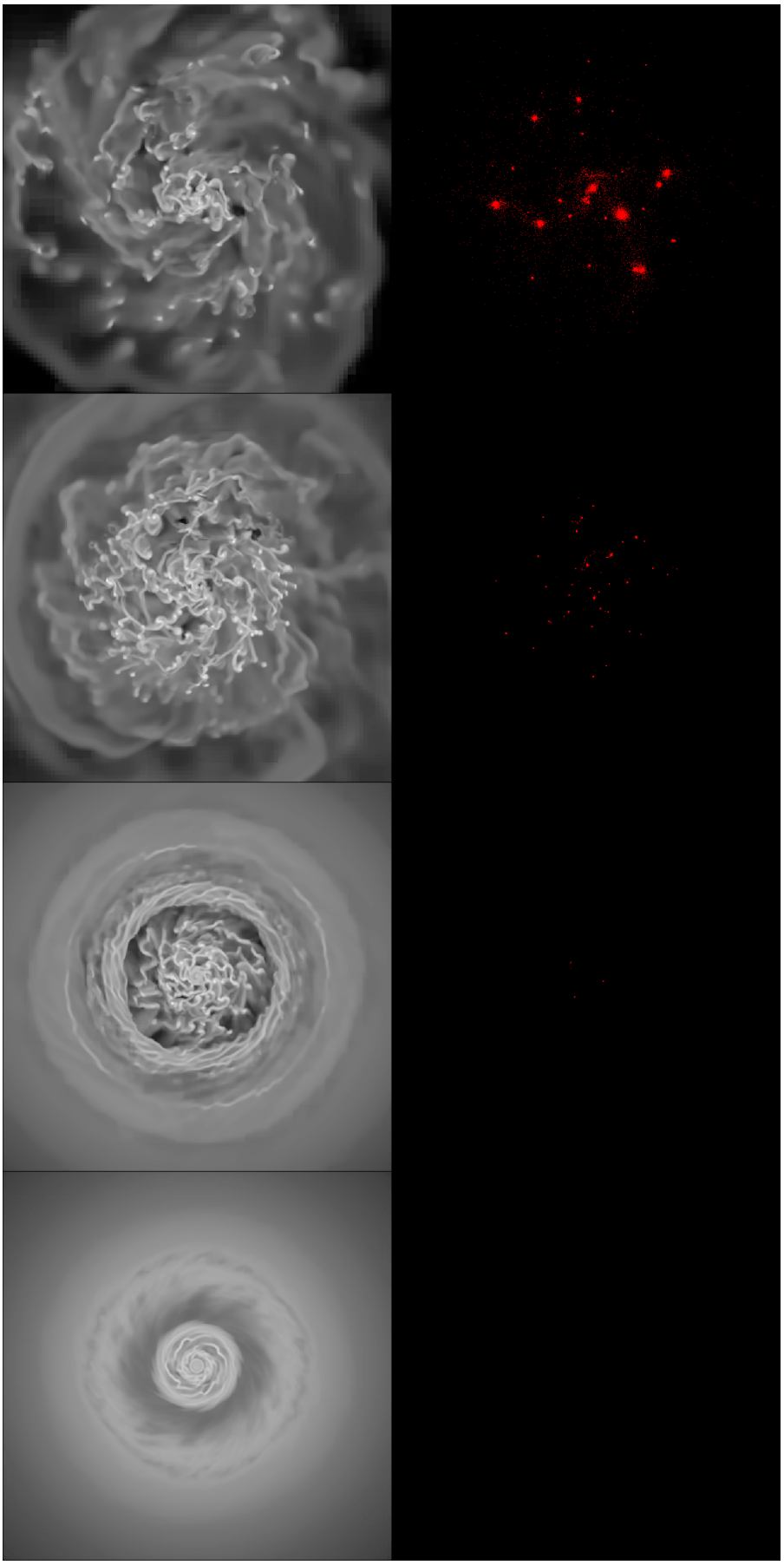


Figure 4.7: Evolution of model BII. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

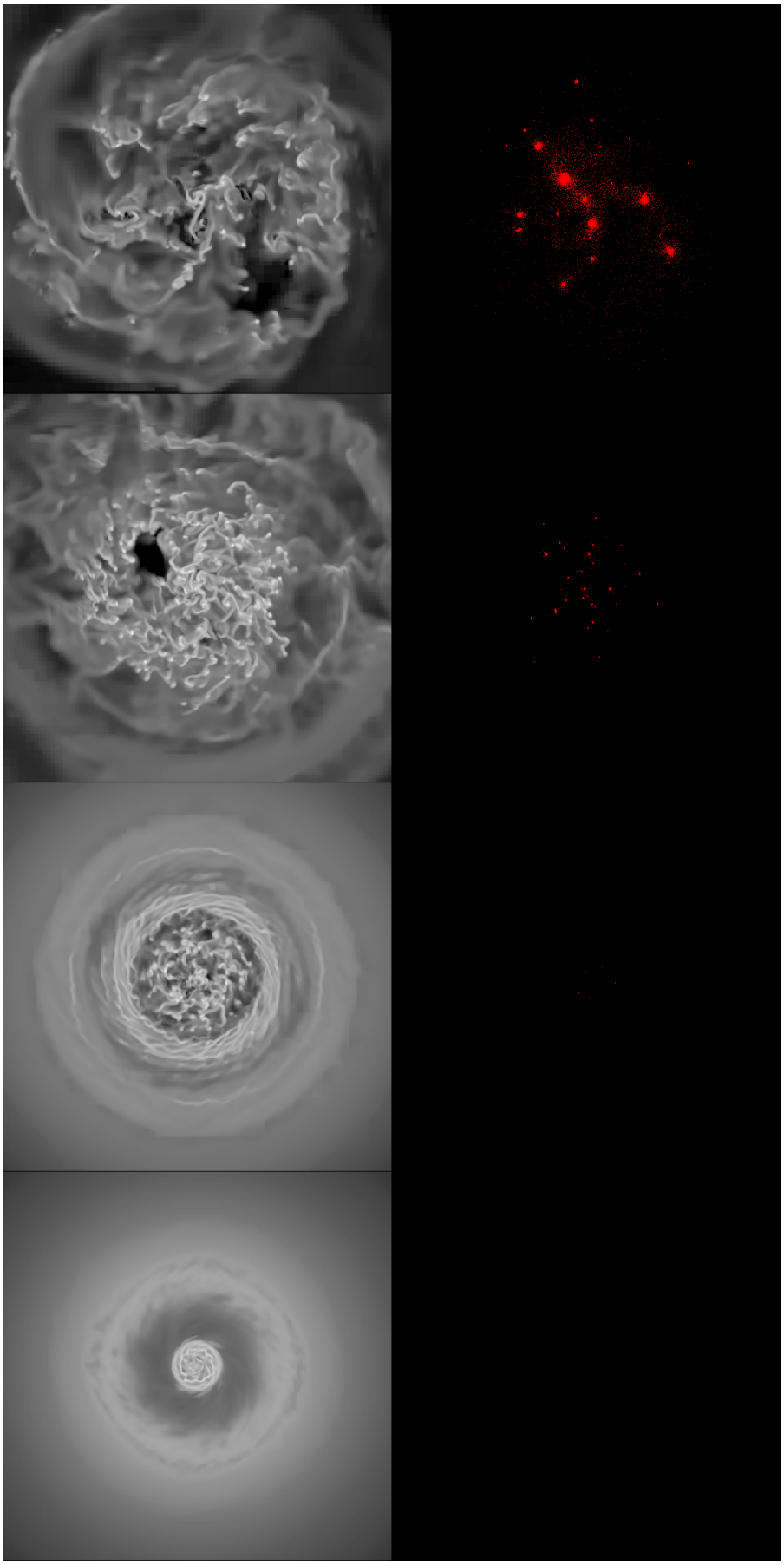


Figure 4.8: Evolution of model BIII. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

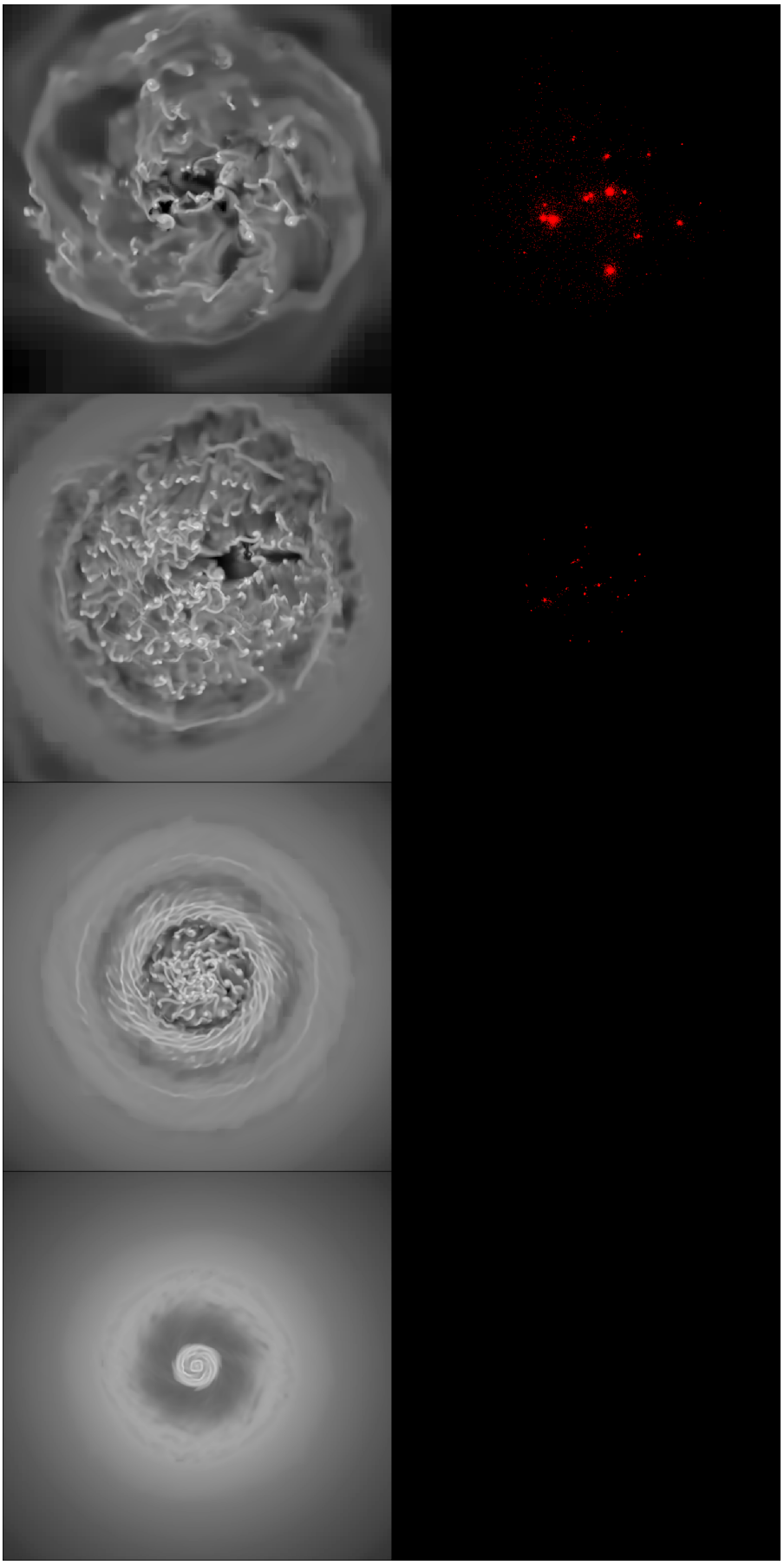


Figure 4.9: Evolution of model BIV. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

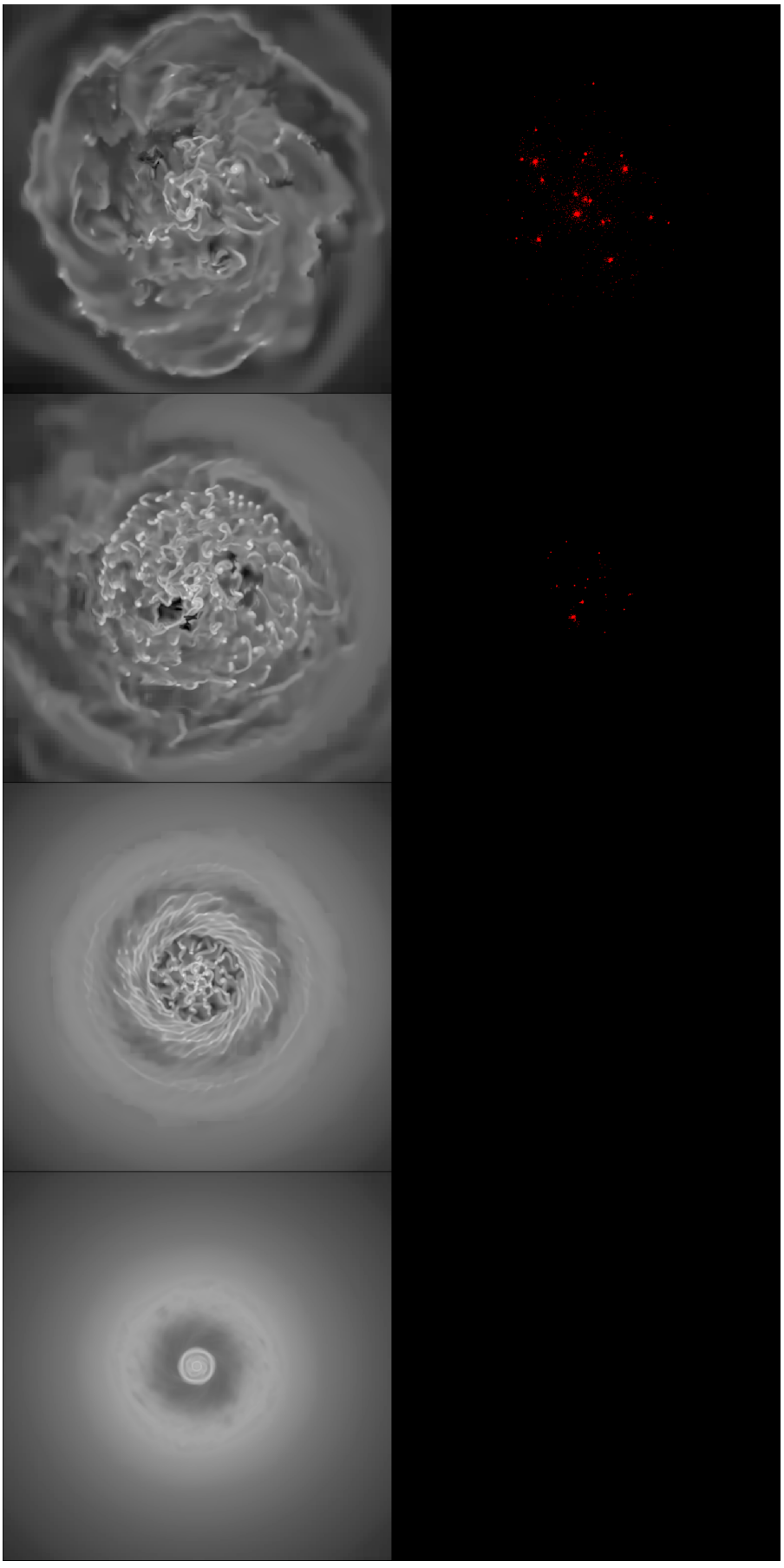


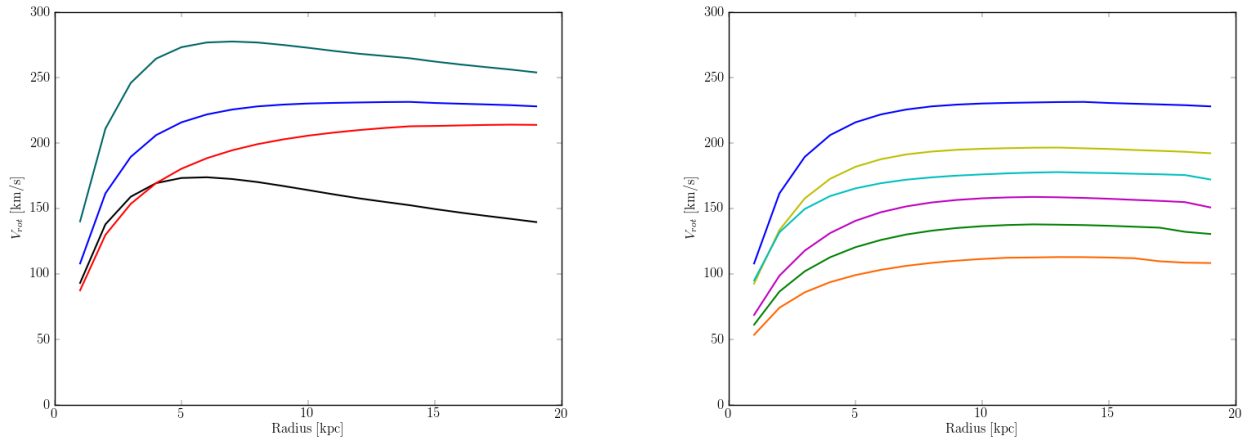
Figure 4.10: Evolution of model BV. Upper row shows density and lower row shows stars. From left to right columns represent snapshots of 30 kpc wide at times $t \sim$100, 200, 400, and 800 Myr.

In the following sections we will study in detail the properties of each simulation, including their dynamical properties in §4.1 (e.g. rotation curves) and the stability of the disks in §4.2. In §4.4 and §4.5 we will show how our galaxies behave related to the formation of stars and massive black hole growth. It is worthwhile to mention that in making plots and figures in Chapter 3 and 4 we have made an extensively use of `yt` [1] (Turk et al. 2011) and `jacques` [2].

4.1 Galaxy dynamics

4.1.1 Rotational velocities

In Figure 4.11 we show the rotational velocity profiles for every galaxy in the Group A (left) and Group B (right) at the beginning of the runs. In general they are flat rotation curves as observed in several galaxies, with a peak ranging from 160 km s^{-1}in simulation AII to 280 km s^{-1}in simulation AI for the first group and from 110 km s^{-1}in simulation BV to 230 km s^{-1}in simulation F for the second group. There are some exceptions as the black curve in Figure 4.11a, which is not completely flat. Instead the velocity start to slowly decrease after reaching its maximum value. The same process happens with the dark cyan rotation curve of the same figure, but not so pronounced. This slowly decay is correlated with the increase in the M_{rot} profile for simulation AII (black curve) in Figure 3.2. The rotation curves calculated for Group B are also related to the criteria for choosing their configuration. As they have similar M_{rot} profiles, they will also have similar rotation curves which is clearly appreciated in the figure.



(a) Group A (b) Group B

Figure 4.11: Initial rotation curves for Group A (*left*) and Group B (*right*). Colors for A are as in Figure 3.2 and for B are as in Figure 3.4

4.1.2 Orbital times

One important parameter to consider is the orbital time of the system, which we can derive from rotation curves. As pointed out in §1.1.1, this is calculated as the time that the gas take to complete one orbit.

[1] http://yt-project.org
[2] http://jacques.enzotools.org/

This is a straightforward calculation from the curves in the previous section. Given a fixed radius, the orbital time is easily computed using

$$t_{orb}(R) = \frac{2\pi R}{v_{rot}(R)} \tag{4.1}$$

It can be seen that having a flat rotation curve as the ones showed in Figure 4.11, t_{orb} will strongly depend on the radius at which its value is calculated. We then have to determined first the radius at which t_{orb} will be computed. Our criteria for choosing a radius is the fraction of stars enclosed by it. So for example, we have determine the orbital time at the radius of the most distant star in the disk which encloses a 100% of stars and we have called it R_{100}. A second and third comparative radius are the ones that enclose 95% and 90% of total stars. The different radius are shown graphically in Figure 4.12 right in yellow, green and blue dashed circles respectively. For comparison it has also been plotted gas density in Figure 4.12 left with a wide of 70 kpc.

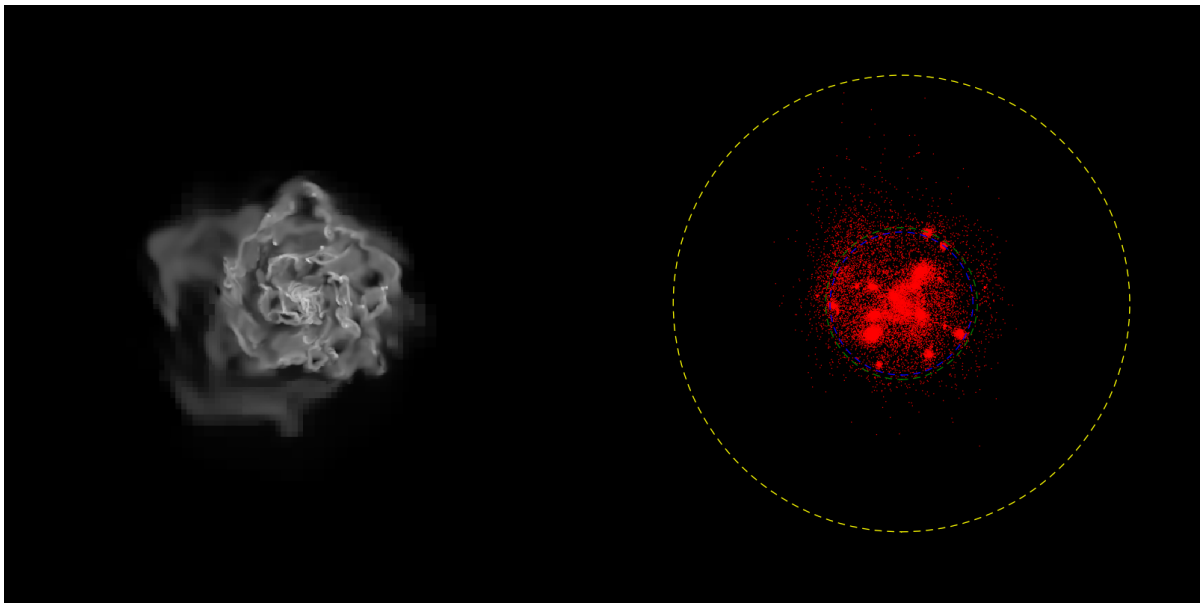


Figure 4.12: Radius enclosing 90% (blue dashed line), 95% (green dashed line) and 100% of stars (yellow dashed line). In the left is plotted a density slice for comparison.

