Numerical Investigation of Galactic Merger Utilizing High Performance Computing Architectures: Ancient Satellite Galaxy and Wandering Supermassive Black Hole

Yohei MIKI

February 2014

Numerical Investigation of Galactic Merger Utilizing High Performance Computing Architectures: Ancient Satellite Galaxy and Wandering Supermassive Black Hole

Yohei MIKI
Doctoral Program in Physics

Submitted to the Graduate School of
Pure and Applied Sciences
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in
Science

at the
University of Tsukuba

Graduate School of Pure and Applied Sciences

Numerical Investigation of Galactic Merger Utilizing High Performance Computing Architectures: Ancient Satellite Galaxy and Wandering Supermassive Black Hole

（高性能計算を駆使した銀河衝突の数値的探求: 過去の衛星銀河と銀河ハローを漂う超巨大ブラックホール）

Yohei MIKI

Doctoral Program in	Physics
Student ID	201130079
Doctor of Philosophy in	Science
Advised by	Masao Mori

Abstract

Recent observations targeting on galactic halos have discovered many signatures of galactic mergers. Galactic merger is one of the key processes in the hierarchical structure formation scenario under the cold dark matter model. Galactic archaeological approaches based on comparison between observed data and theoretical models have contributed to unveil the formation and the evolution history of galaxies. Moreover, observations have revealed some correlations between the physical properties of galaxies and the mass of their central supermassive black holes (SMBHs). Coevolution of galaxies and SMBHs suggested by the observations is a hot research issue recently. State-of-the-art numerical simulation exploiting high performance computing architectures is a powerful and attractive tool to examine the current open-questions by comparing with the observations in detail. We have investigated the physical properties of an ancient satellite galaxy (Part I) and the current location of an expected SMBH wandering in the galactic halo (Part III) in terms of numerical galactic archaeology. Since a numerous parameter studies are necessary to complete these studies, we have developed a highly optimized N-body code efficiently works on a cluster equips many boards of graphics processing units (GPU) in Part II.

Photometric and spectroscopic surveys focused on the halo of the Andromeda galaxy (M31) have found many structures considered to be merger remnants such as the Andromeda giant stellar stream and stellar shells. To unveil a progenitor of the Andromeda giant stellar stream, we have investigated the interaction between an accreting satellite galaxy and M31 using an N-body simulation. A comprehensive parameter study with 247 models has been performed by varying the size and the mass distribution of the progenitor dwarf galaxy. We show that it is crucial the binding energy of the progenitor galaxy to reproduce the Andromeda giant stellar stream and the shell-like structures surrounding M31. As a result of simulations, the progenitor must satisfy a simple scaling relation among the core radius, the total mass and the tidal radius. Using this relation, we have successfully constrained the physical properties of the progenitors which have a mass ranging from $5 \times 10^8 M_\odot$ to $5 \times 10^9 M_\odot$ and a central surface density around $10^3\, M_\odot\, \mathrm{pc}^{-2}$. A detailed comparison between the result and the observed nearby galaxies indicates that the progenitor of the Andromeda giant stellar stream includes a dwarf elliptical galaxy, a dwarf irregular galaxy, and a small spiral galaxy.

We have developed a highly optimized code for collisionless N-body calculations based on direct summation. Our new optimization hides the latency to access the global memory, and the resulting CUDA code has a peak performance of 1006.7 GFlop/s in single precision (assuming 26 floating-point operations per interaction) with a single NVIDIA Tesla M2090 board. Detailed performance analysis clarifies that the

performance metrics of collisionless N-body simulations on GPU are only two quantities: first one is the number of running streaming multiprocessors and another is the clock cycle ratio of the latency to access the global memory and operations to calculate gravitational interaction. To improve the scalability of the OpenMP/MPI hybrid parallelized code, we have reduced the number of communications among multiple GPUs and have overlapped communications with computations to hide the communication time. The results of performance measurements show excellent scalability with superlinear scaling when the number of N-body particles per GPU is less than 10^4 and parallel efficiency approaching unity when the number of N-body particles per GPU is greater than 10^4. The CUDA/OpenMP/MPI code has a peak performance of 255.5 TFlop/s when 256 NVIDIA Tesla M2090 boards are used, which is 75.0% of the theoretical peak performance.

In the hierarchical structure formation scenario, galaxies enlarge through multiple merging events with less massive galaxies. In addition, the Magorrian relation indicates that almost all galaxies host a central SMBH of mass 10^{-3} of its spheroidal component. Consequently, SMBHs likely to wander in the halos of their host galaxies following a galaxy collision, although evidence of this activity is currently lacking. We have investigated a current plausible location of an SMBH wandering in the halo of M31. According to theoretical studies of N-body simulations, some of the many substructures in the M31 halo are remnants of a minor merger occurring about 1 Gyr ago. First, to evaluate the possible parameter space of the infalling orbit of the progenitor, we have performed numerous parameter studies using a GPU cluster, HA-PACS at University of Tsukuba. To reduce uncertainties in the predicted position of the expected SMBH, we then have calculated the time evolution of the SMBH in the progenitor dwarf galaxy from N-body simulations using the plausible parameter sets. The results show that the SMBH lies within the halo ($\sim$20–50 kpc from the M31 center), closer to the Milky Way than the M31 disk. Furthermore, the predicted current positions of the SMBH are restricted to an observational field of $0^\circ.6 \times 0^\circ.7$ in the northeast region of the M31 halo. We also discuss the origin of the infalling orbit of the satellite galaxy and its relationships with the recently discovered vast thin disk plane of satellite galaxies around M31.

Contents

Part I

Physical Properties of Possible Progenitors of Giant Stellar Stream in the M31 Halo

Abstract

To unveil a progenitor of the Andromeda giant stellar stream, we investigate the interaction between an accreting satellite galaxy and the Andromeda galaxy (M31) using an N-body simulation. A comprehensive parameter study with 247 models is performed by varying the size and the mass distribution of the progenitor dwarf galaxy. We show that it is crucial the binding energy of the progenitor galaxy to reproduce the Andromeda giant stellar stream and the shell-like structures surrounding M31. As a result of simulations, the progenitor must satisfy a simple scaling relation among the core radius, the total mass and the tidal radius. Using this relation, we successfully constrain the physical properties of the progenitors which have a mass ranging from $5 \times 10^8 M_\odot$ to $5 \times 10^9 M_\odot$ and a central surface density around $10^3\, M_\odot\, \mathrm{pc}^{-2}$. A detailed comparison between our result and the observed nearby galaxies indicates that the progenitor of the Andromeda giant stellar stream includes a dwarf elliptical galaxy, a dwarf irregular galaxy, and a small spiral galaxy.

Chapter 1 Introduction

In a cold dark matter (CDM) universe, the hierarchical structure formation scenario posits that large galaxies, such as the Milky Way and the Andromeda galaxy (M31), have enlarged through multiple mergers with smaller galaxies. Cosmological N-body simulations of the hierarchical structure formation have revealed a wealth of merger remnants around host galaxies (e.g. Bullock & Johnston 2005, see Fig. 1.1). To verify theoretical predictions from the CDM scenario and therefore test the current cosmology, many observational researchers have focused on merger remnants (e.g., Chiba et al. 2005; Minezaki et al. 2009). In the local universe, theoretically predicted tidal features have been discovered; for example, the Sagittarius

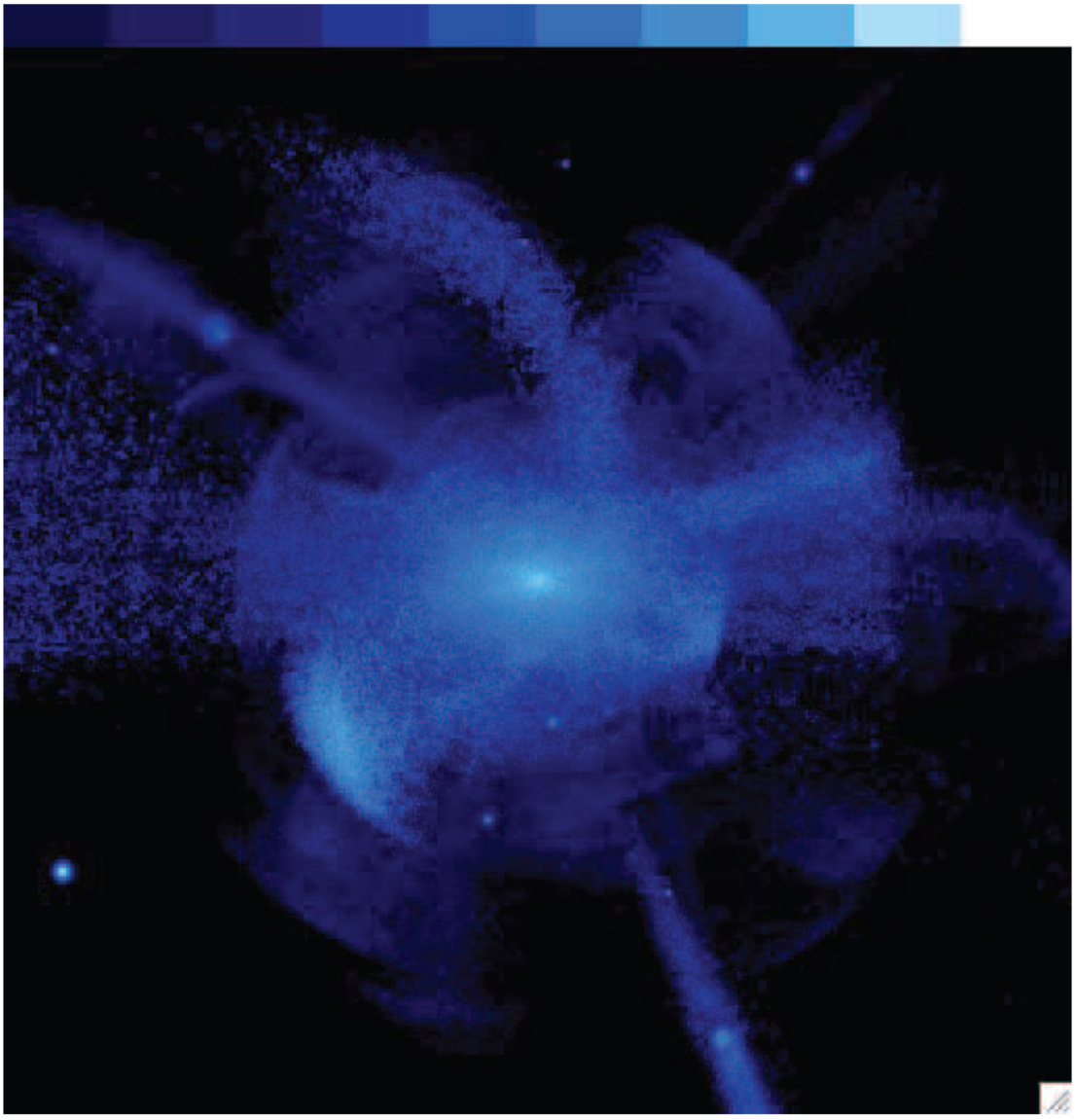


Fig. 1.1: External view of a Milky Way size halo, taken from Bullock & Johnston (2005). The box size is 300 kpc squared.

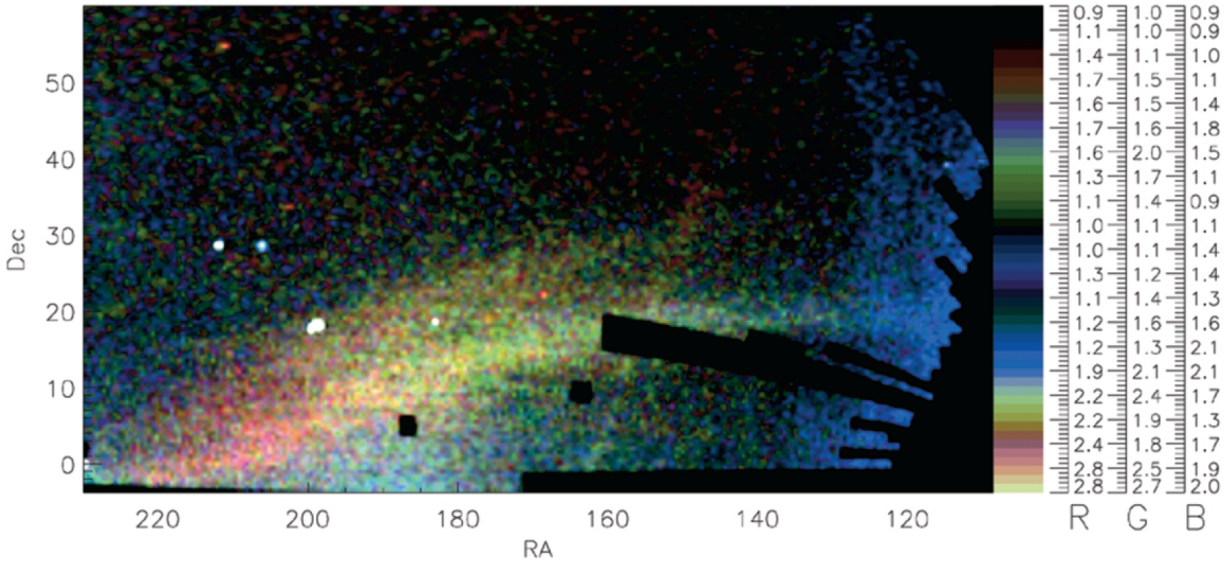


Fig. 1.2: Spatial density of stars in Sloan Digital Sky Survey Data Release 5, taken from Belokurov et al. (2006).

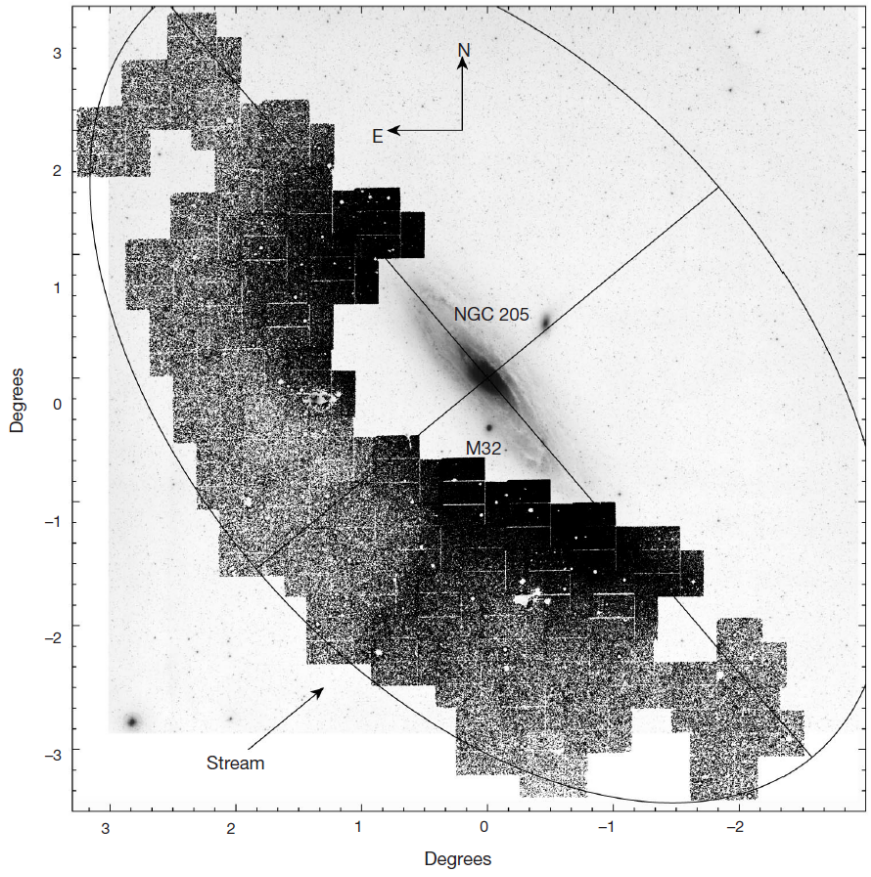


Fig. 1.3: Surface density of red giant branch stars in the southeastern halo of M31, taken from Ibata et al. (2001).

stream (Fig. 1.2) in the Milky Way and the giant stellar stream (Fig. 1.3) in M31.

Further photometric observations of red giant stars near M31 have revealed a giant stellar stream to its south as well as giant stellar shells to the east and the west of its center (Ferguson et al. 2002; McConnachie

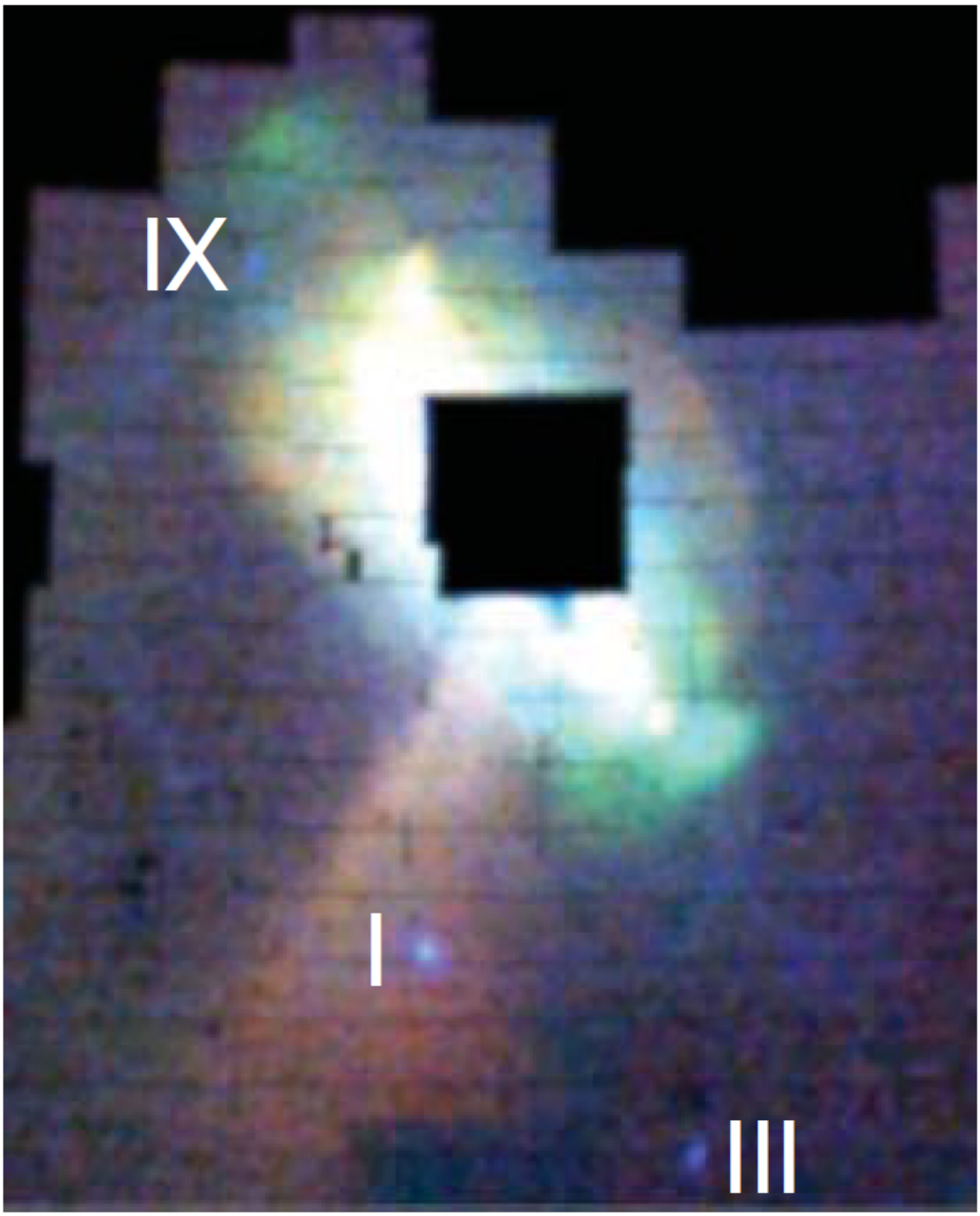
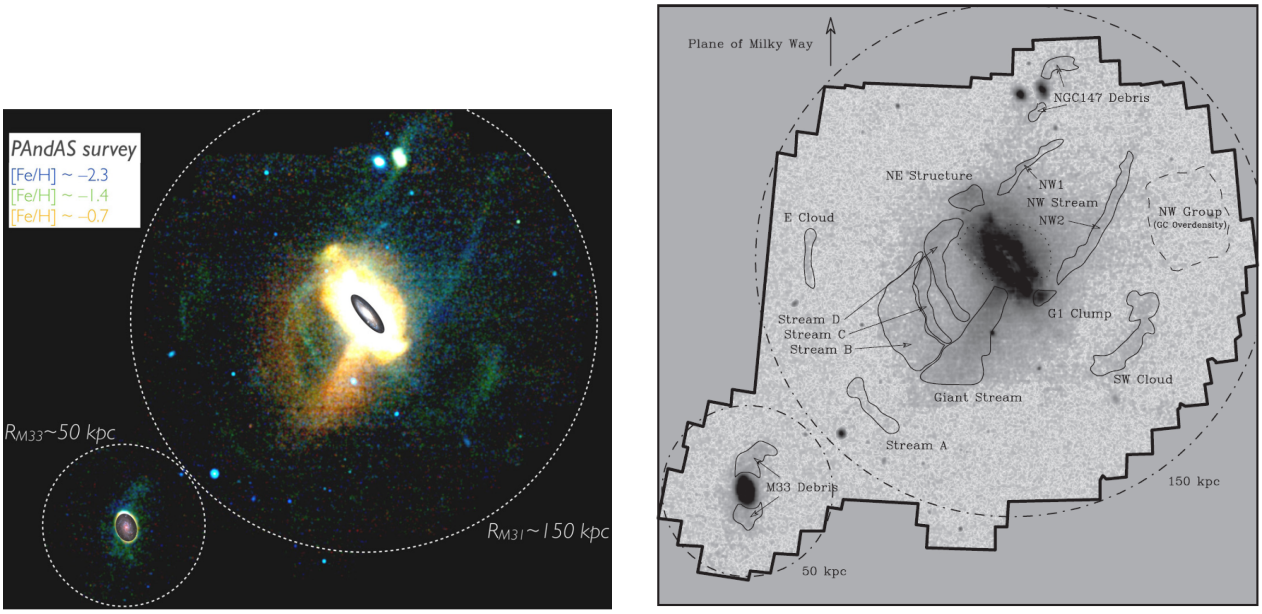


Fig. 1.4: Stellar density map of the M31 halo, taken from McConnachie et al. (2009).

et al. 2003; Ibata et al. 2005; Irwin et al. 2005; McConnachie et al. 2009, Fig. 1.4). The giant stellar stream extends out to over 100 kpc from M31's center (McConnachie et al. 2003). Furthermore, photometric and spectroscopic observations of the spatial distribution and the radial velocity distribution of red giant stars and of the metallicity distribution were performed (Ibata et al. 2004, 2005; Guhathakurta et al. 2006; Kalirai et al. 2006a,b; Ibata et al. 2007; Koch et al. 2008; Gilbert et al. 2007, 2009; Tanaka et al. 2010). A recent photometric observation project targeting M31 and M33, the Pan-Andromeda Archaeological Survey (PAndAS; McConnachie et al. 2009), discovered many substructures in addition to the giant stream and two stellar shells (Martin et al. 2013; Lewis et al. 2013, Fig. 1.5).

The PAndAS project discovered a few tens of satellite galaxies around M31 (McConnachie et al. 2009;



(a) Combined red-green-blue color image around M31 and M33, taken from Martin et al. (2013).

(b) Schematic illustration of the prominent substructures, taken from Lewis et al. (2013).

Fig. 1.5: Panoramic view of PAndAS results.

Richardson et al. 2011; Martin et al. 2013). Recent spectroscopic observations such as Collins et al. (2013) and the SPLASH survey (Spectroscopic and Photometric Landscape of Andromeda's Stellar Halo: Kalirai et al. 2010; Tollerud et al. 2012) obtained kinematic information on newly discovered dwarf spheroidal galaxies. The above mentioned results of recent observations strongly accelerate investigation for the physical properties of M31 dwarf spheroidal galaxies. As another strategy, investigating the physical properties of the progenitor dwarf galaxy is also possible by comparing the observed structures with results of N-body simulation of a galaxy collision with M31. Information related to dynamics of the progenitor dwarf galaxy would be conserved as footprints in the observed structures. Destructive test utilizing N-body simulation has a potential to recover fossil information on dynamics of the progenitor dwarf galaxy imprinted in the observed structures.

N-body simulations of the interaction between the progenitor of the giant stellar stream and M31 (Fardal et al. 2007, 2012; Mori & Rich 2008) suggest that the stream, the northeast shell, and the west shell are tidal debris formed during the pericentric passages of a satellite on a radial orbit. After the first reproduction of the giant stellar stream using N-body simulation by Fardal et al. (2007, Fig. 1.6), many studies based on N-body simulations have devoted to investigating various aspects of the observed structures. Mori & Rich (2008) investigated the dynamical response of the M31 disk in detail and derived mass range of the progenitor dwarf galaxy. Fardal et al. (2008) and Sadoun et al. (2013) showed a collision model of a disk galaxy with M31 also reproduces the observed structures well. Fardal et al. (2013) improved the collision model of Fardal et al. (2007) in various aspects (e.g., the infalling orbit of the progenitor and the mass of M31). Hammer et al. (2010, 2013) proposed an alternative scenario that a past major merger produces M31, the giant stellar stream and the stellar shells. The results of the minor merger scenario based on Fardal et al. (2007) have been compared with results of spectroscopic observations. Observations by Gilbert et al. (2007, 2009) discovered additional structures on phase space predicted by Fardal et al. (2007). Fardal et al. (2012) reported a beautiful agreement of their observation and N-body simulation in the west shell region.

The remainder of this part is organized as follows. In Chapter 2, we describe the M31 model, including a disk, a bulge, and a dark matter halo and the satellite models. In Chapter 3, we present the results of

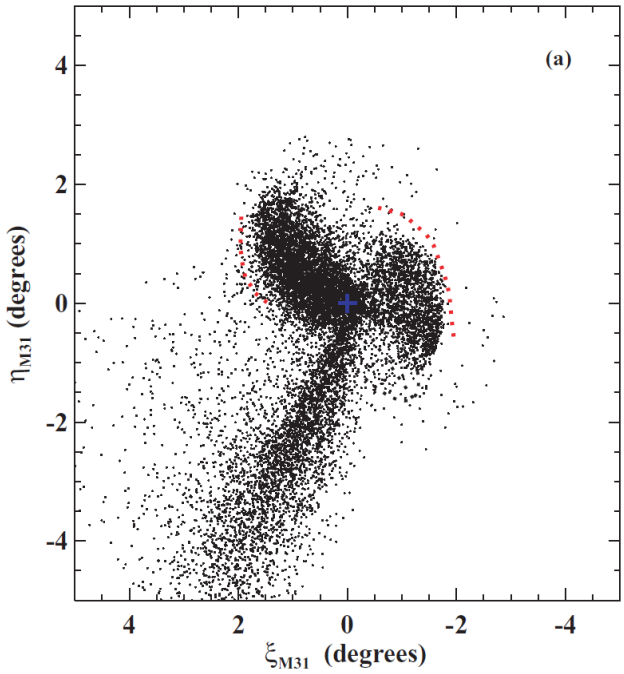


Fig. 1.6: The spatial distribution of simulated satellite debris, taken from Fardal et al. (2007). The red dotted curves show the observed shelf edges.

the numerical simulations and analyze them. In Chapter 4, we compare the results with observations. In Chapter 5, we present the results of convergence tests. Finally, in Chapter 6, we summarize our results.

Chapter 2 Model Description of Interaction between M31 and Satellite Galaxies

We have performed 247 comprehensive N-body simulations of the interaction between M31 and the progenitor dwarf galaxy with different sizes and density profiles to explore the characteristics of the progenitor of the giant stream. We have modeled the gravitational potential of M31 by a fixed potential as described in Section 2.1 and have represented the progenitor dwarf using King spheres (Section 2.2).

2.1 Model of M31

To investigate the dynamical response of the orbiting satellite, we model the dwarf galaxy by a self-consistent N-body realization of stars under the influence of an external force provided by M31. For simplicity, we assume that M31 is composed of three components: a disk, a bulge, and a dark matter halo. It should be noted that Mori & Rich (2008) studied the self-gravitating response of the disk, bulge, and dark matter halo of M31 to an accreting satellite and concluded that satellites less massive than $5 \times 10^9 M_\odot$ had a negligible effect on the gravitational potential of M31. Consequently, in this study, we treat M31 as the source of a fixed gravitational potential.

We model the bulge of M31 as a spherically symmetric mass distribution represented by a Hernquist profile (Hernquist 1990). The corresponding density-potential pair is given by

$$\rho_{\rm b}(r) = \left(\frac{M_{\rm b}}{2\pi {r_{\rm b}}^3}\right) \frac{1}{(r/r_{\rm b})(1+r/r_{\rm b})^3}, \tag{2.1}$$

$$\Phi_{\rm b}(r) = -\frac{GM_{\rm b}}{r_{\rm b}+r}, \tag{2.2}$$

where $M_{\rm b} = 3.24 \times 10^{10} M_\odot$ is the total mass of the bulge; $r_{\rm b} = 0.61\,{\rm kpc}$, its scale radius; and G, the gravitational constant. The density-potential pair of the axisymmetric distribution in cylindrical coordinates (R, z) of the disk is given by

$$\rho_{\rm d}(R,z) = \frac{\Sigma_0}{2z_{\rm d}} \exp\left(-\frac{R}{R_{\rm d}}\right) \exp\left(-\frac{z}{z_{\rm d}}\right), \tag{2.3}$$

$$\Phi_{\rm d}(R,z) = -\frac{2G\Sigma_0}{R_{\rm d} z_{\rm d}} \int_{-\infty}^{\infty} dz' \exp\left(-\frac{|z'|}{z_{\rm d}}\right) \int_0^{\infty} da \sin^{-1}\left(\frac{2a}{\sqrt{+}+\sqrt{-}}\right) a K_0\left(\frac{a}{R_{\rm d}}\right), \tag{2.4}$$

where $r = \sqrt{R^2+z^2}$, $\sqrt{\pm} \equiv \sqrt{(z-z')^2+(a\pm R)^2}$, $\Sigma_0 = 2.0 \times 10^8 M_\odot {\rm kpc}^{-2}$ is the central surface density of the disk, $R_{\rm d} = 5.40\,{\rm kpc}$ is the disk scale radius, $z_{\rm d} = 0.60\,{\rm kpc}$ is the disk scale height, and $K_\alpha(x)$ is the modified Bessel function (cf. Binney & Tremaine 2008). In this case, the total mass of the disk is $M_{\rm d} = 3.66 \times 10^{10} M_\odot$. Finally, we assume that the extended dark matter halo can be adequately modeled as a spherically symmetric system. We adopt the Navarro-Frenk-White profile (Navarro et al. 1996, 1997),

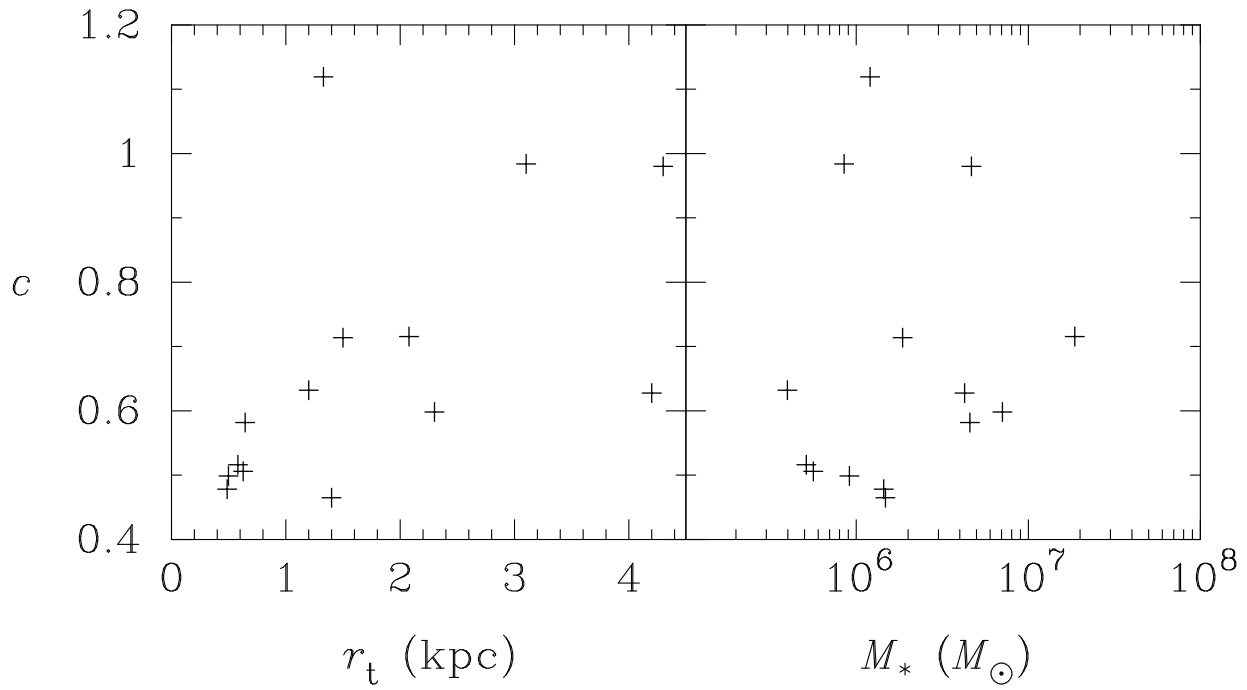


Fig. 2.1: Relationship between the concentration c and the tidal radius r_t (left panel) and the concentration c and the stellar mass M_* (right panel) of local dwarf galaxies. Data are compiled from Woo et al. (2008); McConnachie & Irwin (2006a); Irwin & Hatzidimitriou (1995).

and the density-potential pair is given by

$$\rho_\mathrm{h}(r) = \frac{\delta_\mathrm{c}\rho_\mathrm{c}}{(r/r_\mathrm{h})(1+r/r_\mathrm{h})^2}, \tag{2.5}$$

$$\Phi_\mathrm{h}(r) = -4\pi G\delta_\mathrm{c}\rho_\mathrm{c}{r_\mathrm{h}}^2\left(\frac{r_\mathrm{h}}{r}\right)\ln\left(1+\frac{r}{r_\mathrm{h}}\right), \tag{2.6}$$

where $\delta_\mathrm{c} = 4.41\times10^5$ is the characteristic density relative to the present-day critical density $\rho_\mathrm{c} = 277.72\,h^2\,M_\odot\mathrm{kpc}^{-2}$, $h = 0.71$ is the Hubble constant, and $r_\mathrm{h} = 7.63\,\mathrm{kpc}$ is the halo scale radius. The total mass of the dark matter halo is $M_{200} = 8.8\times10^{11}M_\odot$ within the virial radius $R_{200} = 195\,\mathrm{kpc}$. The specific parameters used here were carefully determined in Geehan et al. (2006) and Fardal et al. (2006, 2007).

2.2 Initial Condition of Satellite

Thus far, spherical progenitor models of the giant stream assumed a Plummer sphere with scale length 1 kpc or a Hernquist sphere to represent the progenitor (Fardal et al. 2007, 2012, 2013; Mori & Rich 2008; Sadoun et al. 2013). Because King profiles provide a tractable family of models with intuitive parameters that have been fitted extensively to nearby dwarf galaxies (Eskridge 1988a,b; Irwin & Hatzidimitriou 1995; McConnachie & Irwin 2006a), we employ King models with different sizes and density profiles to explore the characteristics of the progenitor of the giant stream. Appendix A provides the detailed description of the King model. Figure 2.1 shows the observed properties of the dwarf galaxies in the Local Group (Irwin & Hatzidimitriou 1995; McConnachie & Irwin 2006a), where M_* is the stellar mass; r_t, the tidal radius; $c \equiv \log_{10} r_\mathrm{t}/r_0$, the concentration parameter; and r_0, the core radius of the King model. Based on these properties in the Local Group (Fig. 2.1) and the Virgo cluster (Ichikawa et al. 1986), we have performed a

parameter study by varying the tidal radius from 0.5 to 6.0 kpc and the concentration parameter from 0.1 to 1.5.

To constrain the masses of the progenitor dwarfs of the giant stellar stream, Mori & Rich (2008) estimated the disk heating by the dynamical friction exerting a force opposite to the orbital motion. As a result, they found that the dynamical mass of the progenitor should be less than $5.2 \times 10^9 M_\odot$, because the disk thickness must agree with the observed thickness of M31 after the interaction of the satellite. In addition, the combination of the mass-metallicity relation of Dekel & Woo (2003) and the recent estimation of the heavy element abundance of the stream [Fe/H] $\gtrsim -1$ (Koch et al. 2008) gives a lower mass limit of $5 \times 10^8 M_\odot$ for the stellar mass of the progenitor. Accordingly, the progenitor dwarfs most likely have a total mass in the range of $5 \times 10^8 M_\odot \lesssim M_{\rm sat} \lesssim 5 \times 10^9 M_\odot$. Considering this estimation, we have run simulations for progenitor masses of $10^9 M_\odot$, $2 \times 10^9 M_\odot$, $3 \times 10^9 M_\odot$, and $5 \times 10^9 M_\odot$.

Fardal et al. (2007) reported an N-body simulation of an accreting dwarf satellite within M31's fixed gravitational potential. They obtained orbital properties that are in good agreement with the observed properties of the giant stream. In addition, their simulation reproduced photometric features that they identified as the "western shelf" and the "northeast shelf." Fardal et al. (2012) presented a correspondence between the kinematics of the observed "western shelf" and that of the simulated one. Mori & Rich (2008) also well-reproduced these features using the same initial orbital elements of the progenitor in the case of a full self-gravitating system with a live disk, bulge, and dark matter halo. We believe that Fardal's orbit is quite likely to be the actual orbit of the progenitor; however, the uniqueness of this orbit has not yet been proven. We will investigate this point in detail by performing a large, and systematic parameter study focused on the infalling orbit of the progenitor in Part III. With this fact in mind, we adopt Fardal's orbit in this study, and therefore, the initial position vector and velocity vector for the standard coordinates centered on M31 are $(-34.75,\ 19.37,\ -13.99)\,{\rm kpc}$ and $(67.34,\ -26.12,\ 13.50)\,{\rm km\,s^{-1}}$.

Chapter 3 Simulation Results

We have calculated 247 models in total: 49 for $M_{\rm sat} = 10^9 M_\odot$, 87 for $M_{\rm sat} = 2 \times 10^9 M_\odot$, 57 for $M_{\rm sat} = 3 \times 10^9 M_\odot$, 53 for $M_{\rm sat} = 5 \times 10^9 M_\odot$, and a Plummer model with a total mass of $M_{\rm sat} = 2 \times 10^9 M_\odot$ and effective radius of 1 kpc to check the consistency with Fardal et al. (2007). Each model uses 65,536 particles to represent the King sphere, and the gravitational softening parameter is adopted as $\epsilon = r_0/8$, which is sufficient to resolve the core radius r_0. The direct N-body integration by the second-order Runge-Kutta method with an adaptive time step has been performed on the FIRST simulator at the Center for Computational Sciences, University of Tsukuba. The FIRST simulator is a hybrid PC cluster with 512 Intel Xeon processors and 240 boards of Blade-GRAPE acceleration engine for the gravity calculation. Blade-GRAPE, a specially developed board for gravity calculations, comprises 4 dedicated GRAPE-6 chips (Makino et al. 2003; Fukushige et al. 2005) that are based on a pipeline architecture.

3.1 Dynamical Evolution of Satellites

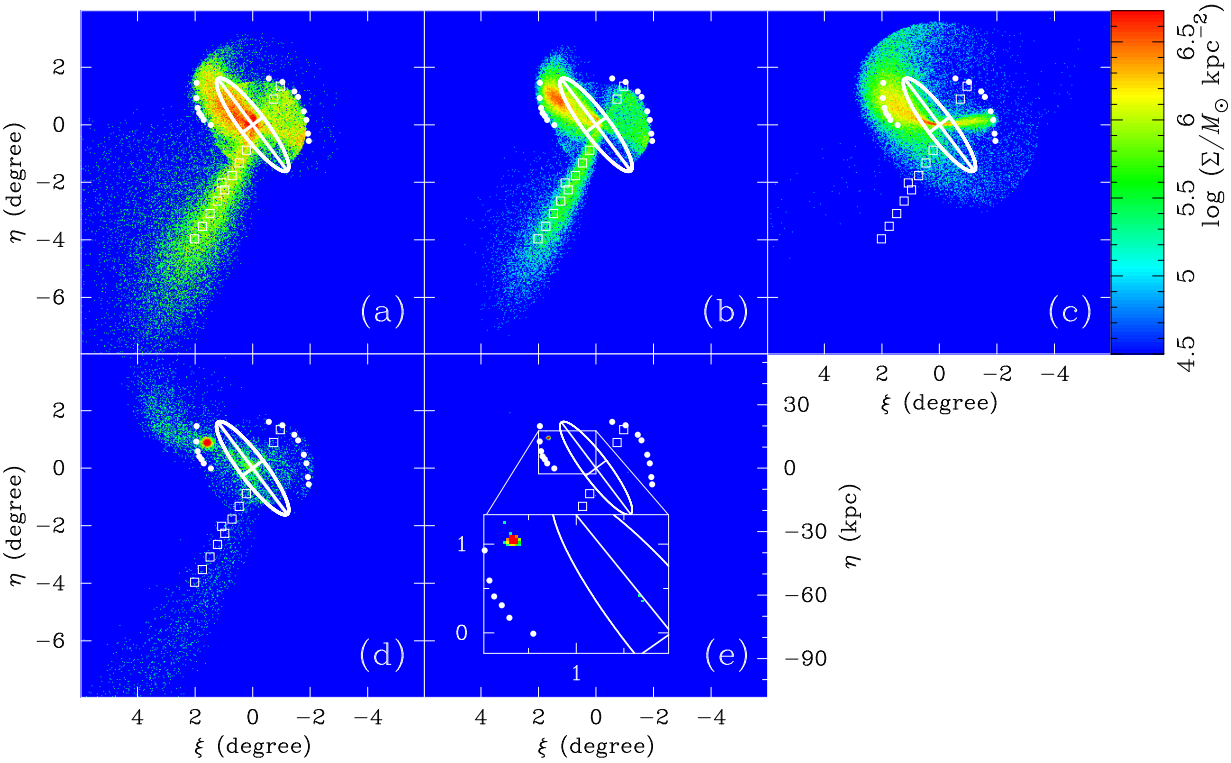


Fig. 3.1: Projected mass-density distribution of the tidal debris for (a) $M_{\rm sat} = 3 \times 10^9 M_\odot$, $c = 0.7$, $r_{\rm t} = 4.5\,{\rm kpc}$, (b) $M_{\rm sat} = 1 \times 10^9 M_\odot$, $c = 0.5$, $r_{\rm t} = 2.0\,{\rm kpc}$, (c) $M_{\rm sat} = 1 \times 10^9 M_\odot$, $c = 0.1$, $r_{\rm t} = 6.0\,{\rm kpc}$, (d) $M_{\rm sat} = 2 \times 10^9 M_\odot$, $c = 1.1$, $r_{\rm t} = 1.5\,{\rm kpc}$, and (e) $M_{\rm sat} = 2 \times 10^9 M_\odot$, $c = 1.1$, $r_{\rm t} = 0.5\,{\rm kpc}$, respectively. Filled circles and open squares show the position of the edge of the shells (Fardal et al. 2007) and the observed areas of the Andromeda Giant Stellar Stream (Font et al. 2006), respectively. Ellipse in each panel corresponds the size of the M31's disk.

Table. 3.1: Parameters of fiducial models

model/panel	$M_{\rm sat}$ ($M_\odot$)	c	W_0 [(1)]	$r_{\rm t}$ (kpc)	r_0 (kpc)	σ_0 (km s^{-1}) [(2)]	ρ_0 ($M_\odot$ pc^{-3}) [(3)]	Σ_0 ($M_\odot$ pc^{-2}) [(4)]	$\Phi_{\rm r0}$ (erg g^{-1}) [(5)]
A	3×10^9	0.7	3.0	4.5	0.96	49.1	6.6×10^{-1}	9.6×10^2	-1.0×10^{14}
B	1×10^9	0.5	1.9	2.0	0.65	40.7	1.4	1.1×10^3	-5.8×10^{13}
C	1×10^9	0.1	0.44	6.0	4.90	22.6	2.9×10^{-2}	9.1×10	-8.8×10^{12}
D	2×10^9	1.1	5.3	1.5	0.12	86.2	9.4×10	2.1×10^4	-4.2×10^{14}
E	2×10^9	1.1	5.3	0.5	0.04	149.3	2.5×10^3	1.9×10^5	-1.2×10^{15}

(1) Dimensionless King parameter at the center of the satellite.

(2) One-dimensional velocity dispersion at the center of the satellite.

(3) Mass density at the center of the satellite.

(4) Column mass density at the center of the satellite.

(5) Potential at the core radius r_0.

Figure 3.1 shows the projected particle positions for typical simulation results with different masses, tidal radii, and concentration parameters. Table 3.1 lists the corresponding parameters for each panel in Fig. 3.1. The ellipsoid in each panel indicates M31's disk size. The edge of the shells defined in Fardal et al. (2007) and the observed areas along the giant stream are indicated by filled circles and open squares, respectively. The first pericentric passage close to the galactic center occurred in 0.8 Gyr ago, and the satellite collided almost head-on with the bulge. The distribution of satellite particles subsequently suffered tidal deformation and stretched out catastrophically in Models A, B, and C. In these models, this debris, while keeping a narrow distribution, expands to a great distance because a large fraction of the satellite particles acquires a high velocity relative to the center of M31. This creates the giant stream. After the second pericentric passage, stellar particles that initially constituted the satellite start to spread out in a fan-like form. A double shell system with roughly constant curvature is sharply defined as seen in Figs. 3.1a, 3.1b, and 3.1c.

Models A and B successfully reproduce the stream and the shells at the east and the west sides of M31. In contrast, Model C does not reproduce the observed structures very well because the stellar stream in Fig. 3.1c is considerably shorter than the observed giant stream that extends out to over 100 kpc from the center of M31 (McConnachie et al. 2003). Furthermore, in this model, the shells have a narrower fan shape than do the observed ones, which have a large central angle. The central angle of the flagellum depends on the velocity dispersion of the progenitor satellite. Head-on collisions of the satellite with the shallower gravitational potential well generate the fan-like debris with the smaller central angle. Because the progenitor of Model C is a less massive fluffy galaxy with a larger core radius, it has a shallower gravitational potential and smaller velocity dispersion than those in Models A and B. Thus, the bunch of stars that are tidally stripped by M31's gravitational potential are not as spread out as in the observed fan-like structures.

Figures 3.1d and 3.1e show the results of Models D and E, respectively. In both cases, the gravitational potential of the satellites is deeper than that in previous models because the progenitor has an appreciably small tidal radius. The distribution of satellite particles in Model D subsequently undergoes a tidal stripping after the first pericentric passage of the galactic center. The debris is drawn out into a long tail similar to the giant stream, but its stellar density is quite low. The stellar particles spread to form double shells in the same manner after the second pericentric passage. The shape of the shells is, however, quite different from the observed structures, and a high-density core of the progenitor still survives at $\xi \sim 2^\circ$ and $\eta \sim 1^\circ$, which is undetectable by the observations. In an extreme case such as that in Model E, there are scarcely any tidal effects on such a granitic satellite with the deep gravitational potential well.

Surface brightness maps in V-band for the typical results and the observed data (Irwin et al. 2005) are shown in Fig. 3.2. Here, we assume that the mass fraction of red giant branch (RGB) stars is 8% using Salpeter's initial mass function. Then, we introduce a mass-to-light ratio for the observed flux so that the giant stream luminosity agrees with the observed luminosity $M_{\rm V} \approx -14$ (Ibata et al. 2001). Figures 3.2a and 3.2b nicely reproduce the observed features such as the Andromeda Stellar Stream and the shell-like structures. However, only Fig. 3.2a shows a clear third shell component. Other three panels show that these models faild to reproduce the observed features.

Figure 3.3 shows the radial profiles of the satellite models listed in Tab. 3.1. The models succeeded to reproduce the observed structures (Models A and B) have a same degree of the volume density ($\sim 1\ M_\odot$ pc^{-3}), the column density ($\sim 10^3\ M_\odot$ pc^{-2}) and the radial velocity dispersion (~ 45 km s^{-1}) in the central region. On the other hand, the central velocity dispersion of Model C is too small and that of Models D and E are too large compared to the above models. This difference corresponds to the difference about the depth of the gravitational potential well. Figure 3.4 compares the radial profiles of the satellite models with that of the Plummer models adopted in Fardal et al. (2007, 2012). The both Plummer models have similar profiles with Model A in this study.

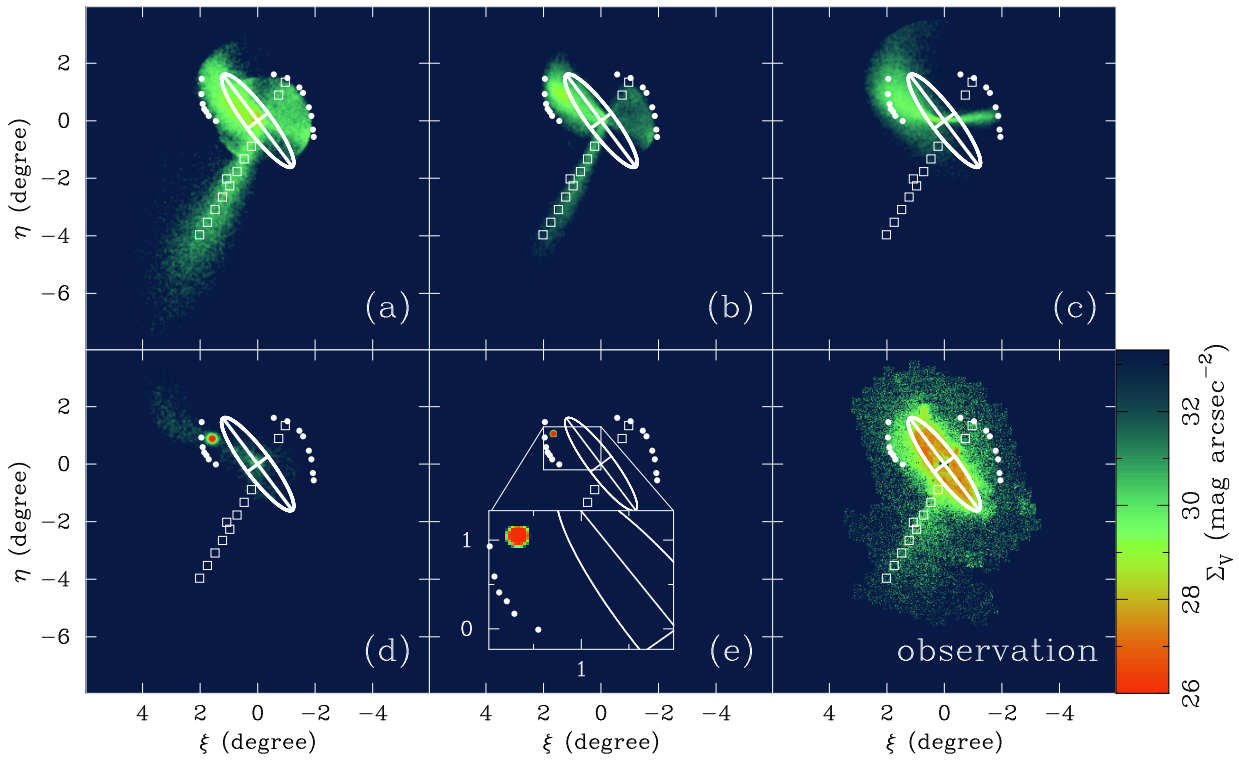
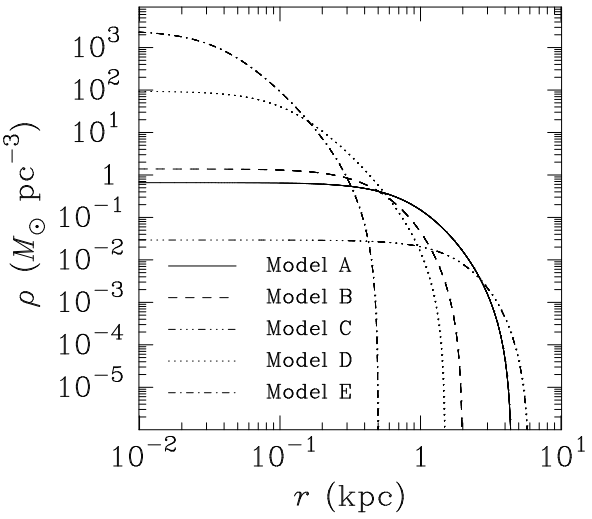
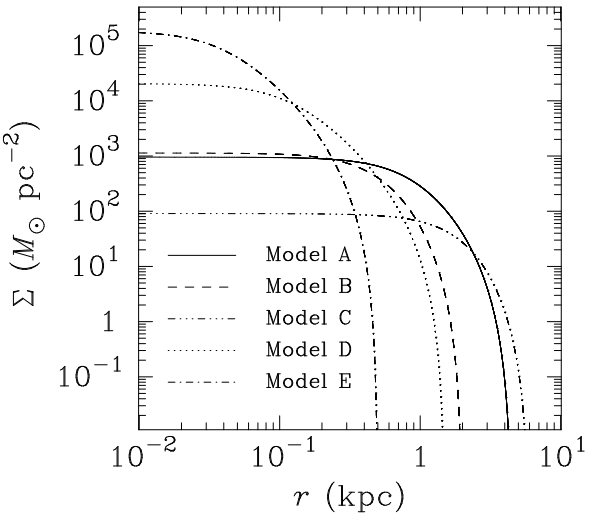


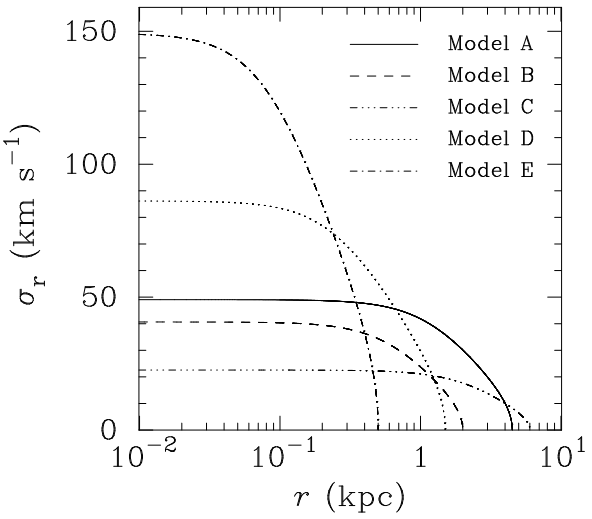
Fig. 3.2: Surface brightness maps. The lower-right panel shows a surface brightness map of the RGB stars around M31 observed by Irwin et al. (2005). The other panels show V-band surface brightness map derived by our simulation results for (a) $M_{\rm sat} = 3 \times 10^9 M_\odot$, $c = 0.7$, $r_{\rm t} = 4.5\,{\rm kpc}$, (b) $M_{\rm sat} = 1 \times 10^9 M_\odot$, $c = 0.5$, $r_{\rm t} = 2.0\,{\rm kpc}$, (c) $M_{\rm sat} = 1 \times 10^9 M_\odot$, $c = 0.1$, $r_{\rm t} = 6.0\,{\rm kpc}$, (d) $M_{\rm sat} = 2 \times 10^9 M_\odot$, $c = 1.1$, $r_{\rm t} = 1.5\,{\rm kpc}$, and (e) $M_{\rm sat} = 2 \times 10^9 M_\odot$, $c = 1.1$, $r_{\rm t} = 0.5\,{\rm kpc}$, respectively. Filled circles and open squares show the position of the edge of the shells (Fardal et al. 2007) and the observed areas of the Andromeda Giant Stellar Stream (Font et al. 2006), respectively. Ellipse in each panel corresponds the size of the M31's disk.



(a) Volume density.



(b) Column density.



(c) Radial velocity dispersion.

Fig. 3.3: Radial profile of satellite models listed in Tab. 3.1. The solid, dashed, triple-dot-dashed, dotted, and dot-dashed curves represent radial profiles for Models A, B, C, D, and E, respectively.

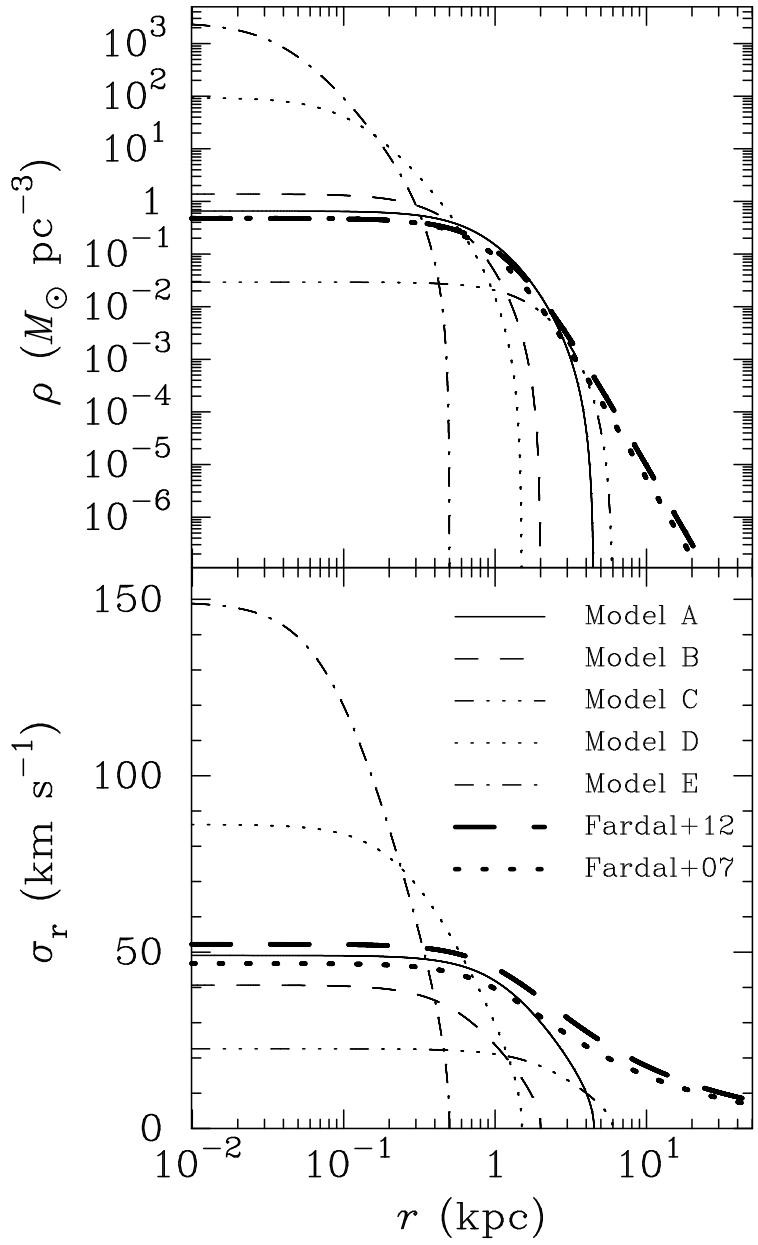


Fig. 3.4: Radial profile of satellite models. The top and the bottle panel reveal the mass density profile and the radial velocity dispersion profile, respectively. Line styles for King models are identical to that in Fig. 3.3. The bold dotted and dashed curves show the Plummer profiles adopted in Fardal et al. (2007) and Fardal et al. (2012), respectively.

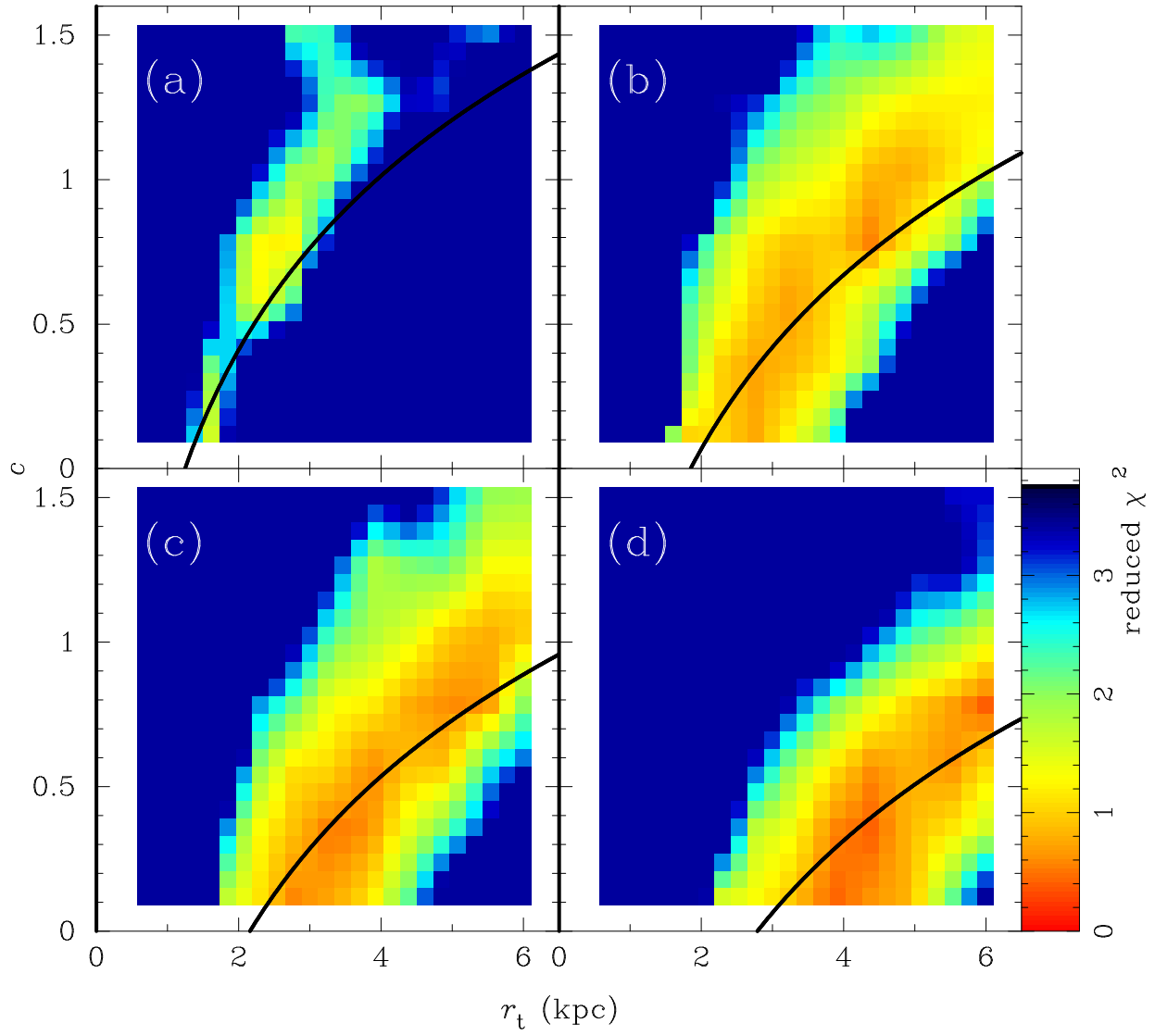


Fig. 3.5: Reduced χ^2 maps of the shapes of the both shells for (a) $M_{\rm sat} = 10^9 M_\odot$, (b) $M_{\rm sat} = 2 \times 10^9 M_\odot$, (c) $M_{\rm sat} = 3 \times 10^9 M_\odot$, and (d) $M_{\rm sat} = 5 \times 10^9 M_\odot$, respectively. The horizontal axis is the tidal radius $r_{\rm t}$ of the progenitor dwarf galaxies, and the vertical axis is the concentration parameter c. Solid line indicates the empirical relationship of the possible progenitor (see text).

3.2 Mock Images of Simulated Tidal Debris

In this section, we show qualitative comparisons between the simulation results and the observations. First, we have tested whether the velocity structure of the giant stellar stream is reproduced within an error range of 3σ; the data were referred from Table 1 in Font et al. (2006), however we do not include the data of Field 8 because of the considerable contamination of the M31's disk components. Second, we have checked the shapes of the northeast and the west shells. The positions of each shell's edge were referred from Table 1 in Fardal et al. (2007). The width of each shell's edge is estimated to be $0^\circ.11$ from the star count map in Irwin et al. (2005). Then, we obtain reduced χ^2 maps of the shapes of the shells in the parameter space of $r_{\rm t}$ and c. In Fig. 3.5, possible parameter regions of the progenitors are indicated by red or orange regions, which have a small, reduced χ^2. "Possible" regions are distributed from low $r_{\rm t}$ – low c regions to high $r_{\rm t}$ – high c regions.

The interpretation of results in the previous section suggests that the key quantity to discriminate success

or fail to reproduce the observed structures is the degree of gravitational binding (i.e., the potential of the satellite). Here, we show this expectation explains the results well. First, the potential depends on the mass and the size of the satellite. The potential is given by $\Phi = -GM/R$, where R is the typical radius (r_0 for the King model) and M is the typical mass. From this relationship, R should be proportional to M to keep the potential constant. Second, the potential also depends on the mass distribution profile of the satellite. If r_t increases, the potential decreases. To conserve the potential of the central region (the most significant radius is the Hill radius: it determines whether stars are bounded or stripped), the central density must increase. This implies that r_0 must decrease against the increase of r_t. Combining the above two dependence, we get a theoretical prospect as

$$r_0 = a \times M_\mathrm{sat} \times {r_\mathrm{t}}^{-1}, \tag{3.1}$$

where a is a remaining fitting parameter. By fitting a for "possible" regions in Fig. 3.5, an empirical relationship (the black line in the figure) is derived as

$$r_0 = 1.0\,\mathrm{kpc} \times \left(\frac{M_\mathrm{sat}}{3 \times 10^9\,M_\odot}\right) \times \left(\frac{r_\mathrm{t}}{4.5\,\mathrm{kpc}}\right)^{-1}. \tag{3.2}$$

The relationship agrees with the results of N-body simulations. Therefore, we conclude that the potential of the satellite is the key quantity to explain the dependence of "possible" regions on M_sat, r_t, and c.

From the above discussion, we confirm "possible" progenitors have the same degree of potential. If the binding is too strong, the progenitors cannot be broken. Even if they are broken, the number of stripped stars is not sufficient to explain the observed luminosity of the structures. If the binding is too weak, then stripped stars cannot spread sufficiently widely to reproduce the observed structures. This is because the stars of such progenitors have small velocity dispersion to support the weak gravitational potential. Therefore, only progenitors have a suitable degree of the binding potential reproduce the observed structures. The relationship between the progenitor's mass and the area of the "possible" domain in the parameter space can be understood from this explanation. If the mass of a progenitor increases, then the gravitational potential increases. Then, the size of the progenitor must be larger to reproduce the observed structures. Therefore, the width of "possible" regions increases if the mass of the progenitor increases.

Chapter 4 Discussion

We compare the empirical relation derived in the previous chapter with the observed properties of nearby galaxies in Section 4.1. In Section 4.2, we examine the velocity structure of the stream in detail.

4.1 Implication of Nearby Dwarf Galaxies

We compare the empirical relation (Eq. (3.2)) with the observed properties of nearby galaxies in Fig. 4.1. The figure plots V-band surface brightness $\mu_{\rm V}$ of nearby galaxies as a function of effective radius $R_{\rm eff}$. An orange band in the horizontal direction shows the empirical relation (Eq. (3.2) for $c = 0.7$ (corresponds to model A at $M_{\rm sat} = 3 \times 10^9 M_\odot$). The empirical relation itself implies that the surface density of the "possible" progenitor galaxy in the central region is independent of the mass to keep the strength of the gravitational binding. Indeed, the central surface density for Models A and B are around $10^3\ M_\odot\ {\rm pc}^{-2}$ (see Tab. 3.1 and Fig. 3.3b). In Fig. 4.1, we assume that the mass-to-light ratio is independent of the mass for simplicity: the width of the band corresponds to 1σ scatter of Faber-Jackson relation to Model A (Toloba et al. 2012; Falcón-Barroso et al. 2011). The orange band contains many observed galaxies; therefore, we conclude that the empirical relation stays in a realistic parameter region. On the other hand, Mori & Rich (2008) derived the mass range of the progenitor dwarf galaxy as $5 \times 10^8 M_\odot \leq M_{\rm sat} \leq 5 \times 10^9 M_\odot$ (a green band in Fig. 4.1). The yellow rectangle in Fig. 4.1 shows an overlapped region of the orange and green bands, which means "possible" parameter region for the progenitor dwarf galaxy of the observed structures. Fig. 4.1 clearly shows that some of the nearby dwarf galaxies have similar photometric properties with the progenitor dwarf galaxy of the observed structures.

Here, we discuss the morphology of the progenitor dwarf galaxy. The yellow rectangle in Fig. 4.1 contains not only dwarf elliptical galaxies, but also dwarf irregulars and small spiral galaxies. We assume the spherical galaxy in this paper; however, the progenitor dwarf galaxy of the observed structures is possibly dwarf irregular or dwarf spiral galaxy. Such a morphological difference can cause the different structures of the tidal debris after the collision with M31, and some of them might become a crucial point to solve current mismatches of N-body simulations with observations (for example, bimodality of the giant stream to be discussed in Section 4.2.3). Fardal et al. (2008) and Sadoun et al. (2013) showed that infall model of a dwarf spiral galaxy toward M31 also reproduces the observed structures well. Kirihara et al. (in preparation) have investigated the relationship between angular momentum of an infalling spiral galaxy and resultant structures and underlying physical mechanisms.

Figures 4.2, 4.3, and 4.4 are essentially same with Fig. 4.1; however, show various dependence on some observable quantities. All the figures clearly reveal that some of the nearby dwarf galaxies have similar photometric properties with the progenitor dwarf galaxy of the observed structures.

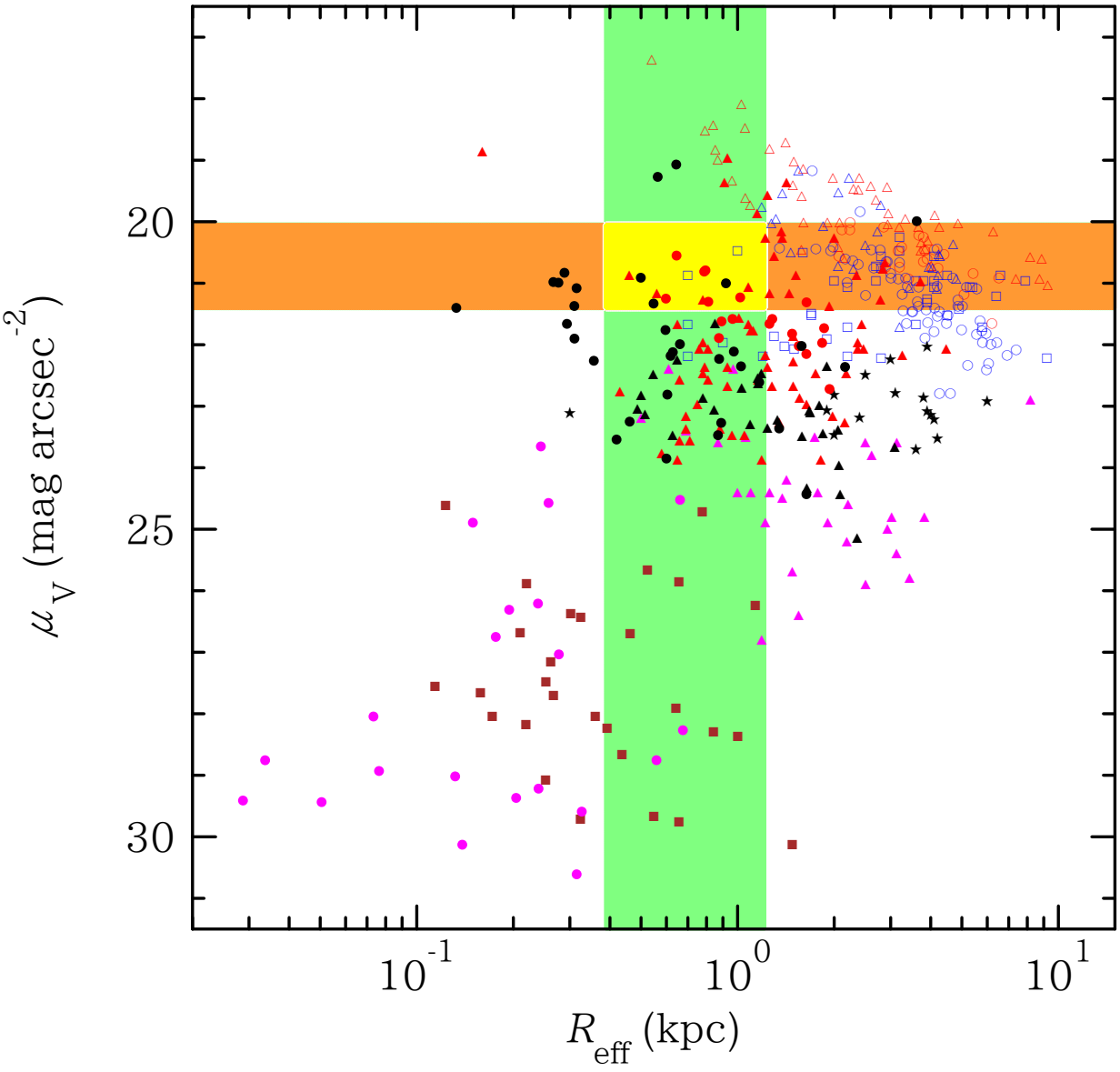


Fig. 4.1: V-band surface brightness $\mu_{\rm V}$ as a function of effective radius $R_{\rm eff}$ (Kormendy relation in V-band). Open symbols are the locations of observed galaxies: E/S0 galaxies (red circles for Kent (1984) and red triangles for Falcón-Barroso et al. (2011)) and spiral galaxies (blue circles for Kent (1984), blue squares for van der Kruit (1987), and blue triangles for Falcón-Barroso et al. (2011)). Stars and magenta triangles show the properties of low surface brightness galaxies studied by de Blok et al. (1995) and Matthews et al. (1998), respectively. Rest symbols show the observed properties for various types of dwarf galaxies: dwarf ellipticals in the Virgo cluster observed by Toloba et al. (2011, 2012, red filled circles), dwarf ellipticals in the Coma cluster observed by Kourkchi et al. (2012b,a, red filled triangles), nearby dwarf galaxies observed by Makarova (1999, black filled circles; most of them are dwarf irregulars), low luminosity dwarf irregulars in the Virgo cluster observed by Heller & Brosch (2001, black filled triangles), dwarf spheroidals in the MW halo (Brasseur et al. 2011; Wolf et al. 2010, magenta circles), and all known Andromeda dwarf spheroidals compiled by Collins et al. (2013, brown squares). An orange band shows the empirical relation (Eq. (3.2): solid curves in Fig. 3.5) under an assumption of a constant mass-to-light ratio with 1σ scatter of Faber-Jackson relation to Model A (Toloba et al. 2012; Falcón-Barroso et al. 2011). A green hatched region is the mass range for the progenitor dwarf galaxy derived by Mori & Rich (2008). Parameter space within a yellow hatched region shows the physical properties of the possible progenitor satellites.

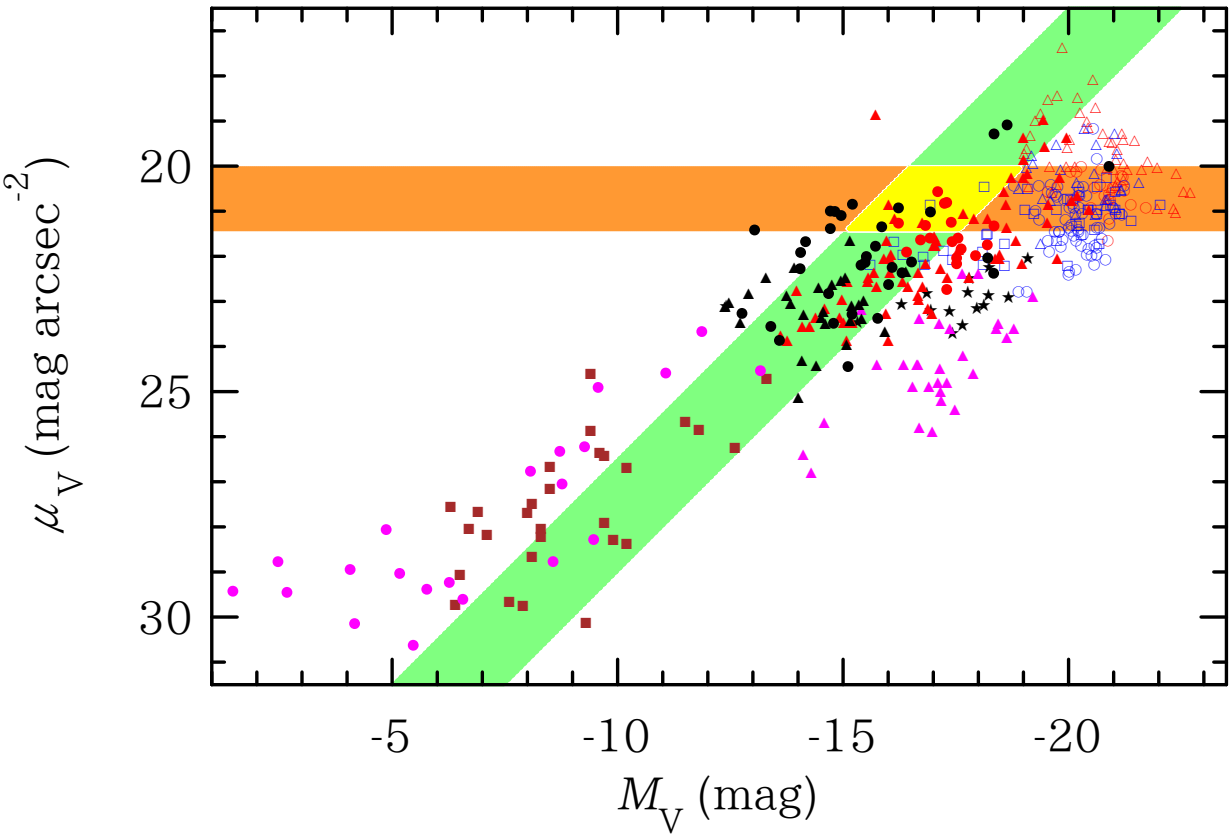


Fig. 4.2: V-band magnitude as a function of V-band surface brightness. All symbols and bands are same with that in Fig. 4.1.

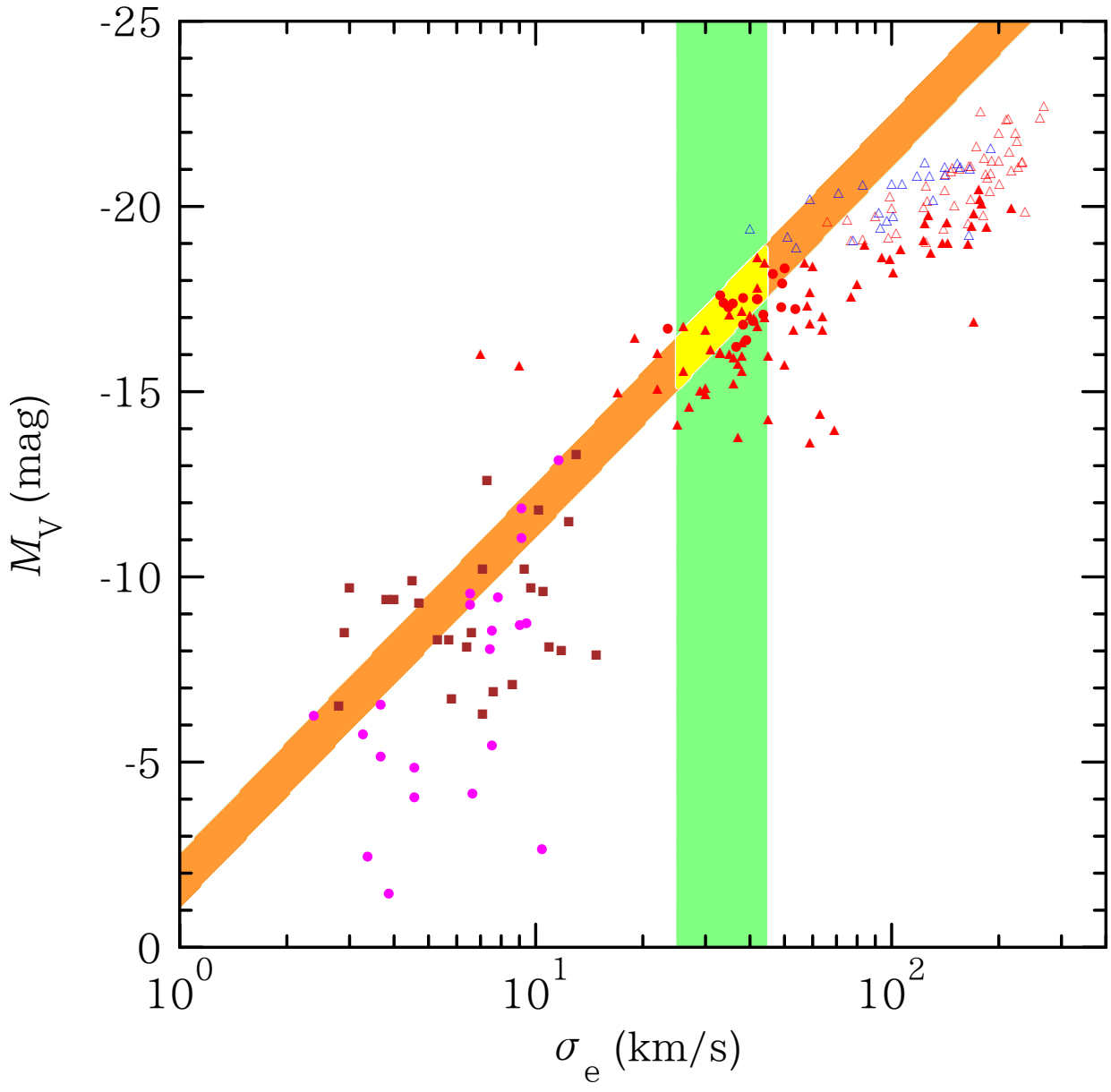


Fig. 4.3: V-band magnitude as a function of one-dimensional velocity dispersion. All symbols and bands are same with that in Fig. 4.1.

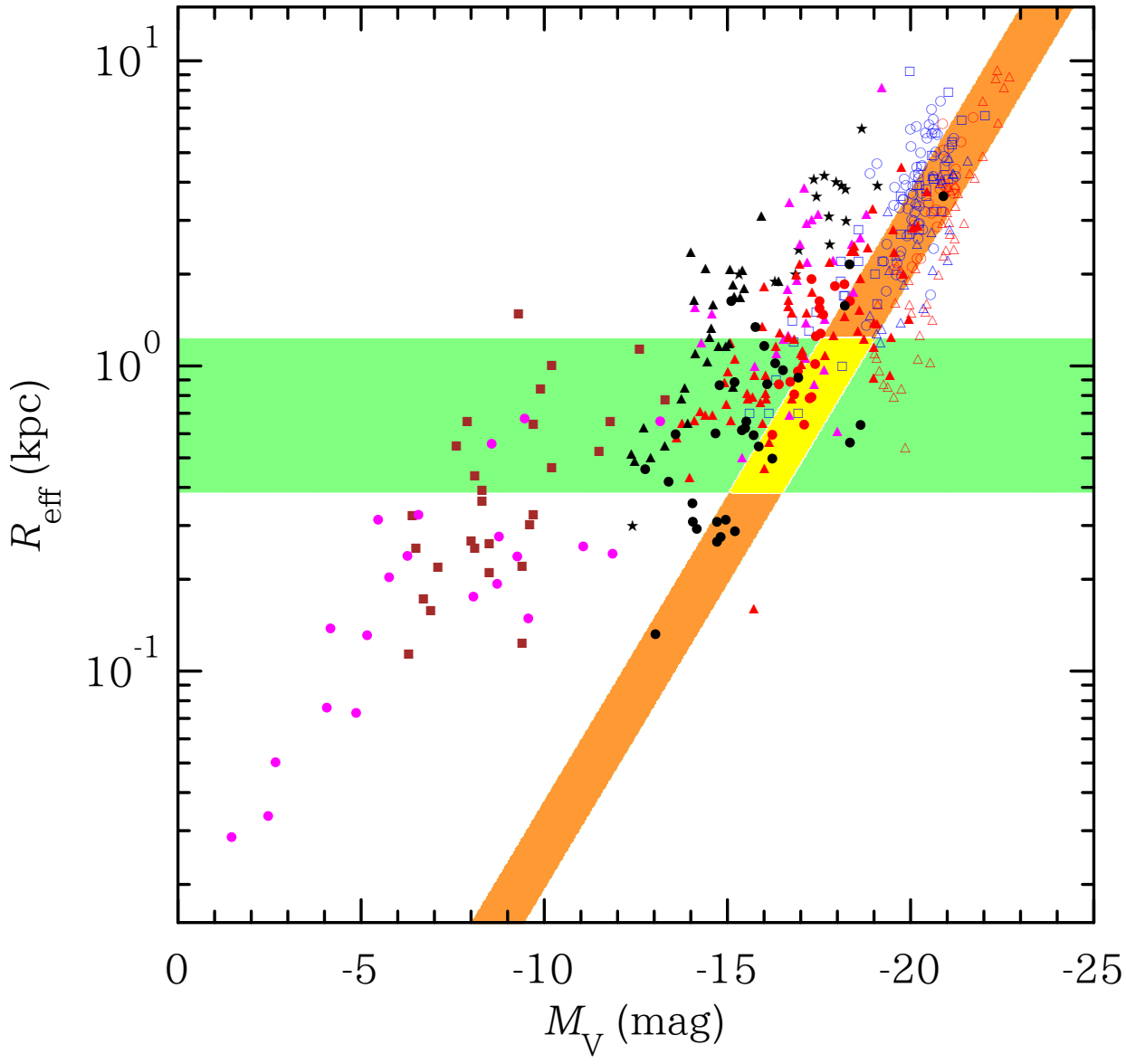


Fig. 4.4: Effective radius as a function of V-band magnitude. All symbols and bands are same with that in Fig. 4.1.