To calculate a representative value per simulation, we have averaged the radius of the last ten output data. Then using Equation 4.1 and rotation curves from Figure 4.11, the calculation of t_{orb} is straightforward. Table 4.1 summarizes the value computed for each radius, where we have assumed the notation R_X is equal to the radius enclosing X% of stars. As seen in Figure 4.12, R_{100} is not a representative radius of star formation, because great fraction of stars included by R_{100} are there not because they were formed in that place, but because they were ejected from the inner regions of the disk. Taking this into account, R_{95} or R_{90} are much better options to be the representative radius. As these last two radius are similar, for simplicity we have chosen R_{95} just to enclose a greater percentage of stars.

	Name	R_{90} [kpc]	$V_{rot}(R_{90})$ [km s^{-1}]	$t_{dyn}(R_{90})$ 10^8 [yr]	R_{95} [kpc]	$V_{rot}(R_{95})$ [km s^{-1}]	$t_{dyn}(R_{95})$ 10^8 [yr]	R_{100} [kpc]	$V_{rot}(R_{100})$ [km s^{-1}]	$t_{dyn}(R_{100})$ 10^8 [yr]
Fiducial	F	8.38	228.67	2.25	8.78	229.16	2.35	27.26	-	-
Group A	AI	7.77	276.65	1.72	9.39	273.74	2.11	23.35	-	-
	AII	10.61	161.61	4.03	10.64	161.48	4.05	44.39	31.98	8.53
	AIII	8.41	201.09	2.57	8.65	201.86	2.63	31.39	-	-
Group B	BI	7.77	193.18	2.47	9.08	194.98	2.86	16.42	194.34	5.19
	BII	8.16	174.17	2.88	8.24	174.29	2.90	18.00	174.59	6.32
	BIII	7.64	153.85	3.05	8.24	155.31	3.26	21.90	-	-
	BIV	6.14	126.98	2.97	7.05	130.56	3.32	12.55	137.62	5.60
	BV	5.34	100.86	3.25	7.88	108.72	4.46	14.03	112.73	7.65

Table 4.1: Orbital times and rotational velocities of the simulated galaxies at different radius

4.2 Stability

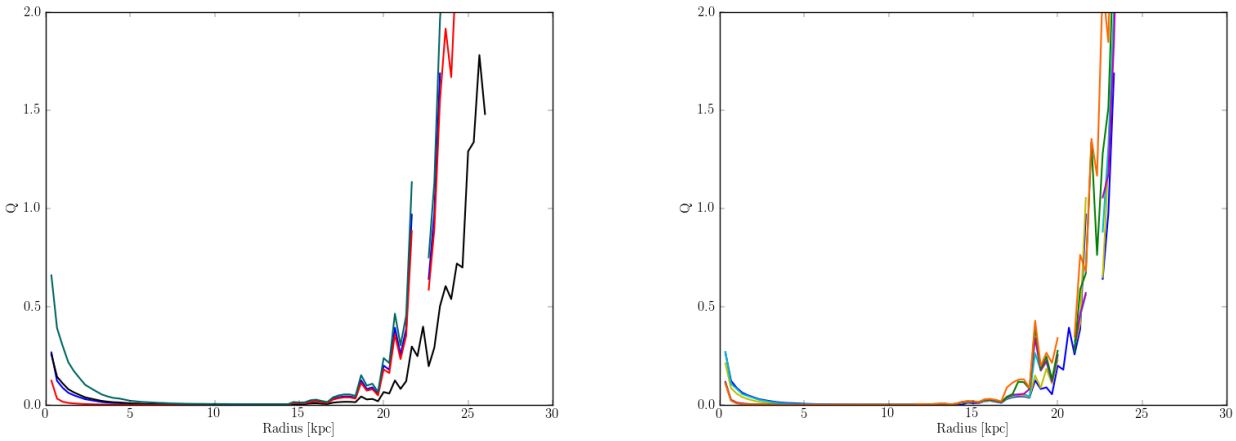
4.2.1 Toomre Q parameter

To analyze the dynamical stability of the simulated disks, we have calculated Toomre Q parameter (Toomre 1964, Goldreich and Lynden-Bell 1965) given by (Leroy et al. 2008):

$$Q = \frac{\kappa\sigma}{\pi G\Sigma_{gas}} \tag{4.2}$$

where κ is the epicyclic frequency, Σ_{gas} the gas surface density and σ the gas velocity dispersion. It gives us a criteria to determine if the galaxy will globally fragment and therefore create stars. In disk galaxies, a critical average value of $Q \sim 1$ has been found. If Q is greater than this value, the disk will be stable. On the contrary, if it is less than unity then it will be unstable.

In Figure 4.13 we can see that initially all our disks have $Q < 1$ for radius at least until 25 kpc, which means that they are unstable and therefore will fragment. As deduced from Table 4.1 a radius of 25 kpc will enclose at least 95% of the stars created during the evolution of the galaxies. This prediction can be compared with Figures 4.2 to 4.6 where it is clearly appreciated that fragmentation and formation of clumps occur where it is predicted.



(a) GroupA (b) GroupB

Figure 4.13: Toomre Q parameter at initial time for Group A (*left*) and Group B (*right*). Colors are s in Figure 3.4

The evolution of the Toomre parameter for model F is shown in Figure 4.14. The solid line represents the profile of Q at t =100 Myr, dashed line at $t = 200$ Myr and dotted line at $t = 400$ Myr. There are two clear tendencies that can be appreciated in the graph: a) the median value of the parameter increases as time goes by, beginning at values $Q << 1$ but increasing to 0.1 at time $t = 100$ Myr, 0.5 at $t = 200$ Myr and reaching 1.2 at $t = 400$ Myr; and b) as the disk evolves the radius at which the Toomre parameter increases to values greater than 2 decreases from ~ 23 kpc to ~ 7 kpc. This last feature is easier to see in Figure 4.2 where it is shown that in the outer regions of the simulations the gas becomes smoother.

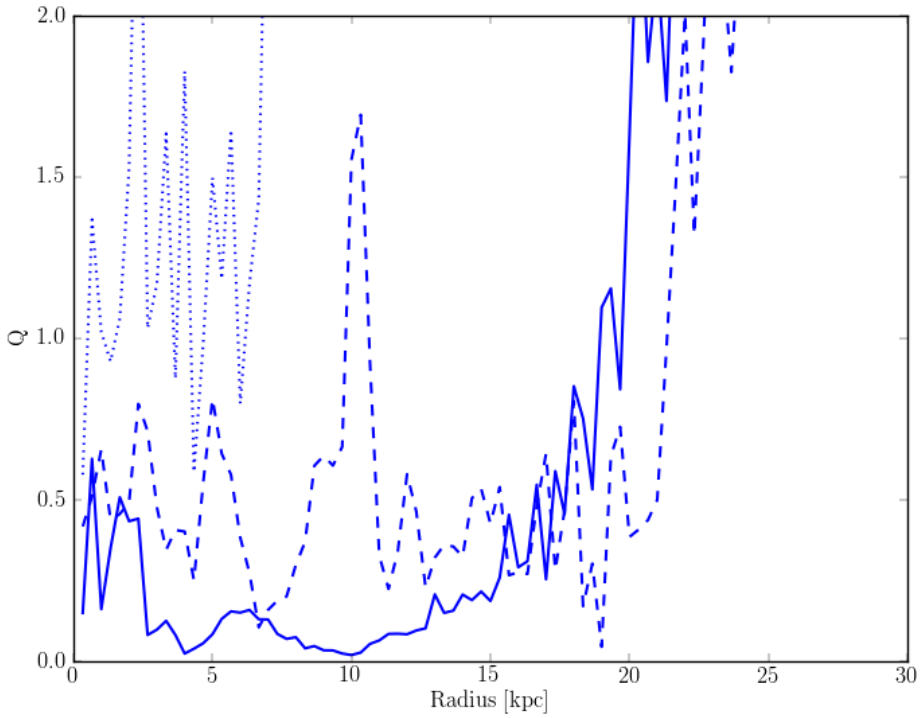


Figure 4.14: Evolution of the Toomre Q parameter for model F. Solid, dashed and dotted lines are Q profile at t=100, 200 and 400 Myr, respectively

4.3 Interstellar Medium Statistics

4.3.1 Probability Density Function

All our models reach a quasi-stationary state where a fragmented and highly turbulent disk is observed. Figure 4.15 shows the density Probability Density Function (PDF) of our fiducial simulation. As in the rest of our models, the high density tail of the density PDF can be fitted by a lognormal PDF (LN-PDF) over almost four orders of magnitude in gas surface density (red dashed line) in concordance with simulations of Wada and Norman (2001), Kravtsov (2003), Tasker and Bryan (2006, 2008), Wada and Norman (2007). It is believed that this LN-PDF could be the origin of the Kennicutt-Schmidt relation (Elmegreen 2002, Wada and Norman 2007). The average densities in our best fit LN-PDF vary from $10^{-0.9} M_{\odot}\mathrm{pc}^{-2}$ to $10^{-0.65} M_{\odot}\mathrm{pc}^{-2}$ while the dispersion stays almost constant at $10^{0.95} M_{\odot}\mathrm{pc}^{-2}$ (see Appendix A.2).

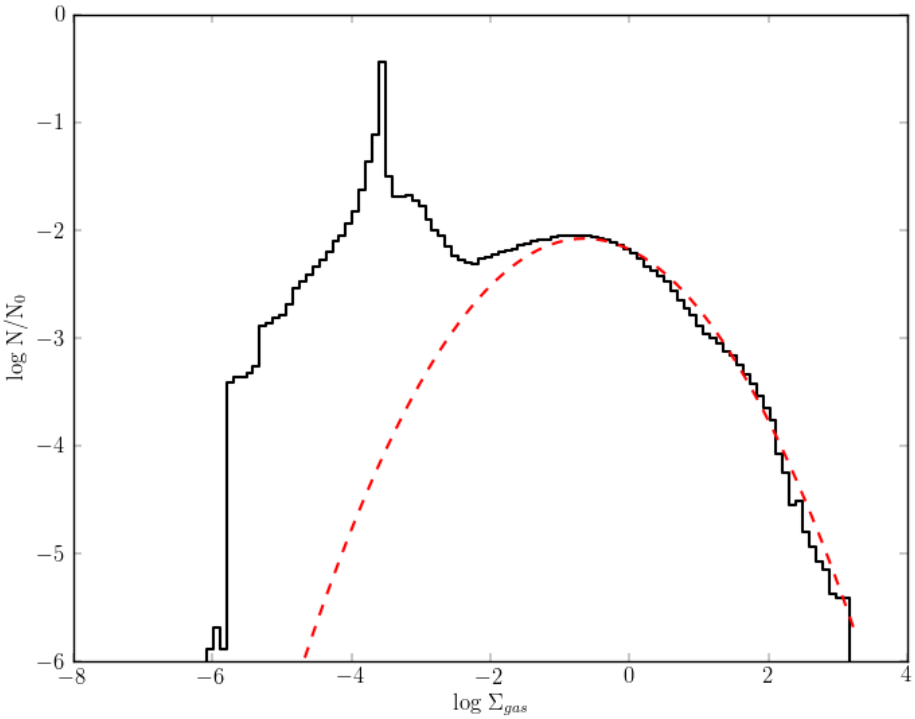


Figure 4.15: Density PDF for simulation F at the end of the run. The red dashed line shows our best lognormal fit to the high density part.

A lognormal probability density function for gas surface density is thought to be a natural outcome of isothermal gas (e.g. Passot and Vázquez-Semadeni 1998, Vázquez-Semadeni et al. 2000). Although the denser cells in our models have temperatures near 300 K, in general temperature lies in a range from 10^2 K to 10^4 K (Figure 4.16). Thus we obtain a lognormal fit at high densities, even though our simulations are not isothermal.

As noted at the beginning of this Chapter, supernovae explosions originates zones of low density seen as "black bubbles" in face-on snapshots (see Figure 4.8). This also can be viewed in the density PDF as a peak in the low density part of the curve. A similar feature was also obtained by Slyz et al. (2005),

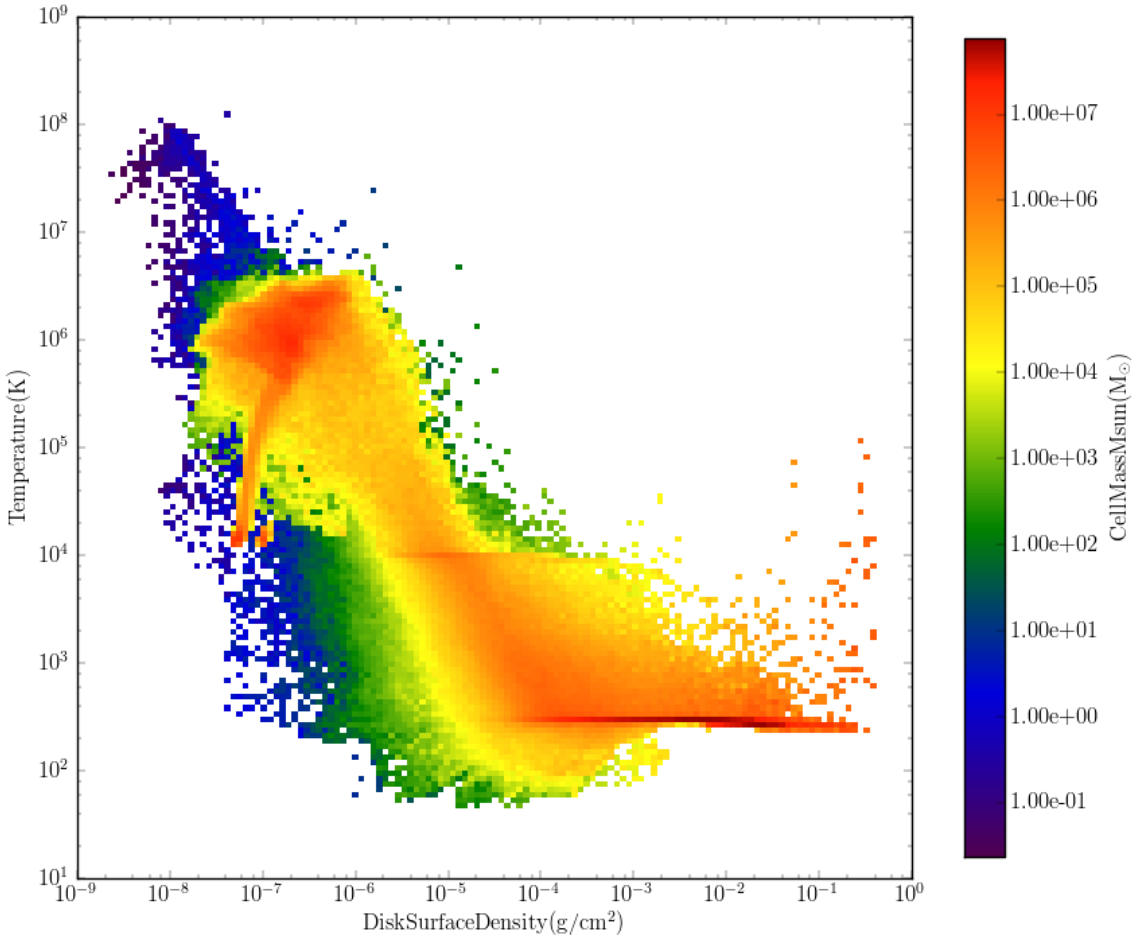


Figure 4.16: Gas surface density and Temperature phase diagram for model F, weighted by cell mass.

although they found a bi-modal PDF for simulations with feedback which is not seen in our case.

4.3.2 Power spectrum

We also analyze the power spectrum of the Interstellar Medium (ISM) in our models. The power spectrum of our fiducial simulation F can be seen in Figure 4.17. Total energy spectrum is plotted in dashed line and solid line is used for radial velocity power spectrum. Both profiles have a similar slope being almost parallel in the whole k range. The shift in the y-axis is in average $\sim 0.5 \sim \log 3$, which can be explained if we assume isotropic turbulent velocity. In that case we have $< v_r^2 >=< v_\theta^2 >=< v_z^2 >$, and then:

$$\begin{gathered} < v^2 >=< v_r^2 > + < v_\theta^2 > + < v_z^2 >= 3 < v_r^2 > \\ \Rightarrow \log < v^2 > \sim \log 3 + \log < v_r^2 > \end{gathered} \tag{4.3}$$

For comparison, in dotted lines we have plotted the slopes of the Kolmogorov $E(k) \sim k^{-5/3}$ (Kolmogorov 1941) and Burger $E(k) \sim k^{-2}$ power spectrum. The former is an indicator of an incompressible flow while the latter indicates a shock-dominated one. Our results agree best with a slope $\beta = -2$, but for larger values of k it is significantly steeper than both of them, reaching values of $\sim$ -4.

As we are modeling disk galaxies, most of their velocity component is in the tangential direction in the plane of the disk. Assuming that will allow us to say that in our case $v_{turb} \sim v_r$. As we plotted the radial velocity as a function of the wavenumber, this will lead us to a direct relation between v_{turb} and the wavelength λ. If we remember Equation 1.7 then we will be able to test that hypothesis. This analysis will be done in more detail in §4.5.

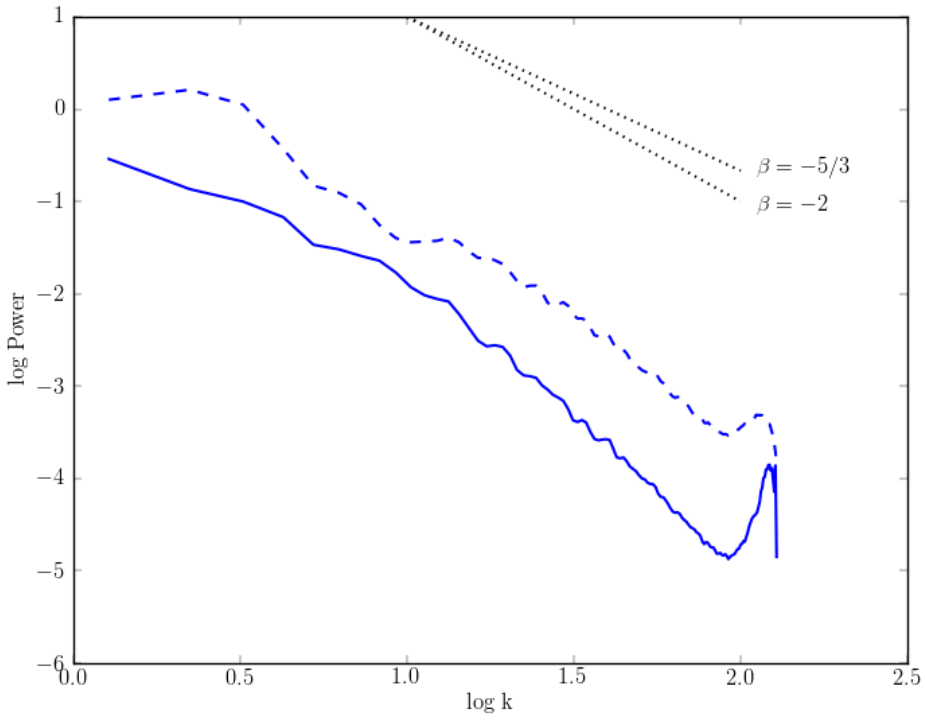


Figure 4.17: Kinetic energy power spectrum ($E(k)$) for simulation F at the end of the run. Solid line is radial velocity power spectrum, while dashed line is total energy power spectrum. For reference, the Kolmogorov and Burger slopes have been plotted as dotted lines.

4.4 Star Formation

4.4.1 Star Formation Rates

Star formation rates are highly variable, commonly oscillating around an average value. Usually effects such as supernovae explosions trigger star formation, provoking high peaks in the SFR curves. This variability can be a problem when plotting star formation rates if we display them in order to compare them all together. Therefore as a previous step, we have fitted a smoother curve to them. By applying a median filter to the original points we reduced the noise and we then calculated a polynomial fit to the points. An example is shown in Figure 4.18 where the original data is plotted as a red solid line while our fit is represented by a blue dashed curve for our fiducial simulation. The rest of the simulations are shown in Appendix A.1. It is important to mention that we use this fit only for plotting purposes, in any other calculation (for example Σ_{SFR} in §4.4.2) we have used the original values.

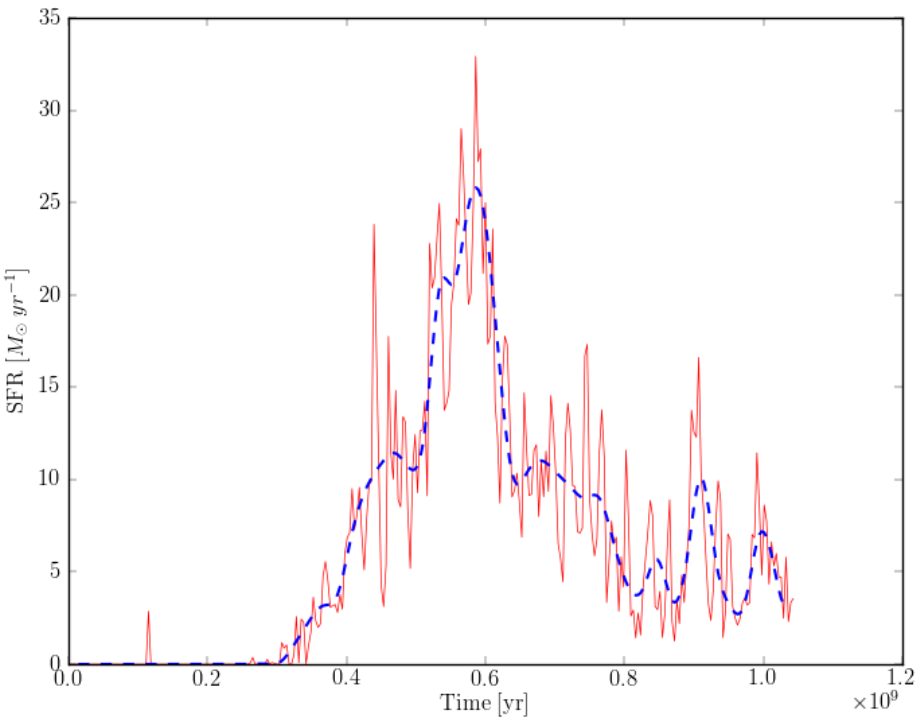
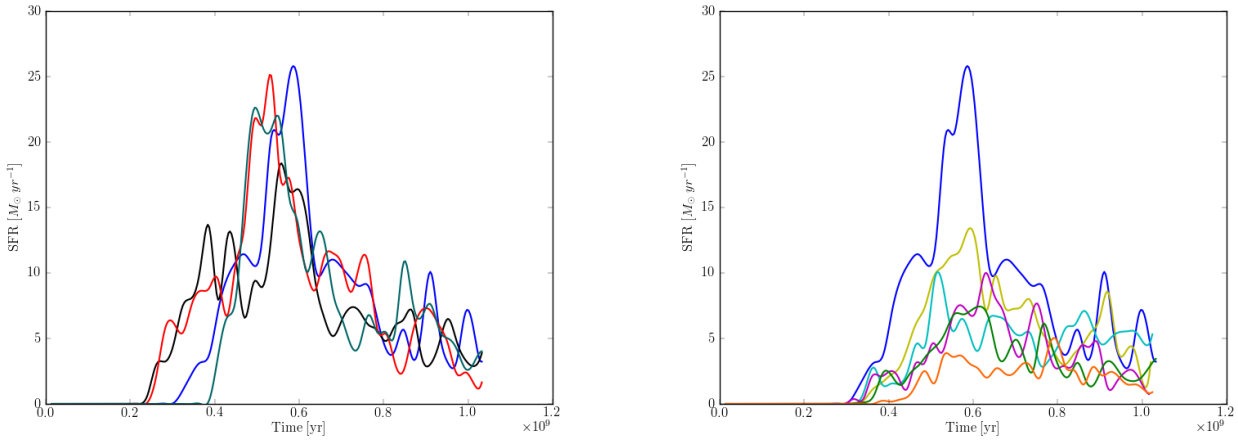


Figure 4.18: Star Formation Rate fit. In red solid line is shown the star formation rate, while in blue dashed line is shown SFR after applying a median filter and a polynomial fit.