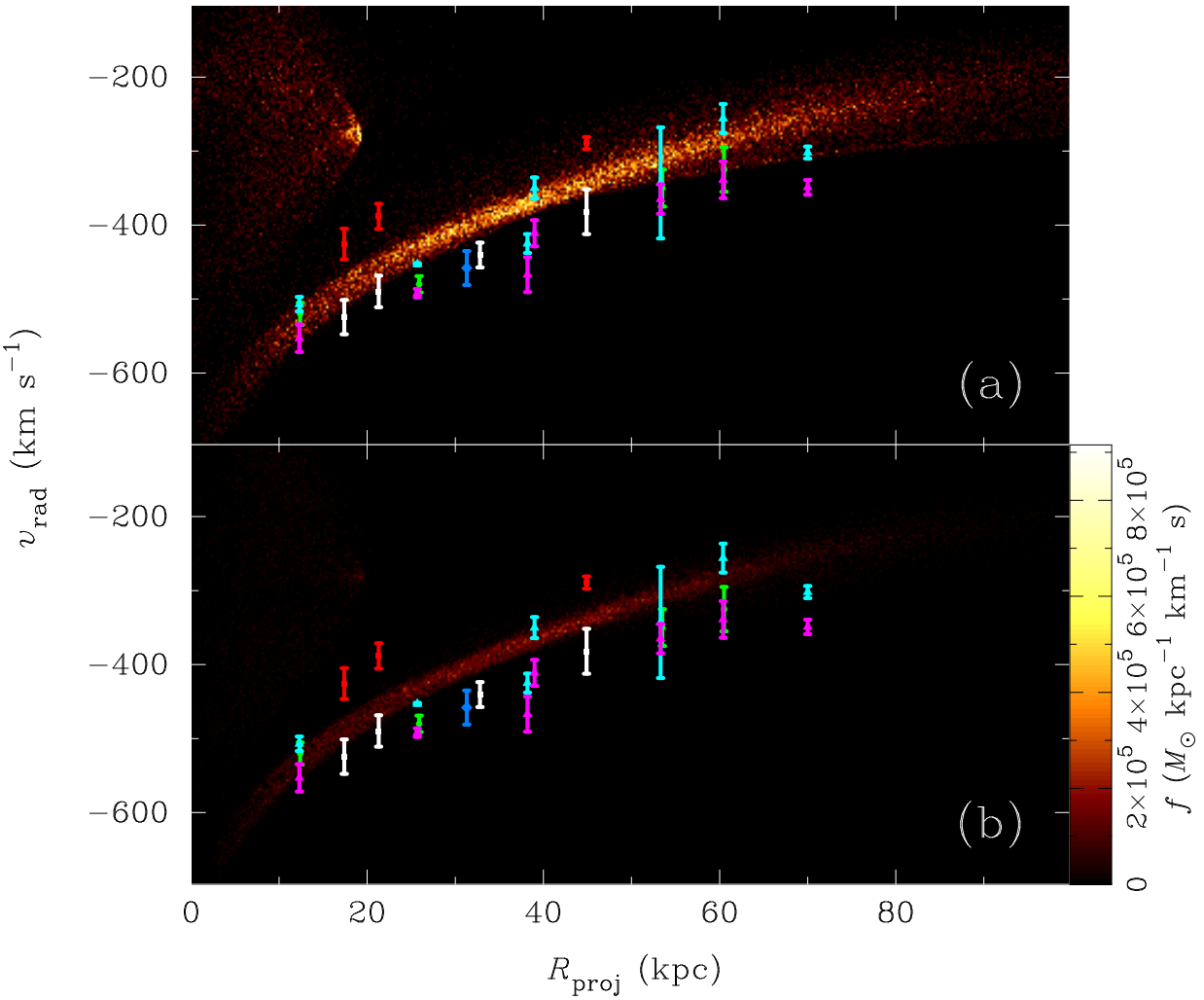


Fig. 4.5: Phase space density maps of the giant stream. Horizontal and vertical axis are the projected distance from the M31 center and the line-of-sight velocity, respectively. The top and bottom panels correspond to Models A and B, respectively. Over-plotted symbols show the observed velocity structure. Green circles represent data observed by Ibata et al. (2004), analyzed by Ferguson et al. (2004) and Kalirai et al. (2006a). A blue diamond shows the observed line-of-sight velocity by Guhathakurta et al. (2006). White or red squares indicate the results of multiple component fit in various observation fields by Gilbert et al. (2009). Cyan or magenta triangles are the analyzed results of double Gaussian fit by Trethewey et al. (in preparation).

4.2 Velocity Structures

In this section, we examine the velocity structure of the giant stream in detail. First, we compare the global structure of the N-body simulations with observations in Section 4.2.1. Then, we focus on individual fields where the detailed velocity structure were observed (Sections 4.2.2 and 4.2.3).

4.2.1 Global Structure

Figure 4.5 shows the velocity structure of the giant stream. Color maps in the figure show the phase space density of the giant stream, and colored symbols represent the observations (Ibata et al. 2004; Ferguson et al. 2004; Kalirai et al. 2006a; Gilbert et al. 2009; Trethewey et al. in preparation). The both panels in Fig. 4.5

well reproduce the observed velocity structure. This correspondence between the N-body simulations and the observations indicates that the simulations reproduce the kinematic properties of the observations. Additional component appeared in Fig. 4.5a which have a density peak around $R_{\rm proj} \sim 20$ kpc and $v_{\rm rad} \sim -300$ km s^{-1} is considered to be the third shell component discussed in the next subsection.

4.2.2 Third Shell Component

In addition to the giant stellar stream and the two stellar shells, Fardal et al. (2007) pointed out that there is the third shell structure originating from the same progenitor. A similar structure was also reported in Mori & Rich (2008). They showed that the third shell component is a forward continuation of the giant stellar stream. In our results of simulations, many parameter sets indicate the third shell component, and these explain the giant stellar stream, the northeast shell, and the west shell (see Fig. 3.1a: $M_{\rm sat} = 3 \times 10^9 M_\odot$, $c = 0.7$, $r_{\rm t} = 4.5\,{\rm kpc}$). However, some parameter sets also nicely reproduce the giant stellar stream, northeast shell, and west shell without the third shell (see Fig. 3.1b: $M_{\rm sat} = 1 \times 10^9 M_\odot$, $c = 0.5$, $r_{\rm t} = 2.0\,{\rm kpc}$). This is clearly different from earlier studies, and it suggests that the observed third shell component might not be a forward continuation of the giant stellar stream. It is important that both parameter sets explain the giant stellar stream, the northeast shell, and the west shell.

To clarify our statement, we have compared the velocity distribution of both cases. Figures 4.6 and 4.7 shows the radial velocity histogram around fields f135 and f207, respectively. The centers of the both fields are $\xi = 0\overset{\circ}{.}9$, $\eta = -1\overset{\circ}{.}1$ (field f135) and $\xi = 0\overset{\circ}{.}2$, $\eta = -1\overset{\circ}{.}3$ (field f207). Figures 4.6 and 4.7 can be compared with Fig. 15 in Gilbert et al. (2007) and Fig. 6 in Gilbert et al. (2009), respectively. The top panel shows a histogram of Model A, and the bottom shows that of Model B. They are fitted by the Kayes mixture-modeling (KMM) algorithm proposed in Ashman et al. (1994). Table 4.1 lists the fitting results of Models A and B in fields f135 and f207. The radial velocity of M31 is $-300\,{\rm km\,s^{-1}}$, with the negative sign implying that the direction of motion is toward us.

Gilbert et al. (2007) reported two more components exist besides the inner spheroid of M31 in field f135: 25% of the total population has $v_{\rm rad} = -449 \pm 55\,{\rm km\,s^{-1}}$, and another 30% has $v_{\rm rad} = -273 \pm 30\,{\rm km\,s^{-1}}$. In Fig. 4.6, two components are clearly observed in Model A: 53.5% are $v_{\rm rad} = -464 \pm 32\,{\rm km\,s^{-1}}$ (solid curve), 46.5% are $v_{\rm rad} = -274 \pm 41\,{\rm km\,s^{-1}}$ (dashed curve). The former is a component of the giant stellar stream, and the latter is considered as the third shell component. The simulated result well explains the observation by Gilbert et al. (2007), except a small difference of the contrast over the giant stream component (0.87 is Model A while 1.2 is the observed value). As shown in Fig. 4.6b, there also exists the giant stellar stream component ($v_{\rm rad} = -466 \pm 23\,{\rm km\,s^{-1}}$, solid curve) with a fraction of 71.9%. The dashed curve in Fig. 4.6b is the kinematically hot component ($v_{\rm rad} = -288 \pm 36\,{\rm km\,s^{-1}}$); this is unlikely the third shell component.

The same analysis in field f207 exhibits similar results (Fig. 4.7). Gilbert et al. (2009) reported two more components exist besides the inner spheroid of M31 in field f207: 31% of the total population has $v_{\rm rad} = -524 \pm 23\,{\rm km\,s^{-1}}$, and another 31% has $v_{\rm rad} = -426 \pm 21\,{\rm km\,s^{-1}}$. Model A has two components as same in field f135: 75.7% are $v_{\rm rad} = -484 \pm 37\,{\rm km\,s^{-1}}$ (solid curve), 24.3% are $v_{\rm rad} = -287 \pm 44\,{\rm km\,s^{-1}}$

Table. 4.1: Fitting results based on the KMM algorithm

field	model	primary component			secondary component		
		$\langle v_{\rm rad}\rangle$ (km s^{-1})	σ (km s^{-1})	fraction (%)	$\langle v_{\rm rad}\rangle$ (km s^{-1})	σ (km s^{-1})	fraction (%)
f135	A	-464	32	53.5	-274	41	46.5
	B	-466	23	71.9	-288	36	28.1
f207	A	-484	37	75.7	-287	44	24.3
	B	-488	36	97.9	-302	36	2.1

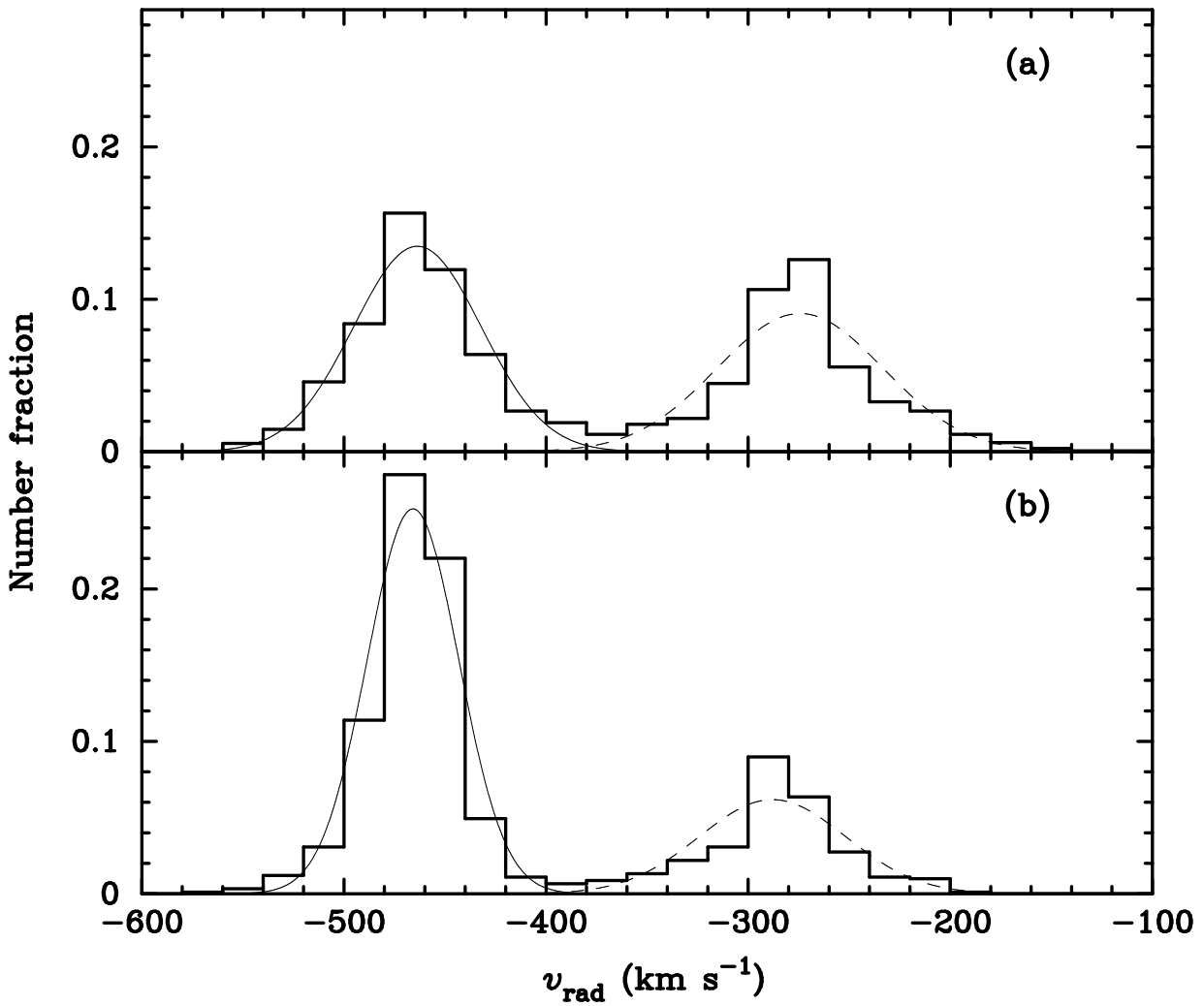


Fig. 4.6: Histogram of the radial velocity around field f135 (Gilbert et al. 2007). Solid and dashed curve show the fitting results based on the KMM algorithm for the primary and secondary component, respectively. In the upper panel (Model A), there are two components: 53.5% are $v_{\rm rad} = -464 \pm 32\,{\rm km\,s^{-1}}$ (solid curve), 46.5% are $v_{\rm rad} = -274 \pm 41\,{\rm km\,s^{-1}}$ (dashed curve). In the bottom panel (Model B), there are also two components: 71.9% are $v_{\rm rad} = -466 \pm 23\,{\rm km\,s^{-1}}$ (solid curve), 28.1% are $v_{\rm rad} = -288 \pm 36\,{\rm km\,s^{-1}}$ (dashed curve).

(dashed curve). On the other hand, Model B has only a single component ($v_{\rm rad} = -488 \pm 33\,{\rm km\,s^{-1}}$, solid curve), and there is no clear third shell component.

Gilbert et al. (2007, 2009) observed the velocity distribution of RGB stars in these regions to verify the existence of the third-shell structure. The result shows there are two components: a giant stellar stream component and a kinematically cold component. This new component is nearly consistent with the results of Fardal et al. (2007) (position, kinematic trends, and [Fe/H] distribution); however, the contrast between the two components does not match. In field f135, the contrast in the simulations is much greater than that in the observation by Gilbert et al. (2007). Furthermore, even if the [Fe/H] distribution is consistent with the stream component, the two components could have different origins. More observations such as the [Fe/Mg] distribution will be required to determine whether the two components have the same or different origins, and it is important to analyse the simulation results in these regions precisely for future comparison.

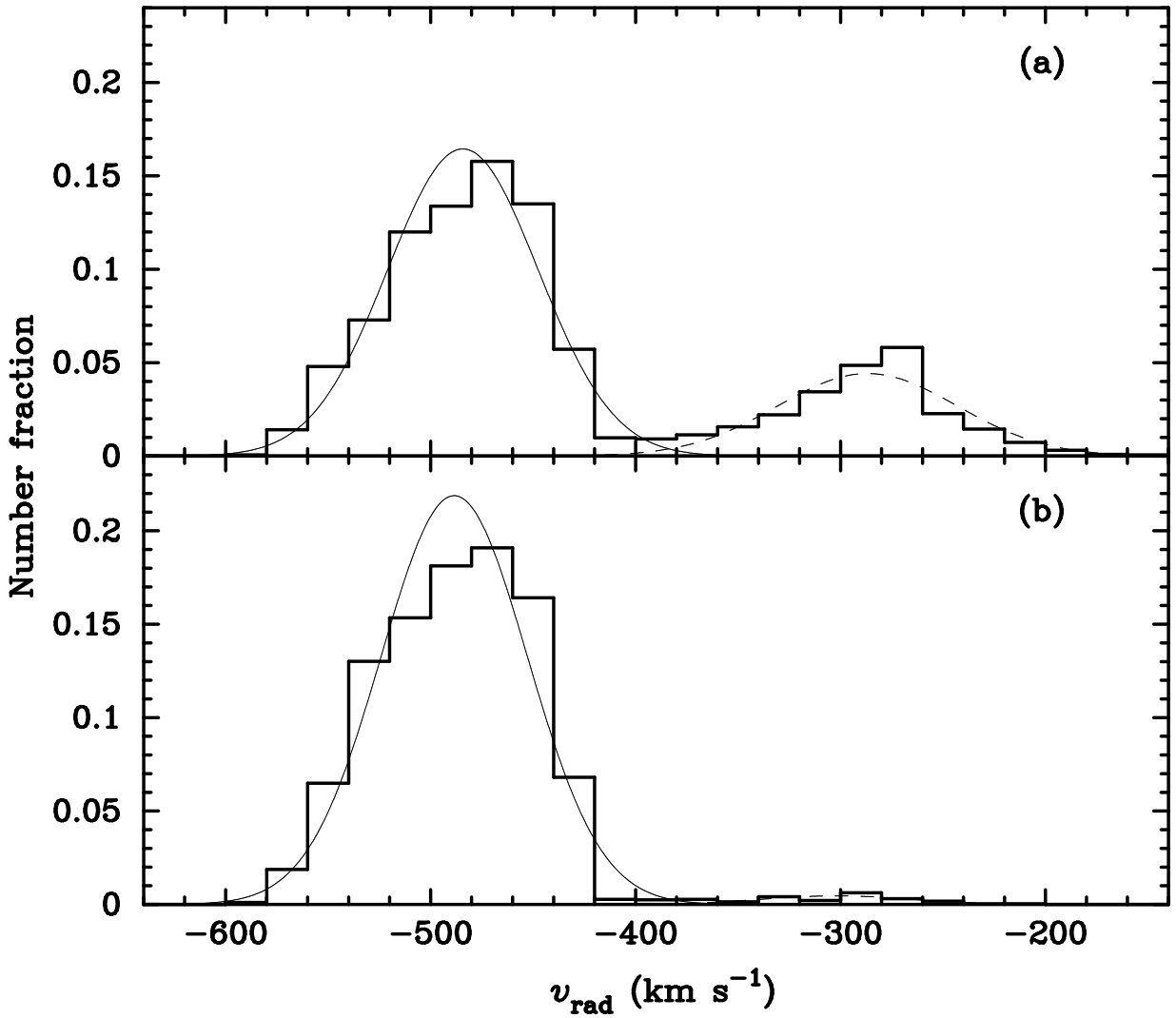


Fig. 4.7: Histogram of the radial velocity around field f207 (Gilbert et al. 2009). Solid and dashed curve show the fitting results based on the KMM algorithm for the primary and secondary component, respectively. In the upper panel (Model A), there are two components: 75.7% are $v_{\rm rad} = -484 \pm 37\,{\rm km\,s^{-1}}$ (solid curve), 24.3% are $v_{\rm rad} = -287 \pm 44\,{\rm km\,s^{-1}}$ (dashed curve). In the bottom panel (Model B), there are also two components: 97.9% are $v_{\rm rad} = -488 \pm 36\,{\rm km\,s^{-1}}$ (solid curve), 2.1% are $v_{\rm rad} = -302 \pm 36\,{\rm km\,s^{-1}}$ (dashed curve).

4.2.3 Bimodality of the Giant Stream

The bimodality of the stream is observed in H13s region (center of this region is $\xi = 0\overset{\circ}{.}4$, $\eta = -1\overset{\circ}{.}5$), and there are two components with peak radial velocity of -520 and $-400\,{\rm km\,s^{-1}}$ (Koch et al. 2008). Gilbert et al. (2009) performed more detailed analysis and reported radial velocities of two components are $-490 \pm 21\,{\rm km\,s^{-1}}$ and $-388 \pm 17\,{\rm km\,s^{-1}}$ for 48% and 27% of the total population, respectively.

Figure 4.8 shows the radial velocity histogram of our results around field H13s. In Koch et al. (2008), both components cannot be considered as the third shell component. Therefore, we can neglect the peak radial velocity component of $\sim -290\,{\rm km^{-1}}$ as the origin of the bimodality. The figure shows that there is no double clear component in the histogram except for the third shell. This is a natural result from the assumptions made in our simulations. We assume King models as the progenitor; therefore, the progenitor is a single component in the phase space. This explains why our simulations do not reveal the observed

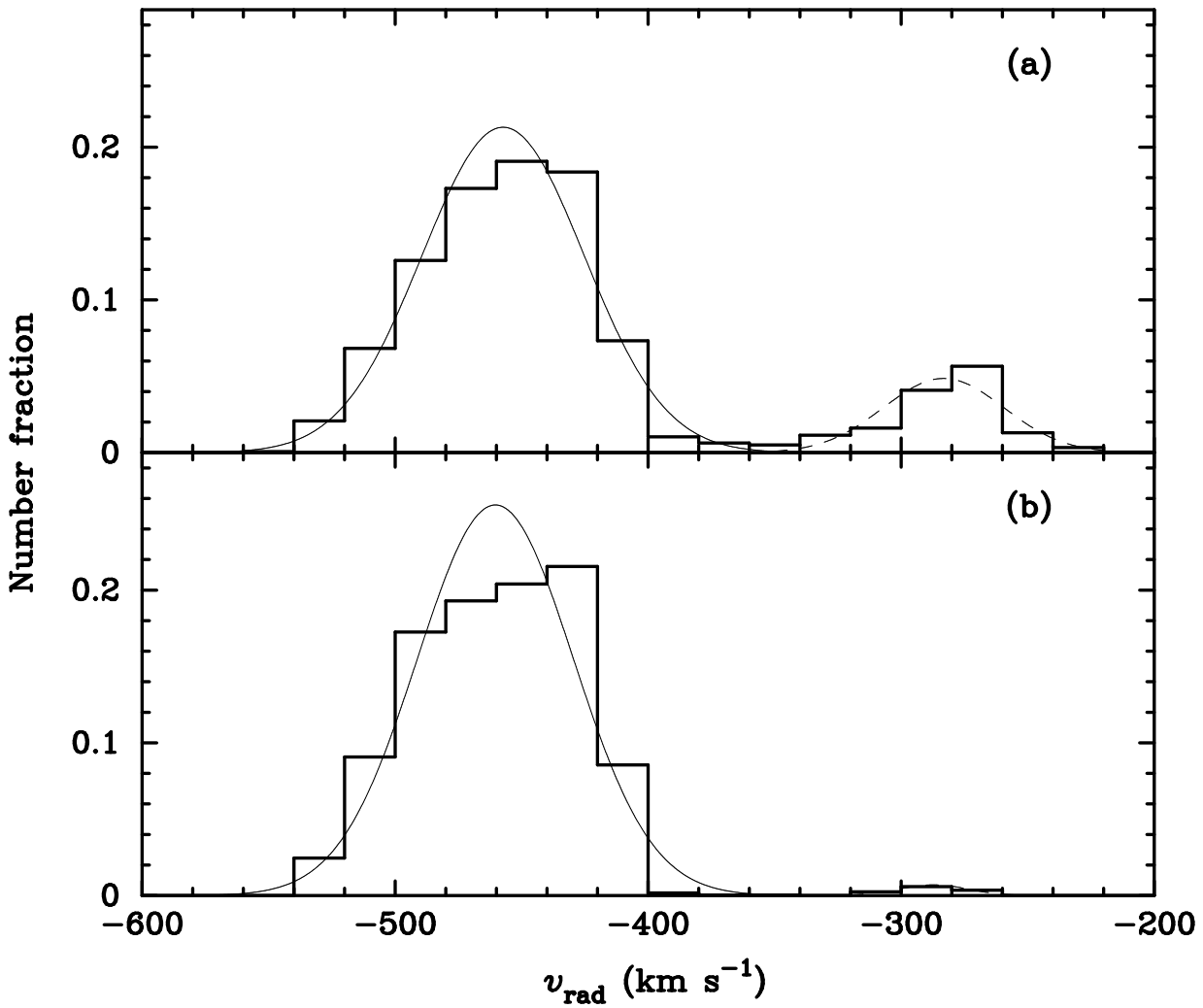


Fig. 4.8: Histogram of the radial velocity around field H13s (Koch et al. 2008). Model A exhibits two clear components have radial velocity of $-457 \pm 32\,\mathrm{km\,s^{-1}}$ (solid curve) and $-283 \pm 24\,\mathrm{km\,s^{-1}}$ (dashed curve) while Model B has only one component of $v_{\rm rad} = -460 \pm 31\,\mathrm{km\,s^{-1}}$.

Table. 4.2: Fitting results based on the KMM algorithm

field	model	primary component			secondary component		
		$\langle v_{\rm rad}\rangle$ (km s^{-1})	σ (km s^{-1})	fraction (%)	$\langle v_{\rm rad}\rangle$ (km s^{-1})	σ (km s^{-1})	fraction (%)
H13s	A	−457	32	85.5	−283	24	14.5
	B	−460	31	98.8	−287	14	1.2
a3	A	−388	21	85.1	−363	32	14.9
	B	−384	17	55.1	−387	23	44.9

separation. In our simulations, we cannot explain the bimodality in field H13s.

The bimodality in field H13s might be attributable to this accretion event. If the progenitor has two or more components in the phase space like as dwarf spirals or dwarf irregulars, the observed bimodality might be reproduced. By this hypothesis, we can explain the splitting of two components. However, we must consider the difference between the radial velocity of the two components. The peak velocity reflects the initial velocity of the center of mass (c.o.m.) of the component. Thus, a difference of the order of $100\,\mathrm{km\,s^{-1}}$ between the peak velocity of the two components implies that the initial velocity of the c.o.m. of

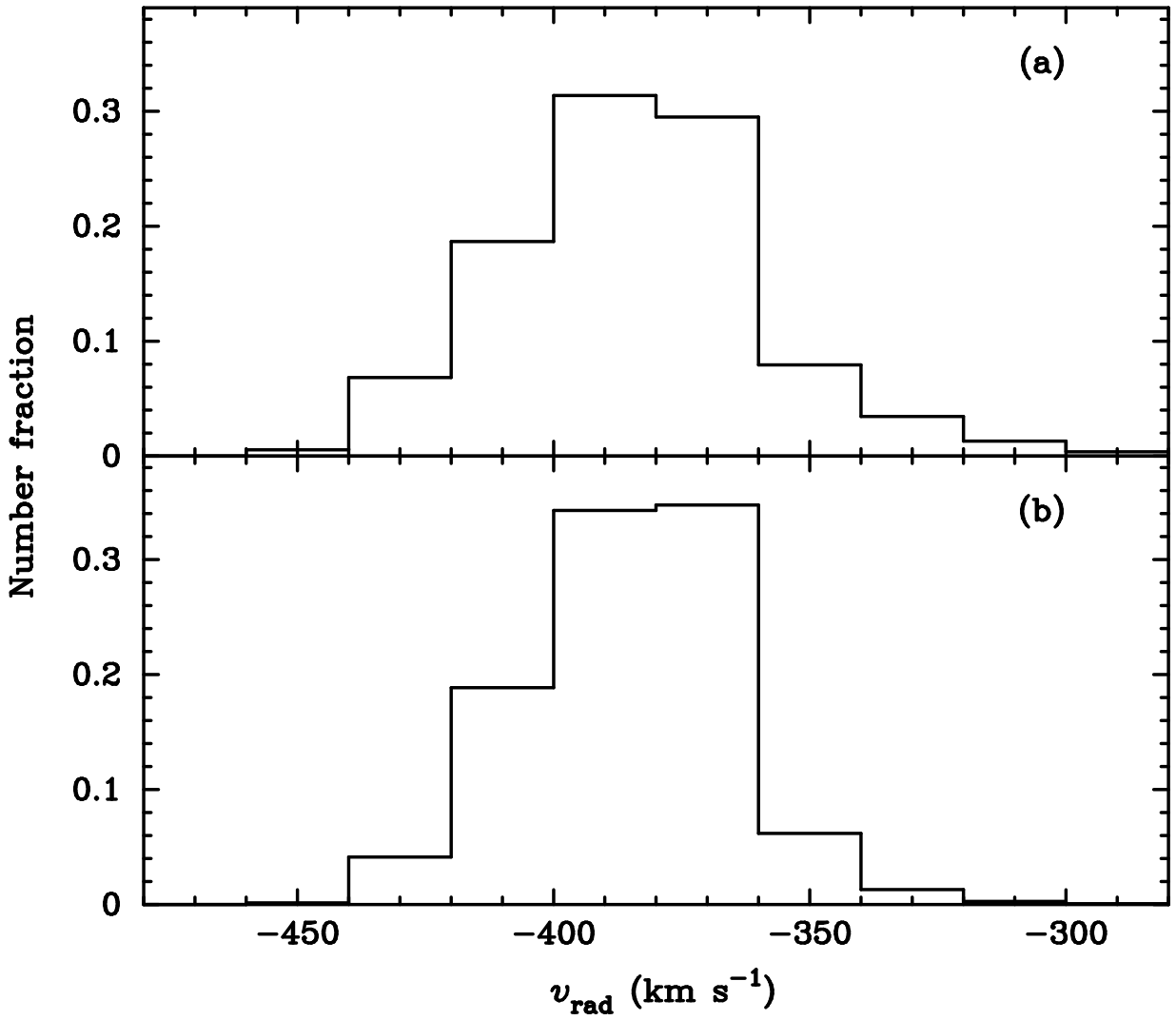


Fig. 4.9: Histogram of the radial velocity around field a3 (Koch et al. 2008). Models A and B show a single component which has radial velocity around $-380\,\mathrm{km\,s^{-1}}$.

the components is considerably different. The observed bimodality might also be attributable to a different accretion event. In this case, the splitting and the difference between peak velocities can be explained. This hypothesis is supported by the analysis of field a3 (center of this region is $\xi = 1\overset{\circ}{.}3$, $\eta = -2\overset{\circ}{.}1$) (Koch et al. 2008). The bimodality of the giant stellar stream is not observed in field a3 (Koch et al. 2008; Gilbert et al. 2009), and our results also do not indicate any bimodality in this region (Fig. 4.9). This result suggests that the H13s region is a special region; hence, the bimodality is not a typical feature of the giant stellar stream. From the RGB star count map in Irwin et al. (2005), we can identify many structures except for the giant stellar stream, the northeast shell, and the west shell (see also Ibata et al. 2007; McConnachie et al. 2009; Martin et al. 2013; Lewis et al. 2013). There is a faint shell-like structure near field H13s. More precise observations of the radial velocity, the metallicity distribution or the abundance pattern near field H13s, and other stream regions will determine the origin of the two observed components.

Chapter 5 Convergence Tests and Implications

In this chapter, we check numerical convergence of results presented in the former chapters. Especially, we focus on the best-fit parameter as the progenitor satellite of the observed structures and report the results in Sections 5.1 and 5.2. Since the result of the high-resolution simulation enables more detailed analysis about the reproduce structures, we here examine the intrinsic metallicity distribution model of the progenitor satellite in Section 5.3.

5.1 Convergence of Spatial Structures

To examine numerical convergence of results, we use 524,288 particles to represent Model A and set the gravitational softening length to be 13 pc. In other words, we use ~ 10 times more particles with $\sim 1/10$ softening length compared with N-body simulations presented in Chapter 2. Figure 5.1 compares the result of the high-resolution simulation with that of the low-resolution simulation. The figure clearly shows that results in the both resolution are almost the same, except for the smoothness of the density distribution and the sharpness of the edges of the reproduced structures. Therefore, we conclude that the low-resolution simulations have sufficient numerical resolution to reproduce the observed structures at least in the shapes of them.

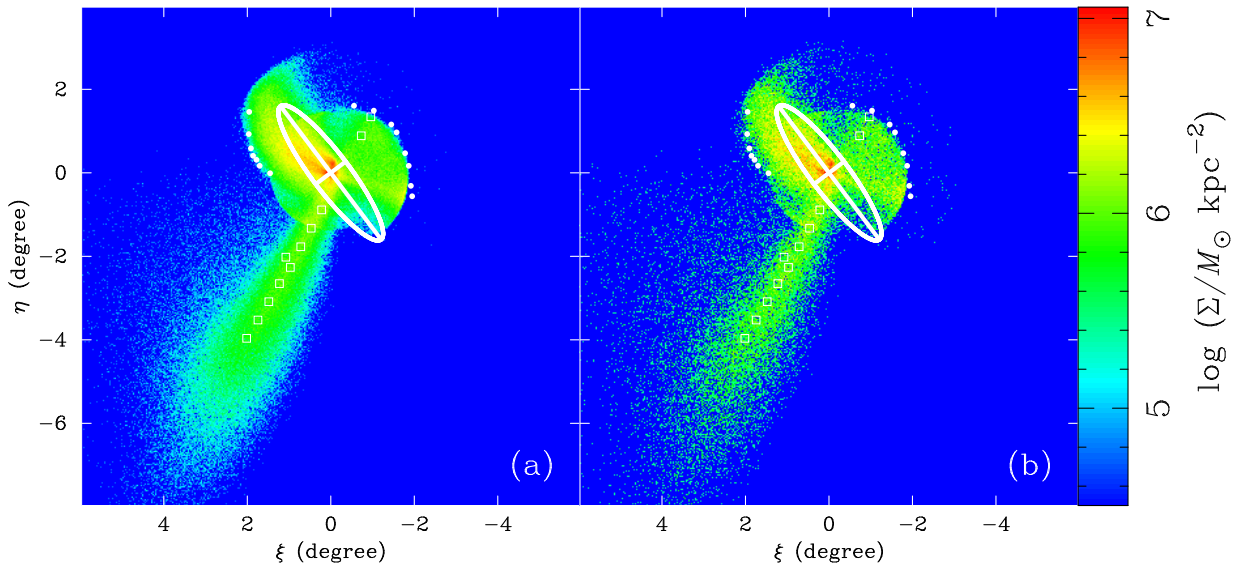


Fig. 5.1: Projected mass-density distribution of the tidal debris for (a) $N = 524{,}288$ model and (b) $N = 65{,}536$ model. The right panel is identical to Fig. 3.1a.

5.2 Convergence of Velocity Structures

Figure 5.2 compares the phase space density distribution of the giant stream between the high and the low resolution models. The similar distribution appeared in Fig. 5.2 implies that the numerical results are well converged. Therefore, we conclude that the low-resolution simulations have sufficient numerical resolution to reproduce the kinematics of the observed structures.

Figures 5.3, 5.4, 5.5, and 5.6 check the numerical convergence of the radial velocity distribution in individual observation fields (field f135, f207, H13s, and a3, respectively). Table 5.1 lists the fitting results based on the KMM algorithm in each observation field. In all the above figures, the top and the bottom panel show the high and the low resolution results, respectively. All the comparisons clearly exhibit beautiful agreements of the both resolution results. This result successfully confirms that the presented numerical results have sufficient resolution even in the low-resolution models.

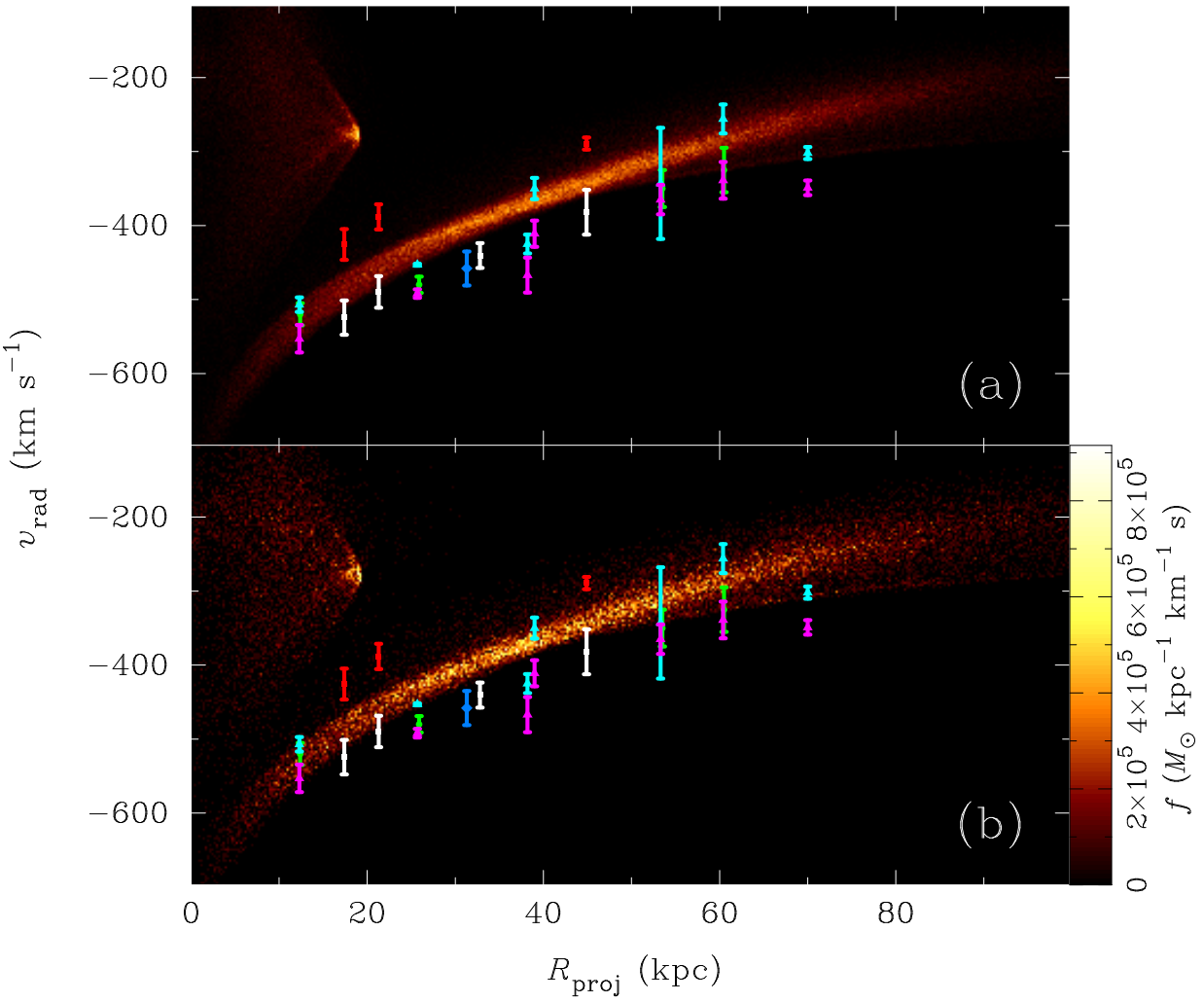


Fig. 5.2: Phase space density maps of the giant stream. Panels (a) and (b) correspond to the high and the low resolution models, respectively. Panel (b) is identical to Fig. 4.5a.

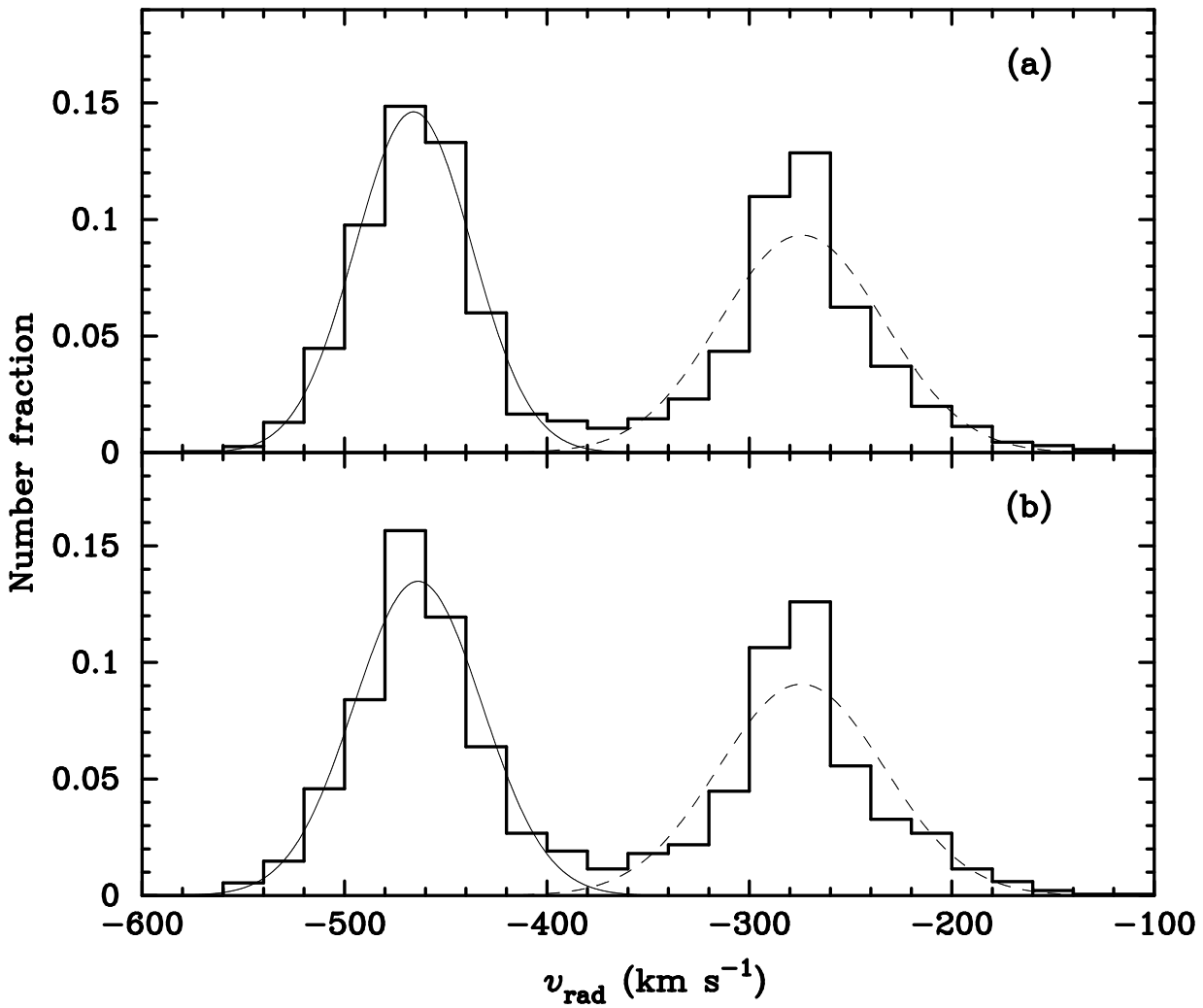


Fig. 5.3: Histogram of the radial velocity around the field f135 (Gilbert et al. 2007). The top and the bottom panels show the high and the low resolution results, respectively. The bottom panel is identical to Fig. 4.6a.

Table. 5.1: Fitting results based on the KMM algorithm

field	N (1)	primary component			secondary component		
		$\langle v_{\rm rad}\rangle$ (km s^{-1})	σ (km s^{-1})	fraction (%)	$\langle v_{\rm rad}\rangle$ (km s^{-1})	σ (km s^{-1})	fraction (%)
f135	524,288	−466	29	52.7	−274	40	47.3
	65,536	−464	32	53.5	−274	41	46.5
f207	524,288	−484	37	74.2	−285	40	25.8
	65,536	−484	37	75.7	−287	44	24.3
H13s	524,288	−457	32	85.4	−281	23	14.6
	65,536	−457	32	85.5	−283	24	14.5
a3	524,288	−388	22	86.6	−351	31	13.4
	65,536	−388	21	85.1	−363	32	14.9

(1) Number of N-body particles.

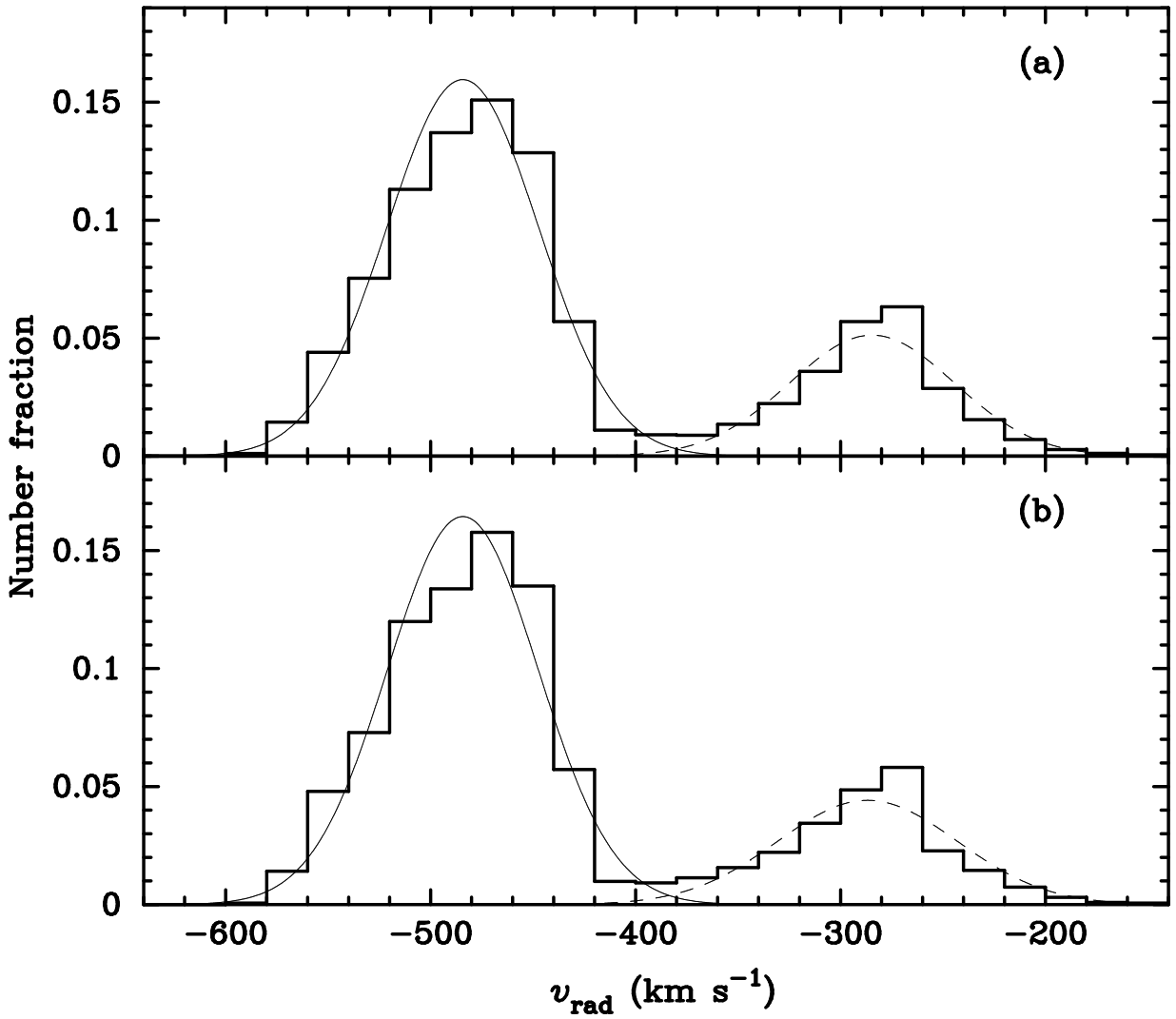


Fig. 5.4: Histogram of the radial velocity around the field f207 (Gilbert et al. 2009). The top and the bottom panels show the high and the low resolution results, respectively. The bottom panel is identical to Fig. 4.7a.

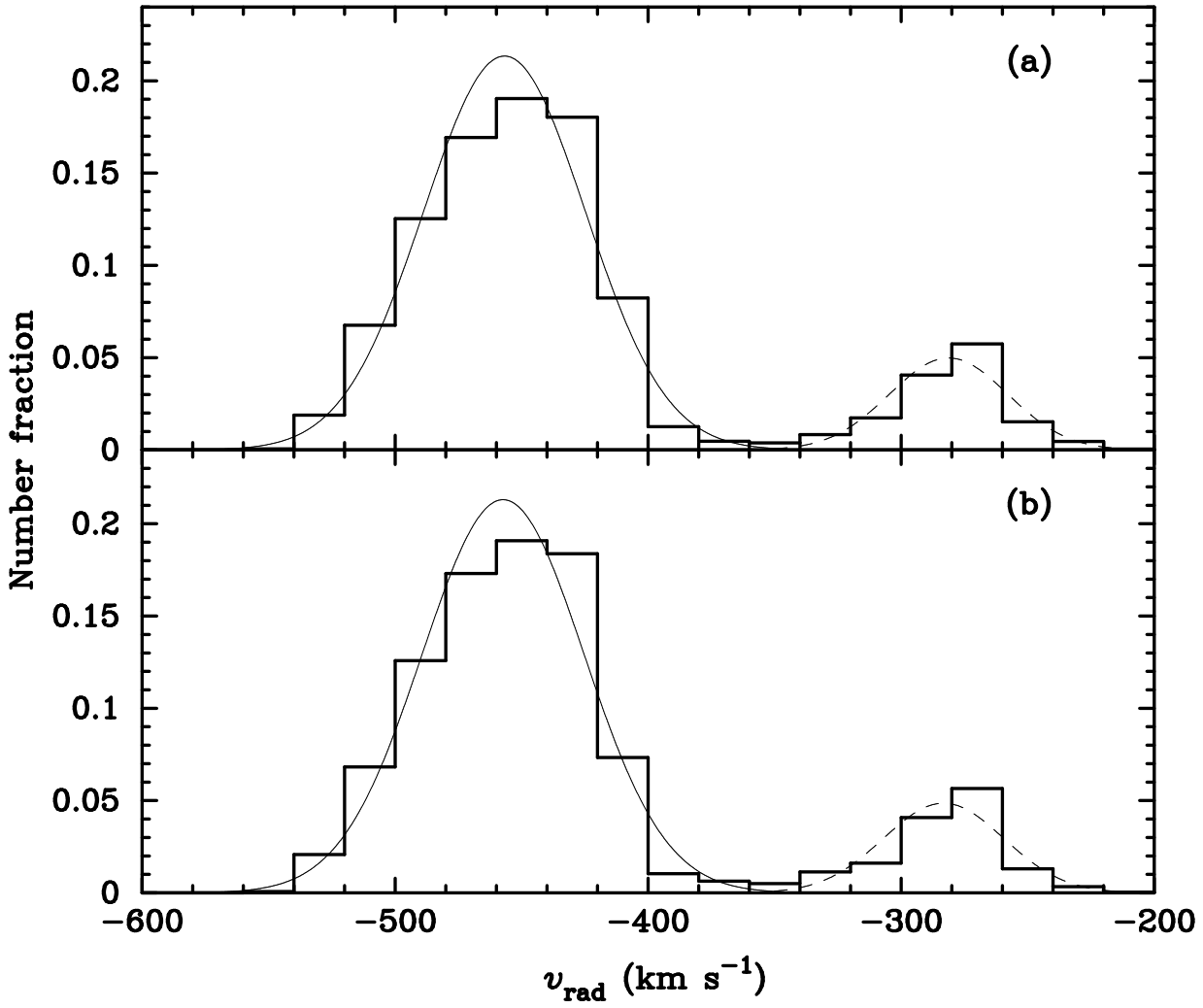


Fig. 5.5: Histogram of the radial velocity around the field H13s (Koch et al. 2008). The top and the bottom panels show the high and the low resolution results, respectively. The bottom panel is identical to Fig. 4.8a.

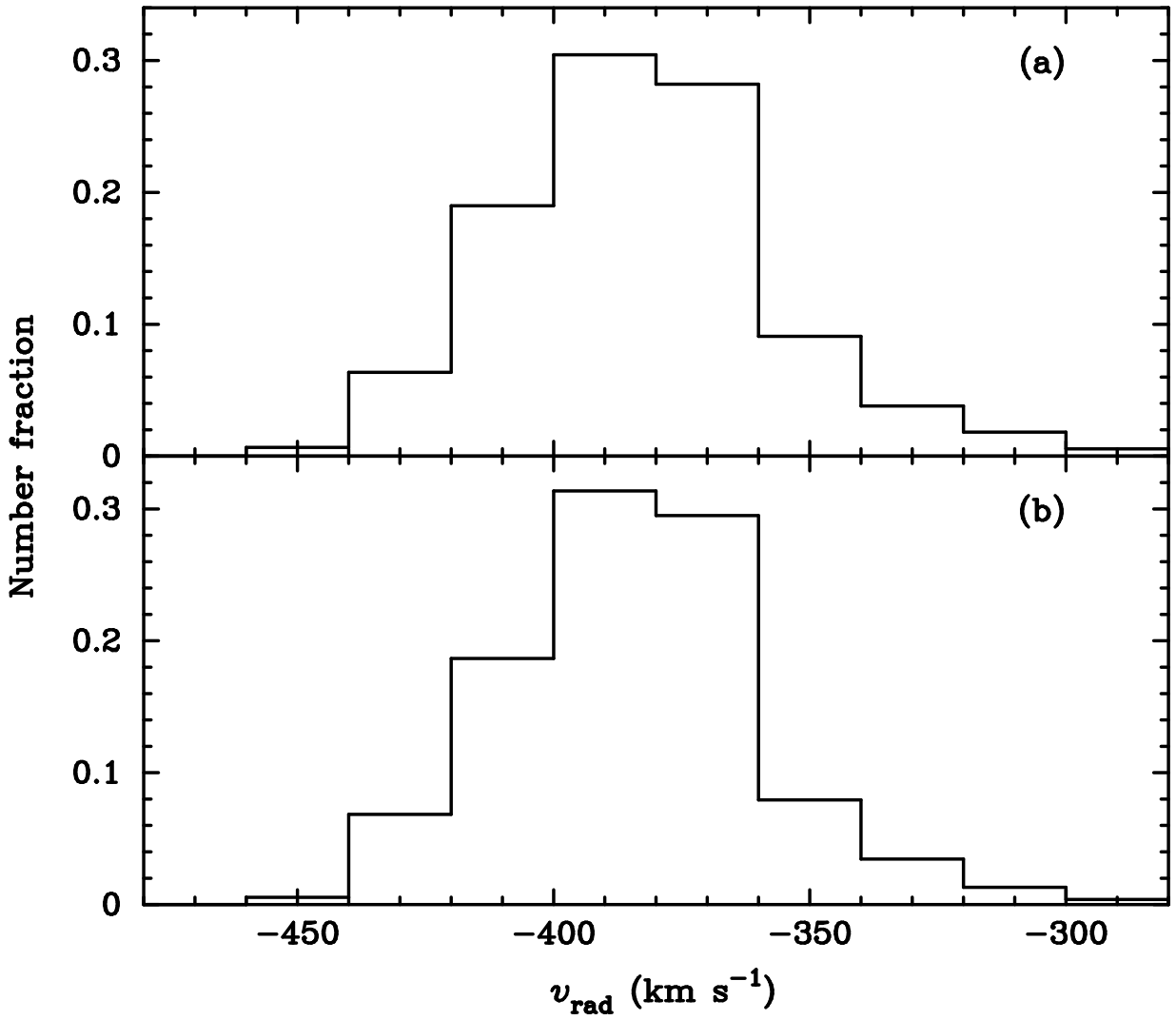


Fig. 5.6: Histogram of the radial velocity around the field a3 (Koch et al. 2008). The top and the bottom panels show the high and the low resolution results, respectively. The bottom panel is identical to Fig. 4.9a.

5.3 Metallicity Gradient of the Progenitor Satellite

The mass-metallicity relation and the metallicity gradient in a galaxy play a significant role to explore the galactic chemical evolution. In this study, we define the metallicity gradient of a galaxy Δ[Fe/H] as

$$\frac{d\ [\mathrm{Fe/H}](r)}{d\ \log(r/r_{\mathrm{e}})}, \tag{5.1}$$

where [Fe/H](r) and r_{e} are the radial profile of the metallicity [Fe/H] and the effective radius, respectively. Recent observational studies reveal that the dwarf elliptical galaxies in the local universe have both the negative metallicity gradient and the positive one (Spolaor et al. 2009; Koleva et al. 2009a,b). Figure 5.7 shows that the observed metallicity gradients are ranging from ~ -0.6 to ~ 0.2 (Koleva et al. 2009b). Since the metallicity gradient is one of the fingerprints of star formation history of galaxies, is a good tool to investigate the evolution history of galaxies.

A large part of observed dwarf ellipticals show a negative metallicity gradient. The formation process of the negative metallicity gradient is naturally explained as follows. The central region of galaxies has a deeper gravitational potential well compared to the outskirts. Consequently, gas as a source of star formation fall into the galactic center, form stars, and finally distribute heavy elements into the interstellar medium via supernova explosions. Hence such a metal-enriched gas then become a source of star formation in the next

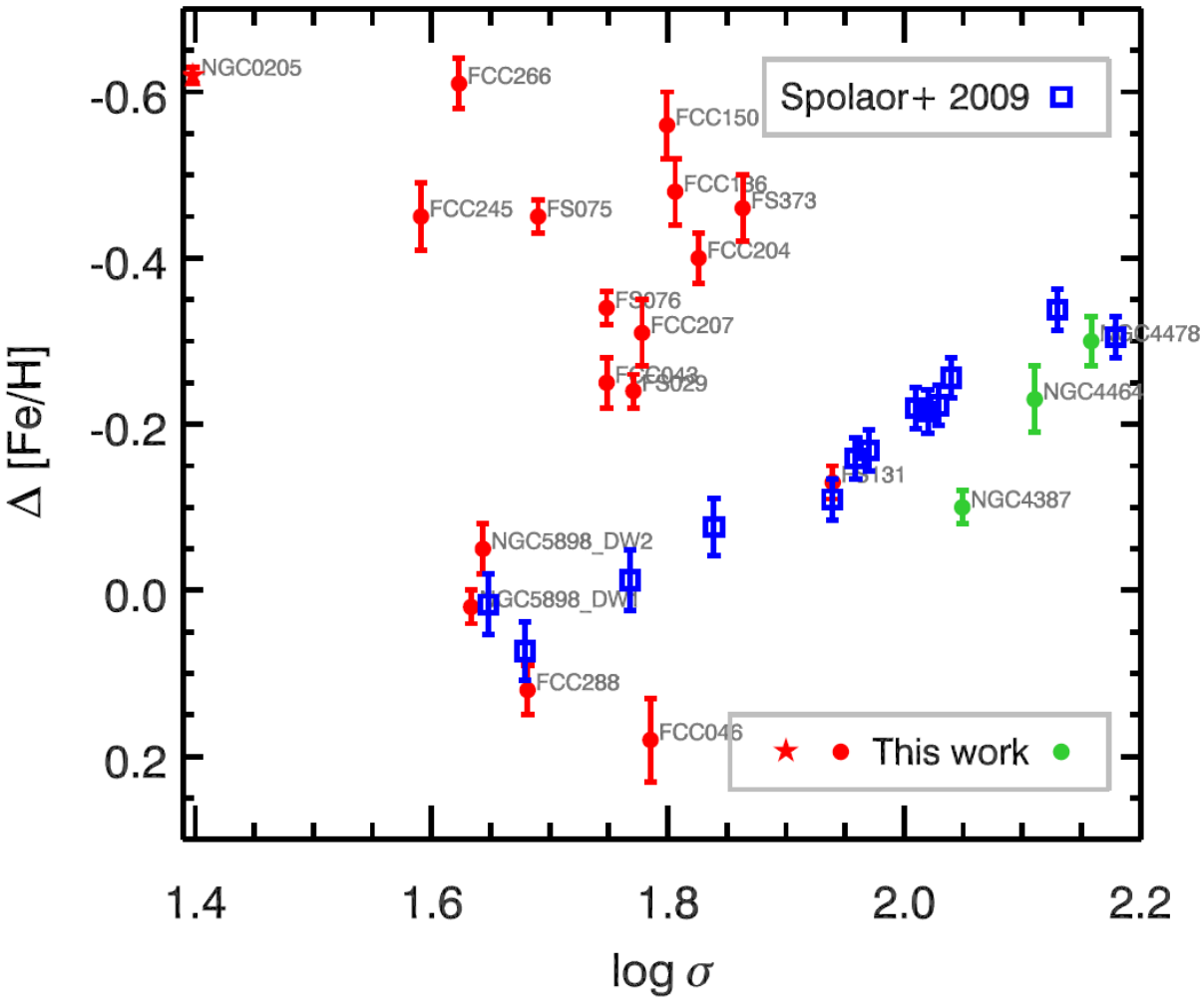


Fig. 5.7: Observed metallicity gradients as a function of one-dimensional velocity dispersion, taken from Koleva et al. (2009b).

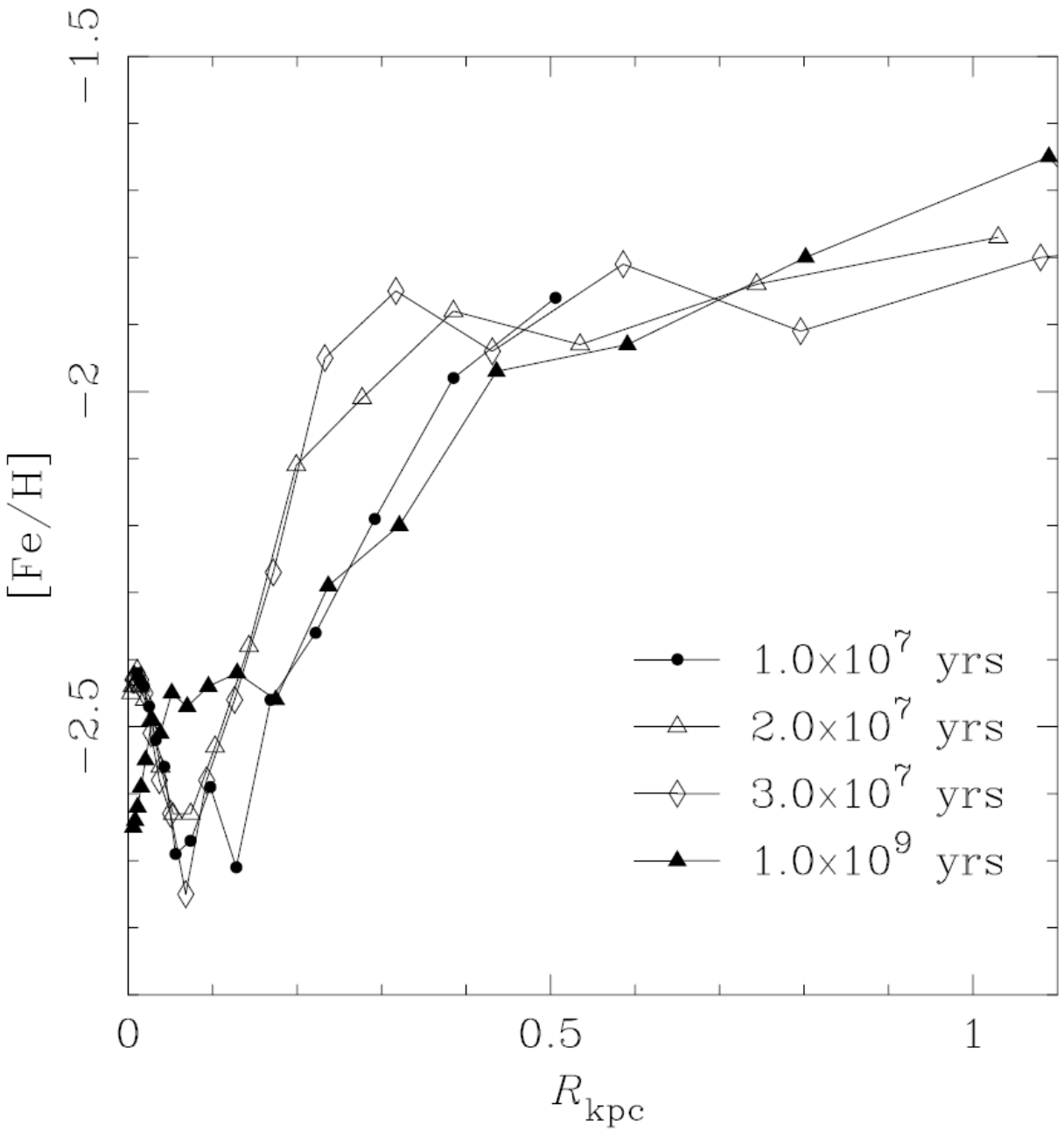


Fig. 5.8: Simulated radial profile of metallicity [Fe/H], taken from Mori et al. (1999).

generation, the metallicity evolution in the galactic center proceeds from recycling star formation and metal enrichment. In contrast, the metallicity evolution in the outskirts of galaxies is relatively slow due to low-density gas reflecting weak gravitational field. As a result, the negative gradient of the metallicity naturally evolves. On the other hand, Mori et al. (1997, 1999) proposed a formation process of the positive gradient. First, star formation and a following supernova explosion occur in the central region of galaxies. Then, the explosion sweeps surrounding interstellar gas and the gas expands as a shell-like structure. A cycle of star formation and metal enrichment starts within the expanding shell. Finally, the positive metallicity gradient is formed (Fig. 5.8). The most important point of this process is that the first supernova explosion in the central region of a galaxy gives a significant influence on the evolution of the whole part of the corresponding galaxy; therefore, considered being effective in less massive galaxies.

Studies focused on merger remnants would be a useful tool to investigate the metallicity gradient of the

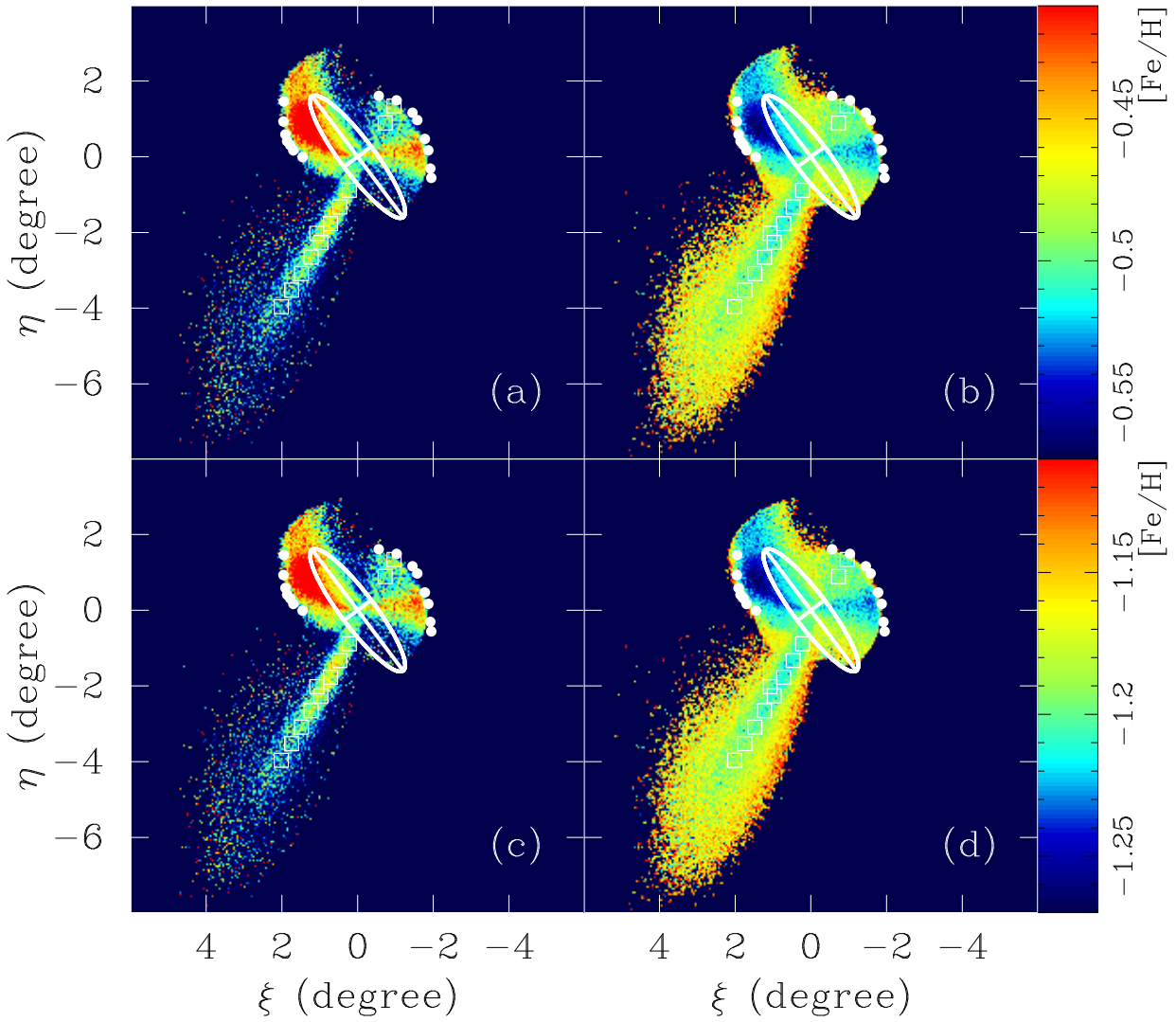


Fig. 5.9: Metallicity distribution for (a) $[\mathrm{Fe/H}]_{\mathrm{mean}} = -0.5$, $\Delta[\mathrm{Fe/H}] = -0.7$, (b) $[\mathrm{Fe/H}]_{\mathrm{mean}} = -0.5$, $\Delta[\mathrm{Fe/H}] = 0.3$, (c) $[\mathrm{Fe/H}]_{\mathrm{mean}} = -1.2$, $\Delta[\mathrm{Fe/H}] = -0.7$, and (d) $[\mathrm{Fe/H}]_{\mathrm{mean}} = -1.2$, $\Delta[\mathrm{Fe/H}] = 0.3$, respectively. Filled circles and open squares show the position of the edge of the shells (Fardal et al. 2007) and the observed areas of the giant stellar stream (Font et al. 2006), respectively. Ellipse in each panel corresponds the size of the M31's disk.

corresponding progenitor. If a satellite galaxy initially had some non-uniform metallicity distribution, then structures formed after galactic merger should also have non-uniform metallicity distribution. Therefore, theoretical studies based on N-body simulations have a potential to connect the former metallicity distribution of the satellite and the current metallicity distribution of merger remnants. Fardal et al. (2008) studied connections between satellite galaxy models which initially have a negative metallicity gradient and resultant metallicity distribution of the giant stellar stream, the east and the west shells. However, no one has studied satellite models of positive metallicity gradient, and there are no restrictions. Owing to relations between the observed metallicity distribution and the metallicity distribution model of the progenitor satellite, the lack of knowledge about the gradient also makes it difficult to determine the mean metallicity. To provide information on metallicity distribution models of the progenitor satellite, we have investigated the metallicity gradient of the progenitor satellite galaxy of the giant stellar stream taking account for negative and positive gradient models.

Figure 5.9 shows spatial metallicity distribution maps with varying metallicity distribution model of the

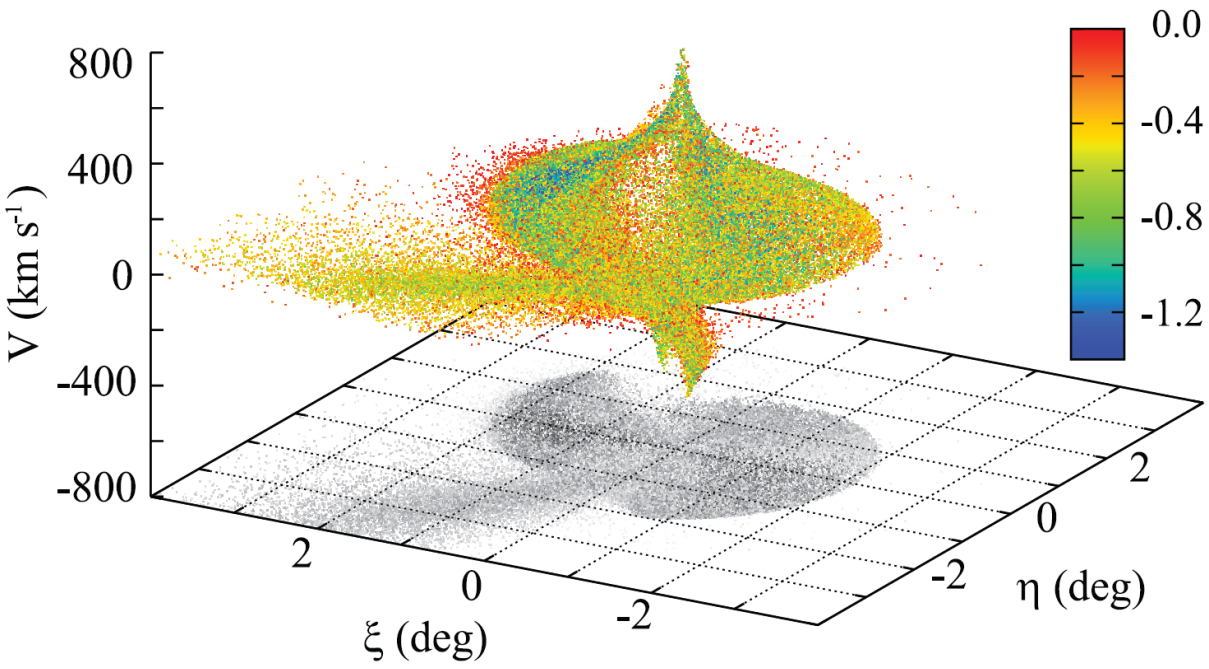


Fig. 5.10: Distribution of satellite debris at the present day, taken from Mori & Rich (2008). The color of particles shows the initial total energy in units of 10^{51} ergs.

progenitor satellite. The top panels in the figure are high metallicity model for the progenitor satellite which assuming mean iron abundance $[\mathrm{Fe/H}]_{\mathrm{mean}}$ of -0.5 while the bottom panels exhibit low metallicity model ($[\mathrm{Fe/H}]_{\mathrm{mean}} = -1.2$). The mean of the observed mass-metallicity relation (Dekel & Woo 2003) infers that the high and the low metallicity models have the stellar mass of $5 \times 10^9\,M_\odot$ and $10^8\,M_\odot$, respectively. The left and the right panels show negative and positive metallicity gradient model (Δ[Fe/H] of -0.7 or 0.3), respectively. Figure 5.9 shows that differences of metallicity distribution models make clear differences in metallicity distribution at the present day. Since particles initially located in the central region of the satellite most likely to exist in the east shell, [Fe/H] observed in the east shell region is relatively higher/lower than that observed in other fields for the negative/positive metallicity gradient model. Similarly, particles initially located on the outskirt of the progenitor satellite tend to locate on an "envelope" region around the bright area of the stream ("core" of the stream); hence, [Fe/H] in the "envelope" of the stream becomes lower/higher than that in the "core" of the stream for the negative/positive metallicity gradient model. The above trend of particle distribution agrees with earlier studies; for example, Mori & Rich (2008) reported that strongly bound particles in the initial condition (blue particles in Fig. 5.10) most likely locate in the east shell.

Figure 5.11 compares the above metallicity distribution models and the observed metallicity distribution. The high metallicity models match with the observations by Guhathakurta et al. (2006); Kalirai et al. (2006a,b); Gilbert et al. (2009) while the low metallicity models are consistent with the observation by Koch et al. (2008). The figure shows that the difference of metallicity gradient Δ[Fe/H] of unity, which is corresponding to the observed variety of the metallicity gradient (Koleva et al. 2009b), does not make a clear difference of the metallicity within inner region ($R_{\mathrm{proj}} \lesssim 40$ kpc). Observations targeting on the outer region might distinguish the sign of the metallicity gradient for the progenitor satellite of the giant stream.

Figure 5.12 shows the radial profiles of the mean metallicity of the "envelope" of the giant stream. The metallicity profiles shown in Fig. 5.12 have a weak dependence on the metallicity gradient. In case of the negative metallicity gradient models, the mean metallicity of the "envelope" is lower than that of the "core" at the inner region while the both metallicity converge at the outer region. The "envelope" mainly contains

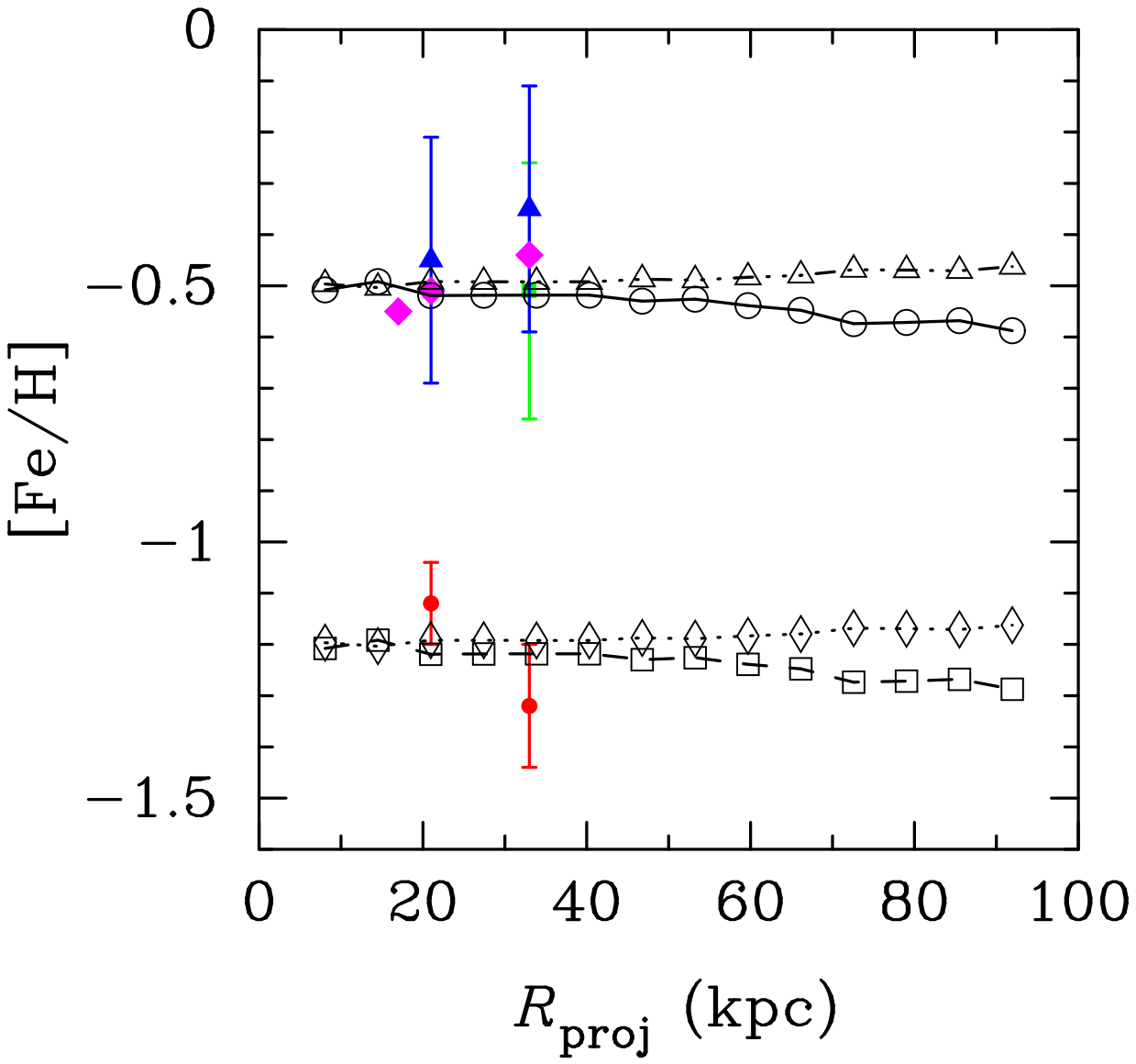


Fig. 5.11: Metallicity [Fe/H] within the "core" of the giant stellar stream as a function of projected radius $R_{\rm proj}$. Open symbols with lines represent metallicity distribution models for the result of N-body simulations shown in Fig. 5.9: (a) by circles with a solid line, (b) by triangles with a triple-dot-dashed line, (c) by squares with a dashed line, and (d) by diamonds with a dotted line. Filled symbols show the observed metallicity: red circles (Koch et al. 2008), a green square (Guhathakurta et al. 2006), blue triangles (Kalirai et al. 2006a,b), and magenta diamonds (Gilbert et al. 2009).

particles initially located in the outer part of the progenitor satellite while the "core" is composed by a mixture of the whole part of the satellite; therefore, the both metallicity converge at the distant region where the initially strong bound particles likely do not locate. The difference about the mean metallicity between the "core" and the "envelope" at the inner halo would be useful to distinguish the sign of the gradient. Gilbert et al. (2009) reported that the metallicity observed in the "core" of the giant stream is 0.17 dex higher than in the "envelope". This trend is the same with negative metallicity gradient models; however, the simulation shows a difference of only ~ 0.1 dex (Fig. 5.9). The observed difference of about 0.2 dex implies much stronger metallicity gradient compared to the observed values for nearby dwarf ellipticals (Spolaor et al. 2009; Koleva et al. 2009a,b).

Fardal et al. (2012) presented results of spectroscopic measurements focused on the west shell (Fig. 5.13).

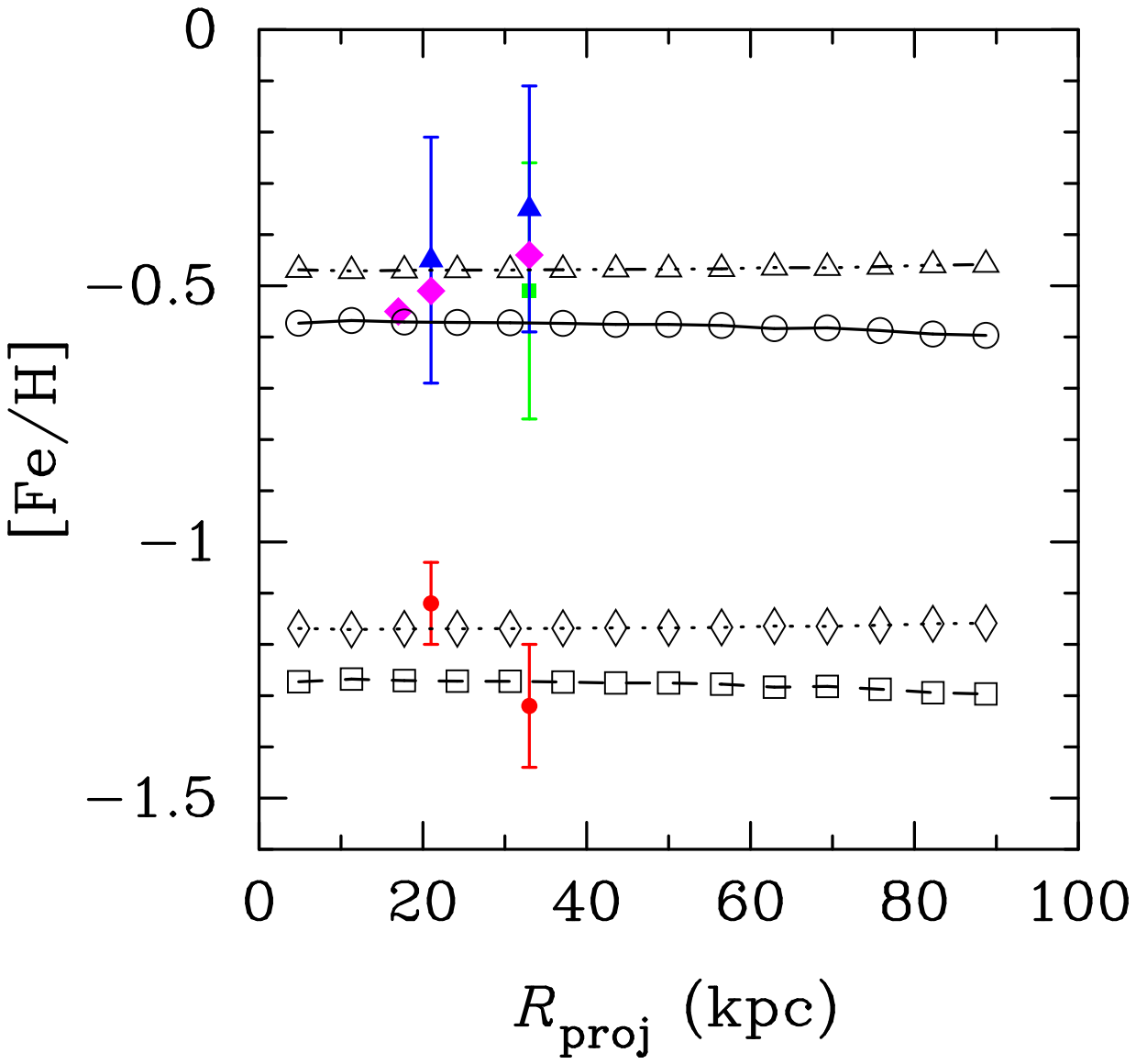


Fig. 5.12: Metallicity [Fe/H] of the "envelope" of the giant stellar stream as a function of projected radius $R_{\rm proj}$. All symbols are the same as Fig. 5.11.