We present Star Formation Rates for Group A and Group B in Figures 4.19a and 4.19b respectively. In general they follow a similar pattern, reaching a peak of star formation at $t \sim 500 - 600$ Myr and then decreasing slowly to a stationary value. One key aspect in Figure 4.19 is the time at which star formation starts for each galaxy. In the case of Group A there are differences of $\sim 2 \times 10^8$yr, where in the case of Group B this difference is no more than $\sim 0.7 \times 10^8$yr.



(a) Group A (b) Group B

Figure 4.19: Star Formation Rates profiles for Group A (*left*) and Group B (*right*). Colors are s in Figure 3.4

4.4.2 Star Formation Laws

We will focus our study in the Kennicutt-Schmidt and Silk laws introduced in §1.1.1. Our aim is to explore the factors that can affect these relations. In order to do that we will calculate star formation rates, gas surface densities and dynamical times for each galaxy at different stages of evolution of the galaxies. Assuming that the want to measure SFR at time t^*, we will consider two different approaches to calculate it: a) the exact star formation rate value at t^*, and b) the total cumulative mass of stars formed until t^*, divided by the total time elapsed from the beginning of the simulation to t^*. From now on, we will adopt the names *instantaneous* star formation rate for the first case, and *average* star formation rate for the second one. It is important to make this distinction because observations are only able to measure instantaneous star formation rates, which can be a bias in the observationally deduced star formation laws.

In Figure 4.20 and 4.21 we start studying the *instantaneous* KS law. These figures show star formation rate surface density versus gas surface density. For reference we first plot the values of these quantities at $t \sim 300$ Myr which is when star formation has already begun for every simulation. Next, we select snapshots every ~ 60 Myr to include them as points in our plots . The two lines drawn in Group A simulations plots are the best fit to the points using a least-squares method (red solid line) and the original KS law from Kennicutt (1998) (black dashed line). The slope of the fit is wrote on the top left of each plot.

Each image is divided in three columns which represent different radius used to calculate surface densities. From left to right the radius are R_{90}, R_{95} and R_{100} (see Figure 4.12). It is clearly inferred that the larger radius we consider, the lower slope we get. Comparing images of the same row we can also say that taking R_{100} will give us a broader range of gas densities. Its explanation is that increasing the radius in the outer regions of the galaxy will not add significant amount of gas. It means that the area enclosed increase more than the gas mass enclosed, thus decreasing the gas surface density.

In the case of Group B, we have only plotted the fitting curve for R_{100}. That radius gives us a wider range of values for gas surface density as mentioned before. This will allow us to obtain a better fit to the points, which is not possible for R_{90} and R_{95} where we have a narrower range of values for Σ_{gas}. In the first two columns, the proximity of the points does not give a good fit making it a poor indicator of the relation. We consider that it is much better to compare our results with the original law (dotted black curve) in the case of R_{90} and R_{95}.

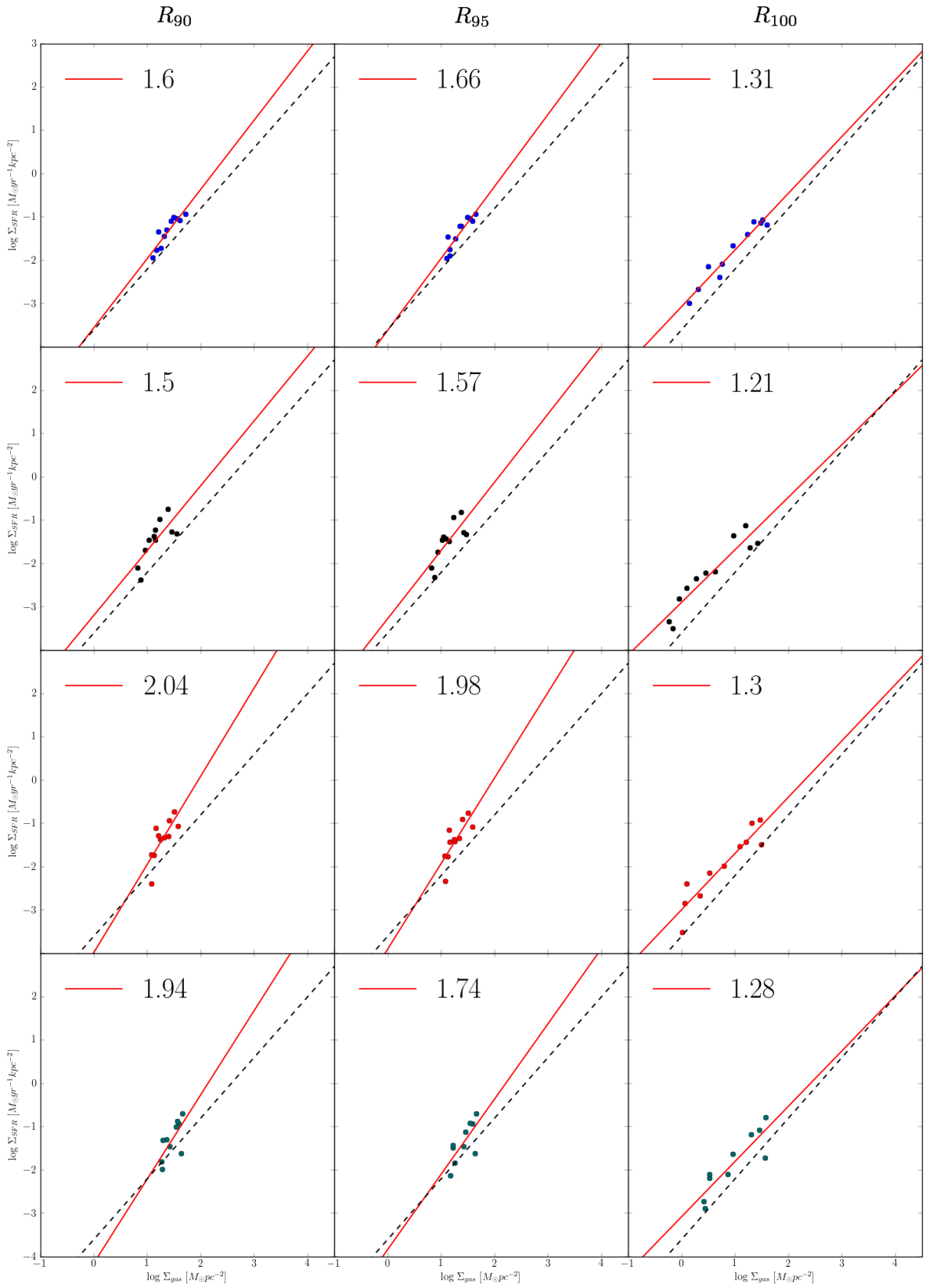


Figure 4.20: Kennicutt-Schmidt relation as a function of radius for Group A. Rows represent R_{90}, R_{95} and R_{100} from left to right. Columns represent models F, AII, AIII and AI from top to bottom. The red solid line shows our best fit to the points and the black dashed line shows the relation from Kennicutt (1998)

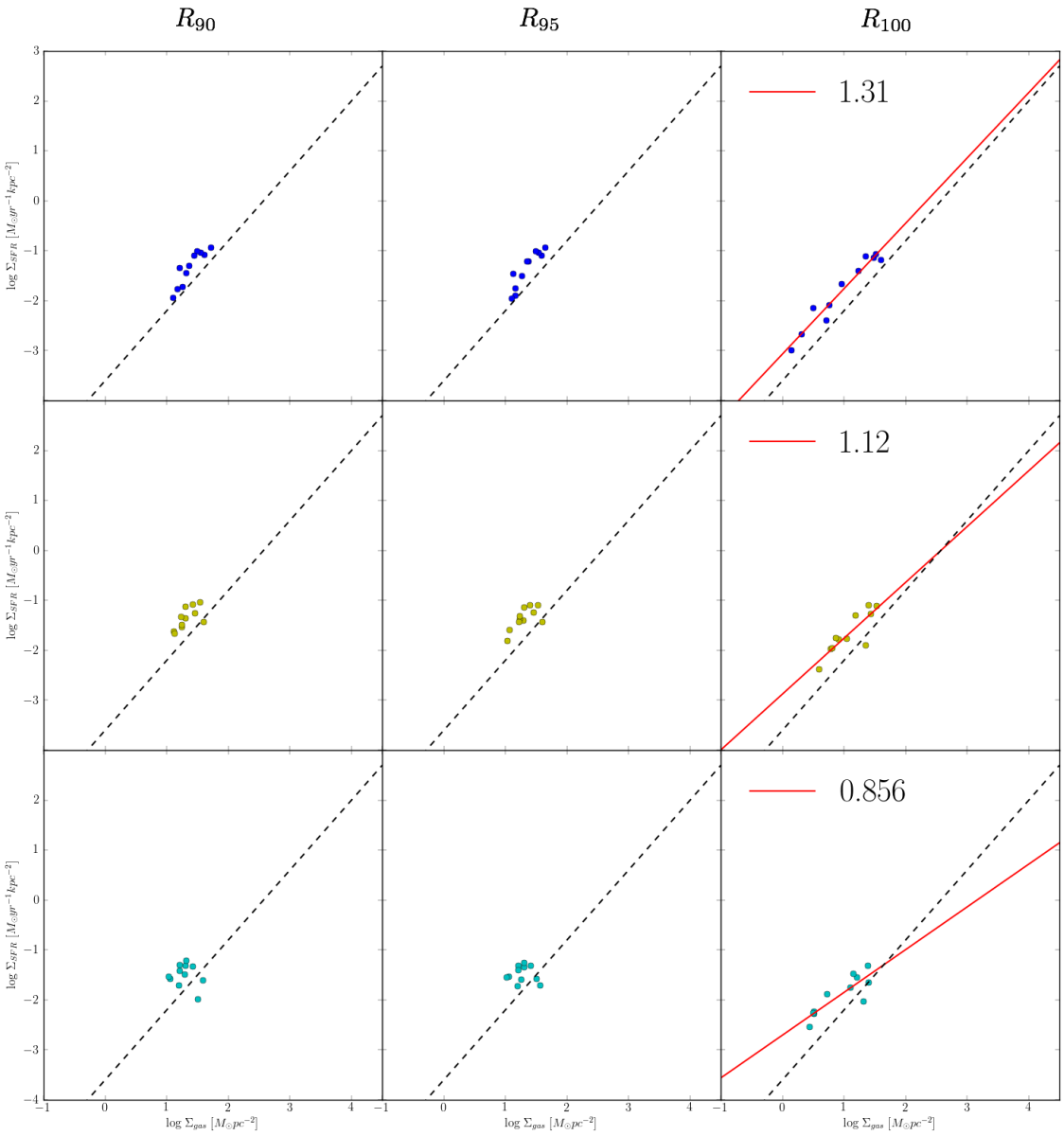


Figure 4.21: Kennicutt-Schmidt relation as a function of radius for Group B. Rows represent R_{90}, R_{95} and R_{100} from left to right. Columns represent models F, BI and BII from top to bottom. The red solid line shows our best fit to the points and the black dashed line shows the relation from Kennicutt (1998)

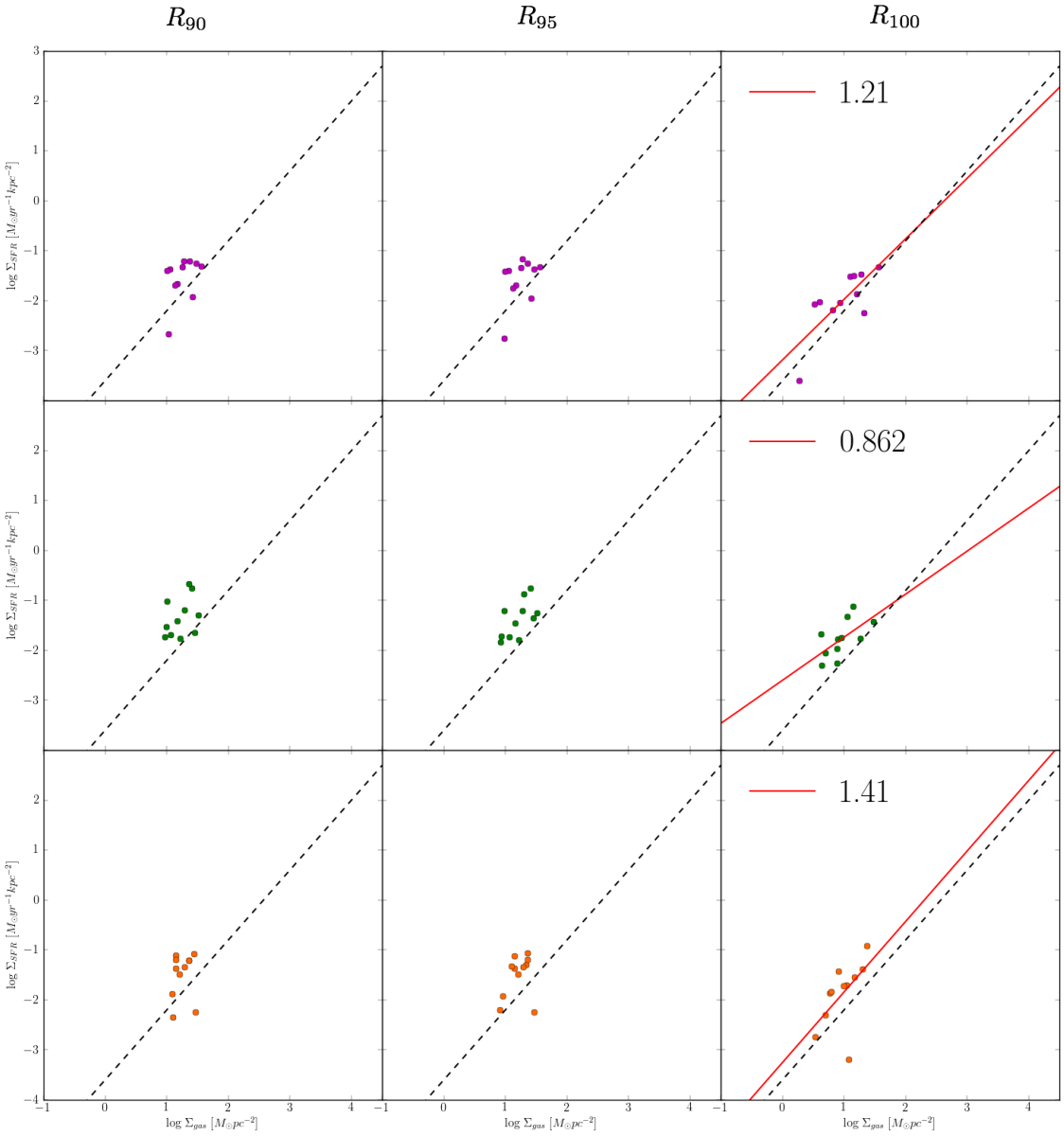


Figure 4.21: *Continuation:* Kennicutt-Schmidt relation as a function of radius for Group B. Rows represent R_{90}, R_{95} and R_{100} from left to right. Columns represent models BIII, BIV and BV from top to bottom. The red solid line shows our best fit to the points and the black dashed line shows the relation from Kennicutt (1998)

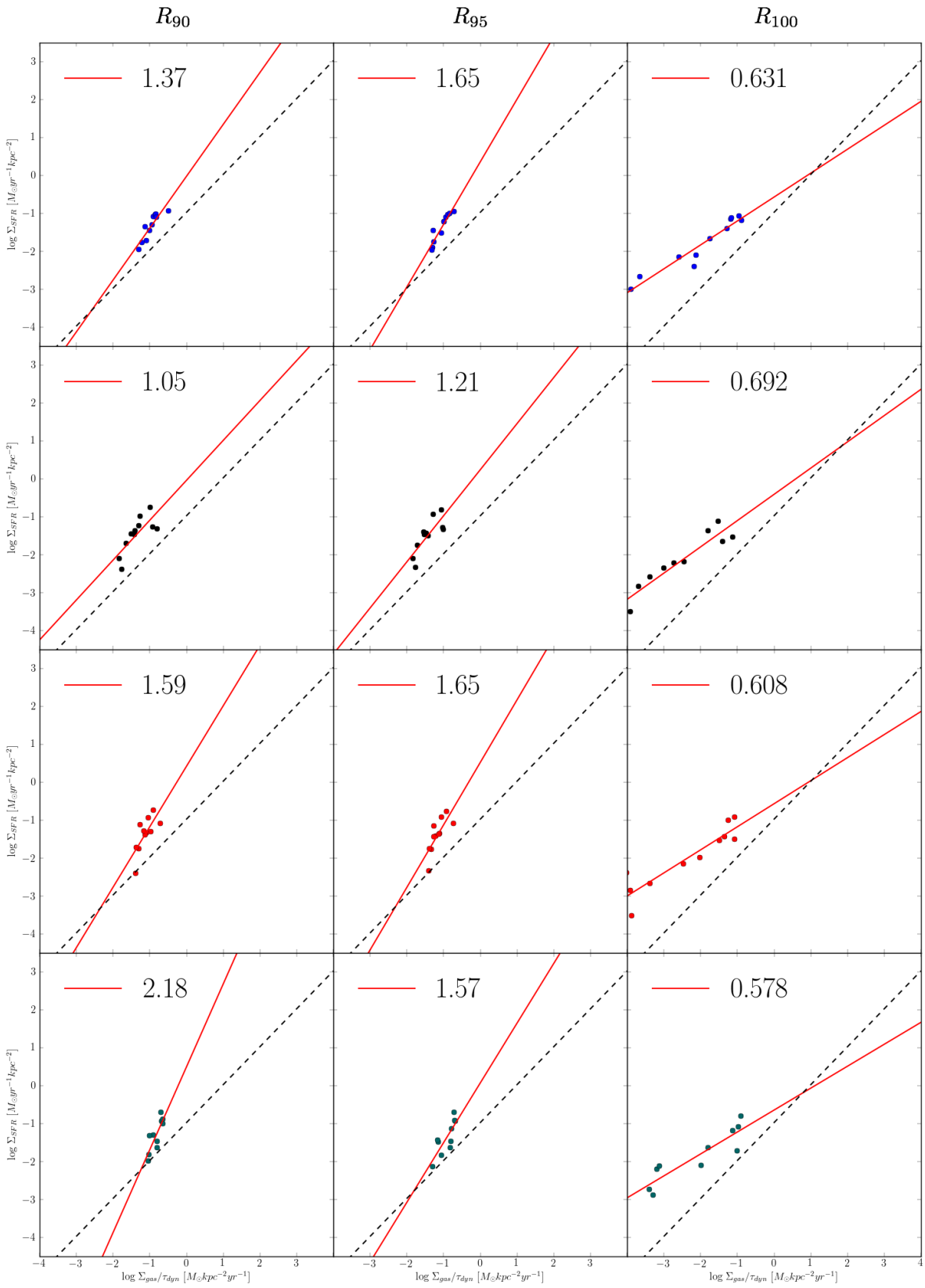


Figure 4.22: Silk relation as a function of radius for Group A. Rows represent R_{90}, R_{95} and R_{100} from left to right. Columns represent models F, AII, AIII and AI from top to bottom. The red solid line shows our best fit to the points and the black dashed line shows the relation from Kennicutt (1998)