They found that the observed metallicity in the west shell was similar to that in the "core" of the giant stream. This result is consistent with results presented in this work (Fig. 5.9). The radial metallicity profile along the minor axis of the simulated results presented in this study is shown in Fig. 5.14. The figure clearly shows that the metallicity of all the models well match with the assumed mean metallicity and have a negligible dependence on the metallicity gradient. This result suggests that spectroscopic observations committed along the minor axis reflect the mean metallicity of the progenitor satellite. In addition, the metallicity of the "core" of the stream at the inner halo also match with the assumed mean metallicity (Fig. 5.11). This agreement of the metallicity in the both regions is consistent with the observations (Gilbert et al. 2009; Fardal et al. 2012, see Fig. 5.13).

As discussed in this section, the metallicity distribution model of the progenitor satellite has not yet been determined. The mean metallicity would be determined by the spectroscopic observations focused on the "core" of the giant stellar stream or the west shell (Figs. 5.11 or 5.14, respectively). Currently, there is the systematic difference about the observed metallicity of the stream between the high metallicity results (Guhathakurta et al. 2006; Kalirai et al. 2006a,b; Gilbert et al. 2009) and the low metallicity results

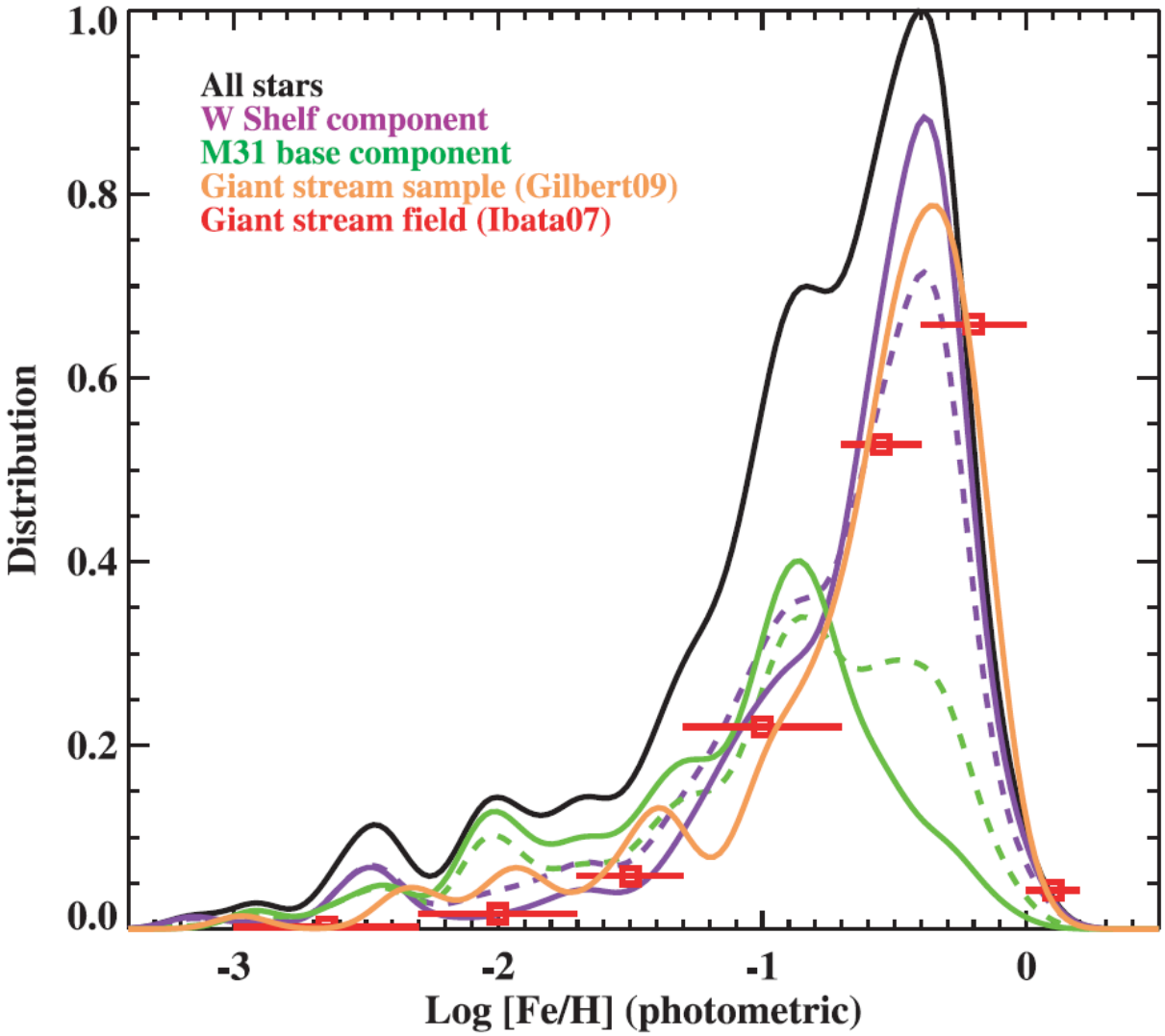


Fig. 5.13: Observed metallicity distribution in the west shell, taken from Fardal et al. (2012). The purple curve represents the contribution from the progenitor satellite of the giant stellar stream. The orange curve and red squares show the metallicity distribution of the "core" of the stream.

(Koch et al. 2008) as appeared in Fig. 5.11. If future observations discovered the origin of this difference and provided reliable results of the metallicity, then we could conclude that either the high or the low metallicity model is a realistic one. As for the metallicity gradient, observational results by Gilbert et al. (2009) suggest that the progenitor satellite had a negative metallicity gradient. However, the required metallicity gradient to reproduce the observed result is about twice steeper than the observed gradients of the nearby dwarf ellipticals (Spolaor et al. 2009; Koleva et al. 2009a,b). To determine the gradient more precisely and constrain the metallicity distribution model of the progenitor satellite, the east shell is the most suitable target of future spectroscopic observations. As clearly shown in Fig. 5.9, the iron abundance in the east shell region is the highest/lowest for the negative/positive metallicity gradient model. Therefore, comparing the metallicity in the east shell and the others would be responsible to recover fossil information on the metallicity distribution of the progenitor satellite. Future spectroscopic observations focused on the east shell would provide useful clues to investigate the metallicity distribution model of the progenitor.

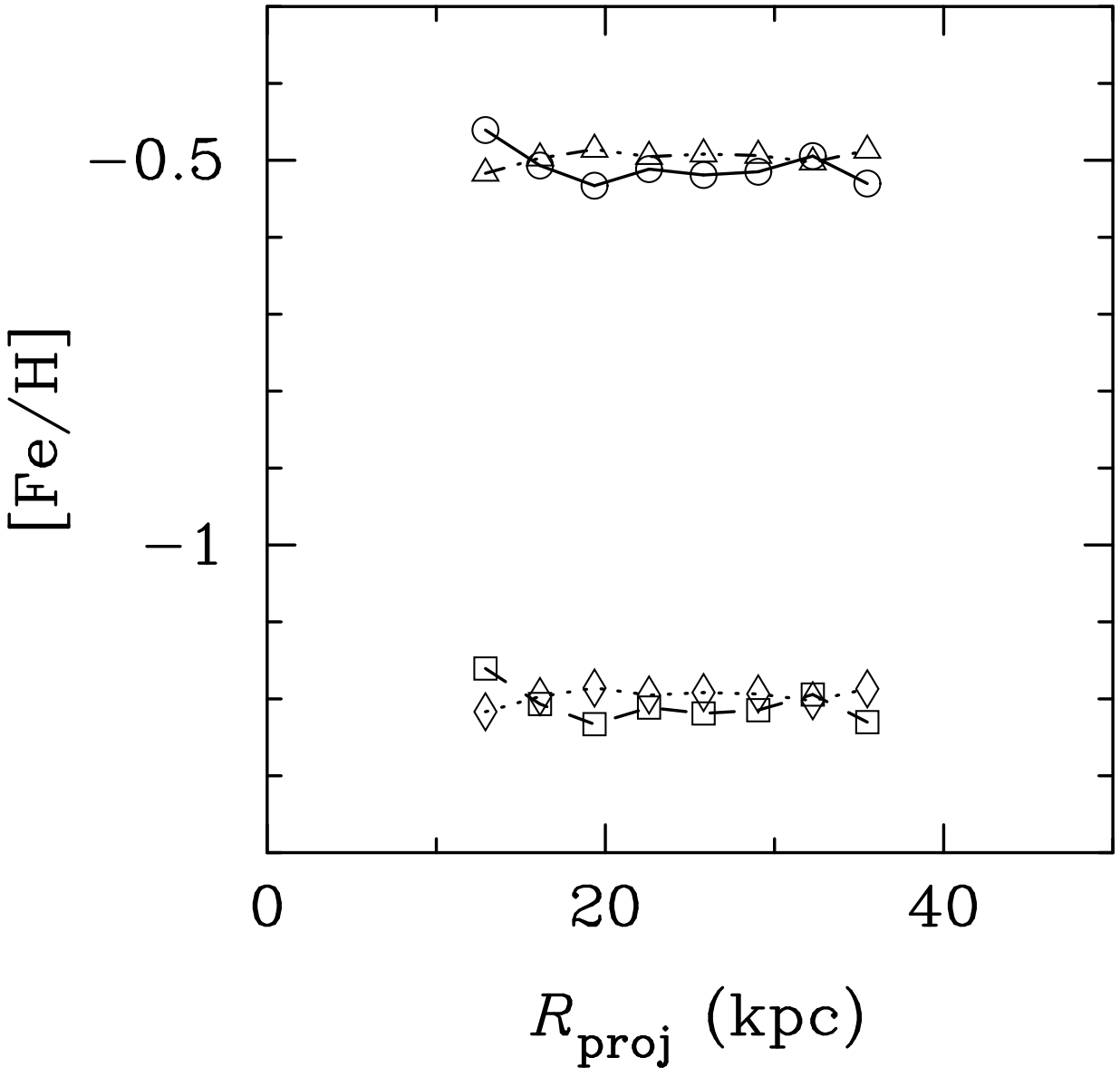


Fig. 5.14: Radial profile of metallicity [Fe/H] of the west shell along the minor axis of the M31 disk. Symbols are the same as Fig. 5.11.

Chapter 6 Conclusion

We have investigated the interaction between an accreting satellite galaxy and M31 using an N-body simulation. A comprehensive parameter study with 247 models is performed by varying the size and the mass distribution of the progenitor dwarf galaxy. We show that it is crucial the binding energy of the progenitor galaxy to reproduce the Andromeda giant stellar stream and the shell-like structures surrounding M31. As a result of simulations, the progenitor must satisfy a simple scaling relation among the core radius, the total mass and the tidal radius. Using this relation, we successfully constrain the physical properties of the progenitors which have a mass ranging from $5 \times 10^8 M_\odot$ to $5 \times 10^9 M_\odot$ and a central surface density around $10^3\, M_\odot\, \mathrm{pc}^{-2}$. A detailed comparison between our result and the observed nearby galaxies indicates that the progenitor of the Andromeda giant stellar stream includes a dwarf elliptical galaxy, a dwarf irregular galaxy, and a small spiral galaxy.

Part II

An N-body Code on a GPU Cluster

Abstract

We have developed a highly optimized code for collisionless N-body calculations based on direct summation. Our new optimization hides the latency to access the global memory, and the resulting CUDA code has a peak performance of 1006.7 GFlop/s in single precision (assuming 26 floating-point operations per interaction) with a single NVIDIA Tesla M2090 board. Detailed performance analysis clarifies that the performance metrics of collisionless N-body simulations on GPU are only two quantities: first one is the number of running streaming multiprocessors and another is the clock cycle ratio of the latency to access the global memory and operations to calculate gravitational interaction.

To improve the scalability of the OpenMP/MPI hybrid parallelized code, we have reduced the number of communications among multiple GPUs and have overlapped communications with computations to hide the communication time. The results of performance measurements show excellent scalability with superlinear scaling when the number of N-body particles per GPU is less than 10^4 and parallel efficiency approaching unity when the number of N-body particles per GPU is greater than 10^4. The CUDA/OpenMP/MPI code has a peak performance of 255.5 TFlop/s when 256 NVIDIA Tesla M2090 boards are used, which is 75.0% of the theoretical peak performance.

Chapter 7 Proposed Algorithm

7.1 Background

In astrophysics, collisionless N-body simulations are one of the most powerful tools for investigating structure formation of large scale structure, formation and evolution history of stellar systems such as galaxies. The fundamental equation of N-body simulations is Newton's equation of motion expressed as

$$\boldsymbol{a}_i = \sum_{j=0, j\neq i}^{N-1} \frac{G m_j \left(\boldsymbol{x}_j - \boldsymbol{x}_i\right)}{\left(\left|\boldsymbol{x}_j - \boldsymbol{x}_i\right|^2 + \epsilon^2\right)^{3/2}}, \tag{7.1}$$

where G is the gravitational constant, N is the number of particles, and m_i, $\boldsymbol{x}_i$ and $\boldsymbol{a}_i$ are the mass, position, and acceleration of the i-th particle, respectively. The gravitational softening parameter ϵ, introduced to avoid divergence due to division by zero, eliminates self-interaction when calculating gravitational force. The amount of computation for this equation is proportional to the number of *i-particles*, N_i, the particles on which gravitational is exerted, and the number of *j-particles*, N_j, the particles that cause gravitational force.

Because a large number of N-body particles are necessary to investigate astrophysical phenomena in detail, many studies have been devoted to achieving fast computation for N-body simulations. Some of the proposed algorithms for reducing the amount of computation include the particle-mesh method and the tree method (Hockney & Eastwood 1988; Barnes & Hut 1986). The computational complexity of the tree method is $O(N \log N)$ because the multipole expansion technique reduces the contribution of N_j.

In astrophysics, there are cases that require direct N-body simulations. For example, direct summation is employed to investigate the long-term evolution of globular clusters because their lifespan is much longer than dynamical time. Inaccurate gravitational force calculations incorrectly characterize the orbital evolution of the stars in such systems because numerous orbital integration steps are necessary for computing the time evolution of these systems. Indeed, a fourth-order Hermite scheme with double precision is often employed to investigate the dynamical evolution of globular clusters. Although the present study is not directly aimed at investigating the dynamical evolution of globular clusters, our optimized implementation of N-body calculations provides a useful tool for research in this field.

Investigations that achieve high performance and scalability for simple and characteristic algorithms constitute an important area of computer science. These investigations propose ways to optimize various complicated applications. An N-body simulation using direct summation is a well-known characteristic problem owing to its computation-intensive nature. Because a computational complexity of $O(N^2)$ severely limits the tractable problem size, computer science makes an essential contribution to the development of other sciences when it proposes performance improvements using recent architectures that can increase the tractable problem size.

One way to reduce the computation time for direct N-body calculations is to use an accelerator. The most famous and among the most successful accelerators for gravitational many-body systems is the GRAPE ("Gravity PipE") series (Okumura et al. 1993; Kawai et al. 2000). Its high performance results from the pipelined and massively parallel architecture design, which enables massive parallelization of gravitational force calculations.

Recently, Graphics Processing Units (GPU) have become one of the most attractive accelerators owing to the development of General Purpose computing on GPU (GPGPU). This recent development of GPGPU have been supported by rapid improvements of programming environment such as CUDA (Nvidia 2012, 2013), OpenCL (Khronos 2011, 2013), and OpenACC (OpenACC 2011, 2013). Furthermore, many GPU clusters, including Titan, Tianhe-1A, Nebulae, TSUBAME 2.5, and HA-PACS, appear on the TOP 500 list[1], which indicates the popularity of GPU clusters. Therefore, improving the performance on GPU clusters is an important problem, particularly because the rapid increase in GPU performance and the development of GPU clusters enable the acceleration of numerical simulations, and thus the effectiveness of accelerator device.

Detailed investigations of the effectiveness of accelerating N-body simulations using GPUs are very important. Many previous studies have addressed the use of GPU to accelerate N-body simulations in various research fields, including direct summation for collisionless systems (Hamada & Iitaka 2007; Nyland et al. 2007; Hamada et al. 2009; Hamada & Nitadori 2010), the tree method for collisionless systems (Hamada et al. 2009; Hamada & Nitadori 2010; Gaburov et al. 2010; Bédorf et al. 2012; Nakasato 2012; Ogiya et al. 2013), and direct summation for collisional systems (Harfst et al. 2007; Portegies Zwart et al. 2007; Belleman et al. 2008; Gaburov et al. 2009).

7.2 Motivation

Many earlier studies reported that high performance can be achieved by massive parallelization about i-particles (Hamada & Iitaka 2007; Nyland et al. 2007; Hamada et al. 2009; Hamada & Nitadori 2010). The source code by Nyland et al. (2007) is included in CUDA SDK for CUDA 3.x and 4.x and samples for CUDA 5.x. The implementation in Nyland et al. (2007) can achieve high level of performance – 930 GFlop/s (Giga Foating-point operations per second) in single precision – on an NVIDIA Tesla M2090 board. Hamada et al. (2009) reported their implementation is slightly (about 6%) faster than that of Nyland et al. (2007).

Their implementation can achieve high performance in high N region, however, achieving high performance in low N region is also essential to combine the tree-method or the individual time steps since the benefits of such algorithms comes from reduced N. Therefore, keeping high performance in low N region is essential, to achieve high performance in case of collaborating with such fast algorithms. For such purpose, Nyland et al. (2007) supports two-dimensional parallelization, combination of parallelization about i-particles and j-particles, of calculating gravitational force. In fact, the implementation of Nyland et al. (2007) succeeded to achieve high performance in low N-region. However, there is a possibility to develop performance due to their poor implementation of force accumulation process. Two-dimensional parallelization also provide force accumulation process, since multiple threads calculate acceleration of a common i-particle, thus results of calculation must be accumulated by all corresponding threads. In such process, synchronization or exclusive control is necessary to account whole threads' results appropriately.

Generally speaking, synchronization and exclusive control prevent obtaining high performance in parallel computing. For GPGPU, the cost of synchronization and exclusive control also high due to its characteristics as a many-core architecture. The accumulation process implemented in Nyland et al. (2007) is as follows.

1. Whole threads store their own results to shared memory.
2. Shole threads are synchronized by `__syncthreads()`.
3. A representative thread adds loaded data from shared memory to its own result.

That is to say, Nyland et al. (2007) uses synchronization and exclusive control, both of them can significantly decrease the performance for small N, which is the parameter region they tried to improve performance.

[1] http://www.top500.org/

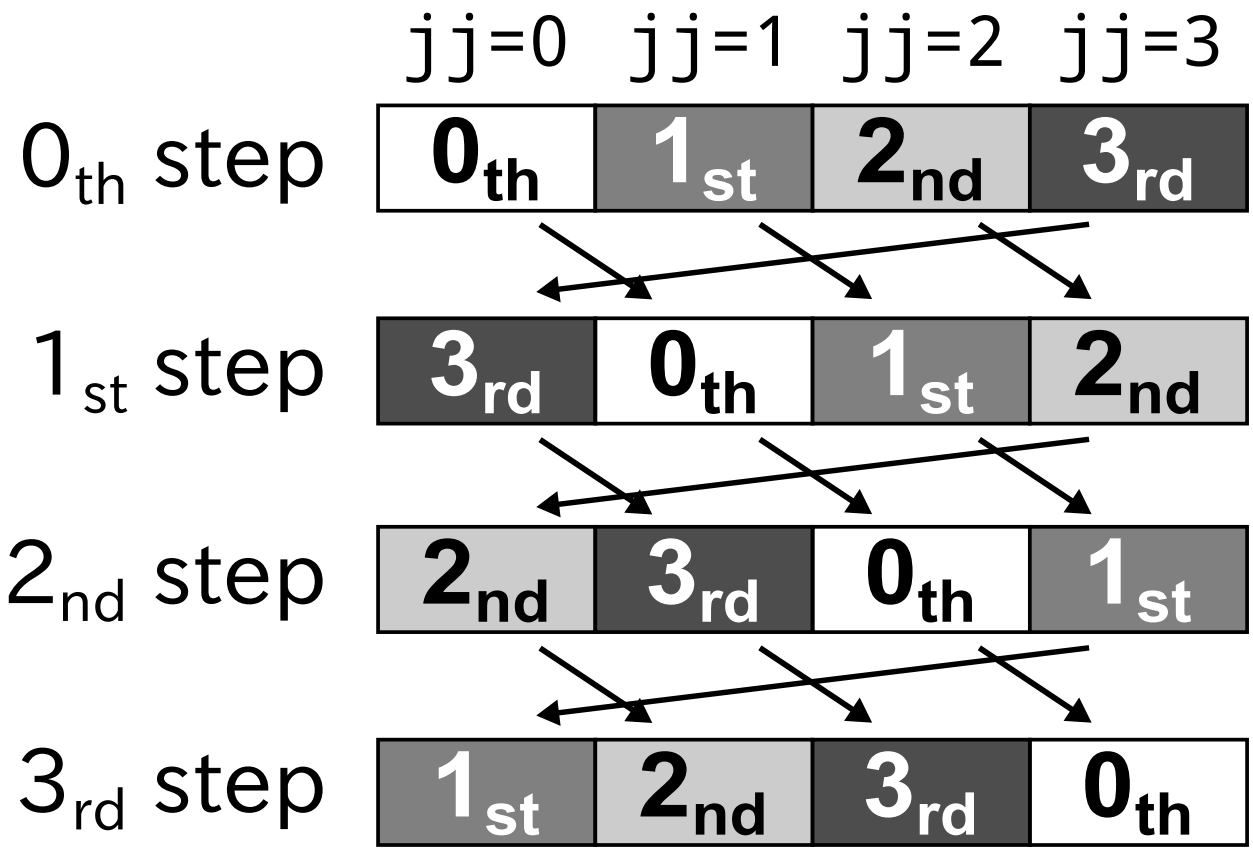


Fig. 7.1: Basic idea of accumulating gravitational acceleration without explicit synchronization.

```
1              float src[4];
2  __shared__  float dst[4];
3
4                        dst[jj]  = src[jj];
5  jj = (jj + 1) % 4;    dst[jj] += src[jj];
6  jj = (jj + 1) % 4;    dst[jj] += src[jj];
7  jj = (jj + 1) % 4;    dst[jj] += src[jj];
```

Fig. 7.2: Pseudo code for accumulating gravitational acceleration without explicit synchronization.

7.3 Proposal

Here, we propose a new technique to accumulate gravitational force without synchronization or usage of atomic operations. Our proposal makes use of the following two features.

1. The target array has 4 components (3 components for gravitational acceleration, and another one for gravitational potential).

2. Threads contained in a warp are synchronized implicitly since all of 32 threads in a warp perform operations at the same time.

Therefore, accumulating gravitational acceleration is possible by shifting component to be accumulated, if a warp contains all relating threads.

Figures 7.1 and 7.2 show the above method to accumulate gravitational acceleration for a case of 4 threads calculate an i-particle, the most straightforward case. Here, we introduce our method in detail for the case. The 4 threads which labeled by an index `jj` = 0, 1, 2, 3 have individual array "`src[4]`" to store their own

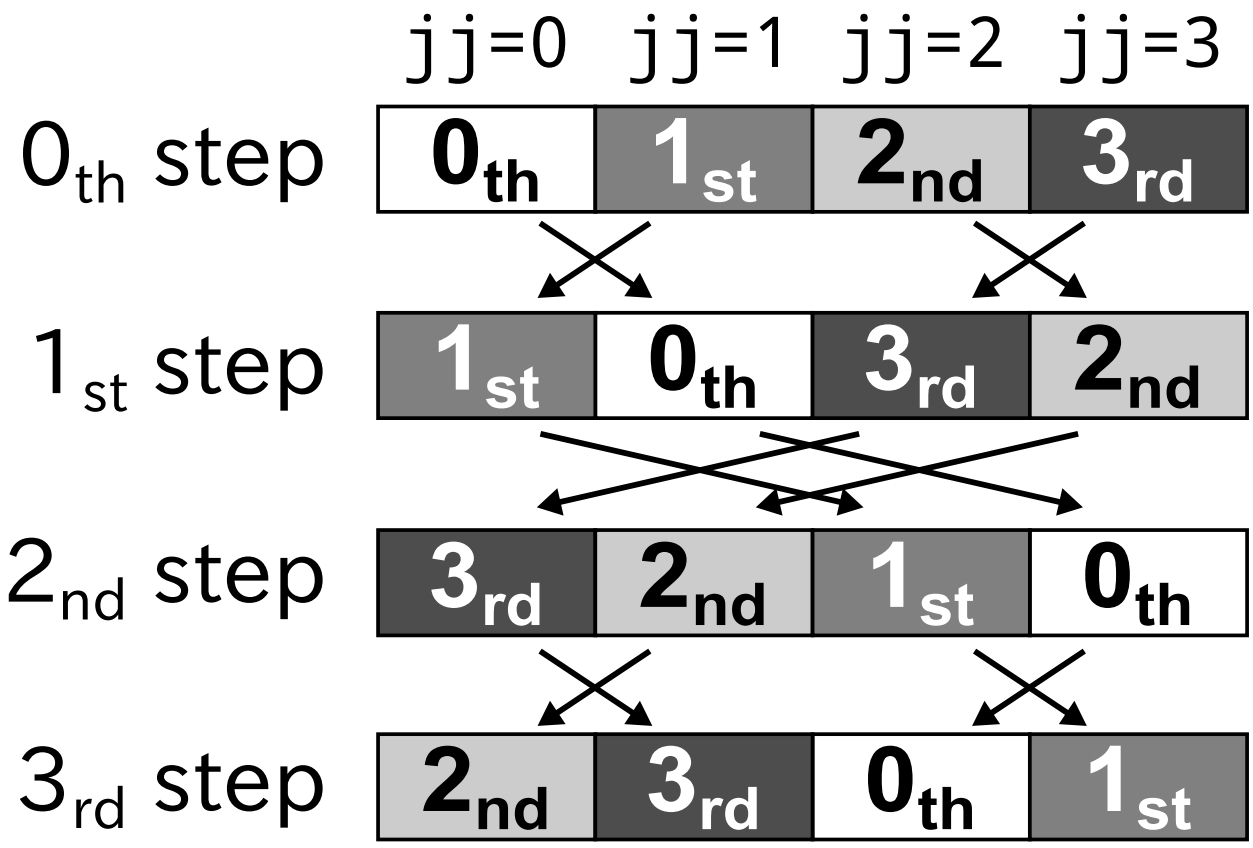


Fig. 7.3: Algorithm to accumulate gravitational acceleration without explicit synchronization in optimized form.

```
1            float src[4];
2 __shared__ float dst[4];
3
4 int jj = threadIdx.x;/* (0, 1, 2, 3) */
5
6          dst[jj]  = src[jj];
7 jj ^= 1; dst[jj] += src[jj];
8 jj ^= 2; dst[jj] += src[jj];
9 jj ^= 1; dst[jj] += src[jj];
```

Fig. 7.4: Pseudo code for accumulating gravitational acceleration without explicit synchronization in optimized form.

results of calculations, and these results must be unified to shared array "`dst[4]`". At the first step, the threads store the data from `src[jj]` to `dst[jj]` (the zeroth step of Fig. 7.1, the fourth line of Fig. 7.1). Then, the array index `jj` is shifted one by one, and the data stored in `src[jj]` is added to `dst[jj]` until whole threads update whole components of gravitational acceleration. Since the whole threads do not touch the same array element at the same time and update all elements at the final step, the accumulation process completes without explicit synchronization.

The previous paragraph explains how to accumulating gravitational acceleration in the most simple form. In this paragraph, we introduce optimized version of the algorithm. At the first step, the threads store the data from `src[jj]` to `dst[jj]` (the zeroth step of Figure 7.3, the sixth line of Figure 7.4). On the next step, exclusive OR (`XOR`) between `jj` and unity is performed, and the data stored in `src[jj]` is added to

```
1            float src[4];
2 __shared__ float dst[4];
3
4 int jj = threadIdx.x;/* (0, 1) */
5
6          dst[jj]  = src[jj];
7 jj ^= 2; dst[jj]  = src[jj];
8 jj ^= 1; dst[jj] += src[jj];
9 jj ^= 2; dst[jj] += src[jj];
```

Fig. 7.5: Pseudo code for $T_{\rm sub} = 2$.

```
1            float src[4];
2 __shared__ float dst[2][4];
3
4 int ii = threadIdx.y;/* (0, 1, ..., 31) */
5 int jj = threadIdx.x;/* (0, 1, ..., 7) */
6 int gid = jj >> 2;
7
8 jj &= 3;
9 ii = (ii << 1) + gid;
10
11          dst[ii][jj]  = src[jj];
12 jj ^= 1; dst[ii][jj] += src[jj];
13 jj ^= 2; dst[ii][jj] += src[jj];
14 jj ^= 1; dst[ii][jj] += src[jj];
15
16 if( !gid )  dst[ii][jj] += dst[ii + 1][jj];
```

Fig. 7.6: Pseudo code for $T_{\rm sub} = 8$.

`dst[jj]` (the first step of Fig. 7.3, the seventh line of Fig. 7.4). The operation `XOR` with unity is taken to flip the lowest bit of the index `jj` with the smallest cost to shift the index of the array to be updated (execution of logical operation for 32-bit integer needs only 1 cycle for GPUs of compute capability 2.0). In the forthcoming step, `src[jj]` is added to `dst[jj]` after performing `XOR` between `jj` and two. At the end of the procedure, each thread updates the one remained component, evaluated by taking `XOR` between `jj` and unity.

Figures 7.5, 7.6, and 7.7 show the pseudo code in remained cases: $T_{\rm sub}$ is two, eight, and sixteen, respectively. Special treatments are required to accumulate gravitational acceleration in such $T_{\rm sub}$ unlike the case of $T_{\rm sub}$ is four (when the number of threads share the same i-particle and that of array elements are identical). First case is that the number of the threads is less than that of the array elements ($T_{\rm sub} = 2$). In this case, storing data to the shared memory is divided into two operations as shown in Fig. 7.5. The latter case is that the number of the threads is greater than that of the array elements ($T_{\rm sub} = 8$, 16). Now, introducing exclusive operations is inevitable. Therefore, Figures 7.6 and 7.7 include some `if` statements in the final phase.

```
1            float src[4];
2  __shared__ float dst[4];
3
4  int ii = threadIdx.y;/* (0, 1, ..., 15) */
5  int jj = threadIdx.x;/* (0, 1, ..., 15) */
6  int gid = jj >> 2;
7
8  jj &= 3;
9  ii = (ii << 2) + gid;
10
11           dst[ii][jj]  = src[jj];
12 jj ^= 1; dst[ii][jj] += src[jj];
13 jj ^= 2; dst[ii][jj] += src[jj];
14 jj ^= 1; dst[ii][jj] += src[jj];
15
16 if( !(gid >> 1) ){
17   dst[ii][jj] += dst[ii + 2][jj];
18   if( gid == 0 ){
19     dst[ii][jj] += dst[ii + 1][jj];
20   }
21 }
```

Fig. 7.7: Pseudo code for $T_{\mathrm{sub}} = 16$.

Chapter 8 Implementation and Performance Optimization

Many earlier studies reported that high performance can be achieved by massive parallelization about i-particles (Hamada & Iitaka 2007; Nyland et al. 2007; Hamada et al. 2009; Hamada & Nitadori 2010). This chapter presents implementation of the CUDA code based on Nyland et al. (2007). Additional optimizations are introduced in Sections 8.1 and 8.2. Values of some parameters which determine performance are determined in the last section.

8.1 Reducing Number of Operations

The innermost loop of N-body simulation requires the value of $\boldsymbol{r}_{ji}{}^2 + \epsilon^2$. By effectively employing fused multiply-add (FMA) operations, the number of operations required to the calculation is reduced. The most important difference is in the calculation of $\mathrm{r}_{ji}{}^2 + \epsilon^2$. Both implementations use a `float3` variable `rji`, a `float` variable `eps2`, and a `float` variable `r2` to store the displacement vector $\boldsymbol{r}_{ji} \equiv \boldsymbol{x}_j - \boldsymbol{x}_i$, ϵ^2, and the calculated value of $\boldsymbol{r}_{ji}{}^2 + \epsilon^2$, respectively, as shown in Figure 8.1. The source codes for the two implementations in Fig. 8.1 appear to be almost identical; however, the generated sets of instructions are quite different. The implementation in Nyland et al. (2007) first performs one multiplication and two FMA operations, followed by one addition. According to the CUDA C Programming Guide (Nvidia 2012), this code's computational cost is four clock cycles. On the other hand, the implementation in this work performs only three FMA operations, which use three clock cycles, and is therefore faster than the implementation in Nyland et al. (2007). The resulting optimization is primarily due to the fact that `r2` is calculated in the innermost loop; this small detail directly enhances performance.

We also eliminate `if`-statements to avoid additional computational costs when dealing with arbitrary numbers of particles. Instead, we add massless particles to fill the i-particle and j-particle arrays when the number of particles is not an integral multiple of the number of threads.

8.2 Hiding Accessing Time to Global Memory

The present study implements an additional optimization to achieve even better performance. Earlier studies have frequently emphasized the importance of utilizing shared memory because reducing the number of global memory accesses is an effective way to improve performance. In Nyland et al. (2007), two

```
1 /* Implementation of Nyland et al. (2007) */
2 r2  =         rji.x * rji.x + rji.y * rji.y + rji.z * rji.z;
3 r2 += eps2;
4
5 /* This work */
6 r2  = eps2 + rji.x * rji.x + rji.y * rji.y + rji.z * rji.z;
```

Fig. 8.1: Calculations of $\boldsymbol{r}_{ji}{}^2 + \epsilon^2$ in Nyland et al. (2007) and this work.

```
1  /* Nyland et al. (2007) */
2  for(jh = 0; jh < Nj; jh += blockDim.x){
3    __syncthreads();
4    body[ii] = jpos[jh + threadIdx.x];
5    __syncthreads();
6    for(j = 0; j < blockDim.x; j++)
7      calc_gravity(...);
8  }
9
10 /* This work */
11 for(jh = 0; jh < Nj; jh += blockDim.x){
12   float4 pj = jpos[jh + threadIdx.x];
13   __syncthreads();
14   body[ii] = pj;
15   __syncthreads();
16   for(j = 0; j < blockDim.x; j++)
17     calc_gravity(...);
18 }
```

Fig. 8.2: Difference between previous implementations and this study's calculation of gravitational force.

`__syncthreads()` instructions are performed immediately before and after an instruction that loads from the global memory and stores to the shared memory, as shown in Figure 8.2. Here the shared memory and global memory each contain a `float4` array –`body[]` and `jpos[]`, respectively– to store the positions of j-particles. In the implementation, the load from `jpos[]` and the store to `body[]` are performed in the same instruction. The two `__syncthreads()` instructions are necessary to maintain consistency while updating the set of j-particles shared by entire threads within a block.

As far as a streaming multiprocessor (SM) contains multiple blocks, the memory access time for a block can be hidden by overlapping the access with the calculations of other blocks. The number of blocks per SM is a few in many cases owing to the limitations of the shared memory's capacity and the number of registers, and a block typically contains several warps. When an SM containing two blocks is used for an N-body calculation, there is only one possibility for overlapping instructions: one block calculates particle–particle interactions and the other block executes the load and store instructions. This is the only possibility because two `__syncthreads()` instructions separate the global memory accesses and particle–particle interaction calculations. Because the `__syncthreads()` instruction synchronizes entire threads within a block, the separate execution allows block-level overlapping but rules out warp-level overlapping of instructions in the implementations shown in Fig. 8.2.

Therefore, the present study augments the possibilities of overlapping instructions through a careful modification of the CUDA code in Fig. 8.2. We separate the instructions that load from the global memory and store to the shared memory. This allows CUDA schedulers to perform warp-level overlapped execution of these instructions since the execution of the load instruction and calculation is not separated. Because the number of blocks per SM cannot exceed the number of warps per SM, the new implementation provides more opportunities to hide slow global memory access time.

```
1  __global__ void calc_gravity(int Ni, float4 *ipos, float4 *acc, int Nj,
       float4 *jpos)
2  {
3    __shared__ float4 body[NTHREADS];
4    int  i =  blockIdx.x * blockDim.x
5           + threadIdx.x;
6    int ii = threadIdx.x;
7
8    /* set i-particles */
9    /* x, y, z, and mass */
10   float4 pi = ipos[i];
11   /* ax, ay, az, and potential */
12   float4 ai = {0.0f, 0.0f, 0.0f, 0.0f};
13
14   int nj = blockDim.x;
15   for(int jh = 0; jh < Nj; jh += nj){
16     /* set j-particles */
17     float4 pj = jpos[jh + ii];
18     __syncthreads();
19     body[ii] = pj;
20     __syncthreads();
21
22 #pragma unroll NUNROLL
23     for(int j = 0; j < nj; j++){
24       pj = body[j];
25
26       float4 rji;
27       rji.x = pj.x - pi.x;
28       rji.y = pj.y - pi.y;
29       rji.z = pj.z - pi.z;
30       rji.w = rsqrtf(eps2 + rji.x * rji.x + rji.y * rji.y + rji.z * rji.z)
             ;
31       /* ai.w += rji.w * pj.w; */
32       rji.w = rji.w * rji.w * rji.w * pj.w;
33       ai.x += rji.x * rji.w;
34       ai.y += rji.y * rji.w;
35       ai.z += rji.z * rji.w;
36     }
37   }
38   atomicAdd(&(acc[i].x), ai.x);
39   atomicAdd(&(acc[i].y), ai.y);
40   atomicAdd(&(acc[i].z), ai.z);
41   atomicAdd(&(acc[i].w), ai.w);
42 }
```

Fig. 8.3: Source code for gravitational attraction calculation for a single GPU.

8.3 Determing Configuration

Figure 8.3 shows the implementation of the presented code in case of T_{sub} is unity. At the end of Fig. 8.3, we used atomic instructions to update information about i-particles' acceleration in the global memory. These atomic instructions are necessary to ensure consistency when the computations are performed on multiple GPUs.

The crucial parameters for enhancing performance in the figure are the number of threads per block

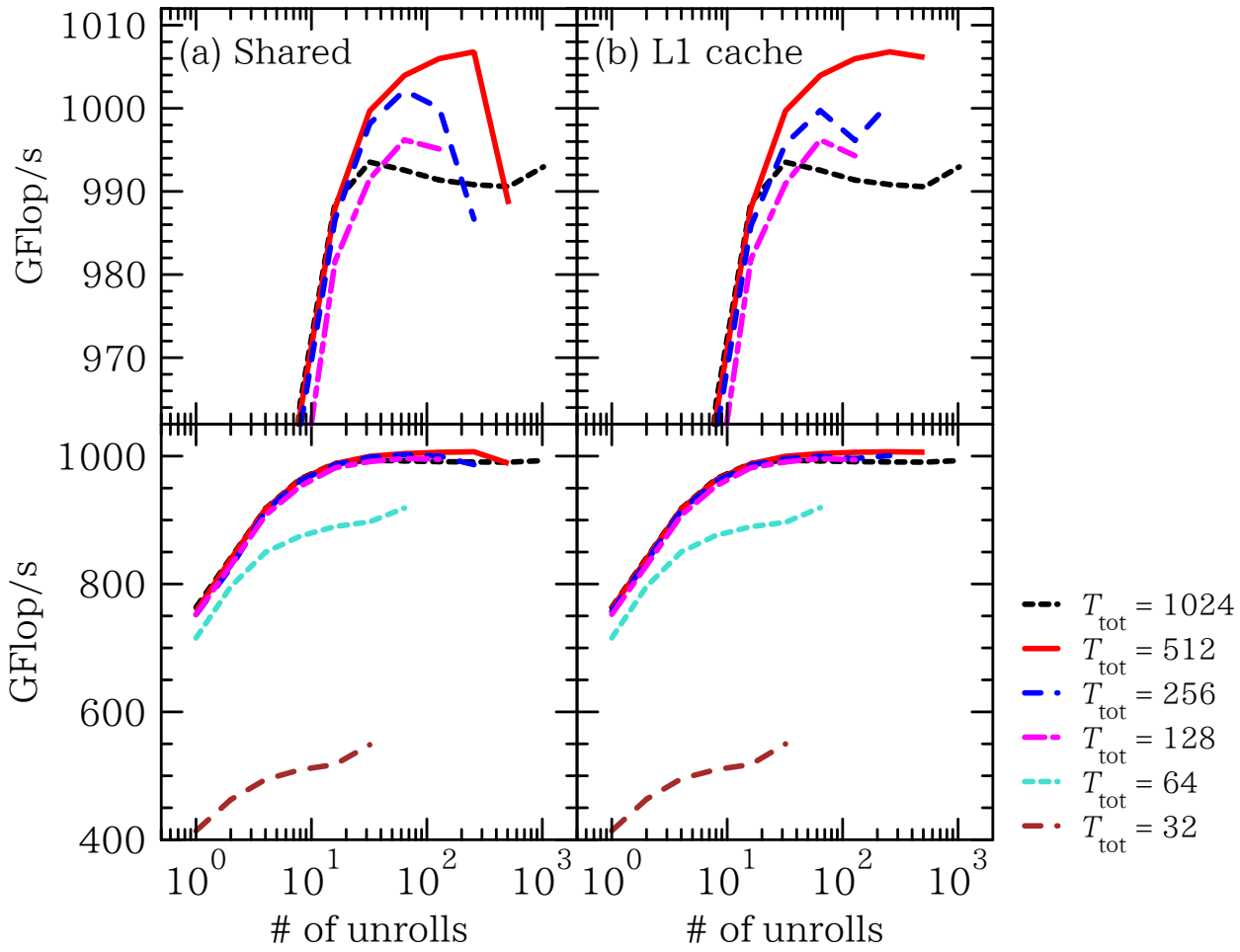


Fig. 8.4: Results of parameter study to determine the optimal parameter sets where the $T_{\rm sub}$ is unity, and the number of N-body particles is 1,048,576. Measured performance in single precision is plotted as a function of the number of unrolls. The brown, cyan, magenta, blue, red, and black lines indicate 32, 64, 128, $\cdots$, and 1,024 threads per block, respectively. The left and right panels show the results for the "shared memory preferred" and "L1 cache preferred" configurations, respectively. The top panels are enlargements of the bottom panels.

(`NTHREADS`), the number of unrolls for the innermost loop (`NUNROLL`), and the cache configuration ("shared memory preferred" or "L1 cache preferred"). In Nyland et al. (2007), it was claimed that the key to high performance is to have a large number of threads per block and a large number of unrolls. However, the complicated relation between these parameters prompts us to determine the optimal parameter settings for achieving the highest performance.

We have performed a parameter study to examine the best configurations on an NVIDIA Tesla M2090 board when the number of N-body particles is 1,048,576 (Table 10.1 provides more detailed information). Figure 8.4 shows the performance measured in single precision as a function of the number of unrolls: the plotted lines show the results for different values of `NTHREADS` ($T_{\rm tot}$ in the legend of Fig. 8.4) and the left/right panels show the results for "shared memory preferred"/"L1 cache preferred" configurations. The top panels show that the optimal number of threads per block is 512. The peak performance for the parameter study is 1006.7 GFlop/s in single precision, which is reached in the case of 256 unrolls and the "L1 cache preferred" configuration. The peak performance in excess of 1 TFlop/s in single precision is due to the optimization introduced in this study. The effects of the optimization itself are not so great, but it is the tipping point for the performance exceeding 1 TFlop/s per single NVIDIA Tesla M2090 board.

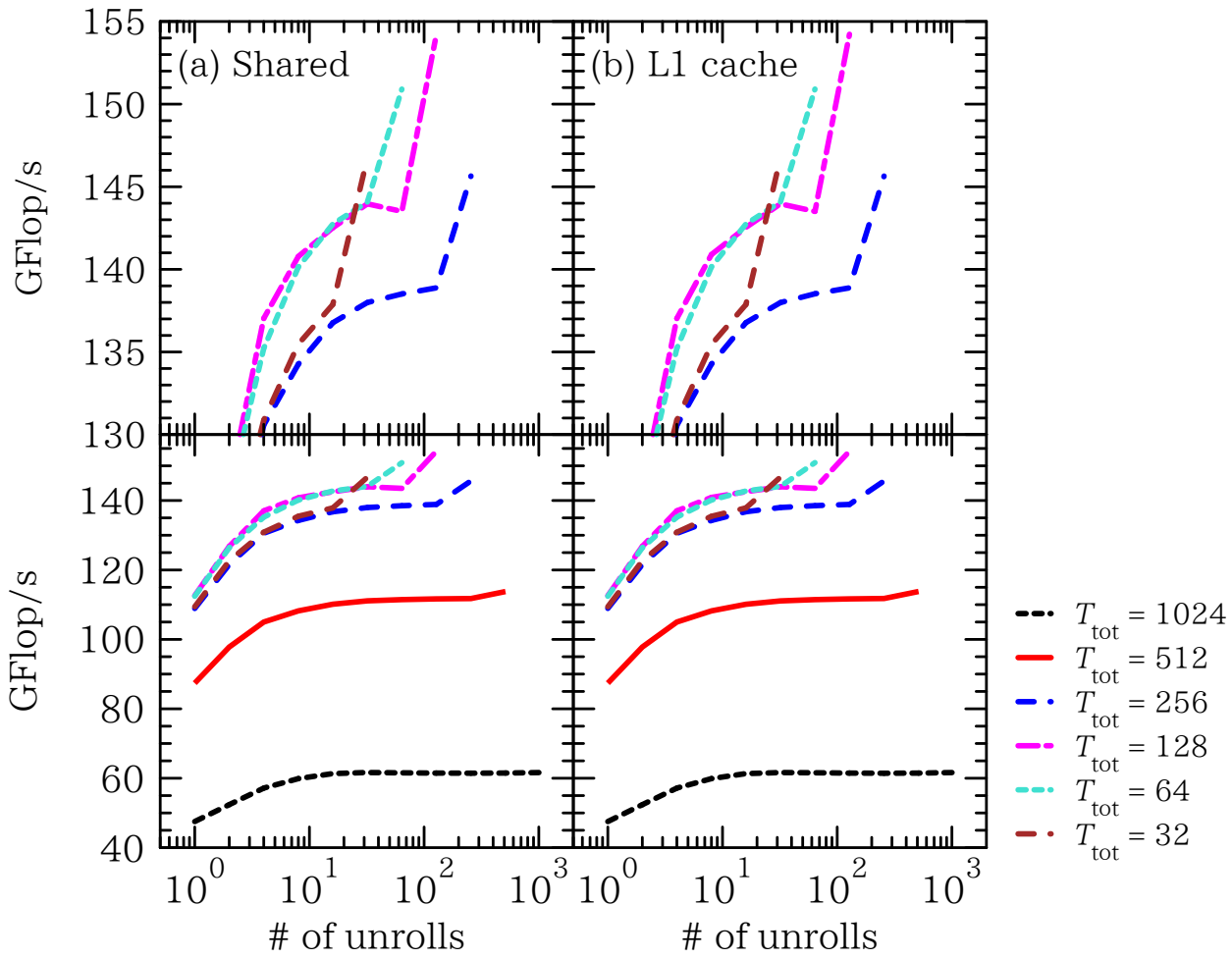


Fig. 8.5: Results of parameter study to determine the optimal parameter sets where the T_{sub} is unity, and the number of N-body particles is 1,024. Legends are identical to Fig. 8.4.

Figures 8.6 – 8.9 shows the performance measured in single precision as a function of the number of unrolls when the number of N-body particles is 1,024 for $T_{\mathrm{sub}} = 1, 2, \cdots, 16$. As a result, "L1 cache preferred" configuration tends to show higher performance compared to "shared memory preferred" configuration. Furthermore, excluding the case of T_{sub} is sixteen, measured performance with $T_{\mathrm{tot}} = 1{,}024$ is very low. This is because, utilizing as many as SMs by reducing T_{tot} improves the performance in case of the number of N-body particles is small. Therefore, increasing T_{sub} allows to use many SMs and increases performance even for the higher value of T_{tot}. Table 8.1 summarizes the optimal configuration in each T_{sub}.

Table. 8.1: The optimal configuration for $N = 1{,}024$

T_{sub}	T_{tot}	# of unrolls	cache configuration	measured performance
1	128	128	L1 cache preferred	154.2 GFlop/s
2	128	32	shared memory preferred	276.5 GFlop/s
4	256	32	shared memory preferred	502.6 GFlop/s
8	512	32	L1 cache preferred	759.8 GFlop/s
16	1,024	16	L1 cache preferred	703.0 GFlop/s

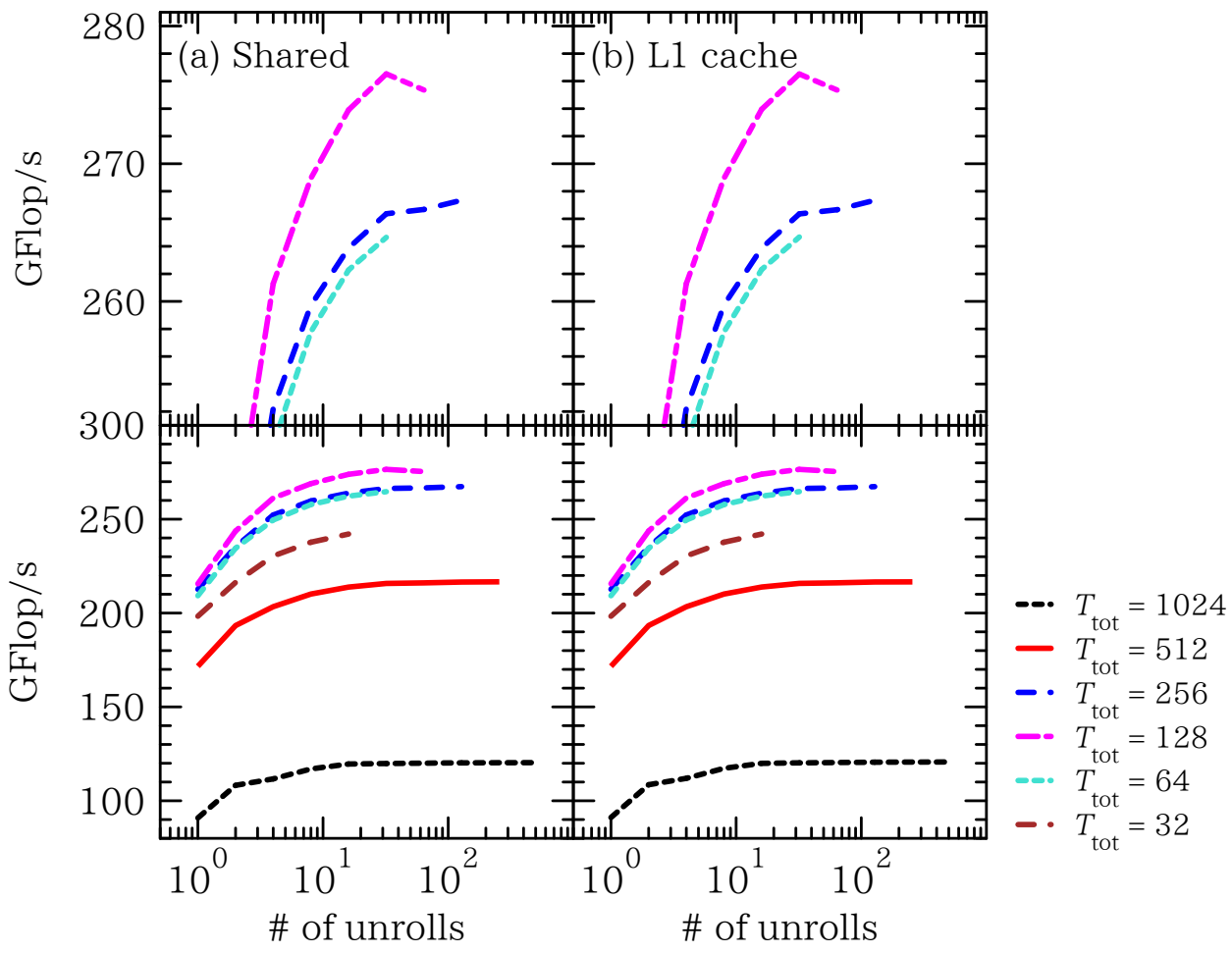


Fig. 8.6: Results of parameter study to determine the optimal parameter sets where the T_{sub} is two, and the number of N-body particles is 1,024. Legends are identical to Fig. 8.4.

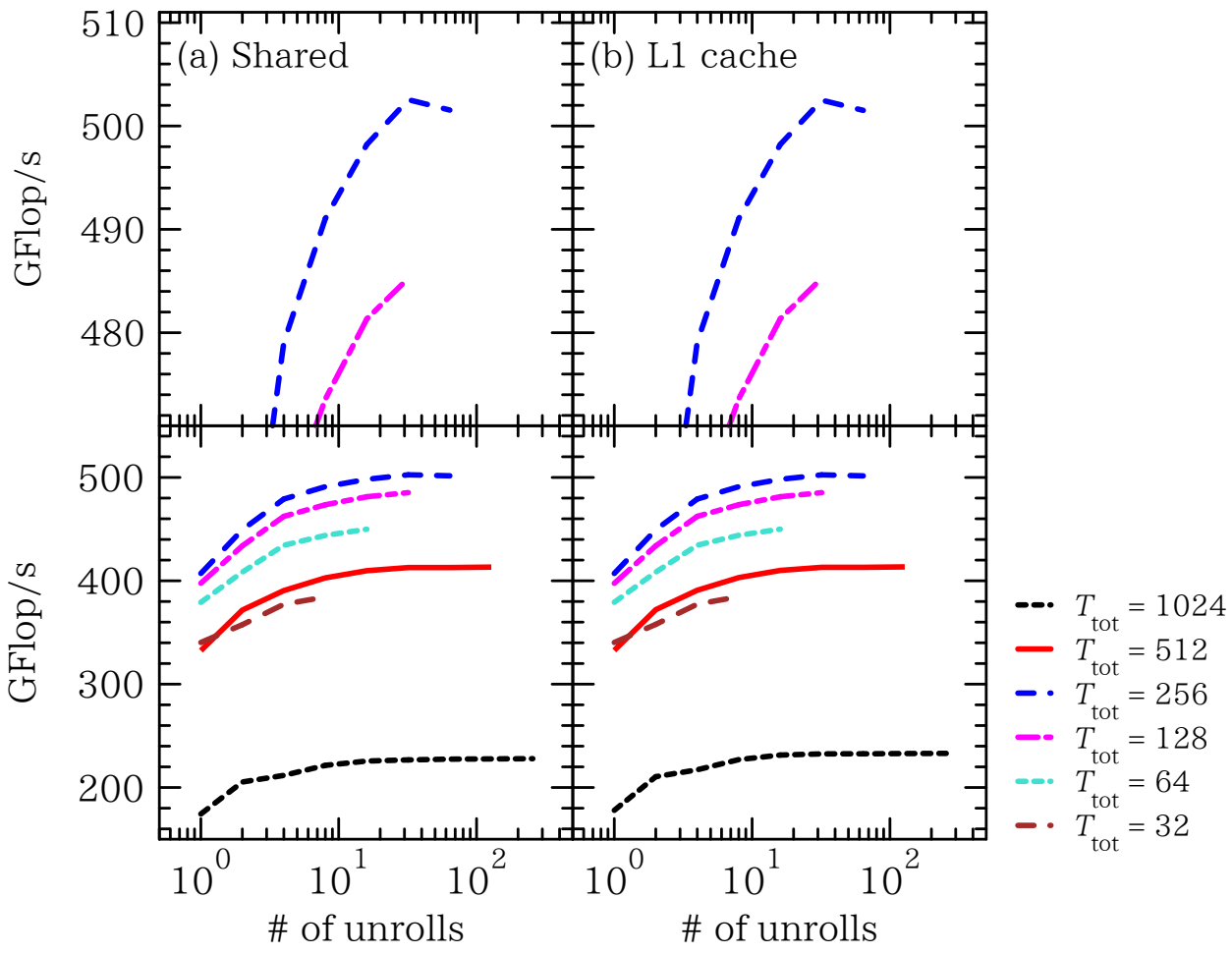


Fig. 8.7: Results of parameter study to determine the optimal parameter sets where the $T_{\rm sub}$ is four, and the number of N-body particles is 1,024. Legends are identical to Fig. 8.4.

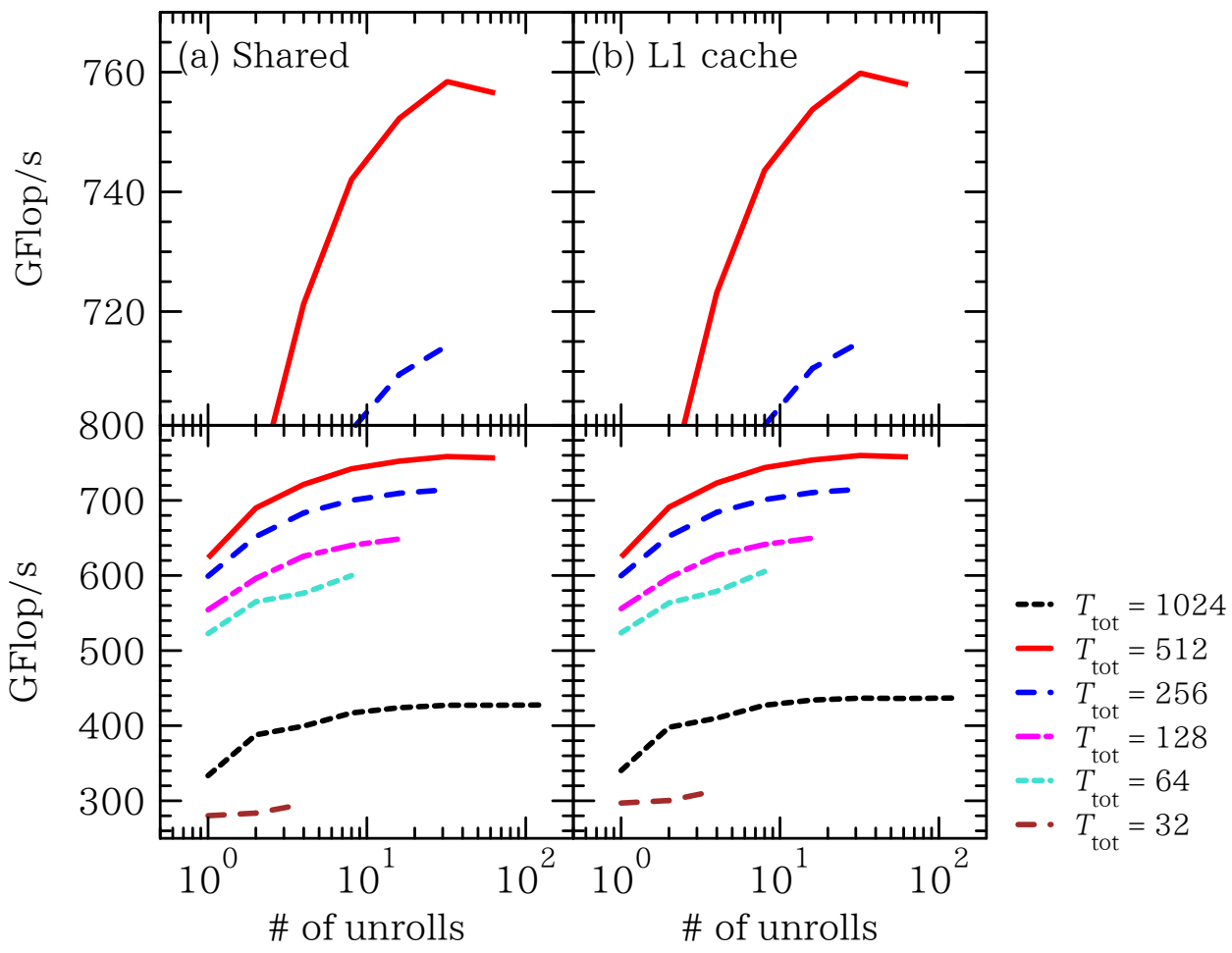


Fig. 8.8: Results of parameter study to determine the optimal parameter sets where the T_{sub} is eight, and the number of N-body particles is 1,024. Legends are identical to Fig. 8.4.

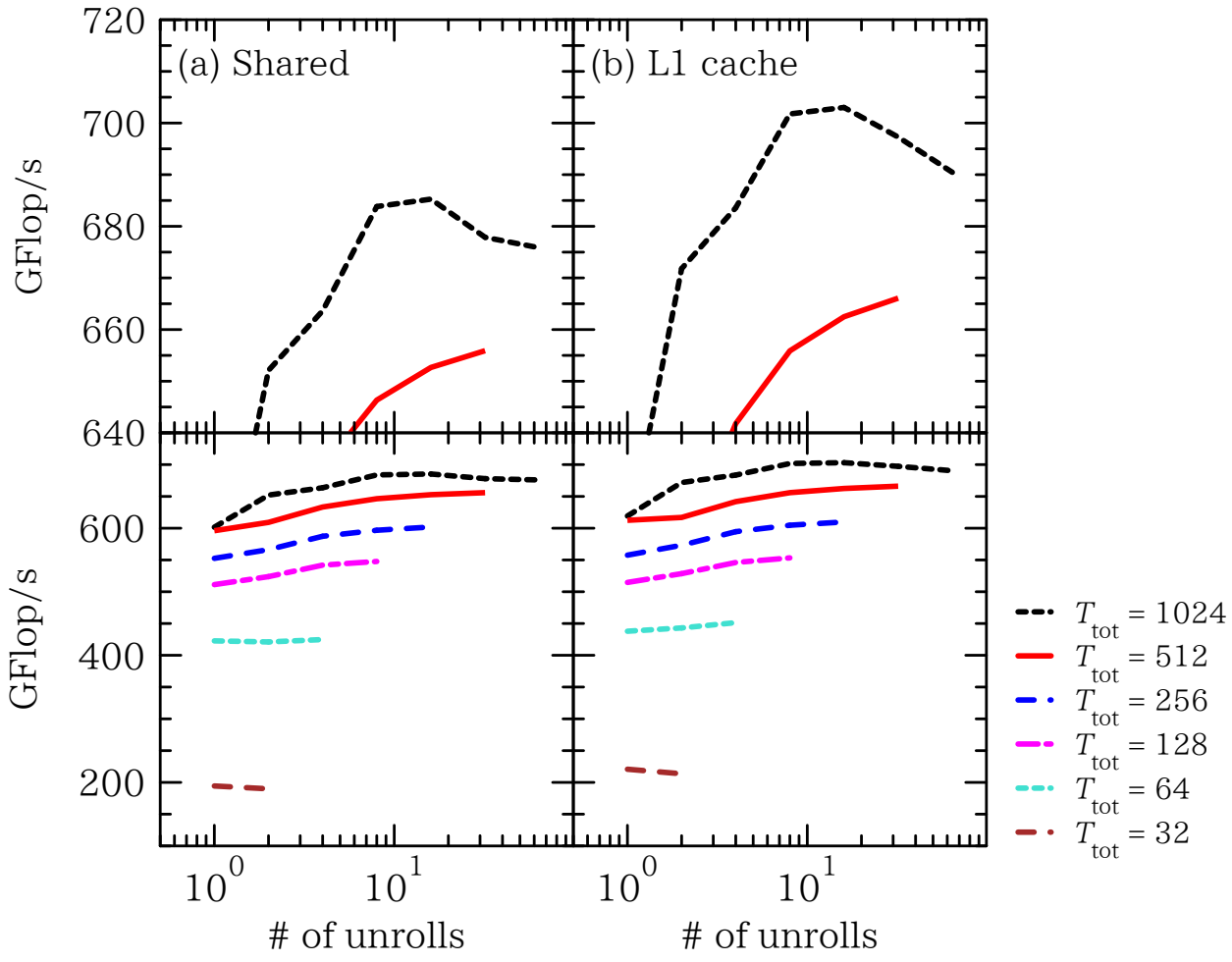


Fig. 8.9: Results of parameter study to determine the optimal parameter sets where the T_{sub} is sixteen, and the number of N-body particles is 1,024. Legends are identical to Fig. 8.4.

Chapter 9 OpenMP/MPI Hybrid Parallelization

9.1 Parallelization based on OpenMP

When the position data for N-body particles are divided between two GPUs, communication between the two devices is necessary for calculating gravitational interaction. Here peer-to-peer memory access between the two devices is a good choice to reduce communication time because peer-to-peer memory access does not require accessing memories through the CPU. Because peer-to-peer memory access requires sharing a pointer to global memory on each device within a process, we have parallelized the code using OpenMP.

An OpenMP thread controls a GPU, and each GPU stores half of the N-body particles: device 0 has position data for particles 0 through $N/2-1$ and device 1 has position data for the remaining N particles. The algorithm for calculating gravitational interactions and orbit integration using a leap-frog integrator is shown in Figure 9.1.

Communication between GPUs (step 3 in Fig. 9.1) reduces the parallel efficiency by interrupting the kernel function's execution (step 4 in Fig. 9.1). Therefore, additional changes are necessary to realize high scalability when many GPUs are used. In the algorithm in Fig. 9.1, the second and third steps can be performed concurrently. This enables overlapped execution of the gravitational interaction calculations and communications between the two GPUs, thus hiding the communication time. We use two CUDA streams to overlap the two instructions: one CUDA stream performs the first and second steps, while the other executes the remaining steps after the first has been executed.

Each of the two implementations below (SendSync in Figure 9.2 and SyncRecv in Figure 9.3) can be used to realize the algorithm in Fig. 9.1 (steps 2 to 4). SendSync is named after the data send and synchronize before the calculation, and SyncRecv is named after the synchronize and data receive before the calculation. It is not obvious which implementations is better; however, it is crucial to determine this for deciding how to implement the code. To this end, we must examine factors such as time needed to synchronize the CUDA

1: Update position data for $N/2$ i-particles on each device.
2: Calculate gravitational interactions among $N/2$ i-particles on each device.
3: Copy the updated position data for $N/2$ particles from the peer device as a new set of j-particles.
4: Calculate gravitational interactions between $N/2$ i-particles and $N/2$ j-particles.
5: Update the velocity data for $N/2$ i-particles on each device

Fig. 9.1: Algorithm for OpenMP parallelization.

1: Calculate gravitational interactions among $N/2$ i-particles on each device (stream ID `sid`).
2: Flip the stream ID `sid` through an exclusive or with unity.
3: Send the position data for j-particles to the peer device (stream ID `sid`).
4: Synchronize instructions related to stream ID `sid`.
5: Calculate gravitational interactions between $N/2$ i-particles and $N/2$ j-particles (stream ID `sid`).

Fig. 9.2: Implementation of "SendSync" mode.

1: Calculate gravitational interactions among $N/2$ i-particles on each device (stream ID `sid`).
2: Flip the stream ID `sid` through exclusive or with unity.
3: Synchronize OpenMP threads to confirm that both devices have already set their own j-particle data.
4: Receive the position data for j-particles from the peer device (stream ID `sid`).
5: Calculate gravitational interactions between $N/2$ i-particles and $N/2$ j-particles (stream ID `sid`).

Fig. 9.3: Implementation of "SyncRecv" mode.

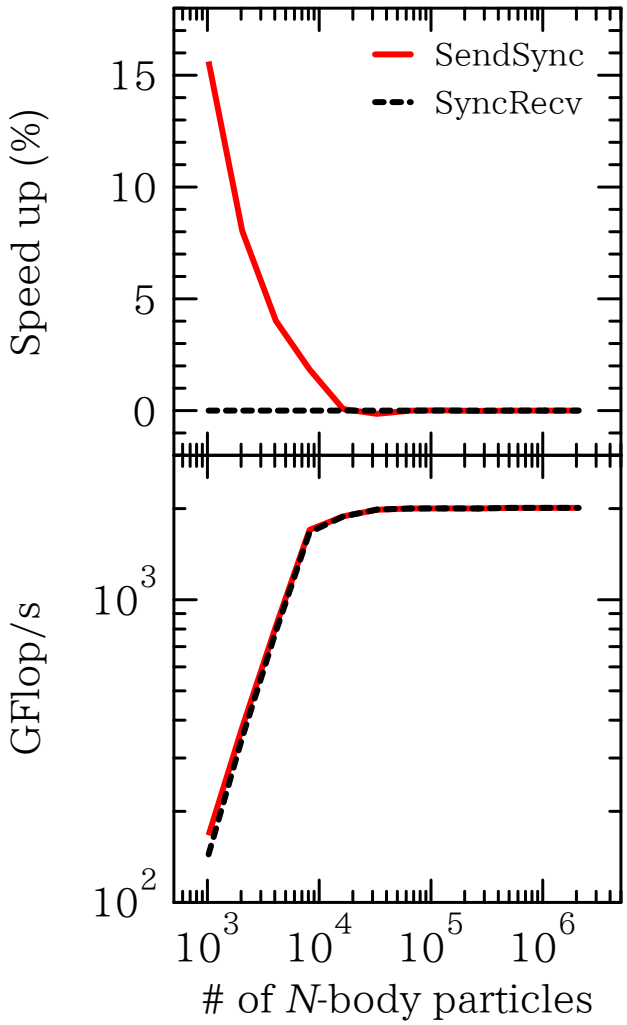


Fig. 9.4: Comparison between the SendSync (solid red lines) and SyncRecv (dotted black lines) modes. The bottom panel shows the measured performance of both modes as a function of the number of N-body particles. The top panel shows the speedup rate for the SendSync mode compared with the SyncRecv mode.

streams and OpenMP threads, which implementation is easier to optimize for the CUDA/C compiler, and how the single GPU code performs.

Figure 9.4 compares the measured performance of the algorithm using SendSync ("SendSync mode") and SyncRecv ("SyncRecv mode"). The red (black) line in the bottom panel shows the measured performance of the SendSync (SyncRecv) mode as a function of the number of N-body particles. The lines in the top panel represent the speedup rates from the SyncRecv mode. The experiment shows that SendSync mode is suitable in the test environment (listed in Tab. 10.1) for the low-N region.

9.2 Parallelization using Message Passing Interface

To handle data distribution on multiple GPUs attached to a distributed computational node, we must use the Message Passing Interface (MPI). It is straightforward and simplest to implement this case as a natural extension of the algorithm presented in Section 9.1. However, we have developed a more sophisticated algorithm to achieve a high degree of scalability by reducing the number of communications between the MPI processes. The overall algorithm is roughly divided into two phases: transfer and accumulation phases.

The transfer phase, shown in Figure 9.5, is a natural extension of the algorithm in Section 9.1 for a multinode environment. In this phase, each MPI process transfers the position data for j-particles to the next MPI process (line 2 of Fig. 9.5) until all MPI processes complete calculation of gravitational interactions for all j-particles. If n_{t} MPI processes are involved in the transfer phase, the total number of communications is $n_{\mathrm{t}} - 1$ owing to the ring-like communication pattern.

To realize overlapped executions of the memory copy instructions from CPU to GPU and calculations on the GPU, we again use two CUDA streams, as described in Section 9.1. In addition, we utilize nonblocking communication functions such as `MPI_Isend` and `MPI_Irecv` to minimize the impacts of communications on the parallel efficiency by overlapping communication among MPI processes with other instructions.

Figure 9.6 shows the accumulation phase that we introduced to reduce the number of communications. The fundamental idea behind the accumulation phase is quite simple: an MPI process sends position data for all known j-particles, namely the MPI process's initial j-particle data and those it has received in earlier stages, to other MPI processes.

Figure 9.7 presents a schematic view of the accumulation phase for eight MPI processes. Each MPI process has two arrays (arrays 0 and 1 in Figures 9.6 and 9.7). In the initial stage (stage 0), two MPI processes constitute a fundamental unit represented by two black boxes and two white boxes in Fig. 9.7. The black boxes (corresponding to array 0 in Fig. 9.6) contain position data for j-particles while the white boxes do not contain any data. In stage 0, every MPI process sends the data in its array 0 to array 1 of the other MPI process in the same unit, as indicated by the arrows (step 3 in Fig. 9.6). Before moving to

1: **for** $s = 0$ to $N_{\mathrm{proc}} - 2$ **do**
2: Send data stored in the array 0 to array 1 on the next process connected as ring structure.
3: Copy position data on array 1 as a new set of j-particles from host process to each device.
4: Calculate gravitational interactions between i-particles and j-particles.
5: Add the data on the array 0 to the array 1.
6: Swap the array 0 and the array 1.
7: **end for**

Fig. 9.5: Algorithm of the transfer phase.

1: **while** data size of j-particles does not exceed the capacity of arrays 0 and 1, **or** the number of j-particles becomes the half of the total number of N-body particles **do**
2: Determine the pair process to exchange position data for N-body particles.
3: Send data stored in array 0 to array 1 on the pair process.
4: Copy position data on array 1 as a new set of j-particles from host process to each device.
5: Calculate gravitational interactions between i-particles and j-particles.
6: Add the data on array 0 to array 1.
7: Swap arrays 0 and 1.
8: **end while**

Fig. 9.6: Accumulation phase algorithm.

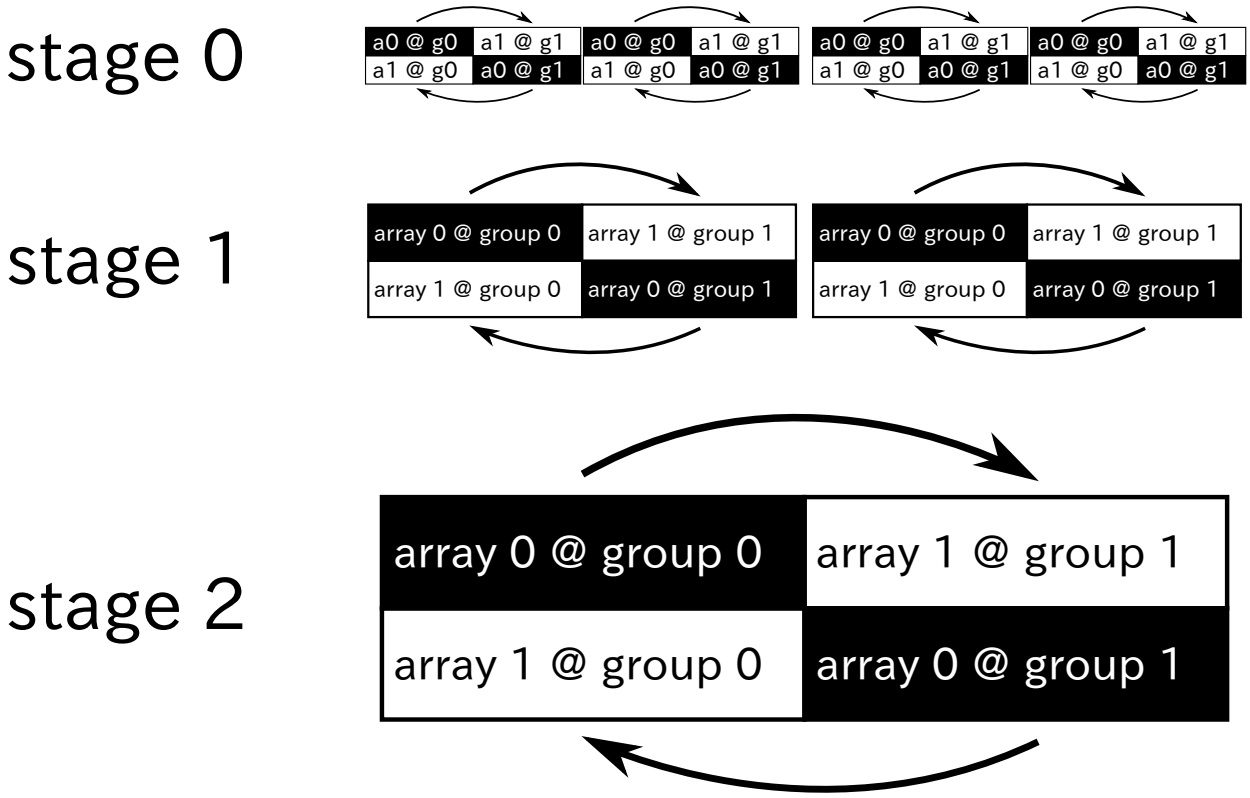


Fig. 9.7: Schematic view of data set growth at each stage of the accumulation phase. The black and white blocks are the source and destination arrays of the MPI communication, respectively.

the next stage, we merge the position data for the j-particles in arrays 0 and 1 in each unit (steps 6 and 7). At the end of this stage, both MPI processes have the same data for j-particles. At the beginning of stage 1, the MPI processes constitute new groups by merging each pair of groups that already share a dataset (groups connected by arrows in Fig. 9.7). After pairs of groups from the previous stage have been unified, the same steps are repeated until the accumulation phase completes.

The transferred data size doubles in each stage, as shown in Fig. 9.7. The number of communications is $\log_2 n_\mathrm{a}$, where n_a is the maximum number of MPI processes constituting a unit in the accumulation phase. This is significantly smaller than $n_\mathrm{a} - 1$ owing to its implementation as a binary tree through the exchange of datasets. The process rank of the target process of the exchanging is given by exclusive or of MPI rank with unity `<< (stage ID)`. It should be noted that the accumulation phase communication pattern is suitable for tree network topologies but not for mesh-torus topologies because MPI processes must also access distant processes.

The accumulation phase can be repeated as long as the size of the dataset for j-particles does not exceed the capacity of the j-particles arrays. When the size reaches the array's capacity, the communication algorithm proceeds to the transfer phase to complete calculation of the gravitational interactions of all j-particles.

Chapter 10 Performance Measurements

10.1 Measurement Environment

We measured the performance of our code on the HA-PACS (Highly Accelerated Parallel Advanced system for Computational Sciences)[1], a GPU cluster at University of Tsukuba. The HA-PACS is equipped with GPUs and CPUs connected by PCI-express Generation 2.0. Each HA-PACS node consists of two Intel Sandy Bridge-EP sockets and four NVIDIA Tesla M2090 boards, and the CPUs support full-bandwidth connection of the GPUs without any performance bottlenecks. The interconnection network employs a dual-rail Infiniband QDR with a full bisection-bandwidth fat-tree configuration. The HA-PACS's peak performance is 1604 TFlop/s in single precision owing to the GPU's high performance of 1427 TFlop/s in single precision. Table 10.1 provides other details about the HA-PACS. Because two GPUs on every HA-PACS socket share the PCI lane, the HA-PACS is a suitable testbed for our implementation using peer-to-peer memory access. The fat-tree network topology, as opposed to mesh-torus interconnections, is suitable for the MPI communication's accumulation phase.

10.2 Performance of CUDA Code

In performance measurements, we measure executing time of calculating gravitational acceleration with direct summation. In other words, communication time between the CPU and the GPU via PCI Express is not included. We study dependence of performance on N_i, N_j, and thread-block structure. Parameter region of N is N_i, N_j = 256, 512, 1,024, $\cdots$, 1,048,576.

In order to compare the performance of our implementation and that of Nyland et al. (2007) under the same condition, we have to take three additional treatments. At first, the gravitational constant G

Table. 10.1: Measurement environment

Number of nodes	268
CPU	Intel Xeon E5-2670
	16 cores per node, 2.6 GHz
RAM	128 GB (DDR 3, 1600 MHz)
GPU	NVIDIA Tesla M2090
	512 CUDA cores, 1.3 GHz
	4 boards per node
Video RAM	6 GB (GDDR 5, ECC on) per GPU
C Compiler	icc 13.0.1.117 (gcc 4.4.5 compatibility)
MPI Library	Intel MPI 4.1.0.024
CUDA toolkit	4.2.9
CUDA Driver	304.54
Interconnection	Infiniband QDR $\times$2 rails
Network topology	Fat-Tree

[1] http://www.ccs.tsukuba.ac.jp/CCS/eng/research-activities/projects/ha-pacs

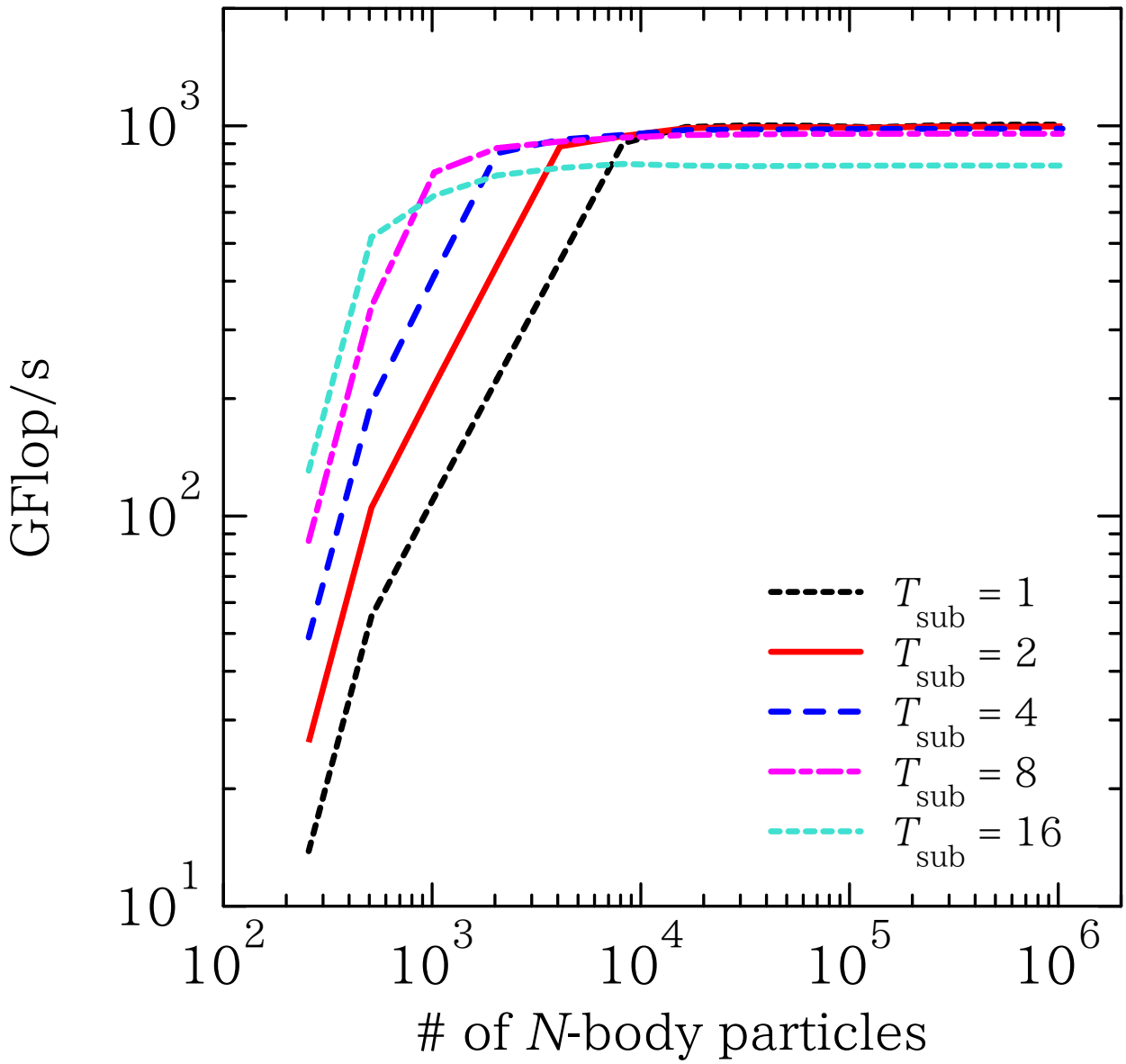


Fig. 10.1: Measured performance of the CUDA code against N.

is assumed to be unity in our implementation as same as Nyland et al. (2007). In addition, we omit calculation about gravitational potential in our implementation, not included in Nyland et al. (2007), as the second treatment. In our implementation, we have separately implemented function for orbit integration of N-body particles and gravitational interaction. This separate implementation is a desirable feature for easily changing implementation of time integration scheme (e.g. leap-frog integrator using fixed time step, or Runge-Kutta method using adaptive time step and so on). Thus, we separate the function in contrast to the all-in-one implementation, calculation of gravitational force and orbit integration is performed in a kernel function, of Nyland et al. (2007). Since this difference of implementation strategy leads the difference of computational amount and an unfair comparison, we omit time integration of position and velocity in our performance measurements as the third treatment. The effect of the first and third treatments on our performance measurements is expected to be negligibly small due to its smallness of computational amount ($O(N)$). On the other hand, effects of omitting the calculation of gravitational potential cannot be negligible due to its computational complexity of $O(N^2)$. However, the additional execution time corresponds to only 1 clock cycle, since the calculation can be performed by one additional FMA operation.

Figure 10.1 shows the measured performance for $T_{\mathrm{sub}} = 1$, 2, 4, 8, and 16. In this measurement, number of threads per block T_{tot} is always 512, so only one parameter T_{sub}, represents a number of threads which share an i-particle, determine the thread-block structure. That is to say, if $T_{\mathrm{sub}} = 2$ then two threads calculate acceleration of the common i-particle and a block calculate acceleration of 256 i-particles. The cache configuration is "L1 cache preferred", and the number of unrolls is $256/T_{\mathrm{sub}}$. The horizontal axis is the number of N-body particles, and the vertical axis shows the performance under the assumption of one interaction corresponds to 26 floating-point operations (see Appendix B). A black dotted line, red full line, blue dashed line, magenta dot-dashed line, and cyan dotted line show the performance of $T_{\mathrm{sub}} = 1$, 2, 4, 8, and 16, respectively.

The figure shows some clear behavior as follows.