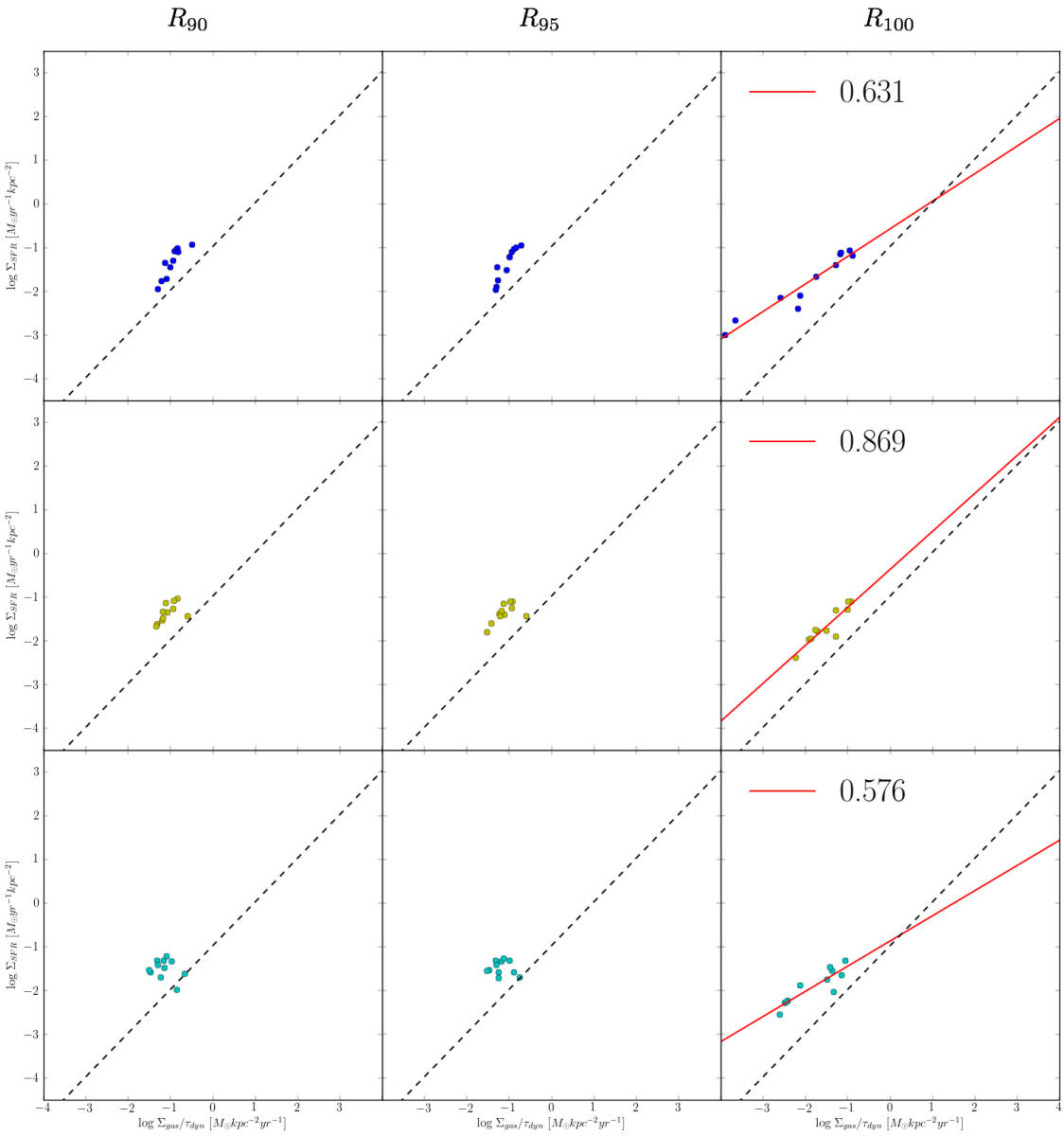


Figure 4.23: Silk relation as a function of radius for Group B. Rows represent R_{90}, R_{95} and R_{100} from left to right. Columns represent models F, BI and BII from top to bottom. The red solid line shows our best fit to the points and the black dashed line shows the relation from Kennicutt (1998)

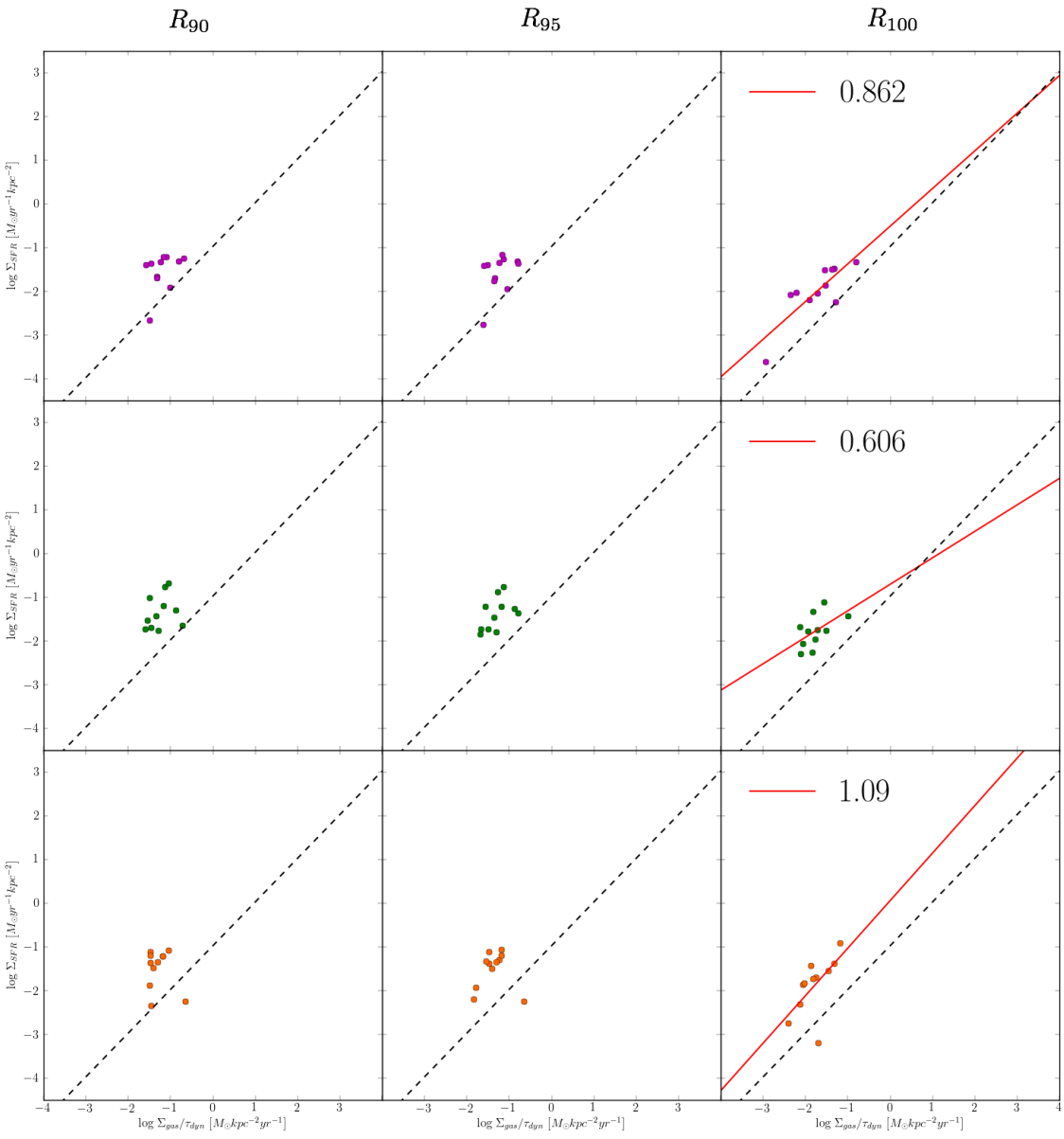
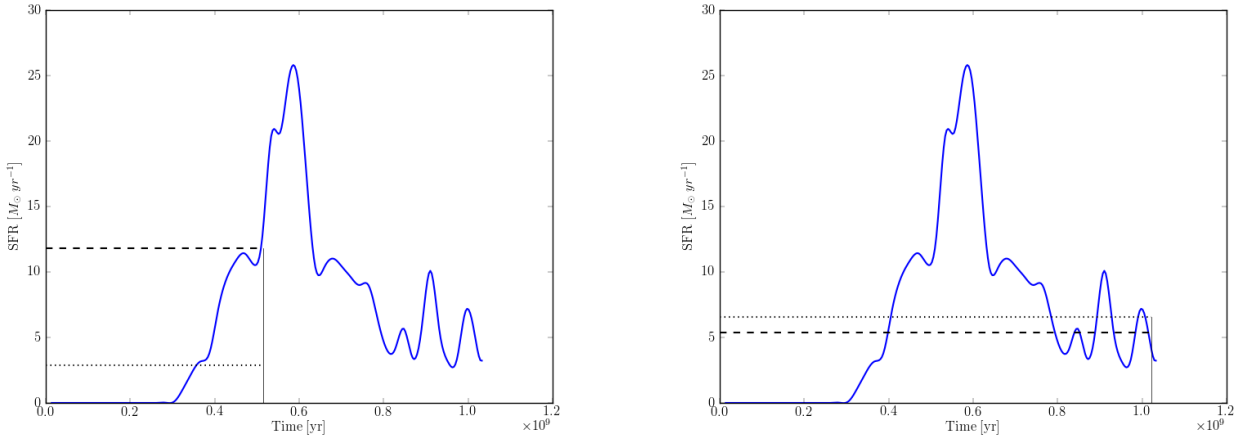


Figure 4.23: *Continuation:* Silk relation as a function of radius for Group B. Rows represent R_{90}, R_{95} and R_{100} from left to right. Columns represent models BIII, BIV and BV from top to bottom. The red solid line shows our best fit to the points and the black dashed line shows the relation from Kennicutt (1998)

A similar analysis for the Silk law is shown in Figures 4.22 and 4.23. Here we have plotted this relation using exactly the same points we used when studying KS relation. From these figures we deduce that both law behave in a equivalent way when increasing radius. In that sense we have the same problem fitting a curve to the data from Group B simulations at R_{90} and R_{95} due to lack of a wider range in gas surface density values. Even though not all of them gives us a poor fit, we have prefer to follow the discussion of KS law and not include the fits for these radius for the sake of consistency.

Once we have studied the dependence of both laws on the radius used for surface densities, a second parameter to be studied is how we measure star formation rates. To introduce the differences between instantaneous and average star formation rates, we will divide the SFR profiles (Figure 4.19) in two sections: before and after the peak. In the first case when SFR is increasing the instantaneous SFR will be greater than the average, because the last one is also influenced by the lower SFRs before the peak. In the second case the average SFR is greater than instantaneous. At those times the average SFR will be greatly influenced for the peak which will give a greater value. We can see this variation illustrated in Figure 4.24 where the dashed and dotted lines represent the value of the instantaneous and average SFR respectively.

Now, to see how it influences star formation laws, we will focus in Figure 4.25 which shows a comparison between the slopes obtained taking instantaneous (upper row) and average (lower row) star formation rates in Kennicutt-Schmidt law (Figure 4.25a) and Silk law (Figure 4.25b). These plots correspond to model F which means that upper rows in Figures 4.25a and 4.25b are exactly the same upper rows as in Figures 4.20 and 4.22 respectively. It is clearly noted that the second method of measuring SFRs gives us lower slopes. As we pointed out before, if average SFR tends to a fixed value then as gas is consumed, gas surface density will decrease while keeping almost constant the value of SFR.

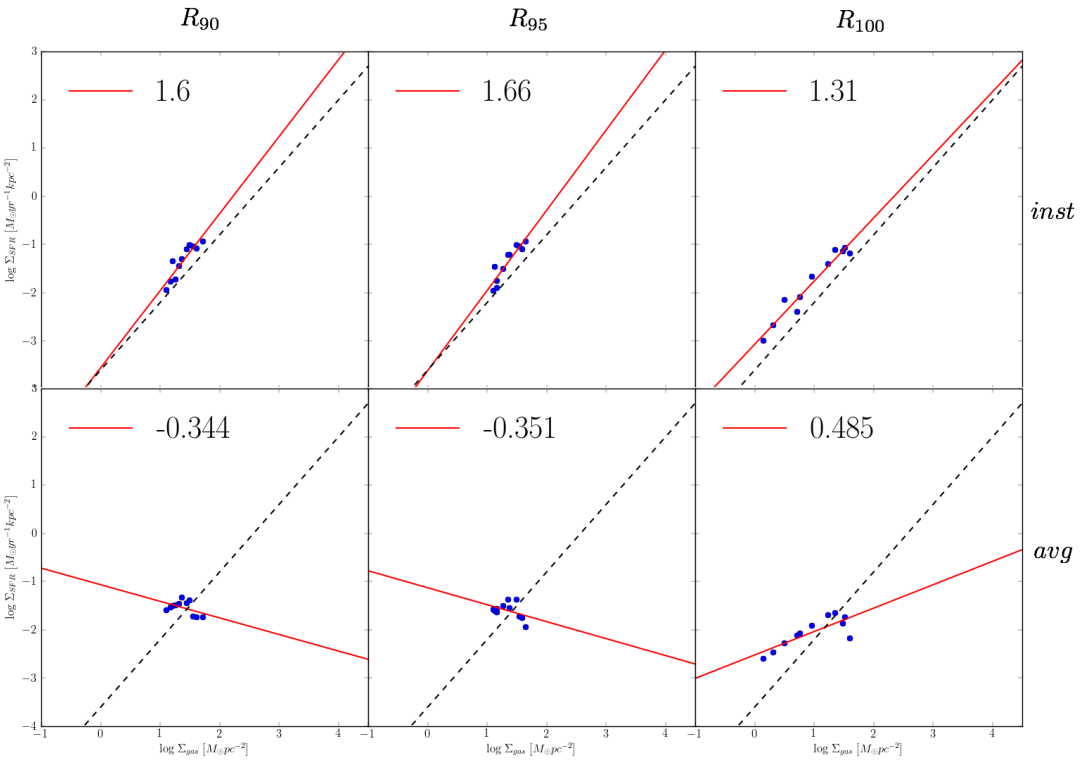


(a) Pre-peak (b) Post-peak

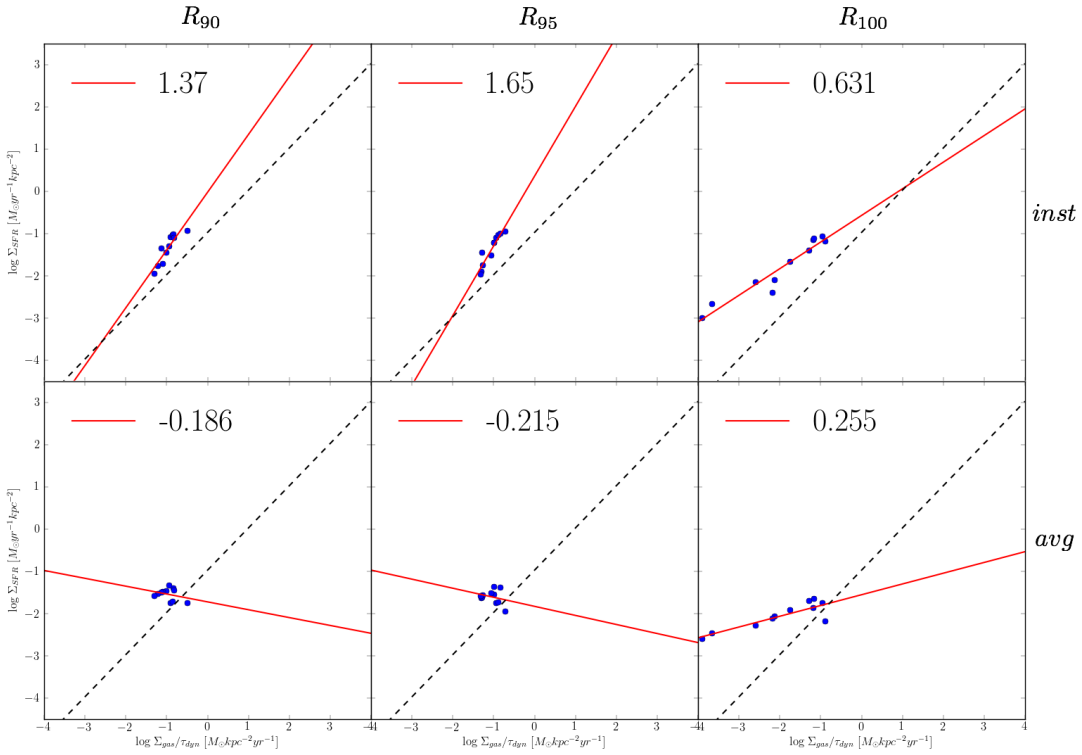
Figure 4.24: Example of differences between average (dotted line) and instantaneous SFR (dashed line) for the case before the peak (*left*) and after the peak (*right*) for model F

4.4.3 Dependency of M_{rot}

Finally, we also want to study how rotational mass affects the Kennicutt-Schmidt and Silk relations. We have plotted all simulations in one unique plot, where each point represent one of the models. Doing that we will be able to compare the behavior of both groups, the one with similar M_{rot} profiles at the



(a) Kennicutt-Schmidt law

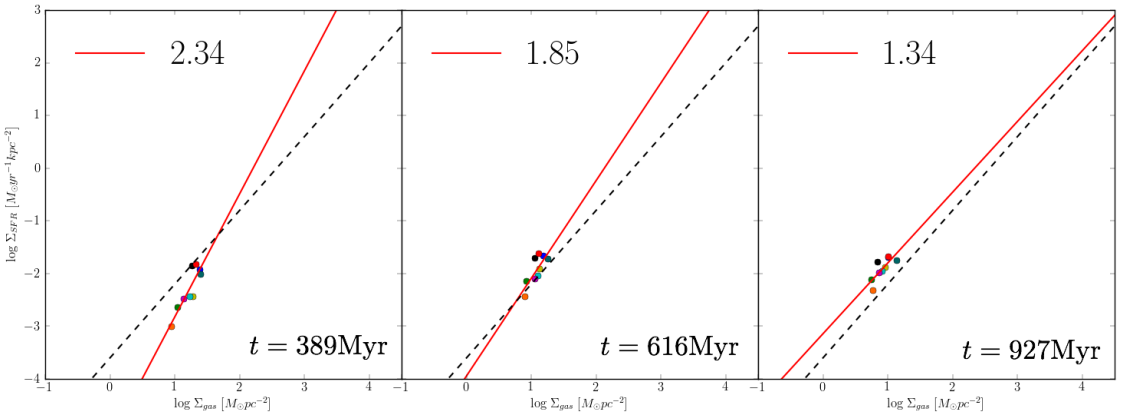


(b) Silk law

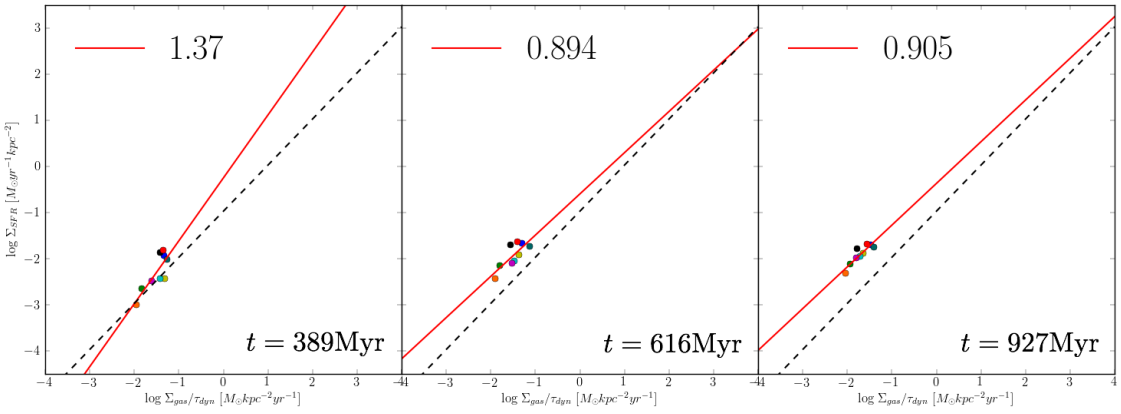
Figure 4.25: Comparison of Star Formation Laws ((a) Kennicutt-Schmidt, (b) Silk) using instantaneous (upper row) and average (lower row) Star Formation Rates for model F

beginning, and the one with different profiles.

In order to do that, we first have to choose a time at which we will analyze the SFR laws. In Figure 4.26 we have plotted three different times: 389, 616 and 927 Myr. It is important to mention that in the y-axis we have plotted the *average* star formation rate. We consider that it is a better indicator of the global SFR, converging to a fixed value as the galaxy evolves and decreasing the scattering in the law. Despite that, the relation does exhibit scattering at initial times, but mainly because galaxies start creating stars at different times (§4.4.1, Figure 4.19) which translates in different SFRs for different values of initial M_{rot}. This is easily seen at t =389 Myr in Figure 4.26, where Group A (blue, black, red and dark cyan dots) have different SFRs, even though they started with the same gas surface density profile. As the galaxies evolve (t =616 and t = 927 Myr) the values of the average SFR start to converge to a common value.



(a) Kennicutt-Schmidt law



(b) Silk law

Figure 4.26: Evolution of star formation laws with time. From left to right the plots represent the star formation laws at t =389, 616 and 927 Myr. Each point represent a model. The colors are as follow: F blue, AI dark cyan, AII black, AIII red, BI yellow, BII cyan, BIII magenta, BIV green and BV orange.

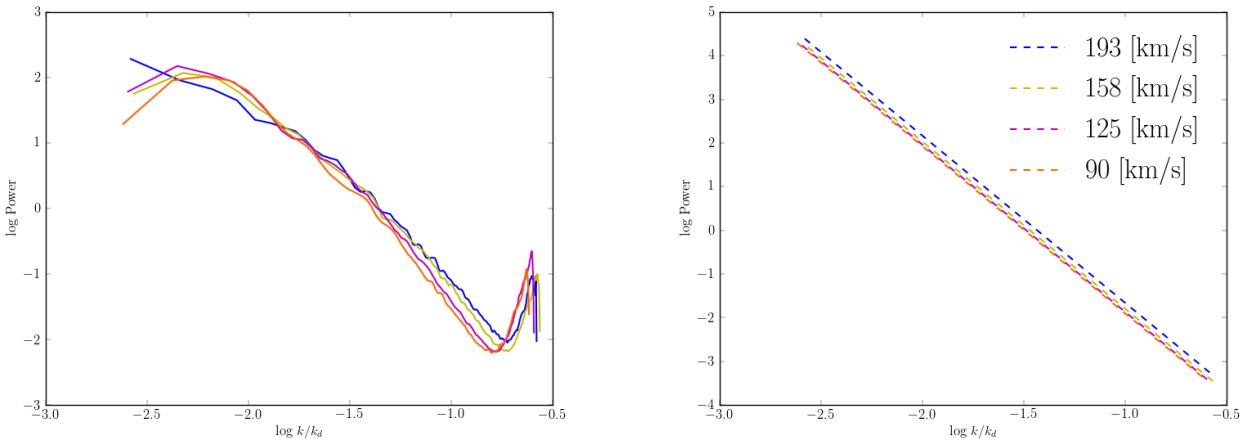
4.5 Massive Black Hole accretion rate

As introduced in §1.2, we will focus our study of Massive Black Hole growth in studying the relation between their radial velocity power spectrum and the rotational velocity of the disk. We already showed the power spectrum for model F at the beginning of this Chapter (Figure 4.17), where we found that they are well-fitted by a power-law with index ∼-4. Now to derive a generic relation of the type

$$v_r^{turb} \sim v_{rot} \left(\frac{\lambda}{R_d}\right)^{-\beta} = v_{rot} \left(\frac{k}{k_d}\right)^{\beta}, \tag{4.4}$$

where $k_d = 2\pi/R_d$, we will also need to study a larger sample of simulated galaxies. Taking in mind that Equation 4.4 involves rotation curves, we will concentrate our study in Group B which is the most dynamically diverse (see Figure 4.11).