1. The performance increase in low N region looks like proportional to N.
2. The measured performance saturate in the large N region.
3. Critical N which determine transition point of performance dependence on N tends to decrease with increasing of T_{sub}.
4. The sustained performance of $T_{\mathrm{sub}} = 16$ is much lower than that of other T_{sub}.

The achieved peak performance of 1006.7 GFlop/s for $N_i = N_j =$ 1,048,576, $T_{\mathrm{sub}} = 1$ corresponds to 75.7 % of NVIDIA Tesla M2090's theoretical peak performance for single-precision floating-point operations.

While the performance measurement shown in Fig. 10.1 focused on achieving high performance in the high N-region, we here show the result of the performance measurement focused in the low N-region. Figure 10.2 shows the result of performance measurement under the configuration listed in Tab. 8.1. Fig. 10.2 exhibits the same trend with Fig.10.1 with performance improvements in the low N-region.

Figures 10.3 and 10.4 compare the measured performance of the presented implementation and Nyland et al. (2007). The both figures clearly show the performance improvements, especially the peak performance.

To evaluate performance improvement from Nyland et al. (2007) more quantitatively, Figure 10.5 provides speed up of the presented implementation from Nyland et al. (2007) against the number of N-body particles. The figure compares the presented implementation and Nyland et al. (2007) for the fastest T_{sub}. The fastest T_{sub} is T_{sub} which can achieve the best performance for given N (e.g. 16 for $N = 256$, 1 for $N =$ 1,048,576). The performance increase reach to 541 % in maximum at N is 256, 6.7 % in minimum at N is 2,048. The figure shows that the presented implementation is always faster than Nyland et al. (2007), and the performance for low N region drastically increase.

10.3 Performance of CUDA/OpenMP/MPI Code

For measurement accuracy, we repeatedly measured the kernel function's total execution time. Since we use two CUDA streams to overlap the calculations and communications, as discussed in Chapter 9, the streams must be synchronized at the end of the kernel function's sequential execution. This is because without the additional synchronization, a kernel function running the previous step could execute concurrently with a kernel function (related to another CUDA stream) running the current step, which would lead to an overestimation of the performance. Therefore, we must add either the `cudaDeviceSynchronize()` function or the `cudaStreamSynchronize()` function to the appropriate CUDA stream.

The execution times for the synchronizing instructions themselves do not allow precise measurements, especially in the low-N region. Thus, we must evaluate their execution times independently and subtract these from the measured execution times. We measured the execution time of the synchronization instruction t_{sync} by measuring the execution time of N_{iter} synchronization instructions (`cudaStreamSynchronize()` for a

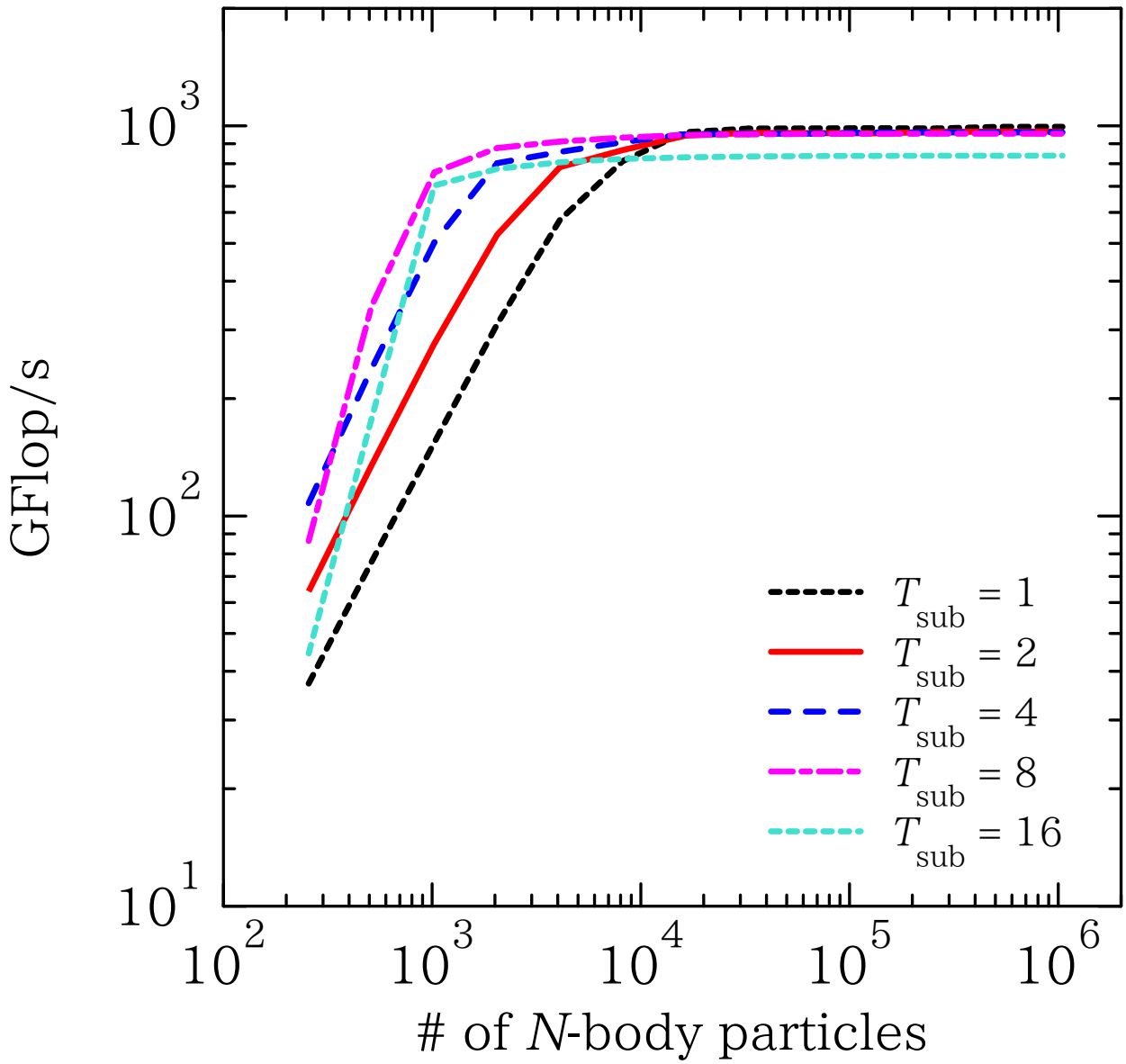


Fig. 10.2: Measured performance of the CUDA code against N focused in achieving high performance in the low N-region.

single GPU, `cudaStreamSynchronize()` plus OpenMP barrier for two GPUs, `cudaStreamSynchronize()` plus OpenMP barrier with `MPI_Barrier()` for the CUDA/OpenMP/MPI implementation). Figure 10.6 shows the resulting measurements: the horizontal axis represents the number of iterations, and the vertical axis represents the total execution time for the repeated execution of synchronize instructions. The black filled squares, red open circles, and other symbols show the measured $t_{\rm sync}$ for 1, 2, and 4 to 256 GPUs, respectively. The plotted lines in Fig. 10.6 are the fitted results based on the least square method. Table 10.2 lists the corresponding values of $t_{\rm sync}/N_{\rm iter}$.

We also measured the launch times for a dummy kernel function on the basis of the same strategy. The results in Figure 10.7 show no significant difference between asynchronous and synchronous launches. The fitted launch times are 2.69×10^{-6} and 2.72×10^{-6} seconds for asynchronous and synchronous launches, respectively.

The maximum number of GPUs used in the performance measurements is 256. The number of N-body particles per GPU ranges from 512 to 1,048,576, which corresponds to a maximum of $256 \times 1048576 =$

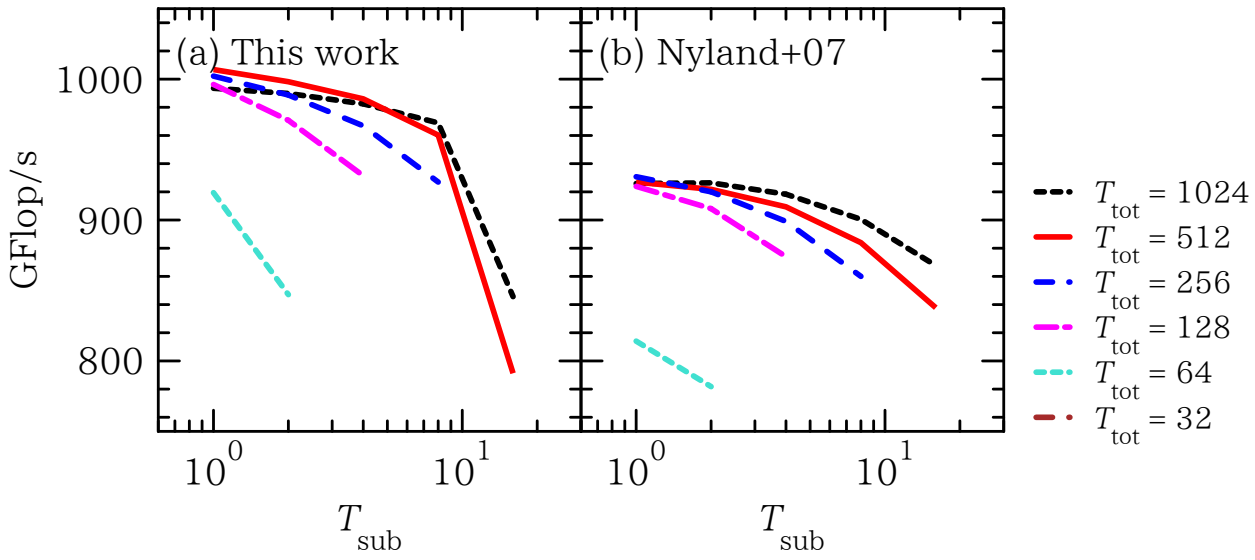


Fig. 10.3: Performance comparison with Nyland et al. (2007) at $N = 1{,}048{,}576$.

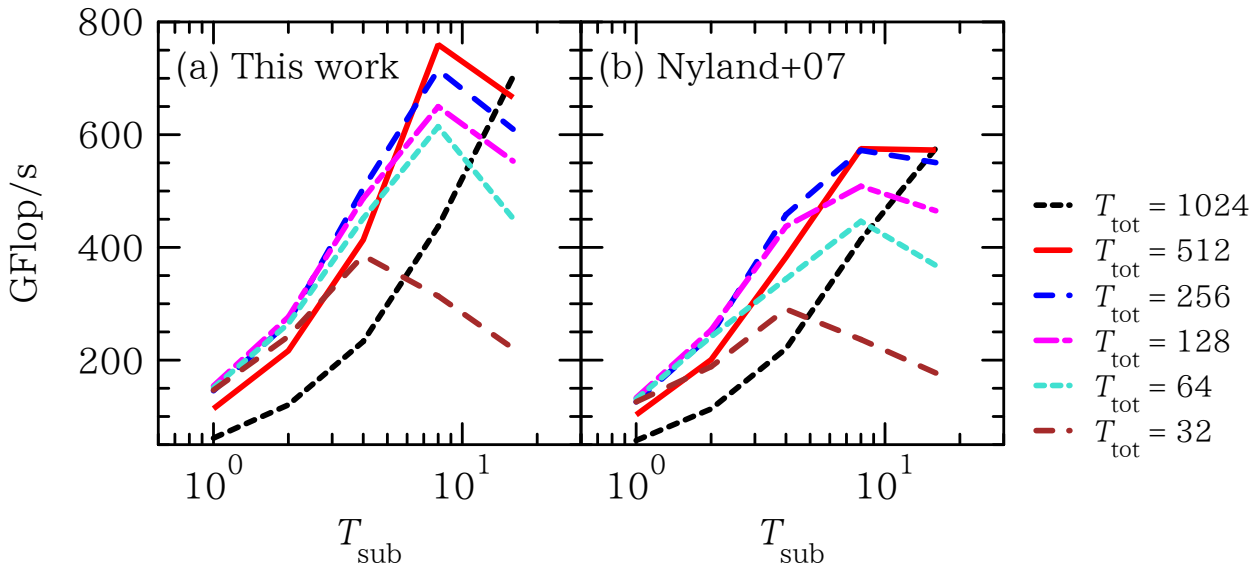


Fig. 10.4: Performance comparison with Nyland et al. (2007) at $N = 1{,}024$.

Table. 10.2: Synchronization cost

Number of GPU(s)	$t_{\rm sync}/N_{\rm iter}$ (sec.)
1	5.93×10^{-7}
2	1.19×10^{-6}
4	7.11×10^{-5}
8	7.84×10^{-5}
16	8.26×10^{-5}
32	8.68×10^{-5}
64	8.88×10^{-5}
128	9.37×10^{-5}
256	9.86×10^{-5}

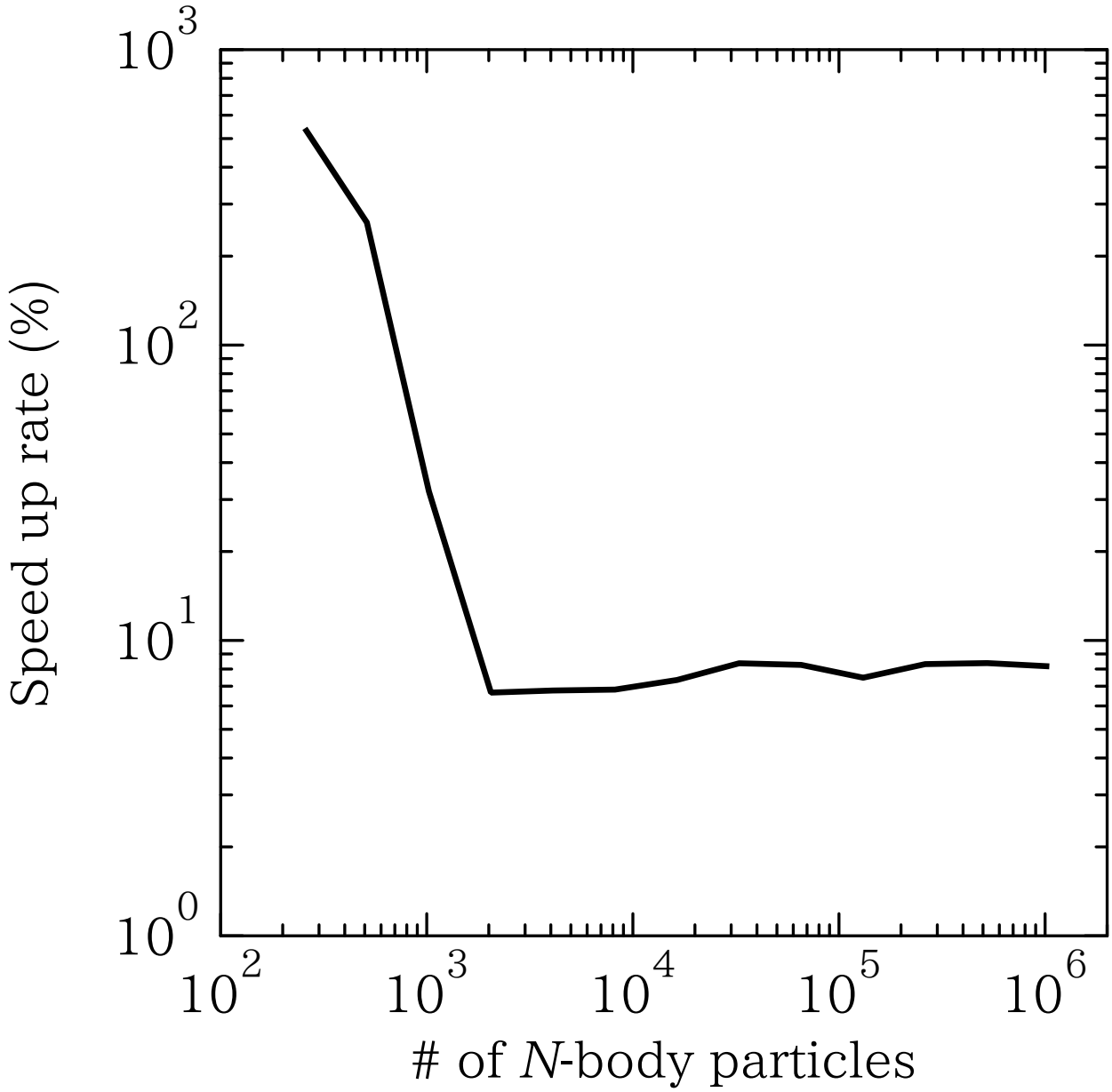


Fig. 10.5: Speed up rate from Nyland et al. (2007).

$2^{28} = 268435456$ particles. Figure 10.8 shows the performance measurements when the $T_{\rm sub}$ is unity. The horizontal axis represents the total number of N-body particles, and the vertical axis represents the measured performance in single precision, assuming 26 floating-point operations for one interaction, which is the most plausible estimate for GPUs with compute capability 2.0 (see discussion in Section 10.2). The lines from the bottom to the top of Fig. 10.8 represent the measured performance in single precision for 1, 2, $\cdots$, 256 GPUs. The peak performance of 255.5 TFlop/s in single precision is reached when $N = 268{,}435{,}456$ and 256 GPUs are used.

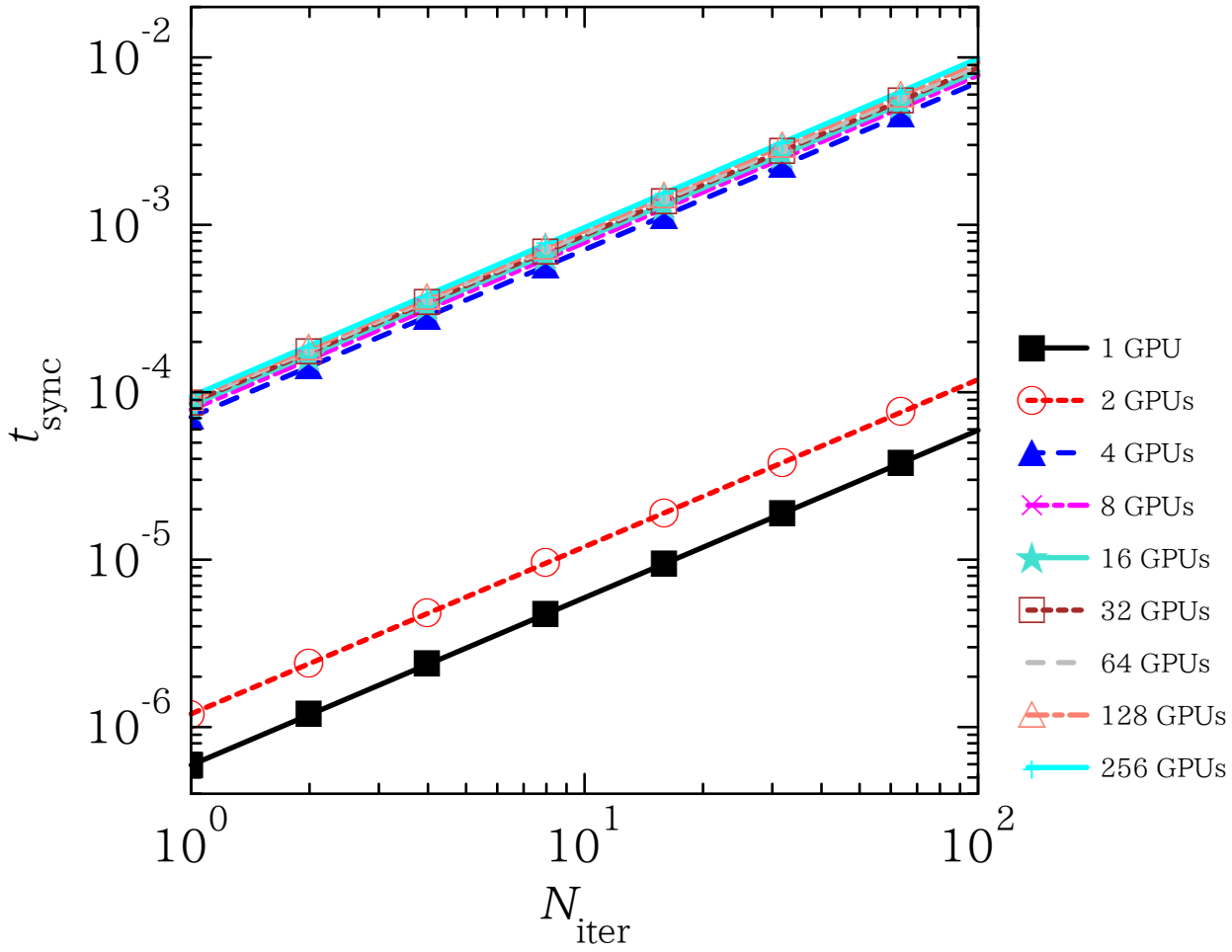


Fig. 10.6: Execution time of the synchronization instruction t_{sync} as a function of the number N_{iter} of synchronization instructions executed. Each symbol shows the measured t_{sync}: black filled squares for a single GPU (CUDA), red open circles for two GPUs (CUDA with OpenMP), and other symbols for 4, 8, $\cdots$, 256 GPU boards (CUDA with OpenMP/MPI). The plotted lines show the fitted results based on the least-square method for each number of GPUs.

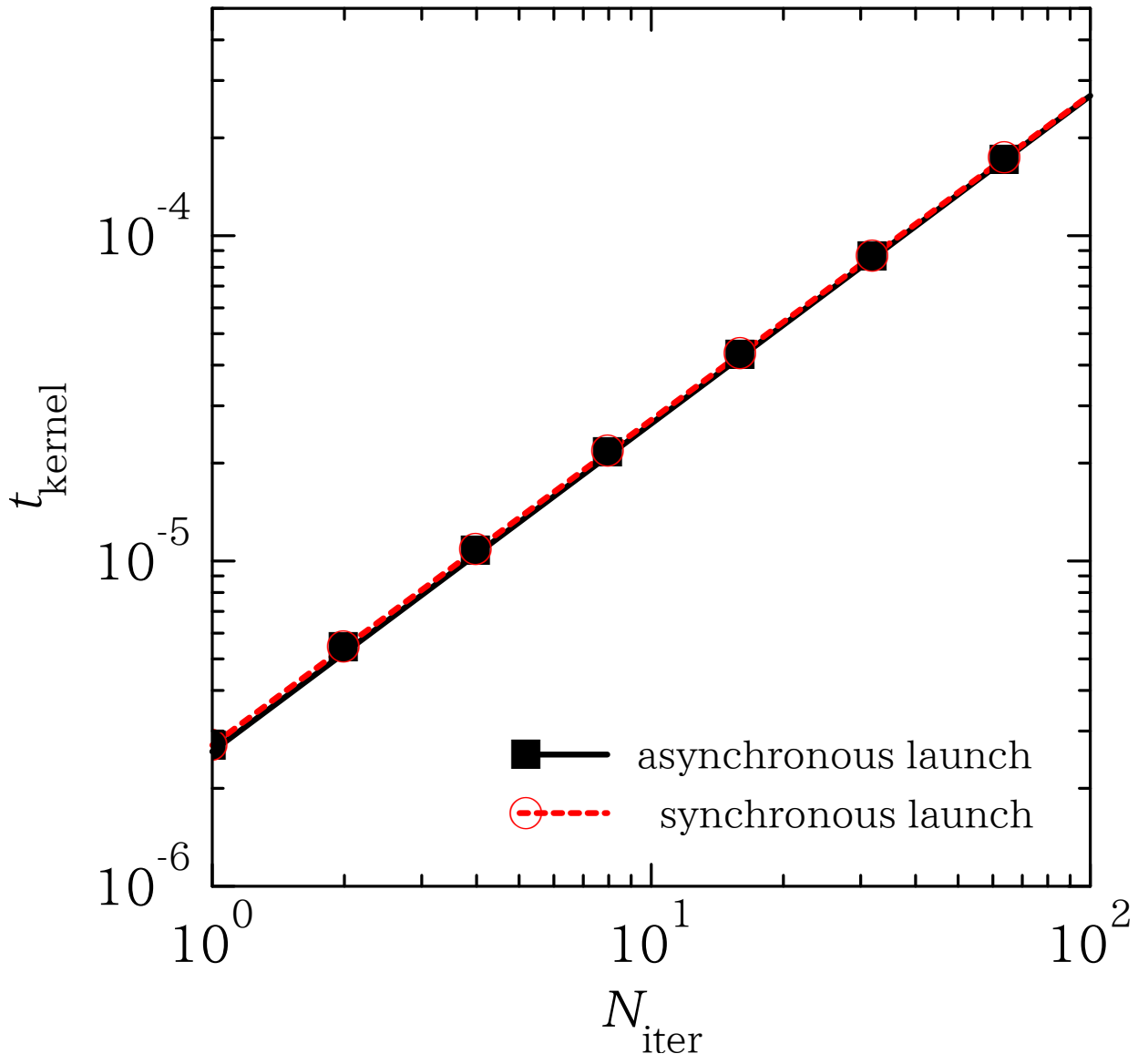


Fig. 10.7: Launch time for a kernel function, t_{kernel}, as a function of the number N_{iter} of launches of a dummy kernel function. Black filled squares and red open circles show t_{kernel} for asynchronous and synchronous launches, respectively. The plotted lines show the fitted results based on the least-square method.

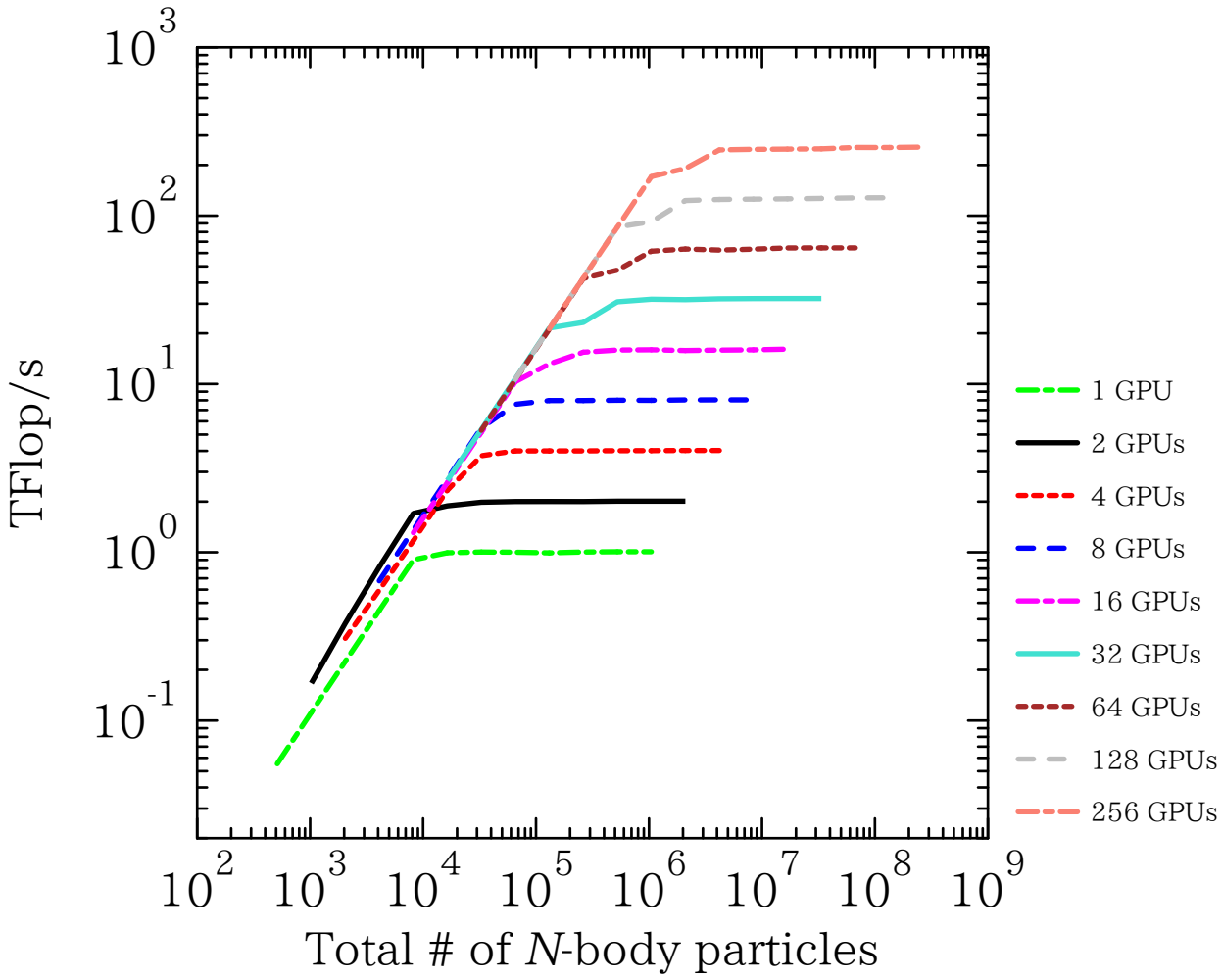


Fig. 10.8: Measured performance in single precision as a function of the total number of N-body particles, under the assumption that one interaction corresponds to 26 floating-point operations. The dot-dashed green line shows the performance for execution by a single GPU, and the solid black line shows the performance for execution by two GPUs employing OpenMP. The remaining lines show the performance for hybrid parallelization based on OpenMP and MPI; the lines from top to bottom show the performance of 256, 128, $\cdots$, 4 GPUs.

Chapter 11 Performance Analysis

11.1 Performance Modeling of CUDA Code

We evaluate the measured performance for the single GPU calculation in this section. Because our implementation overlaps multiple instructions as much as possible to improve performance, analytic modeling of the performance is quite difficult. Thus, we have constructed a model equation with several unknown parameters and have determined that these parameters fit the results of the performance measurements. In what follows, we estimate the total number of clock cycles needed to complete calculation of the gravitational interactions, $C_{\rm all}$.

There is a certain cost $C_{\rm kernel}$ for launching the kernel function to start the computation. In addition, each thread must load position data for i-particles from the global memory before calculating the interactions with j-particles. At this stage, a high latency L (one of the unknown parameters) of approximately $400-800$ clock cycles occurs, according to the NVIDIA CUDA C Programming Guide (Nvidia 2012). The data transfer time from the global memory is negligibly smaller than L owing to the GPU's wide memory bandwidth of 177.6 GB/s.

Once each thread has stored the position data for i-particles in the registers, the gravitational force calculation begins. To achieve a high level of performance, we have divided the force calculation loop into two steps: the first step copies position data for $T_{\rm tot}/T_{\rm sub}$ j-particles from the global memory to the shared memory, and in the second step, threads calculate gravitational interactions among $T_{\rm tot}/T_{\rm sub}$ i-particles and $T_{\rm tot}$ j-particles, where $T_{\rm tot}$ is the number of threads per block. The calculation loop is performed $N_j/T_{\rm tot}$ times to calculate gravitational interactions from with all j-particles. Thus, the number of clock cycles needed to execute the gravitational force calculation in each block is

$$C_{\rm int} = \frac{N_j}{T_{\rm tot}} \times \left(L + \frac{T_{\rm tot}}{T_{\rm sub}} C_{\rm calc} \right) \times \frac{T_{\rm tot}}{N_{\rm core}} = \frac{N_j}{N_{\rm core}} \left(L + \frac{T_{\rm tot}}{T_{\rm sub}} C_{\rm calc} \right). \tag{11.1}$$

The most important unknown parameter in (11.1) is the number of clock cycles for calculating a single interaction, $C_{\rm calc}$. $N_{\rm core}$ is the number of CUDA cores per SM, which is 32 for GPUs with compute capability 2.0. The term $T_{\rm tot}/N_{\rm core}$ represents the warp schedulers automatically dividing the computations into $T_{\rm tot}/N_{\rm core}$ groups owing to the limited number of CUDA cores per SM. Furthermore, if one SM contains multiple blocks, then the latency L is sometimes hidden by overlapping global memory data transfers and calculations of interactions among i-particles and j-particles. In such cases, $C_{\rm int}$ becomes

$$C_{\rm hide} = \frac{N_j}{N_{\rm core}} \max \left(L, \frac{T_{\rm tot}}{T_{\rm sub}} C_{\rm calc} \right). \tag{11.2}$$

At the end of the computations, $T_{\rm sub}$ threads accumulate acceleration data of i-particle to a thread if needed, and the resultant data must be transferred to the global memory in L clock cycles. Clock cycles to complete the accumulation process $C_{\rm acc}$ is a function of $T_{\rm sub}$. To summarize this analysis, the number of clock cycles to complete computation of a block, $C_{\rm block}$, is $C_{\rm int} + C_{\rm kernel} + 2L + C_{\rm acc}(T_{\rm sub})$ or $C_{\rm hide} + (C_{\rm kernel} + 2L + C_{\rm acc}(T_{\rm sub}))/B_{\rm SM}$ when $B_{\rm SM}$ blocks are simultaneously assigned to an SM, is either one or greater than two. In the latter case, the factor $1/B_{\rm SM}$ represents the effect of overlapped memory transfer time due to the multiple blocks.

To evaluate $C_{\rm all}$ using $C_{\rm block}$, we need to determine the number of times the blocks must repeat the computation loop. The total number of blocks, $B_{\rm tot}$, is expressed as $N_i/(T_{\rm tot}/T_{\rm sub})$; therefore, the number of loop iterations, N_{tot}, is represented in terms of the number of SMs within a GPU, $N_{\rm SM}$, as follows:

$$N_{\rm tot} = {\rm ceil}\left(\frac{B_{\rm tot}}{N_{\rm SM}}\right) = {\rm ceil}\left(\frac{T_{\rm sub}N_i}{T_{\rm tot}N_{\rm SM}}\right). \tag{11.3}$$

If $B_{\rm SM} \geq 2$, then gravitational interaction calculations and transfers from the global memory are overlapped in some of the $N_{\rm tot}$ loop iterations. There are $N_{\rm hide} = {\rm floor}(N_{\rm tot}/B_{\rm SM})$ such iterations, and the number of remaining loops is $N_{\rm rem} = N_{\rm tot} - B_{\rm SM}N_{\rm hid}$. As the result, $C_{\rm all}$ is expressed as

$$\begin{aligned} C_{\rm all} = {} & N_{\rm hide}\left[\frac{B_{\rm SM}N_j}{N_{\rm core}}\max\left(L, \frac{T_{\rm tot}}{T_{\rm sub}}C_{\rm calc}\right) + C_{\rm kernel} + 2L + C_{\rm acc}(T_{\rm sub})\right] \\ & + N_{\rm rem}\left[\frac{N_j}{N_{\rm core}}\left(L + \frac{T_{\rm tot}}{T_{\rm sub}}C_{\rm calc}\right) + C_{\rm kernel} + 2L + C_{\rm acc}(T_{\rm sub})\right]. \end{aligned} \tag{11.4}$$

We estimate the unknown parameters $C_{\rm calc}$, L, and $C_{\rm kernel}$. We already have the estimates for L and $C_{\rm kernel}$. The latency L is about $400 - 800$ clock cycles, and the launch time for a kernel function $C_{\rm kernel}$ is the time measured for a dummy kernel launch, calculated as in Section 10.3. The resultant time of 2.7 micro seconds corresponds to 3500 clock cycles. Therefore, we focus on the remaining unknown parameter $C_{\rm calc}$.

First, we evaluate the maximum value of $B_{\rm SM}$ for the implementation. $B_{\rm SM}$ is determined by the number of available registers and the capacity of shared memory per SM, which are 32,768 and 16 KB, respectively, for an NVIDIA Tesla M2090 board with the "L1 cache preferred" option. All threads use 23 registers when $T_{\rm sub}$ is unity; therefore, $B_{\rm SM} \leq 32768/(23 \times 512) \cong 2.8$ when $T_{\rm tot}$ is 512. Each block uses 8 KB to store the position data for 512 particles, since 16 bytes (four single-precision floating-point numbers) are needed to store each position. Therefore, the maximum value of $B_{\rm SM}$ is two for the implementation owing to the limitations of shared memory capacity and the number of registers per SM.

We estimate $C_{\rm calc}$ based on (11.4). Since all quantities related to $N_{\rm tot}$ or $N_{\rm hide}$ are powers of two, $N_{\rm rem}$ becomes zero and the term related to $N_{\rm rem}$ vanishes. Therefore, overlapped gravitational interaction calculations and transfers from the global memory always occur when N_i is a power of two and is greater than $N_{\rm SM}T_{\rm tot}/T_{\rm sub} = 8192/T_{\rm sub}$. Furthermore, if $N_j \gg N_{\rm core}/B_{\rm SM}$, then the execution time for the inside interaction loop becomes much greater than that for the outside loop, and thus the contribution of the term $C_{\rm kernel} + 2L + C_{\rm acc}(T_{\rm sub})$ becomes negligibly small. The measured execution time for the case of $N_i = N_j =$ 1,048,576 and $T_{\rm sub} = 1$ is 28.4 seconds, corresponding to 36.9 billion clock cycles. Thus, according to (11.4), $C_{\rm calc}$ can be estimated to be $C_{\rm all}/(N_iN_j/512) \cong 17.2$ clock cycles.

We must explain why we consider the execution time of 28.4 seconds to be due to $C_{\rm calc}$ rather than L. The arithmetic intensity $T_{\rm tot}C_{\rm calc}/(T_{\rm sub}L)$ is a useful parameter for determining whether the dominant term is $C_{\rm calc}T_{\rm tot}/T_{\rm sub}$ or L. As we have noted, the latency L corresponds to $400-800$ clock cycles. If L is dominant compared to the term $C_{\rm calc}T_{\rm tot}/T_{\rm sub}$ so that the arithmetic intensity is less than unity, then $C_{\rm calc}$ must be smaller than $LT_{\rm sub}/T_{\rm tot} \lesssim 3$. However, this is not the case because calculation of gravitational interaction involves at least three subtractions, six FMA operations, three multiplications, and calculation of an inverse square root. Therefore, the execution time of 28.4 s must represent the contribution of $C_{\rm calc}T_{\rm tot}/T_{\rm sub}$ if $T_{\rm sub}$ is unity.

Here, we discuss the effect of the reduced number of operations appeared in Section 8.1. Since the reduced number of operations is 1, the optimization contributes $(1 + C_{\rm calc})/C_{\rm calc} \cong 1.06$ times of performance increase. The six percent of the performance improvement is the dominant part of the performance increase of 6.7 % in minimum compared to Nyland et al. (2007).

At the end of this section, we examine trends appeared in Fig. 10.1. As we mentioned in Section 10.2, the presented implementation has the below five trends.

1. The performance increase in low N region looks like proportional to N.
2. The performance increase saturate in the large N region.
3. The critical N which determine the transition point of the performance dependence on N tends to decrease with increasing of $T_{\rm sub}$.
4. The sustained performance decreases gradually with $T_{\rm sub}$ increase.
5. The sustained performance of $T_{\rm sub} = 16$ is much lower than that of any other $T_{\rm sub}$.

At first, the performance increase in low N region is due to the dependence on the term $N_{\rm hid}$ of (11.4). Since $N_{\rm hid}$ is approximately expressed as $T_{\rm sub}N_i/(T_{\rm tot}N_{\rm SM}B_{\rm SM}) = T_{\rm sub}N_i/16384$, the increase of N does not mean increase of $N_{\rm hid}$ for the region of $N_i \leq 16384/T_{\rm sub}$. Therefore, the execution time is proportional to N_j only while the amount of computation is proportional to N_iN_j; consequently, the performance increase is proportional to N. Such performance increase continues as long as $N_{\rm hid}$ is less than a few, which corresponds to the condition not to waste 16 SMs of a GPU. When $N_{\rm hid}$ sufficiently grows up, then the performance reaches the peak performance (the second trend). The condition to achieve the peak performance of GPU is that $N_{\rm hid} \gtrsim 16384/T_{\rm sub}$, therefore, the third trend appears due to the term of $1/T_{\rm sub}$.

The origin of the fourth trend is considered to be a decrease of arithmetic intensity $T_{\rm tot}C_{\rm cal}/(T_{\rm sub}L)$ in the interaction loop to calculate gravitational interaction. The overwrapping of the data transfer from global memory and calculation of gravitational interaction becomes easier for the higher value of the arithmetic intensity (i.e. lower $T_{\rm sub}$). Since increasing degree of the overwrapping would contribute for the increase of performance, that would be the origin of the fourth trend. If the arithmetic intensity becomes lower than unity, then the latency due to accessing global memory suppress the performance. According to Nvidia (2012), L is approximately $400 - 800$ clock cycles, and hence the arithmetic intensity being $11 \lesssim T_{\rm tot}C_{\rm cal}/(T_{\rm sub}L) \lesssim 22/T_{\rm sub}$. This value suggests that the access latency to the global memory L would limit the performance when $T_{\rm sub}$ is greater than 11. This is the reason why the sustained performance of $T_{\rm sub} = 16$ is much lower than any other $T_{\rm sub}$.

11.2 Performance Modeling of CUDA/OpenMP/MPI Code

In this section, we analyze the performance when multiple GPUs are used and $T_{\rm sub}$ is unity. We denote the performance for N particles using $N_{\rm GPU}$ GPUs as $P(N; N_{\rm GPU})$. Figure 11.1 plots the parallel efficiency defined as $P(N; N_{\rm GPU})/(P(N; 1) \times N_{\rm GPU})$. Because we have measured the performance of a single GPU only for the region $512 \leq N \leq 1048576$, we assume that $P(N \geq 1048576; 1)$ equals the sustained value $P(1048576; 1)$. The horizontal axis represents the number of N-body particles, and the vertical axis represents the parallel efficiency $P(N; N_{\rm GPU})/(P(N; 1) \times N_{\rm GPU})$. The figure shows the parallel efficiencies for 2, 4, $\cdots$, 256 GPUs from left to right.

Figure 11.1 clearly shows that the parallel efficiency of two GPUs based on OpenMP is greater than the parallel efficiency of the OpenMP/MPI hybrid cases in the low-N region. This is natural because the time needed to synchronize and/or transfer data for OpenMP is significantly shorter than that for OpenMP/MPI (e.g., Fig. 10.6). The saturation point for the increase in parallel efficiency roughly corresponds to the converging point of the SendSync and SyncRecv modes in Fig. 9.4. This suggests that the convergence of the measured performance is the result of the communication time completely being hidden by overlapping execution time.