First we select models F, BI, BIII and BV to plot them all together in a radial velocity power spectrum as a function of k normalized by k_d, as seen in Figure 4.27a. Our definition for R_d is the radius enclosing 95% of total stars (§4.1.2, Figure 4.12). We then proceed to fit a linear curve for the small scales interval $-1.3 < \log k/k_d < -0.9$ obtaining slopes in the range $\beta \sim -3.3$ to $\beta \sim -4.1$. As we are more interested in the zero points of the fit rather than its slope, we adopted the average value of $\beta = -3.83$ for all Group B simulations. With this value, we refitted our simulations getting what is shown in Figure 4.27b. In parallel we also calculated the average value of v_{rot} for each simulation and we show them in the top right corner of the same Figure. There is a clear relation between the zero point and the rotational velocity in the plot, obtaining a greater zero point for a greater circular velocity. This reinforces a relation of the form shown in Equation 4.4. Our findings are similar to the relation found between the area below the power spectrum and the Mach number deduced by Ballesteros-Paredes et al. (2006).



(a) Radial velocity power spectrum

(b) Linear fit of power spectrum with $\beta = -3.83$

Figure 4.27: Radial velocity power spectrum and its best linear fit for models F (blue), BI (yellow), BIII (magenta) and BV (orange). In the upper right corner of plot (b) it is also shown the average rotational velocity of each model.

Once we have velidated Equation 4.4, making a few assumptions we can recover the $M_{BH} - \sigma$ and $M_{BH} - M_{bulge}$ relations. Following Escala (2006), first we will start assuming an α-disk, which is a good approximation for our high turbulent disks. Besides that, as the disk is turbulence-dominated, the more relevant quantity dynamically speaking is v^{turb}, so the disk scale-height will be parameterized as a function of this velocity. We then obtain $\nu = \alpha\Omega h^2 = \alpha(v^{turb})^2/\Omega$ for the viscosity, where we can assume

an isotropic velocity and replace v^{turb} by $\sqrt{3}v_r^{turb}$ as we demonstrated in §4.3.2. We can then write the mass accretion rate as:

$$\dot{M} = \frac{6\sqrt{6}\alpha(v_r^{turb})^3}{GQ}, \tag{4.5}$$

where $Q = v^{turb}\Omega/\pi G\Sigma_{gas}$ is the "turbulent" Toomre's parameter.

Replacing our result of Equation 4.4 with $\beta = -3.83$ in Equation 4.5, we obtain:

$$\dot{M} \sim \frac{6\sqrt{6}\alpha}{GQ}\left(\frac{\lambda}{R_d}\right)^{11.5} v_{rot}^3. \tag{4.6}$$

To follow our reasoning, we need to determine the mass of the black hole, which will be given by $M_{BH} = \eta\dot{M}t_{gas}$ where η is the fraction of gas within the radius of influence that finally ends inside the Schwarzschild radius of the black hole, and t_{gas} is the gas lifetime. Remembering Equation 1.4, we have already argued that the characteristic timescale of the gas is an orbital time, as Silk law pointed out. Thus, we can express $t_{gas} = t_{orb} = \xi R_e/v_{rot}$ with R_e the effective radius of the host spheroid. Replacing the values of $\dot{M}$ and t_{gas} we get:

$$M_{BH} \sim \frac{6\sqrt{6}\alpha}{Q}\left(\frac{\lambda}{R_d}\right)^{11.5} \frac{R_e v_{rot}^2}{G} \tag{4.7}$$

Considering that $M_{dyn} = R_e v_{rot}^2/G$ it is straightforward to deduce the $M_{BH} - M_{bulge}$ relation. In order to obtain the $M_{BH} - \sigma$ relation, it is enough to recall the Faber-Jackson relation which says that $R_e \propto \sigma^2$. In the case of our disks, we have that the radial velocity is a good estimator of the dispersion of the disk σ, and hence we can replace it using $v_{rot} \approx \sigma$, obtaining finally $M_{BH} \sim \sigma^4$

Chapter 5

Conclusion and outlook

Using the Adaptive Mesh Refinement code Enzo, we performed nine simulations of local galaxies to study the factors that alter the Kennicutt-Schmidt and Silk relations. Our models are three-components galaxies, which are composed by gas, stars and dark matter. We use an exponential profile in the radial direction in cylindrical coordinates, coupled with a sech^2 in the vertical direction to model the gas disk. The other two components are modeled as time-independent external potentials by a Miyamoto-Nagai and a Navarro-Frenk-White profile respectively. Among the physical processes included it is worth to mention that we introduce radiative cooling via cooling curves down to $T = 300$ K, we allow stars to form in cells that fulfill a series of requirements, and that these stars also inject energy to the surrounding material through stellar feedback. The simulations were performed in a box of 666 h^{-1} kpc using periodic boundary conditions and a maximum level of refinement of seven additional subgrids. This choice allowed us to reach a resolution of $\sim$40 pc, where we used refinement by baryon mass and refinement to resolve Jeans length for at least four cells. We let them evolve during ~ 1 Gyr in a ΛCDM universe, to ensure that each model evolves at least two orbits at a radius that encloses the region where stars are formed.

Our models were divided in two groups. Group A is characterized by different initial rotational mass scale (M_{rot}) profiles and the same gas surface density initial profile, in contrast to Group B where all galaxies have a similar M_{rot} profile while varying their Σ_{gas} profiles. In both groups we included our fiducial simulation as a reference for comparison. In each model we obtained a similar evolution, with their inner regions fragmenting first and their outer regions later. In the same way, stars start being created at times ~ 50 Myr in the galaxies with lower t_{orb}. What we get once our simulations have evolved is a highly turbulent Interstellar Medium, characterized by a lognormal Probability Density Function at high densities and a power spectrum steeper than Kolmogorov and Burger models.

In order to study star formation, we have plotted KS and Silk laws at different stages in the evolution of our models, using both instantaneous and average star formation rates. In our best fit models for the star formation laws, we obtained slopes ranging from 0.8 to 2.1 for both relations. The two most important factors affecting the slope of the relations are:

a) Radius: We used the radius enclosing 90, 95 and 100 per cent of stars to calculate surface densities. For greater radius we obtained lower slopes. Our explanation is that on one hand, increasing radius does not increase the amount of gas considerably thus obtaining a lower gas surface density. On the other hand, in the case of star formation rate it increases proportionally to the area, thus keeping the SFR surface density with no great variations. Adding both effects, we obtain that an increment in the radius means a decrement of Σ_{gas} while keeping Σ_{SFR} almost constant, reducing therefore the slope.

b) Star Formation Rate: Other factor that greatly influences both laws is the interval of time we consider to measure SFR, in what we have called the instantaneous and average SFR. The former introduces more scattering in the relations as the SFR curves have large oscillations, while the latter converges

to a fixed value as the galaxies evolves, and hence decreasing scattering. We have also examined the differences between both SFRs at the beginning and once the simulations have reach a stationary condition, which is translated in a lower slope for the case of using the average star formation rate in KS and Silk laws. This leads us to the conclusion that on one hand the instantaneous SFR is a more representative value to study the evolution of a single galaxy, but on the other hand the average SFR is a more representative value of the galaxy as a global quantity, specially useful when comparing the different models between them.

The dependence on the rotational mass M_{rot} is much clear at the early stages of the simulations, influencing strongly the time at which simulations starts forming stars. This is also translated in different SFRs for simulations even when they started with the same amount of gas. We propose as a future work to investigate if this effect can be reproduced in longer time scales. We think that varying M_{rot} profiles much more than we did in this study will provoke differences greater than $\sim 2 \times 10^8$ yr in the time when first stars are formed as seen in Group A. These time delays could have a greater impact on the global SFR of the galaxy.

Our simulations also allowed us to study Massive Black Hole growth, through the analysis of the ISM. Using power spectrum plots, we have shown that the radial velocity power spectrum of the galaxy is clearly related to the circular velocity. We have also demonstrated that the velocity in our simulations is nearly isotropic, so we can write turbulent velocity as a function of one of its components. This relation between turbulent, radial and circular velocity will ultimately lead us to the relation between the mass of a black hole and the velocity dispersion of its host galaxy, explaining the origin of the so called $M_{BH} - \sigma$ relation.

Bibliography

Athanassoula, E., Nov. 1992. The existence and shapes of dust lanes in galactic bars. MNRAS259, 345–364.

Ballesteros-Paredes, J., Gazol, A., Kim, J., Klessen, R. S., Jappsen, A.-K., Tejero, E., Jan. 2006. The Mass Spectra of Cores in Turbulent Molecular Clouds and Implications for the Initial Mass Function. ApJ637, 384–391.

Berger, M. J., Colella, P., May 1989. Local adaptive mesh refinement for shock hydrodynamics. Journal of Computational Physics 82, 64–84.

Berger, M. J., Oliger, J., Mar. 1984. Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations. Journal of Computational Physics 53, 484.

Bigiel, F., Leroy, A., Walter, F., Brinks, E., de Blok, W. J. G., Madore, B., Thornley, M. D., Dec. 2008. The Star Formation Law in Nearby Galaxies on Sub-Kpc Scales. AJ136, 2846–2871.

Binney, J., Tremaine, S., 2008. Galactic Dynamics: Second Edition. Princeton University Press.

Bloemen, J. B. G. M., Strong, A. W., Mayer-Hasselwander, H. A., Blitz, L., Cohen, R. S., Dame, T. M., Grabelsky, D. A., Thaddeus, P., Hermsen, W., Lebrun, F., Jan. 1986. The radial distribution of galactic gamma rays. III - The distribution of cosmic rays in the Galaxy and the CO-H2 calibration. A&A154, 25–41.

Boquien, M., Lisenfeld, U., Duc, P.-A., Braine, J., Bournaud, F., Brinks, E., Charmandaris, V., Sep. 2011. Studying the spatially resolved Schmidt-Kennicutt law in interacting galaxies: the case of Arp 158. A&A533, A19.

Bouché, N., Cresci, G., Davies, R., Eisenhauer, F., Förster Schreiber, N. M., Genzel, R., Gillessen, S., Lehnert, M., Lutz, D., Nesvadba, N., Shapiro, K. L., Sternberg, A., Tacconi, L. J., Verma, A., Cimatti, A., Daddi, E., Renzini, A., Erb, D. K., Shapley, A., Steidel, C. C., Dec. 2007. Dynamical Properties of z ˜ 2 Star-forming Galaxies and a Universal Star Formation Relation. ApJ671, 303–309.

Bryan, G. L., Norman, M. L., 1997. Simulating X-Ray Clusters with Adaptive Mesh Refinement. In: D. A. Clarke & M. J. West (Ed.), Computational Astrophysics; 12th Kingston Meeting on Theoretical Astrophysics. Vol. 123 of Astronomical Society of the Pacific Conference Series. p. 363.

Burkert, A., Silk, J., Jun. 2001. Star Formation-Regulated Growth of Black Holes in Protogalactic Spheroids. ApJ554, L151–L154.

Cen, R., Ostriker, J. P., Nov. 1992. Galaxy formation and physical bias. ApJ399, L113–L116.

Ceverino, D., Klypin, A., Apr. 2009. The Role of Stellar Feedback in the Formation of Galaxies. ApJ695, 292–309.

Chang, P., Murray-Clay, R., Chiang, E., Quataert, E., Oct. 2007. The Origin of the Young Stars in the Nucleus of M31. ApJ668, 236–244.

Colella, P., Woodward, P. R., Sep. 1984. The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations. Journal of Computational Physics 54, 174–201.

Daddi, E., Elbaz, D., Walter, F., Bournaud, F., Salmi, F., Carilli, C., Dannerbauer, H., Dickinson, M., Monaco, P., Riechers, D., May 2010. Different Star Formation Laws for Disks Versus Starbursts at Low and High Redshifts. ApJ714, L118–L122.

Dedner, A., Kemm, F., Kröner, D., Munz, C.-D., Schnitzer, T., Wesenberg, M., Jan. 2002. Hyperbolic Divergence Cleaning for the MHD Equations. Journal of Computational Physics 175, 645–673.

Di Matteo, T., Colberg, J., Springel, V., Hernquist, L., Sijacki, D., Mar. 2008. Direct Cosmological Simulations of the Growth of Black Holes and Galaxies. ApJ676, 33–53.

Elmegreen, B. G., Mar. 1989. A pressure and metallicity dependence for molecular cloud correlations and the calibration of mass. ApJ338, 178–196.

Elmegreen, B. G., May 1997. Theory of Starbursts in Nuclear Rings. In: Franco, J., Terlevich, R., Serrano, A. (Eds.), Revista Mexicana de Astronomia y Astrofisica Conference Series. Vol. 6 of Revista Mexicana de Astronomia y Astrofisica, vol. 27. p. 165.

Elmegreen, B. G., Sep. 2002. Star Formation from Galaxies to Globules. ApJ577, 206–220.

Escala, A., Sep. 2006. On the Fueling of Massive Black Holes and the Properties of Their Host Spheroids. ApJ648, L13–L16.

Escala, A., Dec. 2007. Toward a Comprehensive Fueling-controlled Theory of the Growth of Massive Black Holes and Host Spheroids. ApJ671, 1264–1271.

Escala, A., Jul. 2011. A Law for Star Formation in Galaxies. ApJ735, 56.

Escala, A., Larson, R. B., Sep. 2008. Stability of Galactic Gas Disks and the Formation of Massive Clusters. ApJ685, L31–L34.

Faber, S. M., Jackson, R. E., Mar. 1976. Velocity dispersions and mass-to-light ratios for elliptical galaxies. ApJ204, 668–683.

Ferrarese, L., Merritt, D., Aug. 2000. A Fundamental Relation between Supermassive Black Holes and Their Host Galaxies. ApJ539, L9–L12.

Gebhardt, K., Bender, R., Bower, G., Dressler, A., Faber, S. M., Filippenko, A. V., Green, R., Grillmair, C., Ho, L. C., Kormendy, J., Lauer, T. R., Magorrian, J., Pinkney, J., Richstone, D., Tremaine, S., Aug. 2000. A Relationship between Nuclear Black Hole Mass and Galaxy Velocity Dispersion. ApJ539, L13–L16.

Genzel, R., Tacconi, L. J., Gracia-Carpio, J., Sternberg, A., Cooper, M. C., Shapiro, K., Bolatto, A., Bouché, N., Bournaud, F., Burkert, A., Combes, F., Comerford, J., Cox, P., Davis, M., Schreiber, N. M. F., Garcia-Burillo, S., Lutz, D., Naab, T., Neri, R., Omont, A., Shapley, A., Weiner, B., Oct. 2010. A study of the gas-star formation relation over cosmic time. MNRAS407, 2091–2108.

Gingold, R. A., Monaghan, J. J., Nov. 1977. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. MNRAS181, 375–389.

Gnedin, N. Y., Kravtsov, A. V., May 2010. On the Kennicutt-Schmidt Relation of Low-Metallicity High-Redshift Galaxies. ApJ714, 287–295.

Goldreich, P., Lynden-Bell, D., 1965. I. Gravitational stability of uniformly rotating disks. MNRAS130, 97.

Häring, N., Rix, H.-W., Apr. 2004. On the Black Hole Mass-Bulge Mass Relation. ApJ604, L89–L92.

Kenney, J. D., 1987. Molecular gas in Virgo cluster spiral galaxies. Ph.D. thesis, Massachusetts Univ., Amherst.

Kennicutt, Jr., R. C., Sep. 1989. The star formation law in galactic disks. ApJ344, 685–703.

Kennicutt, Jr., R. C., May 1998. The Global Schmidt Law in Star-forming Galaxies. ApJ498, 541.

Kennicutt, Jr., R. C., Calzetti, D., Walter, F., Helou, G., Hollenbach, D. J., Armus, L., Bendo, G., Dale, D. A., Draine, B. T., Engelbracht, C. W., Gordon, K. D., Prescott, M. K. M., Regan, M. W., Thornley, M. D., Bot, C., Brinks, E., de Blok, E., de Mello, D., Meyer, M., Moustakas, J., Murphy, E. J., Sheth, K., Smith, J. D. T., Dec. 2007. Star Formation in NGC 5194 (M51a). II. The Spatially Resolved Star Formation Law. ApJ671, 333–348.

Kolmogorov, A., 1941. The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers. Akademiia Nauk SSSR Doklady 30, 301–305.

Kravtsov, A. V., Jun. 2003. On the Origin of the Global Schmidt Law of Star Formation. ApJ590, L1–L4.

Krumholz, M. R., McKee, C. F., Sep. 2005. A General Theory of Turbulence-regulated Star Formation, from Spirals to Ultraluminous Infrared Galaxies. ApJ630, 250–268.

Krumholz, M. R., McKee, C. F., Tumlinson, J., Jul. 2009. The Star Formation Law in Atomic and Molecular Gas. ApJ699, 850–856.

Kuzmin, G. G., 1956. ,. AZh33, 27.

Lapidus, A., Nov. 1967. A Detached Shock Calculation by Second-Order Finite Differences. Journal of Computational Physics 2, 154.

Leroy, A. K., Walter, F., Brinks, E., Bigiel, F., de Blok, W. J. G., Madore, B., Thornley, M. D., Dec. 2008. The Star Formation Efficiency in Nearby Galaxies: Measuring Where Gas Forms Stars Effectively. AJ136, 2782–2845.

Li, Y., Haiman, Z., Mac Low, M.-M., Jul. 2007. Correlations between Central Massive Objects and Their Host Galaxies: From Bulgeless Spirals to Ellipticals. ApJ663, 61–70.

Li, Y., Mac Low, M.-M., Klessen, R. S., Feb. 2005. Control of Star Formation in Galaxies by Gravitational Instability. ApJ620, L19–L22.

Lucy, L. B., Dec. 1977. A numerical approach to the testing of the fission hypothesis. AJ82, 1013–1024.

Marconi, A., Hunt, L. K., May 2003. The Relation between Black Hole Mass, Bulge Mass, and Near-Infrared Luminosity. ApJ589, L21–L24.

Martin, C. L., Kennicutt, Jr., R. C., Jul. 2001. Star Formation Thresholds in Galactic Disks. ApJ555, 301–321.

Miyamoto, M., Nagai, R., 1975. Three-dimensional models for the distribution of mass in galaxies. PASJ27, 533–543.

Monaghan, J. J., 1992. Smoothed particle hydrodynamics. ARA&A30, 543–574.

Narayanan, D., Krumholz, M. R., Ostriker, E. C., Hernquist, L., Apr. 2012. A general model for the CO-H_2 conversion factor in galaxies with applications to the star formation law. MNRAS421, 3127–3146.

Navarro, J. F., Frenk, C. S., White, S. D. M., Dec. 1997. A Universal Density Profile from Hierarchical Clustering. ApJ490, 493.

Norman, M. L., Bryan, G. L., 1999. Cosmological Adaptive Mesh Refinement^{CD}. In: S. M. Miyama, K. Tomisaka, & T. Hanawa (Ed.), Numerical Astrophysics. Vol. 240 of Astrophysics and Space Science Library. p. 19.

Norman, M. L., Winkler, K.-H. A., 1986. 2-D Eulerian Hydrodynamics with Fluid Interfaces, Self-Gravity and Rotation. In: Winkler, K.-H. A., Norman, M. L. (Eds.), Astrophysical Radiation Hydrodynamics. p. 187.

O'Shea, B. W., Bryan, G., Bordner, J., Norman, M. L., Abel, T., Harkness, R., Kritsuk, A., Mar. 2004. Introducing Enzo, an AMR Cosmology Application. ArXiv Astrophysics e-prints.

Passot, T., Vázquez-Semadeni, E., Oct. 1998. Density probability distribution in one-dimensional polytropic gas dynamics. Phys. Rev. E58, 4501–4510.

Plummer, H. C., Mar. 1911. On the problem of distribution in globular star clusters. MNRAS71, 460–470.

Robertson, B., Hernquist, L., Cox, T. J., Di Matteo, T., Hopkins, P. F., Martini, P., Springel, V., Apr. 2006. The Evolution of the M_{BH}-σ Relation. ApJ641, 90–102.

Robertson, B. E., Kravtsov, A. V., Jun. 2008. Molecular Hydrogen and Global Star Formation Relations in Galaxies. ApJ680, 1083–1111.

Rosen, A., Bregman, J. N., Feb. 1995. Global Models of the Interstellar Medium in Disk Galaxies. ApJ440, 634.

Sarazin, C. L., White, III, R. E., Sep. 1987. Steady state cooling flow models for normal elliptical galaxies. ApJ320, 32–48.

Schmidt, M., Mar. 1959. The Rate of Star Formation. ApJ129, 243.

Schmidt, M., Apr. 1963. The Rate of Star Formation. II. The Rate of Formation of Stars of Different Mass. ApJ137, 758.

Shlosman, I., Begelman, M. C., Frank, J., Jun. 1990. The fuelling of active galactic nuclei. Nature345, 679–686.

Shlosman, I., Frank, J., Begelman, M. C., Mar. 1989. Bars within bars - A mechanism for fuelling active galactic nuclei. Nature338, 45–47.

Silk, J., May 1997. Feedback, Disk Self-Regulation, and Galaxy Formation. ApJ481, 703.

Slyz, A. D., Devriendt, J. E. G., Bryan, G., Silk, J., Jan. 2005. Towards simulating star formation in the interstellar medium. MNRAS356, 737–752.

Somerville, R. S., Hopkins, P. F., Cox, T. J., Robertson, B. E., Hernquist, L., Dec. 2008. A semi-analytic model for the co-evolution of galaxies, black holes and active galactic nuclei. MNRAS391, 481–506.

Springel, V., Sep. 2010. Smoothed Particle Hydrodynamics in Astrophysics. ARA&A48, 391–430.

Springel, V., Hernquist, L., Feb. 2003. Cosmological smoothed particle hydrodynamics simulations: a hybrid multiphase model for star formation. MNRAS339, 289–311.

Stone, J. M., Norman, M. L., Jun. 1992. ZEUS-2D: A radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. I - The hydrodynamic algorithms and tests. ApJS80, 753–790.

Tan, J. C., Jun. 2000. Star Formation Rates in Disk Galaxies and Circumnuclear Starbursts from Cloud Collisions. ApJ536, 173–184.

Tasker, E. J., Bryan, G. L., Apr. 2006. Simulating Star Formation and Feedback in Galactic Disk Models. ApJ641, 878–890.

Tasker, E. J., Bryan, G. L., Feb. 2008. The Effect of the Interstellar Model on Star Formation Properties in Galactic Disks. ApJ673, 810–831.

Teyssier, R., Chapon, D., Bournaud, F., Sep. 2010. The Driving Mechanism of Starbursts in Galaxy Mergers. ApJ720, L149–L154.

Toomre, A., Aug. 1963. On the Distribution of Matter Within Highly Flattened Galaxies. ApJ138, 385.

Toomre, A., May 1964. On the gravitational stability of a disk of stars. ApJ139, 1217–1238.

Tremaine, S., Gebhardt, K., Bender, R., Bower, G., Dressler, A., Faber, S. M., Filippenko, A. V., Green, R., Grillmair, C., Ho, L. C., Kormendy, J., Lauer, T. R., Magorrian, J., Pinkney, J., Richstone, D., Aug. 2002. The Slope of the Black Hole Mass versus Velocity Dispersion Correlation. ApJ574, 740–753.

Truelove, J. K., Klein, R. I., McKee, C. F., Holliman, II, J. H., Howell, L. H., Greenough, J. A., Nov. 1997. The Jeans Condition: A New Constraint on Spatial Resolution in Simulations of Isothermal Self-gravitational Hydrodynamics. ApJ489, L179.

Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T., Norman, M. L., Jan. 2011. yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data. ApJS192, 9.

van Leer, B., Mar. 1977. Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection. Journal of Computational Physics 23, 276.

Vázquez-Semadeni, E., Gazol, A., Scalo, J., Sep. 2000. Is Thermal Instability Significant in Turbulent Galactic Gas? ApJ540, 271–285.

Vonneumann, J., Richtmyer, R. D., Mar. 1950. A Method for the Numerical Calculation of Hydrodynamic Shocks. Journal of Applied Physics 21, 232–237.

Wada, K., Norman, C. A., Jan. 2001. Numerical Models of the Multiphase Interstellar Matter with Stellar Energy Feedback on a Galactic Scale. ApJ547, 172–186.

Wada, K., Norman, C. A., May 2007. Density Structure of the Interstellar Medium and the Star Formation Rate in Galactic Disks. ApJ660, 276–287.

Wang, H.-H., Klessen, R. S., Dullemond, C. P., van den Bosch, F. C., Fuchs, B., Sep. 2010. Equilibrium initialization and stability of three-dimensional gas discs. MNRAS407, 705–720.

Wang, P., Abel, T., May 2009. Magnetohydrodynamic Simulations of Disk Galaxy Formation: The Magnetization of the Cold and Warm Medium. ApJ696, 96–109.

Wong, T., Blitz, L., Apr. 2002. The Relationship between Gas Content and Star Formation in Molecule-rich Spiral Galaxies. ApJ569, 157–183.

Appendix A

Individual profiles

A.1 Star formation rate fits

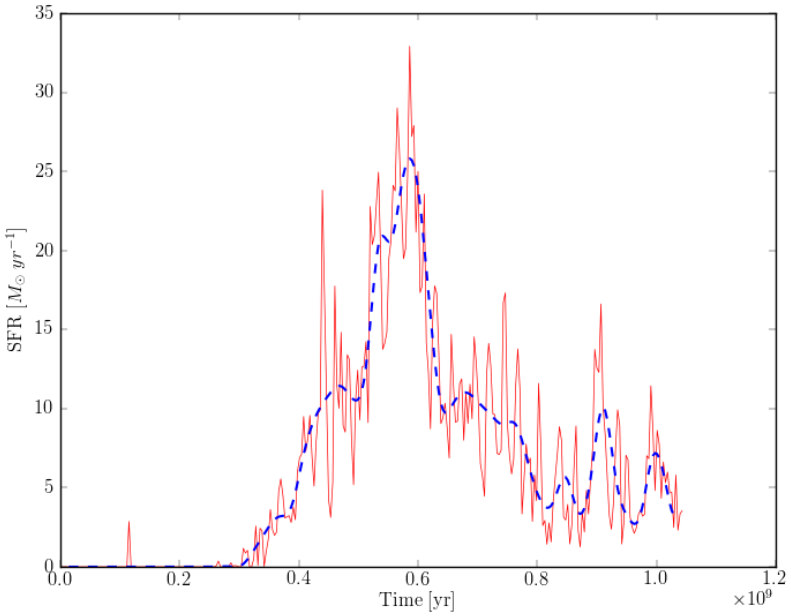


Figure A.1: Least-square fit of the Star Formation Rate for model F

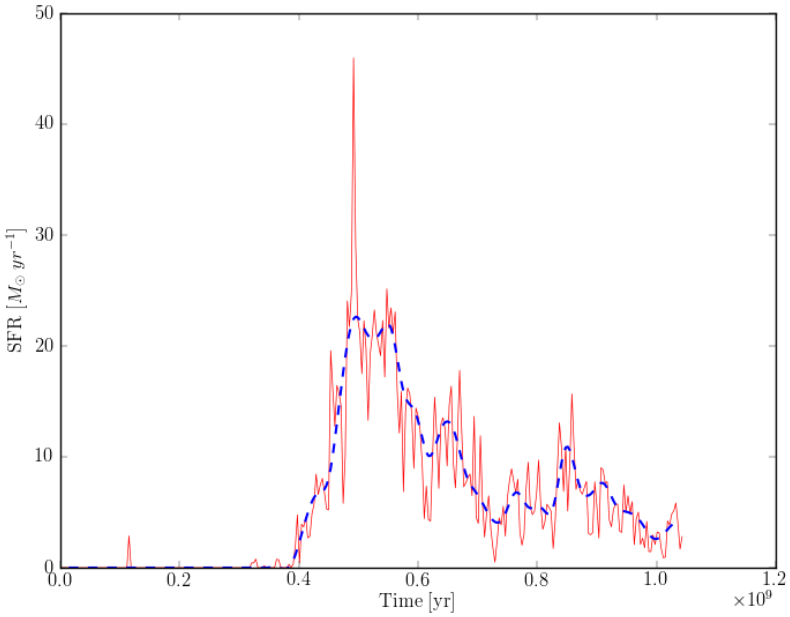


Figure A.1: Least-square fit of the Star Formation Rate for model AI

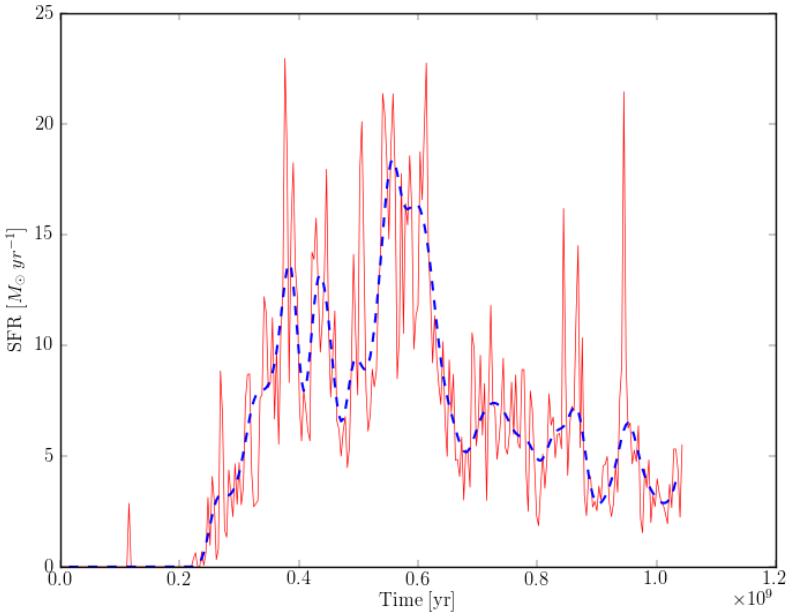


Figure A.1: Least-square fit of the Star Formation Rate for model AII

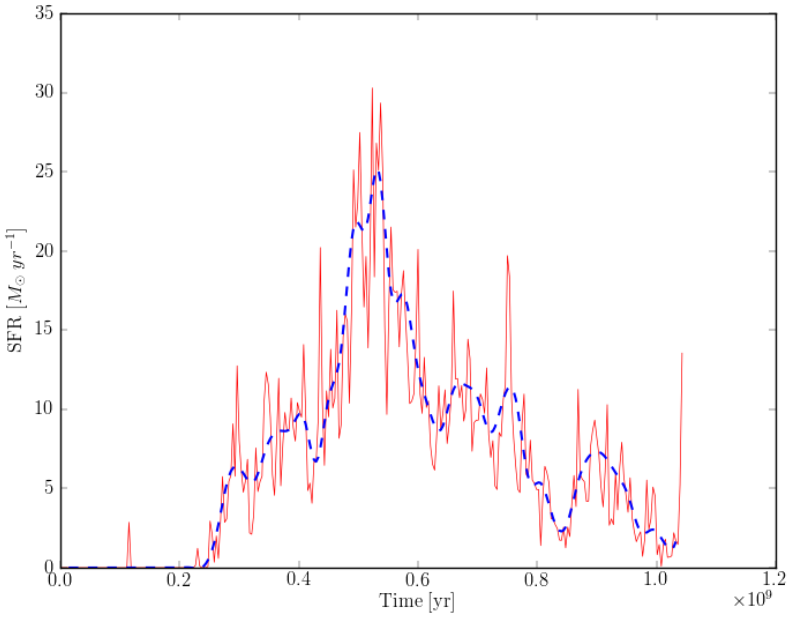


Figure A.1: Least-square fit of the Star Formation Rate for model AIII

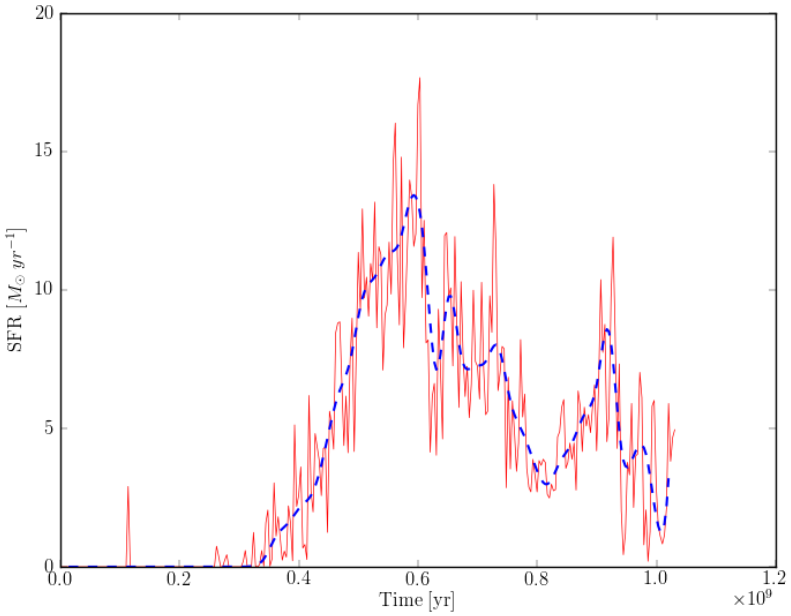


Figure A.1: Least-square fit of the Star Formation Rate for model BI

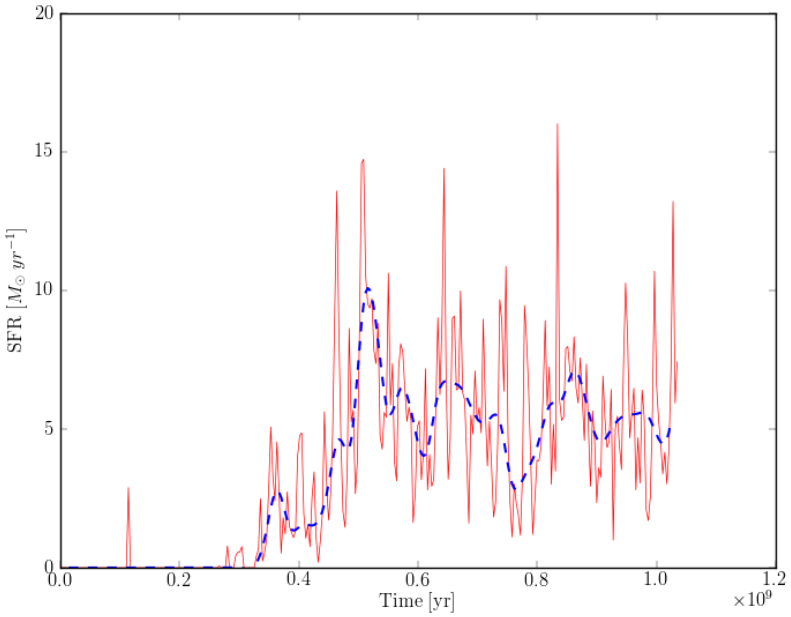


Figure A.1: Least-square fit of the Star Formation Rate for model BII

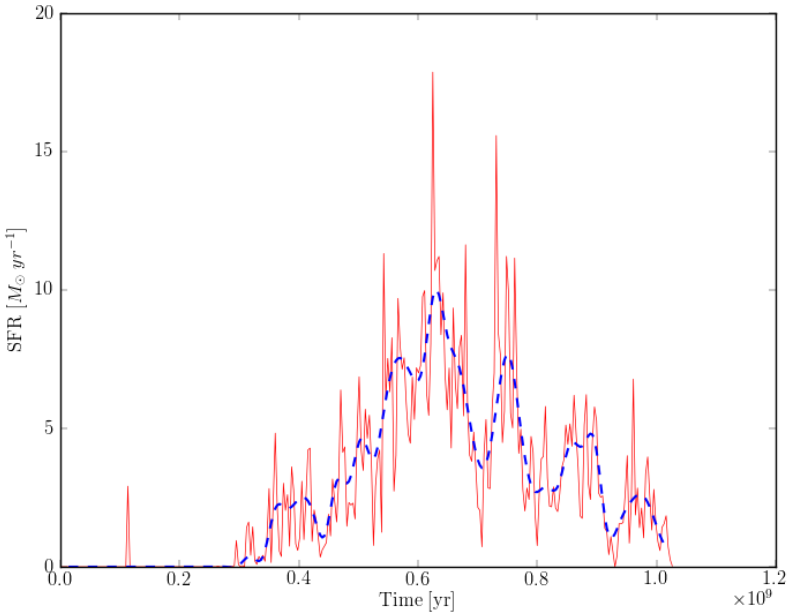


Figure A.1: Least-square fit of the Star Formation Rate for model BIII

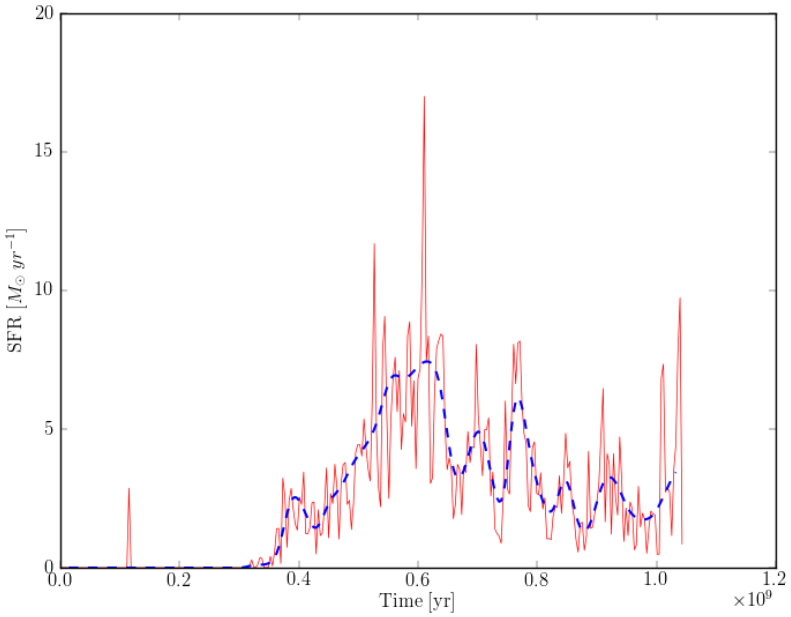


Figure A.1: Least-square fit of the Star Formation Rate for model BIV

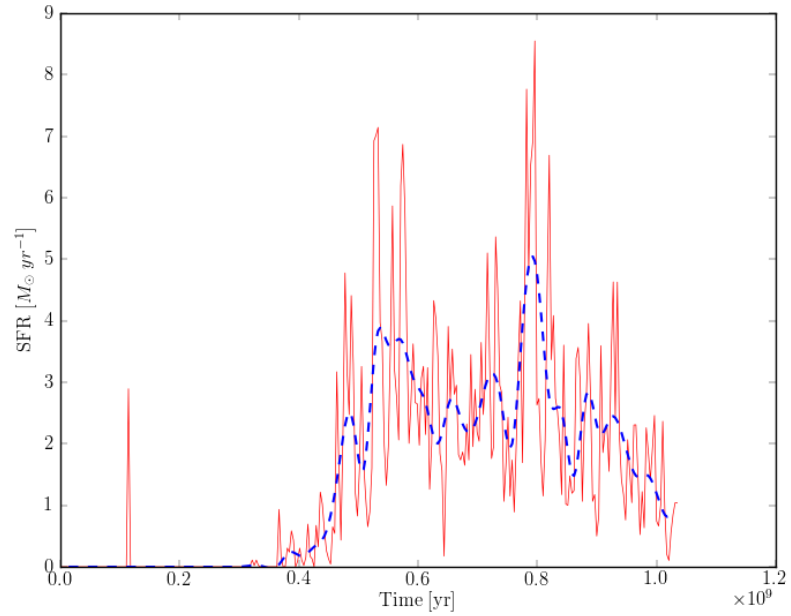


Figure A.1: Least-square fit of the Star Formation Rate for model BV

A.2 Probability Density Functions

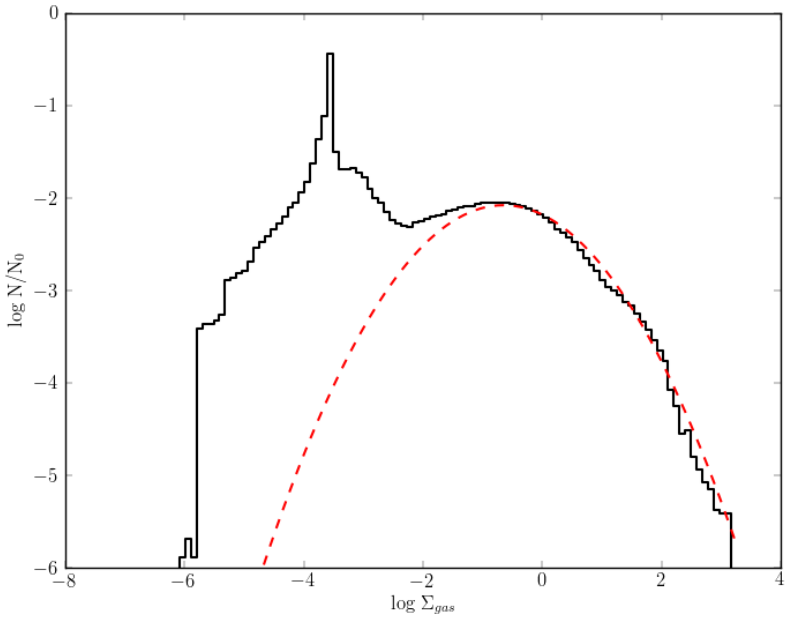


Figure A.2: Probability density function for model F

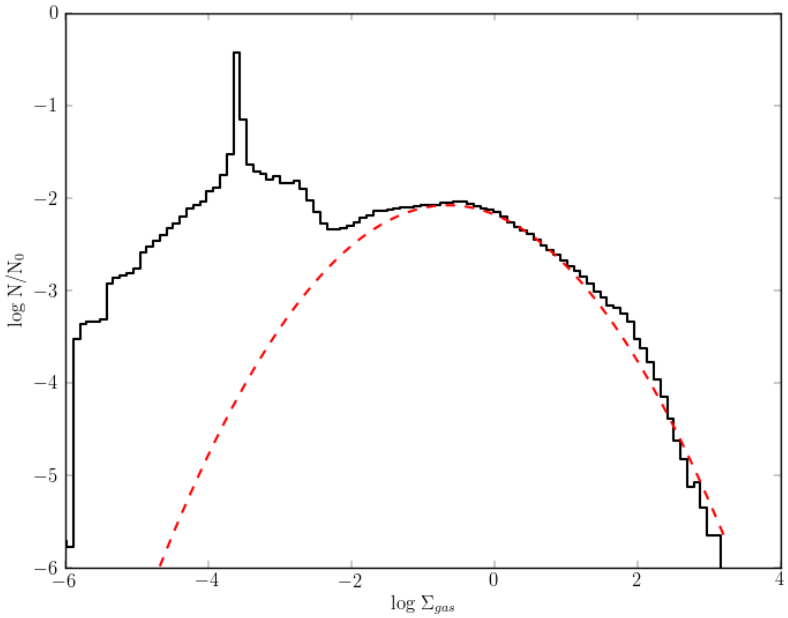


Figure A.2: Probability density function for model AI

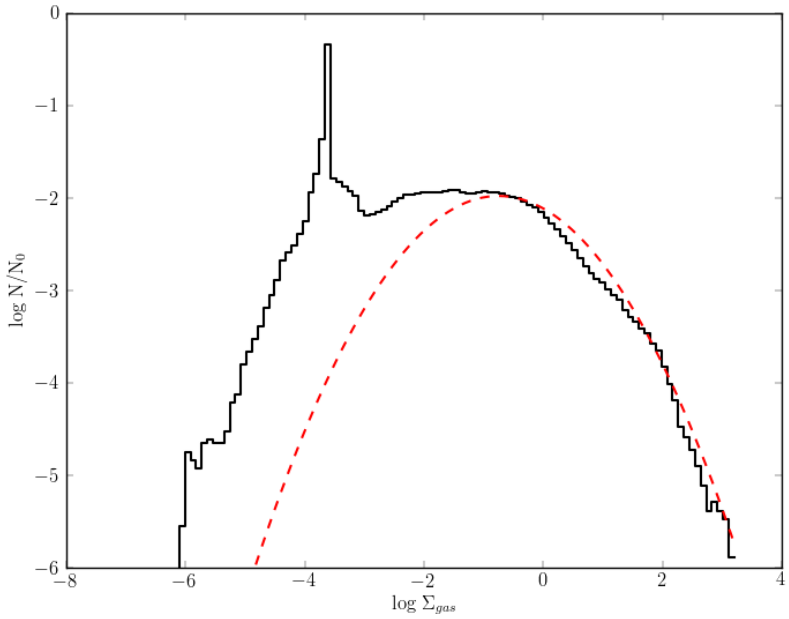


Figure A.2: Probability density function for model AII

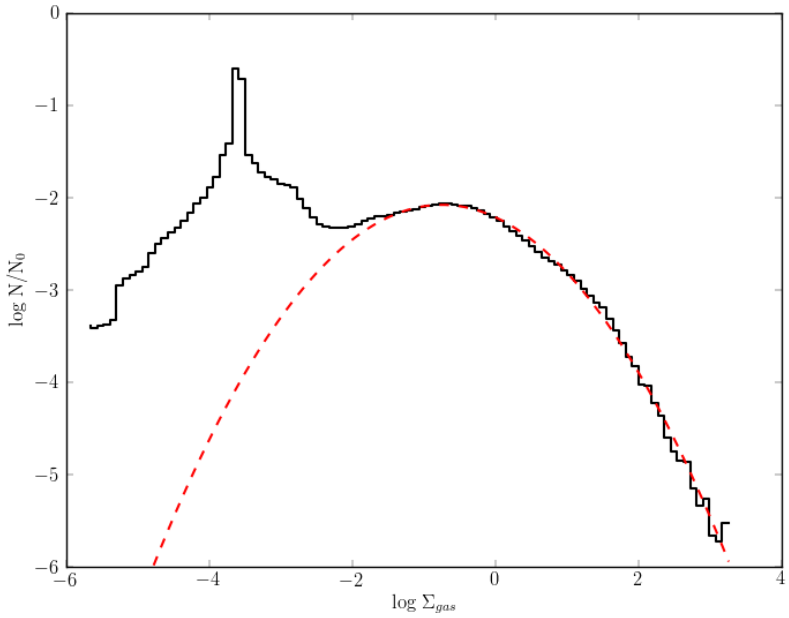


Figure A.2: Probability density function for model AIII

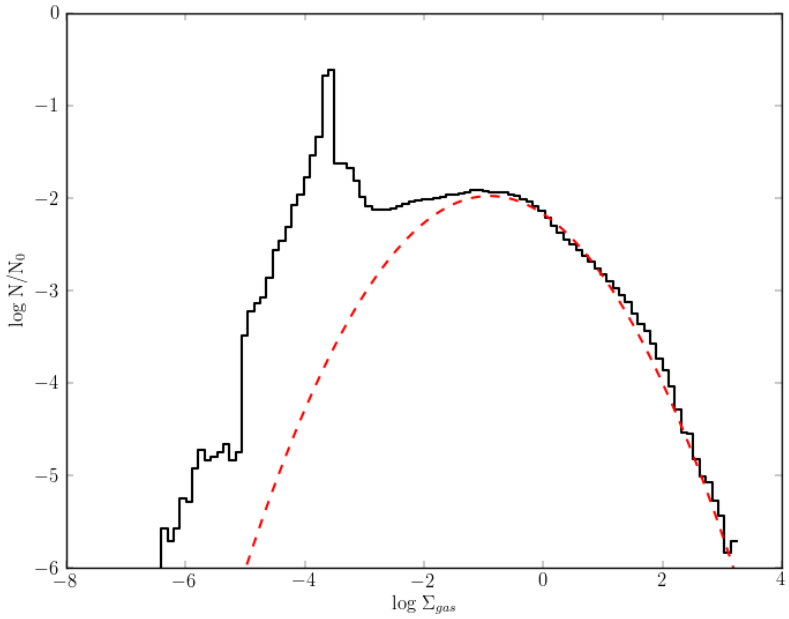


Figure A.2: Probability density function for model BI

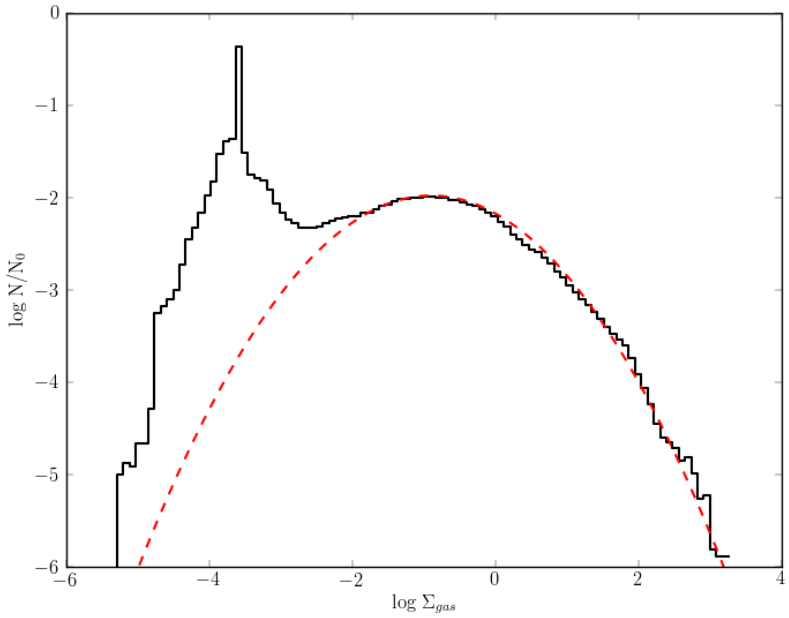


Figure A.2: Probability density function for model BII

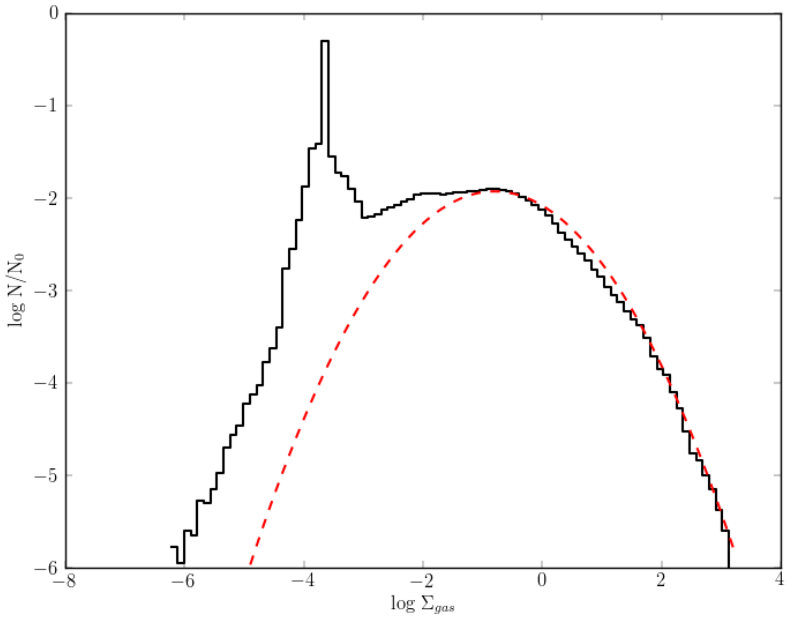


Figure A.2: Probability density function for model BIII

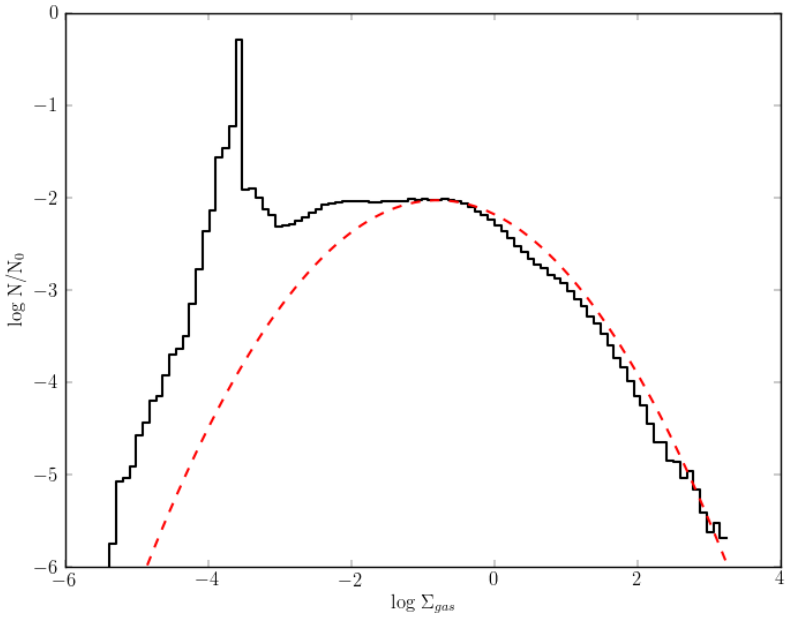


Figure A.2: Probability density function for model BIV

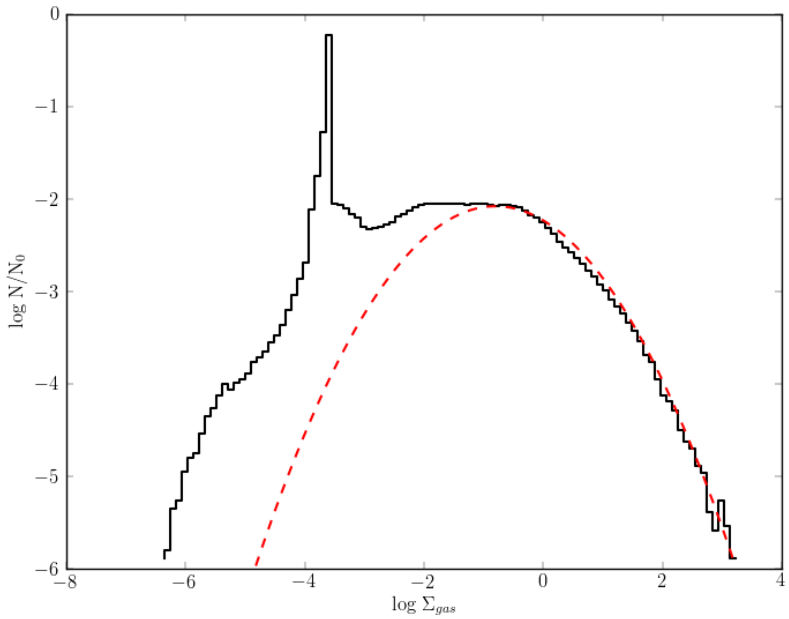


Figure A.2: Probability density function for model BV

A.3 Power spectrum

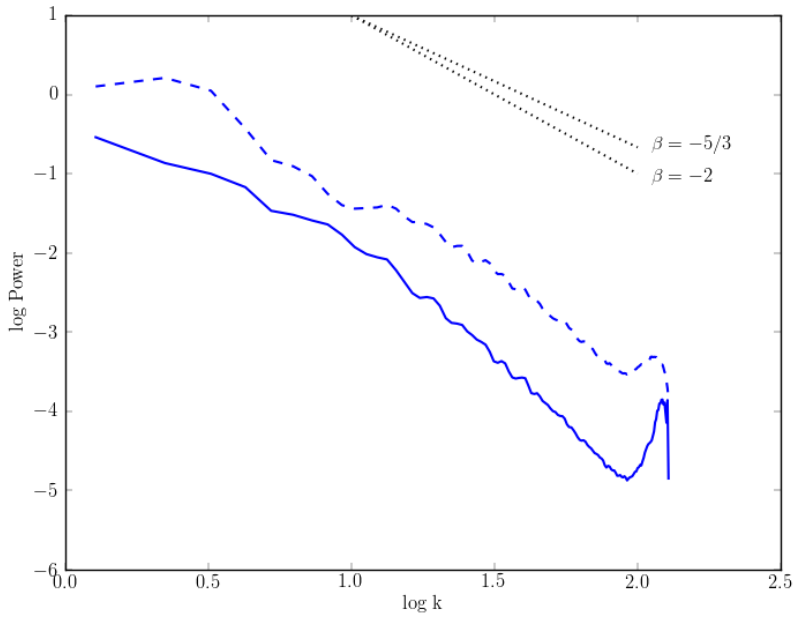


Figure A.3: Power spectrum for model F

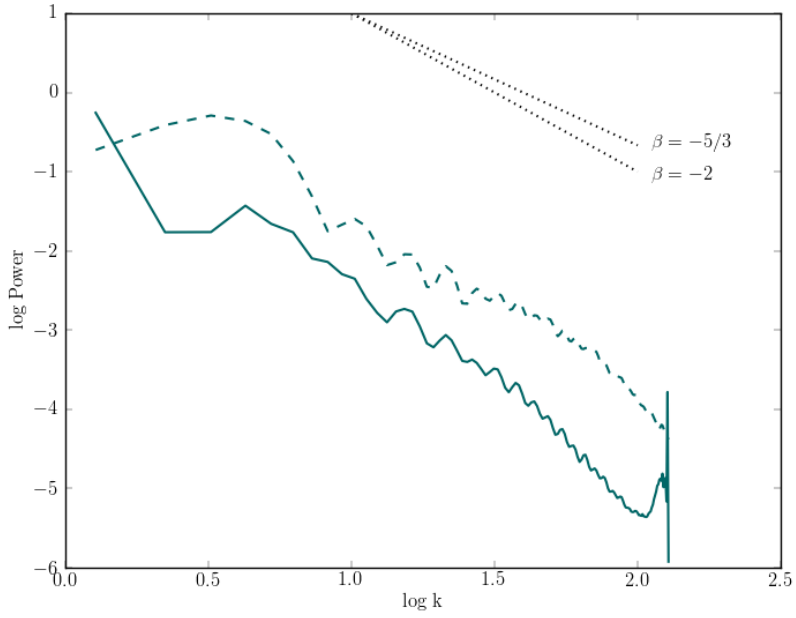


Figure A.3: Power spectrum for model AI

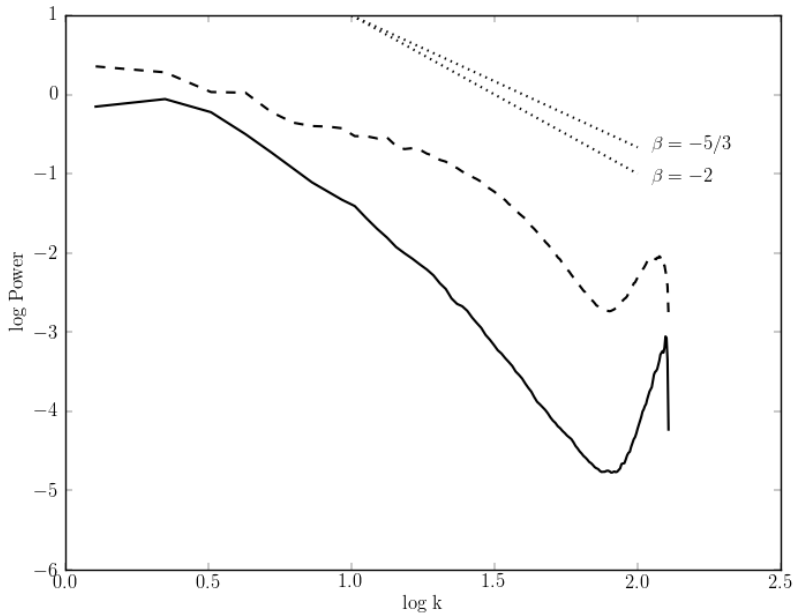


Figure A.3: Power spectrum for model AII

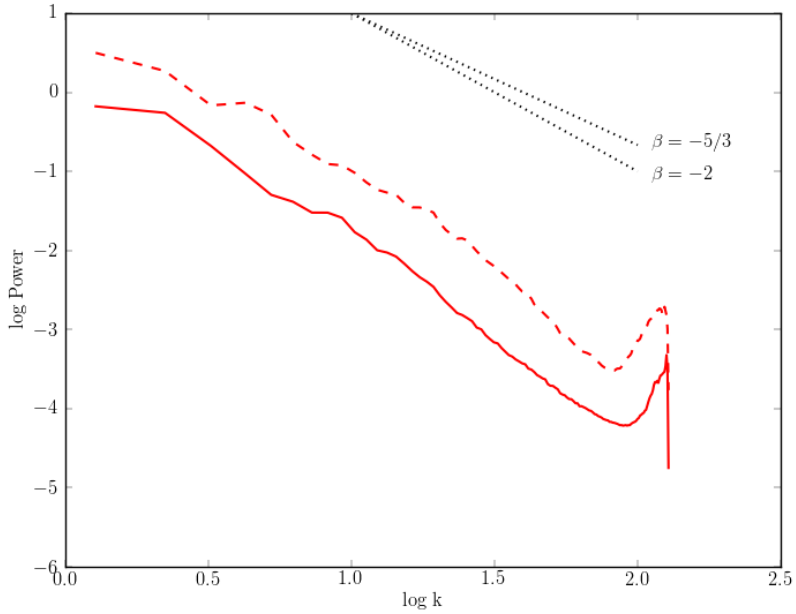


Figure A.3: Power spectrum for model AIII

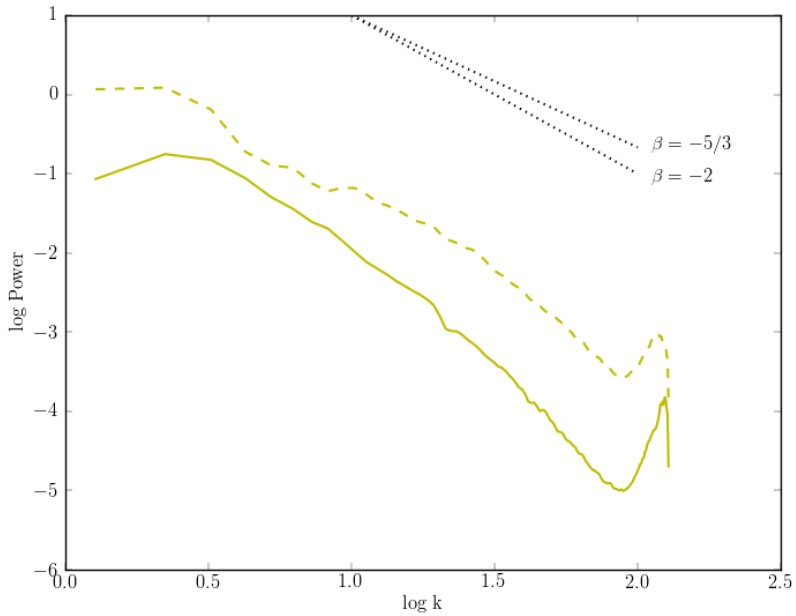


Figure A.3: Power spectrum for model BI

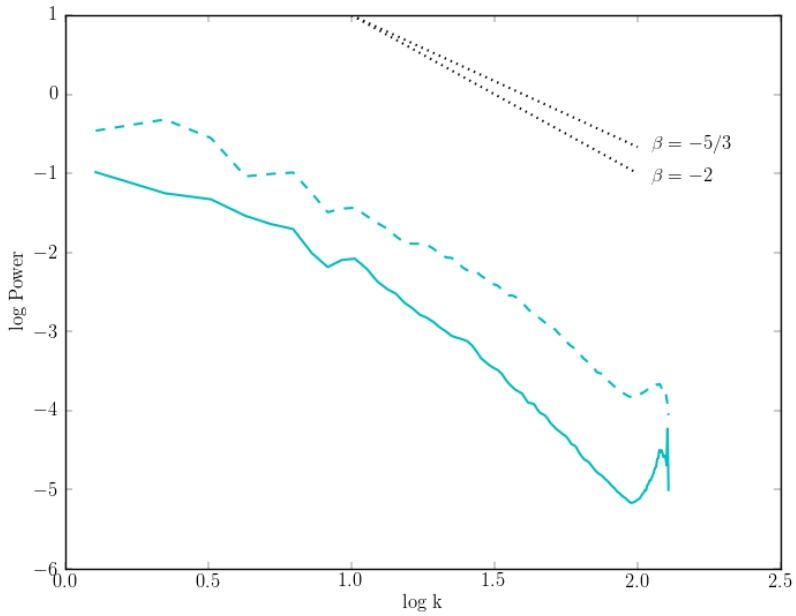


Figure A.3: Power spectrum for model BII

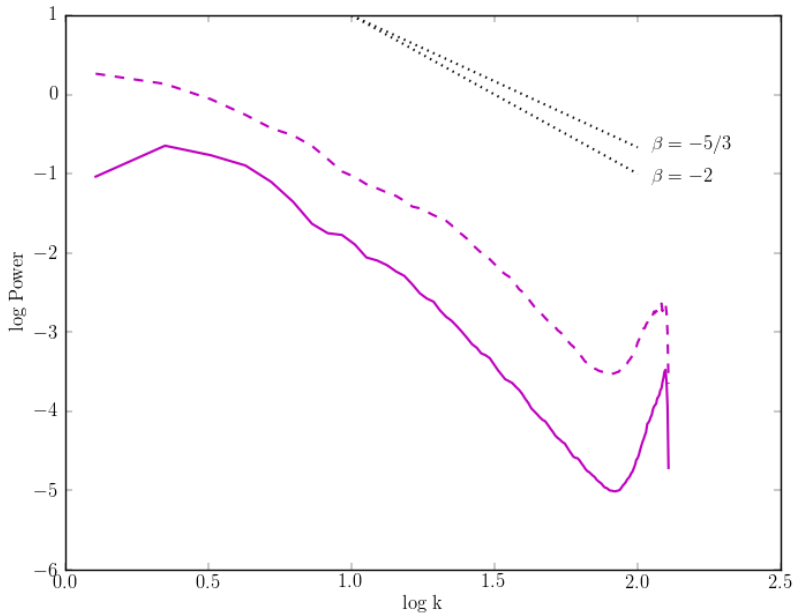


Figure A.3: Power spectrum for model BIII

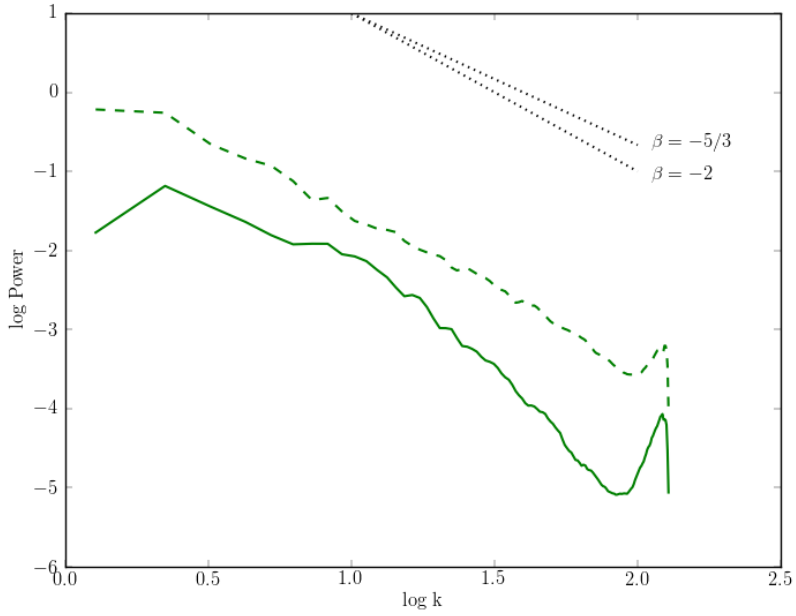


Figure A.3: Power spectrum for model BIV

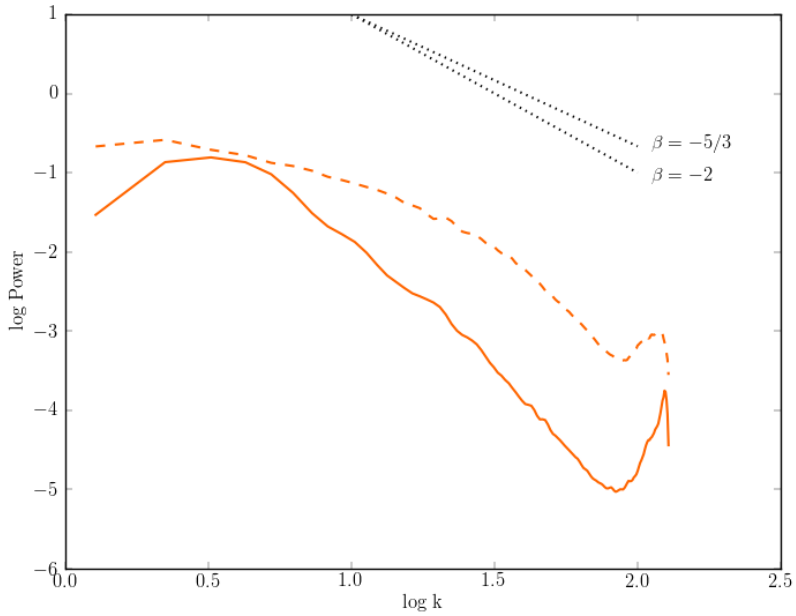


Figure A.3: Power spectrum for model BV

Appendix B

Codes

B.1 yt new fields

```
def _DiskSurfaceDensity(field, data):
    return data["Density"]*data["dz"]
def _ConvertDiskSurfaceDensity(data):
    return data.convert("cm")
add_field("DiskSurfaceDensity", function=_DiskSurfaceDensity,
           convert_function=_ConvertDiskSurfaceDensity,
           units=r"\rm{g}/\rm{cm^2}")

def _ParticleDiskRadius(field, data):
    center = data.get_field_parameter("center")
    DW = data.pf.domain_right_edge - data.pf.domain_left_edge
    radius = na.zeros(data["particle_position_x"].shape, dtype='float64')
    for i, ax in enumerate('xy'):
        r = na.abs(data["particle_position_%s" % ax] - center[i])
        radius += na.minimum(r, na.abs(DW[i]-r))**2.0
    na.sqrt(radius, radius)
    return radius

def _DiskRadius(field, data):
    center = data.get_field_parameter("center")
    DW = data.pf.domain_right_edge - data.pf.domain_left_edge
    radius = na.zeros(data["x"].shape, dtype='float64')
    for i, ax in enumerate('xy'):
        r = na.abs(data[ax] - center[i])
        radius += na.minimum(r, na.abs(DW[i]-r))**2.0
    na.sqrt(radius, radius)
    return radius

def _ConvertDiskRadiusCGS(data):
    return data.convert("cm")
add_field("ParticleDiskRadius", function=_ParticleDiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiusCGS, units=r"\rm{cm}",
          particle_type = True,
          display_name = "Particle Disk Radius")
add_field("DiskRadius", function=_DiskRadius,
```

```
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiusCGS, units=r"\rm{cm}")

def _ConvertDiskRadiusMpc(data):
    return data.convert("mpc")
add_field("DiskRadiusMpc", function=_DiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiusMpc, units=r"\rm{Mpc}",
          display_name = "Disk Radius")
add_field("ParticleDiskRadiusMpc", function=_ParticleDiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiusMpc, units=r"\rm{Mpc}",
          particle_type=True,
          display_name = "Particle Disk Radius")

def _ConvertDiskRadiuskpc(data):
    return data.convert("kpc")
add_field("ParticleDiskRadiuskpc", function=_ParticleDiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiuskpc, units=r"\rm{kpc}",
          particle_type=True,
          display_name = "Particle Disk Radius")
add_field("DiskRadiuskpc", function=_DiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiuskpc, units=r"\rm{kpc}",
          display_name = "Disk Radius")

def _ConvertDiskRadiuskpch(data):
    return data.convert("kpch")
add_field("ParticleDiskRadiuskpch", function=_ParticleDiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiuskpc, units=r"\rm{kpc}/\rm{h}",
          particle_type=True,
          display_name = "Particle Disk Radius")
add_field("DiskRadiuskpch", function=_DiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiuskpc, units=r"\rm{kpc}/\rm{h}",
          display_name = "Disk Radius")

def _ConvertDiskRadiuspc(data):
    return data.convert("pc")
add_field("ParticleDiskRadiuspc", function=_ParticleDiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiuspc, units=r"\rm{pc}",
          particle_type=True,
          display_name = "Particle Disk Radius")
add_field("DiskRadiuspc", function=_DiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiuspc, units=r"\rm{pc}",
          display_name="Disk Radius")

def _ConvertDiskRadiusAU(data):
    return data.convert("au")
add_field("ParticleDiskRadiusAU", function=_ParticleDiskRadius,
          validators=[ValidateParameter("center")],
```

```
          convert_function = _ConvertDiskRadiusAU, units=r"\rm{AU}",
          particle_type=True,
          display_name = "Particle Disk Radius")
add_field("DiskRadiusAU", function=_DiskRadius,
          validators=[ValidateParameter("center")],
          convert_function = _ConvertDiskRadiusAU, units=r"\rm{AU}",
          display_name = "Disk Radius")

add_field("ParticleDiskRadiusCode", function=_ParticleDiskRadius,
          validators=[ValidateParameter("center")],
          particle_type=True,
          display_name = "Particle Disk Radius (code)")
add_field("DiskRadiusCode", function=_DiskRadius,
          validators=[ValidateParameter("center")],
          display_name = "Disk Radius (code)")

def obtain_rvec(data):
    center = data.get_field_parameter('center')
    coords = na.array([data['x'],data['y'],data['z']], dtype='float64')
    new_shape = tuple([3] + [1]*(len(coords.shape)-1))
    r_vec = coords - na.reshape(center,new_shape)
    return r_vec # axis 0 is the x,y,z

def obtain_velocities(data):
    if data.has_field_parameter("bulk_velocity"):
        bv = data.get_field_parameter("bulk_velocity")
    else: bv = na.zeros(3, dtype='float64')
    xv = data["x-velocity"] - bv[0]
    yv = data["y-velocity"] - bv[1]
    zv = data["z-velocity"] - bv[2]
    return xv, yv, zv

def _DiskRadialVelocity(field, data):
    center = data.get_field_parameter("center")
    bulk_velocity = data.get_field_parameter("bulk_velocity")
    if bulk_velocity == None:
        bulk_velocity = na.zeros(3)
    new_field = ( (data['x']-center[0])*(data["x-velocity"]-bulk_velocity[0])
                + (data['y']-center[1])*(data["y-velocity"]-bulk_velocity[1])
                )/data["DiskRadiusCode"]
    if na.any(na.isnan(new_field)): # to fix center = point
        new_field[na.isnan(new_field)] = 0.0
    return new_field
def _DiskRadialVelocityABS(field, data):
    return na.abs(_DiskRadialVelocity(field, data))
def _ConvertDiskRadialVelocityKMS(data):
    return 1e-5
add_field("DiskRadialVelocity", function=_DiskRadialVelocity,
          units=r"\rm{cm}/\rm{s}",
          validators=[ValidateParameter("center")])
add_field("RadialVelocityABS", function=_DiskRadialVelocityABS,
          units=r"\rm{cm}/\rm{s}",
          validators=[ValidateParameter("center")])
add_field("DiskRadialVelocityKMS", function=_DiskRadialVelocity,
          convert_function=_ConvertDiskRadialVelocityKMS, units=r"\rm{km}/\rm{s}",
```

```
          validators=[ValidateParameter("center")])
add_field("DiskRadialVelocityKMSABS", function=_DiskRadialVelocityABS,
          convert_function=_ConvertDiskRadialVelocityKMS, units=r"\rm{km}/\rm{s}",
          validators=[ValidateParameter("center")])

def _ParticleDiskVelocityMagnitude(field, data):
    """M{|v|}"""
    bulk_velocity = data.get_field_parameter("bulk_velocity")
    if bulk_velocity == None:
        bulk_velocity = na.zeros(3)
    return ( (data["particle_velocity_x"]-bulk_velocity[0])**2.0 + \
             (data["particle_velocity_y"]-bulk_velocity[1])**2.0 )**(1.0/2.0)
add_field("ParticleDiskVelocityMagnitude", function=_ParticleDiskVelocityMagnitude,
          particle_type=True,
          take_log=False, units=r"\rm{cm}/\rm{s}")

add_field("ParticleDiskVelocityMagnitude", function=_ParticleDiskVelocityMagnitude,
          particle_type=True,
          take_log=False, units=r"\rm{cm}/\rm{s}")

def _DiskVelocityMagnitude(field, data):
    """M{|v|}"""
    bulk_velocity = data.get_field_parameter("bulk_velocity")
    if bulk_velocity == None:
        bulk_velocity = na.zeros(3)
    return ( (data["x-velocity"]-bulk_velocity[0])**2.0 + \
             (data["y-velocity"]-bulk_velocity[1])**2.0 )**(1.0/2.0)
add_field("DiskVelocityMagnitude", function=_DiskVelocityMagnitude,
          take_log=False, units=r"\rm{cm}/\rm{s}")

def _DiskVelocitySquared(field, data):
    return  data["DiskVelocityMagnitude"]**2
add_field("DiskVelocitySquared", function=_DiskVelocitySquared,
  units=r"\rm{cm}^2/\rm{s}^2")

def _DiskTangentialVelocity(field, data):
    return na.sqrt(data["DiskVelocityMagnitude"]**2.0
                   - data["DiskRadialVelocity"]**2.0)
add_field("DiskTangentialVelocity",
          function=_DiskTangentialVelocity,
          take_log=False, units=r"\rm{cm}/\rm{s}")

def _DiskAngularVelocity(field, data):
    r_vec = obtain_rvec(data)
    xv, yv, zv = obtain_velocities(data)
    diskr_vec = na.array([r_vec[0],r_vec[1]], dtype='float64')
    diskv_vec = na.array([xv,yv], dtype='float64')
    mom = na.cross(diskr_vec, diskv_vec, axis=0)
    return mom/data["DiskRadiusCode"]**2
def _convertDiskAngularVelocity(data):
    return 1.0/data.convert("cm")
add_field("DiskAngularVelocity", function=_DiskAngularVelocity,
           convert_function=_convertDiskAngularVelocity,
           units=r"\rm{s}^{-1}", validators=[ValidateParameter('center')])
```

B.2 Preamble and definitions

For every program used, the first lines were to preload the following packages:

```
from yt.config import ytcfg; ytcfg["yt","serialize"] = "False"
from yt.mods import *
from yt.analysis_modules.star_analysis.api import *
import numpy
import math
import scipy as sc
from scipy import interpolate
from scipy import ndimage
import pylab as pl
pl.ion()
from matplotlib import rc
fontsize=12
rc('text', usetex=True)
rc('font', **{'family':'serif','serif':'Computer Modern Roman', 'size':fontsize})
rc('axes', labelsize=fontsize)
rc('legend', fontsize=fontsize, numpoints=1, frameon=False)
rc('xtick', labelsize=fontsize)
rc('ytick', labelsize=fontsize)
rc('lines', lw=1.5, mew=0.3)
rc('grid', linewidth=0.5)
```

We also define the following variables:

```
start_data           = 01
end_data             = numpy.array([226])
path                 = '/Volumes/SIMULATIONS/enzo-2.1.1/'
simulations          = ['g1e10s4e10dm1e12']
colors               = ['b']
total_sim            = len(simulations)

disk_radius          = numpy.array(range(1,60))/3.
initial_radius       = disk_radius[0]
delta_radius         = initial_radius/2.
final_radius         = disk_radius[len(disk_radius)-1]
disk_height          = 5                  # kpc
specified_center     = [0.5, 0.5, 0.5]
percentage_of_stars  = 0.95
```

B.3 Profiles

```
Q                    = numpy.zeros(len(disk_radius), float)
sigma_c              = numpy.zeros(len(disk_radius), float)
V_rot                = numpy.zeros(len(disk_radius), float)
M_dyn                = numpy.zeros(len(disk_radius), float)
Omega                = numpy.zeros(len(disk_radius), float)
M_rot                = numpy.zeros(len(disk_radius), float)
disk_sigma_gas       = numpy.zeros(len(disk_radius), float)
ring_vtan            = numpy.zeros(len(disk_radius), float)
ring_dispersion      = numpy.zeros(len(disk_radius), float)
```

```
ring_sigma_gas       = numpy.zeros(len(disk_radius), float)

for sim in range(total_sim):

   if PLOT_PROPERTY(SFR):
      dir   = path+simulations[sim]+"/DD%04i/" %(end_data[sim])
      file  = simulations[sim]+"_%04i" %(end_data[sim])
      pf    = load(dir+file)
      my_disk=pf.h.disk(specified_center, [0, 0, 1], final_radius/pf["kpc"],
         disk_height/pf["kpc"])
      sfr = StarFormationRate(pf, data_source=my_disk)
      mediansfr=ndimage.median_filter(sfr.Msol_yr, size=10)
      newtime= [sfr.time[i] for i in range(len(sfr.time)) if i % 6 == 3]
      newsfr= [mediansfr[i] for i in range(len(mediansfr)) if i % 6 == 3]
      f=interpolate.interp1d(newtime, newsfr, kind='cubic')
      interpsfr=f(sfr.time[(sfr.time>newtime[0]) & (sfr.time<newtime[len(newtime)-1])])
      pl.xlabel(r'Time [yr]')
      pl.ylabel(r'SFR [$M_\odot~yr^{-1}$]')
      pl.plot(sfr.time-sfr.time[0], sfr.Msol_yr, 'r-', linewidth=0.5)
      pl.plot(sfr.time[(sfr.time>newtime[0]) &
            (sfr.time<newtime[len(newtime)-1])]-sfr.time[0], interpsfr,
            label=simulations[sim], color=colors[sim])
      pl.ylim(ymin=0)
      for actual_data in [221]:
         dir   = path+simulations[sim]+"/DD%04i/" %(actual_data)
         file  = simulations[sim]+"_%04i" %(actual_data)
         pf    = load(dir+file)
         dd    = pf.h.all_data()
         print pf.current_time
         my_disk=pf.h.disk(specified_center, [0, 0, 1], final_radius/pf["kpc"],
            disk_height/pf["kpc"])
         sfr = StarFormationRate(pf, data_source=dd)
         inst_SFR=sfr.Msol_yr[len(sfr.time)-1]
         avg_SFR=sfr.Msol_cumulative[len(sfr.time)-1]/
            (sfr.time[len(sfr.time)-1]-sfr.time[0])
         pl.plot([0, sfr.time[len(sfr.time)-1]-sfr.time[0]],
            [inst_SFR, inst_SFR], 'k--')
         pl.plot([0, sfr.time[len(sfr.time)-1]-sfr.time[0]],
            [avg_SFR, avg_SFR], 'k:')
         pl.plot([sfr.time[len(sfr.time)-1]-sfr.time[0],
            sfr.time[len(sfr.time)-1]-sfr.time[0]], [0, avg_SFR], 'k-', linewidth=0.5)
      location=1
      pl.savefig('./profiles/SFR_avg_vs_inst_'+simulations[sim]+'.png')
      pl.show()
      pl.close()

   dir   = path+simulations[sim]+"/DD%04i/" %(end_data[sim])
   file  = simulations[sim]+"_%04i" %(end_data[sim])
   pf    = load(dir+file)
   for i in range(len(disk_radius)):
      my_disk=pf.h.disk(specified_center, [0, 0, 1], disk_radius[i]/pf["kpc"],
         disk_height/pf["kpc"])
      ring=numpy.where((disk_radius[i] - delta_radius < my_disk["DiskRadiuskpc"])
         & (my_disk["DiskRadiuskpc"] <= disk_radius[i] + delta_radius))
      V_rot[i]=numpy.mean(my_disk["DiskTangentialVelocity"][ring])
```

```
      Omega[i]=V_rot[i]/(disk_radius[i]*kpc)
      M_dyn[i]=V_rot[i]**2*(disk_radius[i]*kpc)/Grav/Msun

      disk_area=na.pi*(disk_radius[i]*kpc)**2
      ring_area=na.pi*((disk_radius[i] + delta_radius)**2-(disk_radius[i]
         - delta_radius)**2)*kpc**2
      disk_gas_mass=my_disk.quantities["TotalQuantity"]("CellMassMsun",
         lazy_reader=True)[0]
      ring_gas_mass=my_disk["CellMassMsun"][ring]
      disk_sigma_gas[i]=disk_gas_mass*Msun/disk_area

      M_rot[i]=na.pi**4*Grav**2.0*disk_sigma_gas[i]**3.0/4.0/Omega[i]**4/Msun

      ring_sigma_gas[i]=sum(ring_gas_mass)*Msun/ring_area
      ring_vtan[i]=sum(my_disk["DiskTangentialVelocity"][ring]*ring_gas_mass)/
         sum(ring_gas_mass)
      ring_velocity_squared=sum(my_disk["DiskVelocitySquared"][ring]*ring_gas_mass)/
         sum(ring_gas_mass)
      ring_dispersion[i]=na.sqrt(ring_velocity_squared)-ring_vtan[i]
      if i > 0 :
         dVdR=(ring_vtan[i]-ring_vtan[i-1])/((disk_radius[i]-disk_radius[i-1])*kpc)
         kappa=na.sqrt(2*ring_vtan[i]**2/(disk_radius[i]*kpc)**2*(1+
            disk_radius[i]*kpc/ring_vtan[i]*dVdR))
         Q[i]=ring_dispersion[i]*kappa/(na.pi*Grav*ring_sigma_gas[i])
         sigma_c[i]=alpha*kappa*ring_dispersion[i]/3.36/Grav
```

B.4 PDF

```
for sim in range(total_sim):

   N     = numpy.zeros(n_bins, float)
   d_m   = numpy.zeros(n_bins, float)

   dir   = path+simulations[sim]+"/DD%04i/" %(end_data)
   file  = simulations[sim]+"_%04i" %(end_data)
   pf    = load(dir+file)
   dd    = pf.h.all_data()
   density = numpy.log10(dd["DiskSurfaceDensity"]*pc**2/Msun)
   density = density.reshape(-1)

   dmin, dmax = dd.quantities["Extrema"]("DiskSurfaceDensity", non_zero=True,
      lazy_reader=False)[0]
   dmin=dmin*pc**2/Msun
   dmax=dmax*pc**2/Msun
   if LOG_PLOT:
      dmin=numpy.log10(dmin)
      dmax=numpy.log10(dmax)
   delta=(dmax-dmin)/n_bins
   for i in range(0,n_bins):
      d_i=delta*i+dmin
      d_f=delta*(i+1)+dmin
      d_m[i]=delta*(i+1./2.)+dmin
      if LOG_PLOT:
```

```
      d_i=10**d_i
      d_f=10**d_f
      d_m[i]=10**d_m[i]
   N[i]=float(len(numpy.where((d_i<=dd["DiskSurfaceDensity"]*pc**2/Msun)
      & (dd["DiskSurfaceDensity"]*pc**2/Msun<d_f))[0]))/float(len(density))

pl.step(numpy.log10(d_m),numpy.log10(N), 'k')
```

B.5 Star formation law plots

```
global_SFR             = numpy.zeros((total_points[sim], number_of_radius), float)
avg_SFR                = numpy.zeros((total_points[sim], number_of_radius), float)
sigma_gas              = numpy.zeros((total_points[sim], number_of_radius), float)
V_rot                  = numpy.zeros((total_points[sim], number_of_radius), float)
t_dyn                  = numpy.zeros((total_points[sim], number_of_radius), float)
Omega                  = numpy.zeros((total_points[sim], number_of_radius), float)
M_dyn                  = numpy.zeros((total_points[sim], number_of_radius), float)
M_rot_1                = numpy.zeros((total_points[sim], number_of_radius), float)

for index in range (total_points[sim]):
   dir   = path+simulations[sim]+"/DD%04i/" %((index*every_data)+start_data)
   file  = simulations[sim]+"_%04i" %((index*every_data)+start_data)
   pf    = load(dir+file)
   dd    = pf.h.all_data()
   stars = (dd["particle_type"] == PARTICLE_TYPE_STAR)
   star_radius = sorted(dd["ParticleDiskRadiuskpc"][stars])
   number_of_particles = len(star_radius)
   if len(star_radius) == 0:
      continue
   for k in range(number_of_radius):
      if percentage_of_stars[k]==1:
         rpart_min, rpart_max = dd.quantities["Extrema"]("ParticleDiskRadiuskpc",
            non_zero=True, lazy_reader=False)[0]
      else:
         rpart_max =
            star_radius[int(math.floor(percentage_of_stars[k]*number_of_particles))]
      my_disk =
         pf.h.disk(specified_center, [0, 0, 1], rpart_max/pf["kpc"],
            disk_height/pf["kpc"])
      sfr = StarFormationRate(pf, data_source=my_disk)
      if INST_KS:
         global_SFR[index,k]=sfr.Msol_yr[len(sfr.time)-1]/PI/rpart_max**2
      elif AVG_KS:
         avg_SFR[index,k] = sfr.Msol_cumulative[len(sfr.time)-1]/
            (sfr.time[len(sfr.time)-1]-sfr.time[0])/PI/rpart_max**2
      sigma_gas[index,k]=my_disk.quantities["TotalQuantity"]("CellMassMsun",
         lazy_reader=True)[0]/PI/(rpart_max*1000.0)**2
      if SFRTDYN_PLOT:
         epsilon=0.5
         annulus=numpy.where((rpart_max - epsilon < my_disk["DiskRadiuskpc"])
            & (my_disk["DiskRadiuskpc"] < rpart_max + epsilon))
         V_rot[index,k]=numpy.mean(my_disk["DiskTangentialVelocity"][annulus])
         t_dyn[index,k]=2*na.pi*rpart_max*kpc/V_rot[index,k]/sec_per_year
```