On the other hand, the behavior of the OpenMP/MPI hybrid parallelized code shown in Fig. 11.1 is not the same as that of the CUDA/OpenMP code. To understand this, we have plotted the parallel efficiency with weak scaling, $P(N; N_{\rm GPU})/(P(N/N_{\rm GPU}; 1) \times N_{\rm GPU})$, in Figure 11.2. The horizontal axis represents the

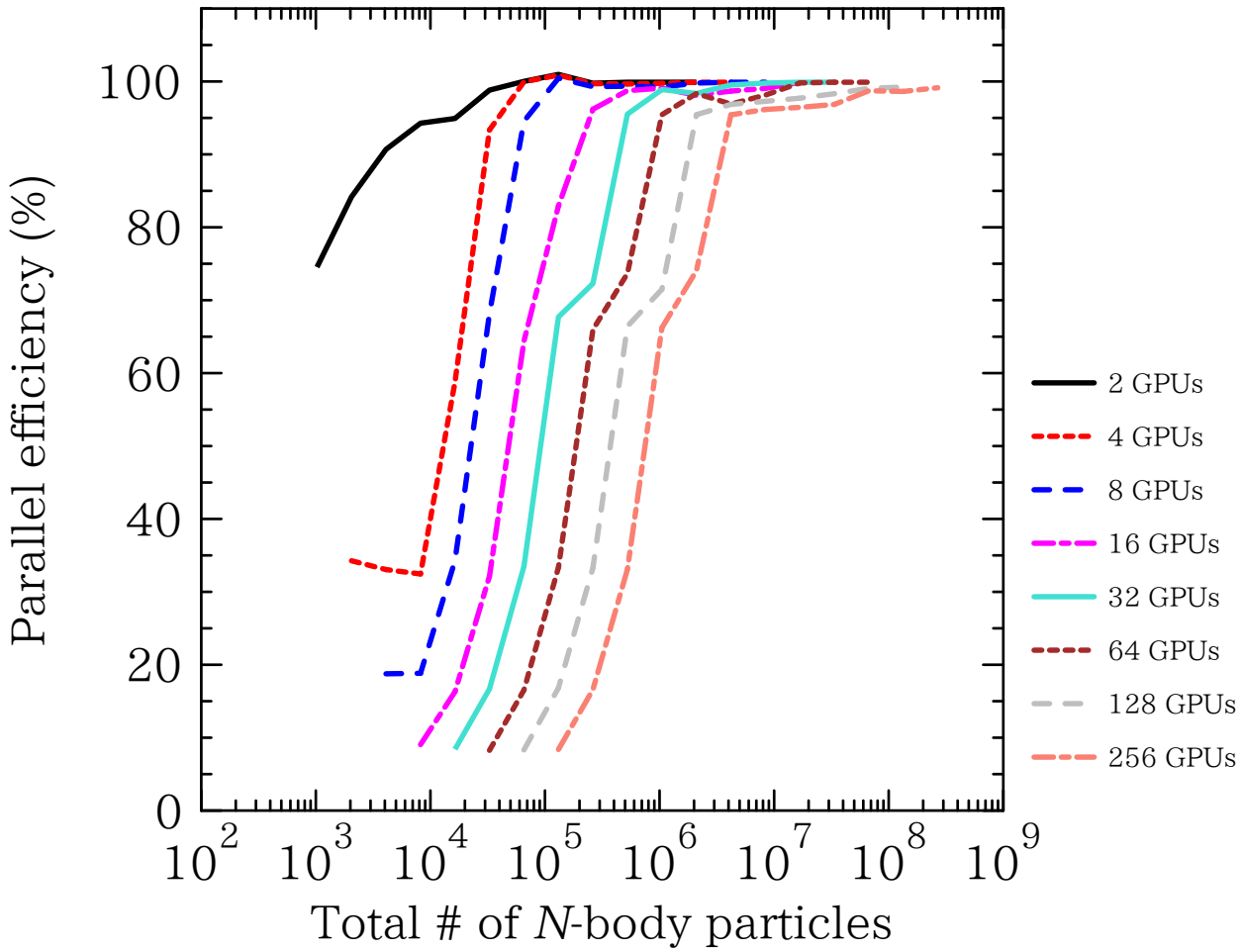


Fig. 11.1: Parallel efficiency $P(N;N_{\rm GPU})/(P(N;1)\times N_{\rm GPU})$ plotted as a function of the total number of N-body particles shows strong scaling. The solid black line represent the efficiency of using two GPUs based on OpenMP, and the other plots show the parallel efficiency for the hybrid parallelized cases using OpenMP/MPI: from left to right, these represent 4, 8, $\cdots$, 256 GPUs.

number of particles per GPU, and the vertical axis represents the parallel efficiency. The figure shows the parallel efficiency for 2, 4, $\cdots$, 256 GPUs from left to right, the same as in Fig. 11.1. Since the performance of the kernel function in the measurement of $P(N;N_{\rm GPU})$ is the same as that in the measurement of $P(N/N_{\rm GPU};1)$, Fig. 11.2 is more suitable than Fig. 11.1 for discussing the performance trends.

The parallel efficiency converges to unity when the number of particles per GPU is greater than 10^4. This is not surprising because we have overlapped communications among the multiple GPUs and calculations of gravitational interaction to hide the communication time. Owing to the overlap, communication and synchronization among the multiple processes do not impact the measured performance per GPU, so that the performance of the single GPU performance is retained. This is also supported by the similarity of the measured performance curves in Fig. 10.8. The behavior of the measured performance per GPU for the OpenMP/MPI hybrid parallelized CUDA code is roughly similar to that for the single GPU code.

The region where the number of N-body particles per GPU is less than 10^4 is quite interesting. The measured parallel efficiency reaches 1.5 when more than four GPUs are used. We consider this superlinear scaling to be a by-product of the overlapped communications and calculations. To overlap calculations within a GPU and communications between the CPU and GPU, the instructions should belong to different CUDA streams. Therefore, we utilize two CUDA streams.

The advantage of using multiple CUDA streams is that this opens the possibility of overlapping when

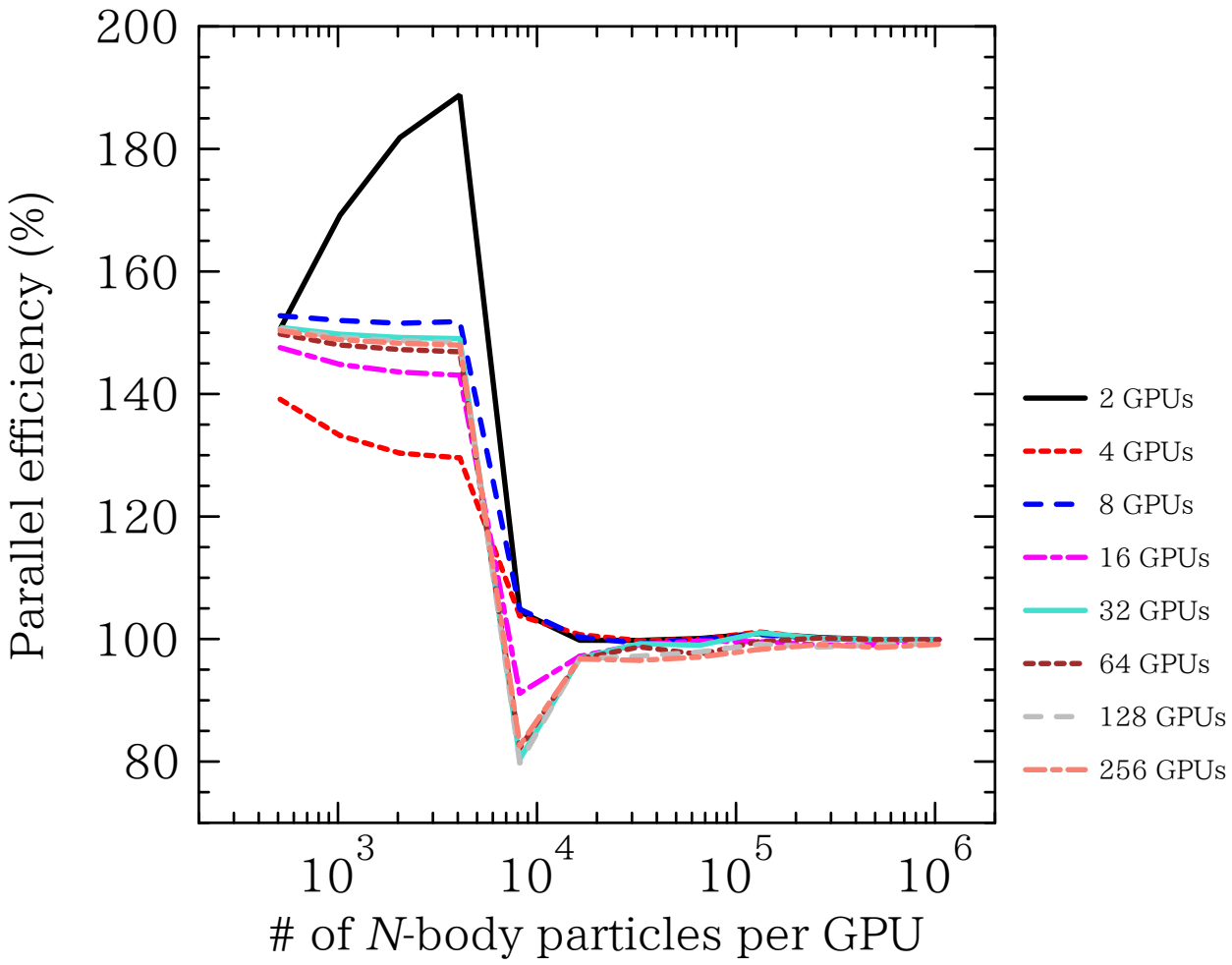


Fig. 11.2: Parallel efficiency $P(N; N_{\mathrm{GPU}})/(P(N/N_{\mathrm{GPU}}; 1) \times N_{\mathrm{GPU}})$ plotted as a function of the number of particles per GPU shows weak scaling. The lines are same as those for the strong scaling in Fig. 11.1.

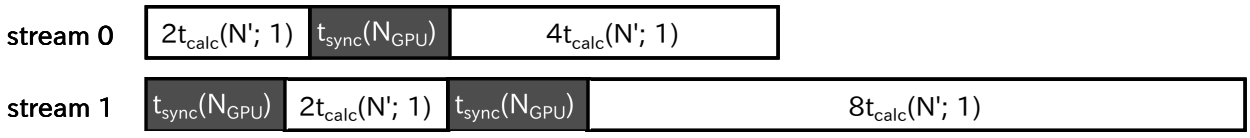


Fig. 11.3: Schematic view of the overlapped execution model for different CUDA streams. The horizontal axis represents time, and each block shows the instructions executed: white blocks represent calculations and dark blocks represent synchronization.

the number of N-body particles per GPU is greater than 10^4. However, there is an additional benefit when the number of N-body particles per GPU is less than 10^4. As discussed in the previous section, some of the SMs do not calculate gravitational interaction when this number is less than 16,384. This means that the remaining SMs simultaneously perform the kernel function related to the second CUDA stream. This study uses two CUDA streams; therefore, the maximum expected speedup ratio is two. This is the reason for the superlinear scaling that appears when the number of N-body particles per GPU is less than 10^4. The measured parallel efficiency fails to reach two owing to the existence of additional instructions such as communications between the host and device and synchronization of related devices.

To assess the speedup ratio from the execution of two CUDA streams, we have constructed a simple model of kernel execution in the accumulation phase. The accumulation phase is the dominant phase of the MPI

Table. 11.1: Execution times for the kernel function

N	time (sec.)
512	1.23×10^{-4}
1,024	2.44×10^{-4}
2,048	4.85×10^{-4}
4,096	9.68×10^{-4}
8,192	1.93×10^{-3}
16,384	7.03×10^{-3}

communication layer when the number of N-body particles per GPU is small. Figure 11.3 shows an outline of the model. The executed instructions for each CUDA stream are shown as a function of time. The white boxes labeled $t_{\rm calc}(N';1)$ denote the kernel execution time for $N' \equiv N/N_{\rm GPU}$ particles on a single GPU, and the dark boxes labeled $t_{\rm sync}(N_{\rm GPU})$ represent the time needed to synchronize $N_{\rm GPU}$ GPU boards.

The necessary time for calculation depends on N_i and N_j, as discussed in Section 11.1. Because the number of j-particles doubles in each communication, the kernel execution time also doubles after each communication. The single exception is the initial step for stream 0. This difference represents the parallelization using OpenMP introduced in Section 9.1. Table 11.1 lists the measured execution times of the kernel function for a single GPU. The time for synchronization has already been estimated, as shown in Tab. 10.2. We ignore the time needed to transfer data between the host and a device or among multiple MPI processes because it is small compared with kernel execution time and synchronization time.

The total execution time for the model shown in Fig. 11.3 can be expressed as

$$t_{\rm calc}(N;1)\left(1+\sum_{i=0}^{\log_4(N_{\rm GPU}/2)}4^i\right)+t_{\rm sync}(N_{\rm GPU})\log_4(N_{\rm GPU}) \tag{11.5}$$

for cases where $N_{\rm GPU}$ is not a power of four corresponding to stream 0 in the figure, and as

$$2t_{\rm calc}(N;1)\sum_{i=0}^{\log_4(N_{\rm GPU}/2)}4^i+t_{\rm sync}(N_{\rm GPU})\log_4(N_{\rm GPU}) \tag{11.6}$$

for cases where $N_{\rm GPU}$ is a power of four corresponding to stream 1. Therefore, the theoretical parallel efficiency is

$$\left[\left(1+\sum_{i=0}^{\log_4(N_{\rm GPU}/2)}4^i\right)/N_{\rm GPU}+\frac{t_{\rm sync}(N_{\rm GPU})\log_4(N_{\rm GPU})}{t_{\rm calc}(N/N_{\rm GPU};1)N_{\rm GPU}}\right]^{-1} \tag{11.7}$$

and

$$\left[\frac{2}{N_{\rm GPU}}\sum_{i=0}^{\log_4(N_{\rm GPU}/2)}4^i+\frac{t_{\rm sync}(N_{\rm GPU})\log_4(N_{\rm GPU})}{t_{\rm calc}(N/N_{\rm GPU};1)N_{\rm GPU}}\right]^{-1} \tag{11.8}$$

for streams 0 and 1, respectively.

Because Equations 11.7 and 11.8 are too complicated for discussing the fundamental properties of the parallel efficiency, we examine two complementary limits. First, we assume $t_{\rm sync}(N_{\rm GPU})$ is zero to evaluate the parallel efficiency's upper limit. In this case, the estimated theoretical parallel efficiency is 2.0, 1.33, 1.6, 1.45, 1.52, 1.49, and 1.51 when 4, 8, 16, 32, 64, 128, and 256 GPUs are used, respectively. Next, we assume that $t_{\rm sync}(N_{\rm GPU})$ is twice $t_{\rm calc}(N;1)$ for $N \leq 16384$ and $N_{\rm GPU} \geq 4$ in order to evaluate the lower limit. In this case, the theoretical parallel efficiency is 1.0, 1.0, 1.14, 1.23, 1.33, 1.39, and 1.44 for 4, 8, 16, 32, and 64, 128, and 256 GPUs are used, respectively. Although these estimates do not explain all the data in detail, they provide a rough explanation of the trends in Fig. 11.2. Furthermore, these theoretical estimates suggest that the parallel efficiency approaches 1.5 as the number of GPUs increases.

Chapter 12 Conclusion

We have developed a highly optimized code for collisionless N-body calculations based on direct summation.

Our new optimization, aimed at hiding the latency to access the global memory, enables a peak performance in excess of 1 TFlop/s per single NVIDIA Tesla M2090 board. The performance of the CUDA code peaks at 1006.7 GFlop/s in single precision when $N = 1{,}048{,}576$ (assuming 26 floating-point operations per interaction), which is 75.7% of the theoretical peak performance.

To improve the scalability of the OpenMP/MPI hybrid parallelized CUDA code, we have reduced the number of communications among the multiple GPUs and have overlapped communications with computations to hide the communication time. The performance of the code peaks at 255.5 TFlop/s in single precision when $N = 268{,}435{,}456$ and 256 NVIDIA Tesla M2090 boards are used, which is 75.0% of the theoretical peak performance and 99.1% of the parallel efficiency.

The code has excellent scalability with a superlinear scaling of 1.5 when the number of N-body particles per GPU is less than 16,384, and the parallel efficiency approaches unity when the number of N-body particles per GPU is greater than 16,384. Finally, we have presented a performance model that explains these trends well.

Part III

Hermitage of Wandering Black Hole in the M31 Halo

Abstract

In the hierarchical structure formation scenario, galaxies enlarge through multiple merging events with less massive galaxies. In addition, the Magorrian relation indicates that almost all galaxies are occupied by a central supermassive black hole (SMBH) of mass 10^{-3} of its spheroidal component. Consequently, SMBHs are expected to wander in the halos of their host galaxies following a galaxy collision, although evidence of this activity is currently lacking.

We have investigated a current plausible location of an SMBH wandering in the halo of the Andromeda galaxy (M31). According to theoretical studies of N-body simulations, some of the many substructures in the M31 halo are remnants of a minor merger occurring about 1 Gyr ago. First, to evaluate the possible parameter space of the infalling orbit of the progenitor, we have performed numerous parameter studies using a graphics processing unit (GPU) cluster. To reduce uncertainties in the predicted position of the expected SMBH, we then have calculated the time evolution of the SMBH in the progenitor dwarf galaxy from N-body simulations using the plausible parameter sets.

Our results show that the SMBH lies within the halo ($\sim$20–50 kpc from the M31 center), closer to the Milky Way than the M31 disk. Furthermore, the predicted current positions of the SMBH are restricted to an observational field of $0^\circ.6 \times 0^\circ.7$ in the northeast region of the M31 halo. We also discuss the origin of the infalling orbit of the satellite galaxy and its relationships with the recently discovered vast thin disk plane of satellite galaxies around M31.

Chapter 13 Introduction

Galaxies are known to be host universally a supermassive black hole (SMBH) in their central region (e.g. Kormendy & Richstone 1995). Furthermore, the mass of the spheroidal component of galaxies is correlated with the mass of their central SMBHs (Magorrian et al. 1998; Marconi & Hunt 2003). Figure 13.1 shows the Magorrian relation that indicates galaxies host an SMBH of mass $M_{\rm BH}$ about 10^{-3} times the mass of the spheroidal component of its host galaxy $M_{\rm sph}$. A similar relation between $M_{\rm BH}$ and the velocity dispersion

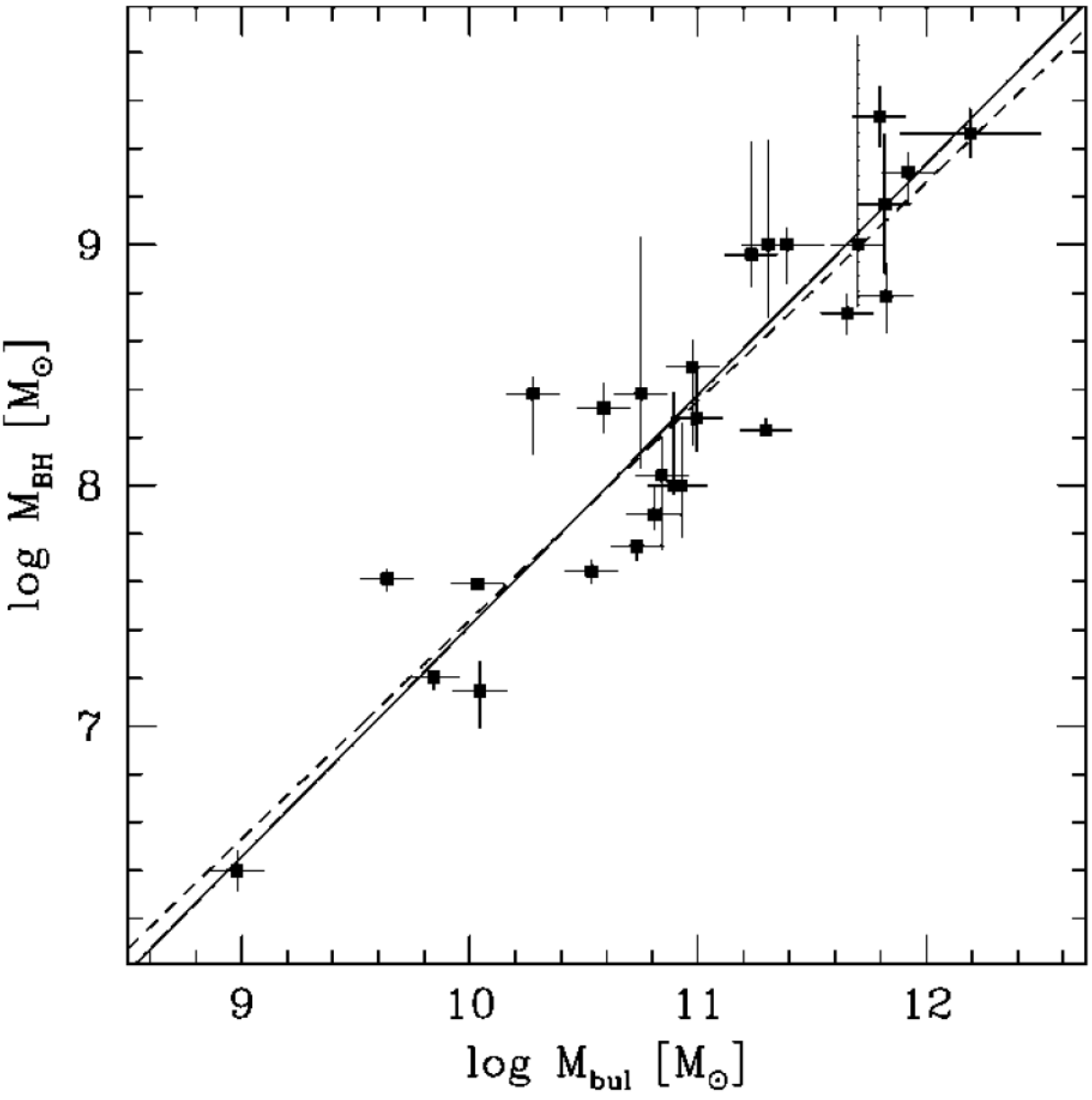


Fig. 13.1: The mass of central SMBHs as a function of the bulge mass of host galaxies, taken from Marconi & Hunt (2003).

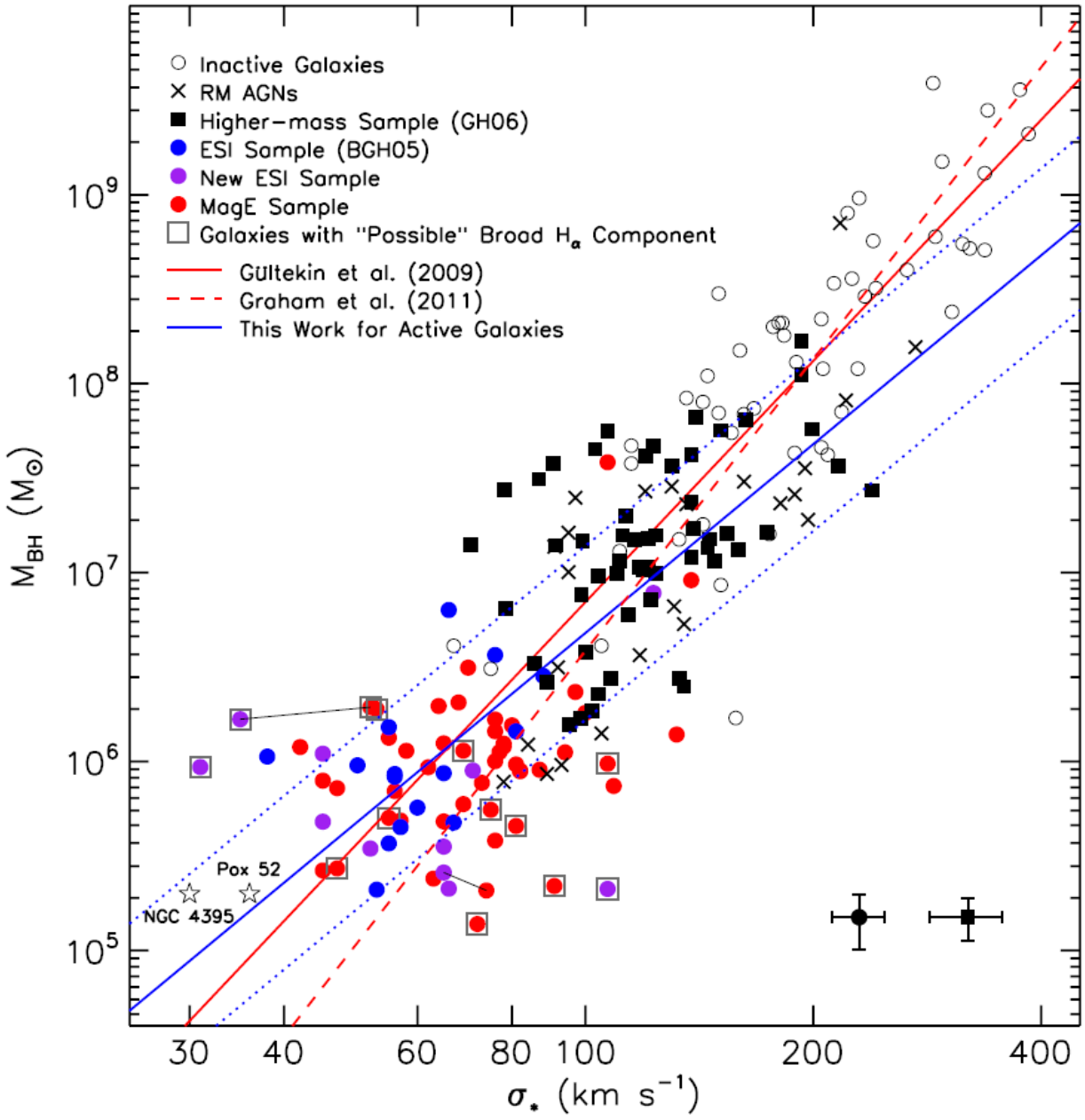


Fig. 13.2: The mass of central SMBHs as a function of the stellar velocity dispersion of host galaxies, taken from Xiao et al. (2011).

σ of the host galaxy, called the $M_{\rm BH} - \sigma$ relation, has been confirmed down to $M_{\rm BH} \sim 10^5 M_\odot$ (Barth et al. 2005; Xiao et al. 2011, see Fig. 13.2). These tight correlations between the mass of SMBHs and the properties of host galaxies ranging over four orders of magnitude in $M_{\rm BH}$ strongly suggests that galaxies coevolve with their central SMBHs. However, the coevolution process of galaxies and SMBHs is largely unknown.

In the hierarchical structure formation scenario, galaxies collide and merge with other galaxies and subsequently with less massive galaxies, causing their central SMBHs to drift within the halo region of their host galaxy. In other words, SMBHs wander in the halo of their host galaxy after galaxy merging events and finally sink toward the central region of the host galaxy under dynamical friction. Therefore, SMBHs can either be centralized in their host galaxy, as in active galactic nuclei, or reside outside of the nucleus (Bellovary et al. 2010; Inoue et al. 2013), although evidence of the latter class of SMBHs is currently lacking. Bellovary

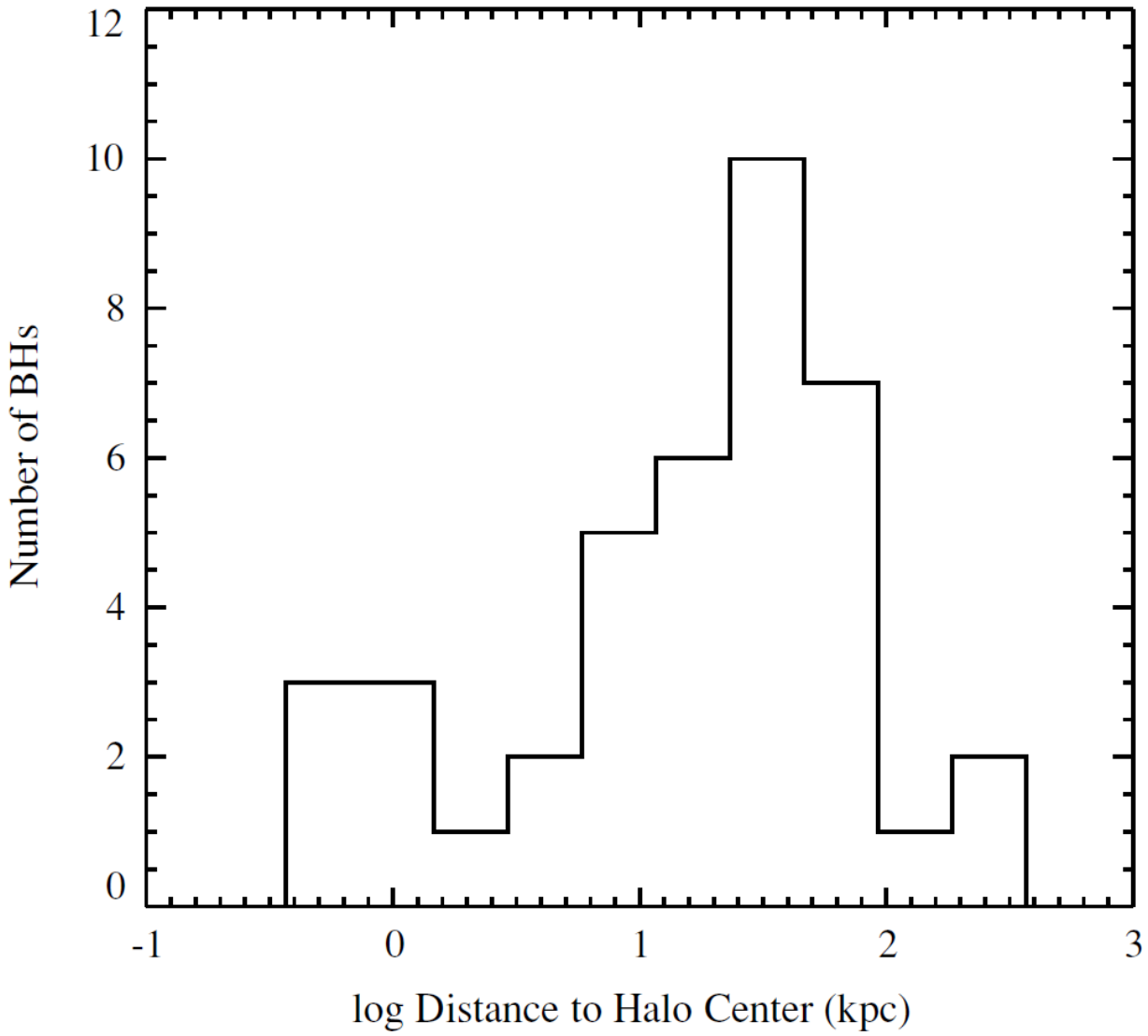


Fig. 13.3: The radial distribution of 40 SMBHs derived by cosmological SPH/N-body simulations, taken from Bellovary et al. (2010).

et al. (2010) investigated the radial distribution of SMBHs using SPH+N-body cosmological simulations of the Milky Way sized halo. Figure 13.3 shows that a few tens of SMBHs locate in the halo region. Inoue et al. (2013) investigated the distribution of intermediate mass black holes (IMBHs; $10^2 M_\odot \lesssim M_{\rm BH} \lesssim 10^6 M_\odot$) in the simulated Milky Way sized halo (Diemand et al. 2008) using a semi-analytic modeling. Figure 13.4 shows that many IMBHs locate in the halo region. The observational search for wandering SMBHs has recently attracted great interest (Farrell et al. 2009; Wiersema et al. 2010). A hyperluminous X-ray source named HLX-1 appeared in Fig. 13.5 is a candidate of an off-center MBH (Godet et al. 2012).

The hierarchical structure formation scenario naturally predicts that many wandering SMBHs exist in the galactic halos. However, such SMBHs have not yet been confirmed. Two simple explanations of this mismatch are as follows: 1. timescale to sink into the galactic center is relatively short, and/or 2. SMBH activity is very low due to low gas density in the galactic halos. If the latter explanation is true, then many undetected SMBHs would move in the galactic halos. To detect such SMBHs, a high sensitivity survey which cover a whole region of the galactic halo could become a powerful tool, but it seems to be a highly unrealistic strategy. Therefore, observations focused on a narrow area where the wandering SMBHs would locate are considered to be a possible plan to detect them. In this study, we theoretically investigate the

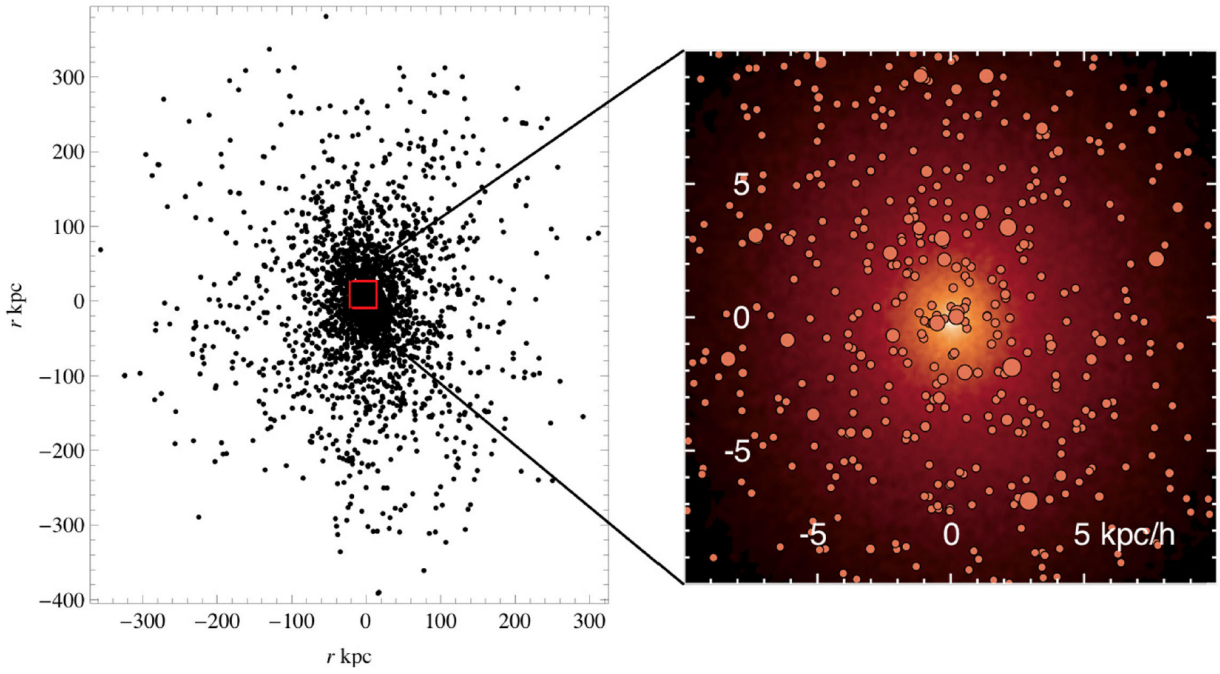


Fig. 13.4: Distribution of IMBHs in the simulated Milky Way sized halo, taken from Inoue et al. (2013).

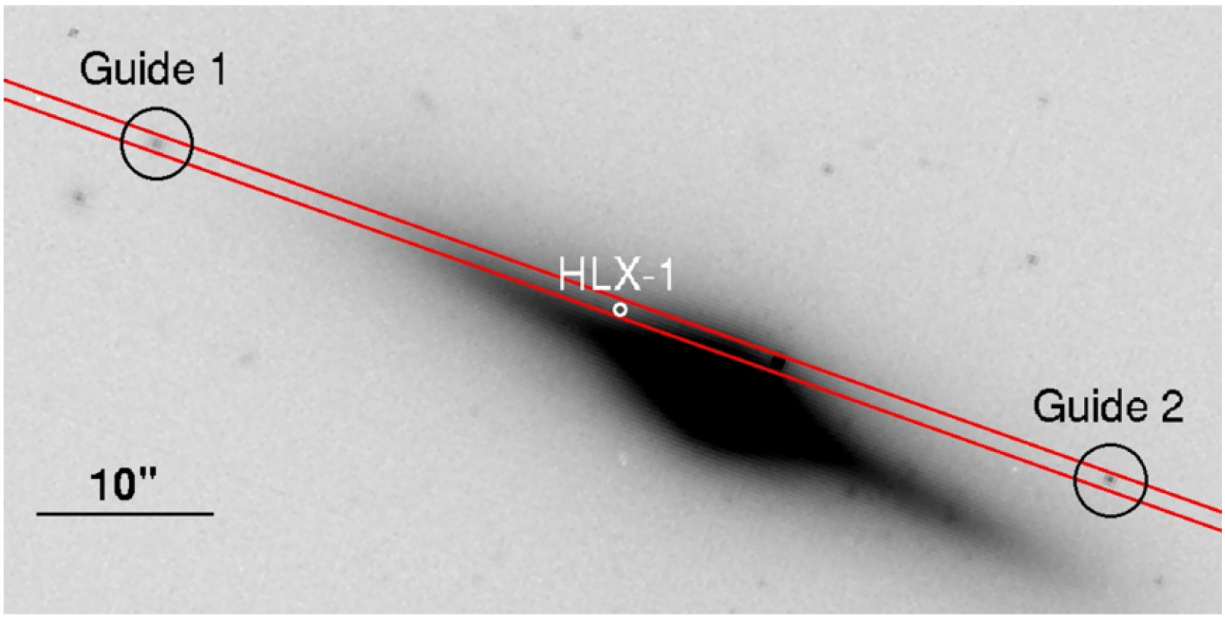


Fig. 13.5: The optical image of a galaxy ESO 243-49, taken from Wiersema et al. (2010). A white circle indicates a hyperluminous X-ray source "HLX-1".

probable positions of such SMBHs. The current SMBH position could be predicted by calculating orbital evolution of SMBHs based on theoretical models that reproduce observed merger remnants well. A suitable test bed for our proposed method is M31. In the M31 halo, observations discovered a giant stellar stream and two stellar shells. Theoretical studies using N-body simulations successfully explained the origin of them and restrain the orbit and the physical properties of the progenitor (see Part I).

The Magorrian relation suggests that the progenitor dwarf galaxy had hosted an SMBH of mass $M_{\rm BH}$ about 10^{-3} times the mass of the spheroidal component of the host (Magorrian et al. 1998; Marconi & Hunt 2003). Since the dynamical mass of the progenitor is estimated to be of order $10^9 M_\odot$ (Fardal et al. 2007,

2013; Mori & Rich 2008, Part I), the progenitor likely harbored an SMBH of mass up to $10^6 M_\odot$, assuming that the progenitor consisted solely of a spheroidal stellar component. The $M_{\rm BH} - \sigma$ relation (Xiao et al. 2011) also suggests the progenitor satellite had possessed an SMBH whose mass is 10^{5-6} $M_\odot$. Thus, an SMBH should be currently wandering among the merger remnants. Finding such an SMBH will help to elucidate how SMBHs coevolve with galaxies. This study uses N-body simulations to predict the current likely positions of the SMBH and thereby guide future observational detections.

Using N-body simulations, we first constrain the infalling orbit of the progenitor satellite galaxy (Chapter 14). In Chapter 15, we investigate the current plausible position of the expected SMBH in the M31 halo and derive an observational field for future observations. The physical relationships between this merger event and the satellite galaxy distribution around M31 are discussed in Chapter 16. The study is summarized in Chapter 17.

Chapter 14 Infalling Orbit of the Satellite

In this chapter, low-resolution N-body simulations are conducted over a wide parameter range to restrict the infalling orbit of the progenitor satellite galaxy. Following this investigation, N-body simulations are conducted at higher resolution in Chapter 15. The numerical modeling and analysis are discussed in Section 14.1 and Section 14.2, and simulation results are presented in Section 14.3.

Before detailing our numerical modeling technique, we emphasize how it differs from that of an earlier parameter study focused on the infalling orbit of the satellite (Fardal et al. 2013). Fardal et al. (2013) sought the parameter set that best reproduced the observed structures. They were interested in the physical properties of M31 and its progenitor satellite. Consequently, they focused on a very narrow region of parameter space around the best-fit configuration. By contrast, we seek to restrict the region in which an SMBH of unknown location wanders around the M31 halo. This requires a wider and more systematic exploration of the parameter phase space to identify all plausible parameter ranges. Our systematic parameter study over a wide parameter region complements the study of Fardal et al. (2013).

14.1 Numerical Modeling of M31 and the Satellite

We simulate an accreting satellite dwarf galaxy interacting with M31 by using N-body simulations, concentrating on the infalling orbit of the progenitor dwarf galaxy. We assume an axisymmetric fixed potential model composed of a Hernquist bulge (Hernquist 1990), an exponential disk, and an NFW halo (Navarro et al. 1996) for M31 (Geehan et al. 2006; Fardal et al. 2007). This assumption of the fixed potential model is appropriate because the dynamical response of M31's disk to progenitor collision is negligible if the dynamical mass of the progenitor is below $5 \times 10^9 M_\odot$ (Mori & Rich 2008). Our numerical simulations are performed in a Cartesian coordinate system (x, y, z) whose origin represents the center of M31. The z axis is directed along our line-of-sight, and the x and y axes point east and north on the sky plane, respectively. This coordinate system has been commonly adopted in earlier studies of M31. The distance from Earth to M31 is assumed as 780 kpc (McConnachie et al. 2003); thus, 1° corresponds to a physical scale of 13.6 kpc. The heliocentric velocity of M31 toward the line-of-sight, east and south on the sky plane is assumed as -300 km s^{-1} (de Vaucouleurs et al. 1991), 127 km s^{-1}, and 75 km s^{-1} (Sohn et al. 2012; van der Marel et al. 2012), respectively.

By restricting the area in which the SMBH exists, this study aims to determine the observational field for future observational detections. The greatest contributor to the uncertainty in the current SMBH position is the uncertainty of the infalling orbit of the progenitor dwarf galaxy. Therefore, we should perform a large parameter study in the six-dimensional phase space to constrain the orbit of the infalling satellite to that of the observed structures. Since a six-dimensional parameter space is excessively large for an exhaustive search, even by recent high-performance computer architectures, we reduce the number of dimensions as follows. First, to ensure that the satellite interacts with M31, we fix the initial distance of the infalling satellite as 7.63 kpc from the center of M31 (corresponding to the scale radius of a dark matter halo; Fardal et al. 2007). In addition, we model M31 as an axisymmetric system. Imposing these conditions reduces the parameter space to four dimensions (the altitude of the initial position and the initial velocity vector); however, it is still very large.

To realize a sufficiently wide parameter space, the parameter sets are distributed on a relatively coarse grid defined in M31-centric spherical coordinates. Since M31 is axisymmetric, the azimuthal angle (around the rotation axis of the M31 disk) of the initial satellite position is simply related to the observational angle. Instead of performing multiple N-body simulations at different azimuthal angles, we "observe" snapshots of N-body simulations by rotating around the axis with a $\sim 3^\circ$ bin width. To determine the altitude of the initial position, the northern hemisphere of M31 is covered in 6° increments. We focus on a merger occurring immediately prior to the giant stellar stream in the southern hemisphere; thus, we consider the initial orbital position to lie in the northern hemisphere. The possibility that the satellite entered from the opposite side is discussed in Chapter 16.

The grid width of the infalling velocity of the satellite in the radial direction is ~ 13 km s^{-1}. Two characteristic velocities are 550, and 440 km s^{-1}, respectively, specifying the escape velocity at the initial position (7.63 kpc from the center of M31), and the velocity required to set the apoapsis of the satellite at $r = 100$ kpc. Since the giant stellar stream extends beyond 100 kpc from the center of M31 (McConnachie et al. 2003), the infalling velocity of the satellite should exceed 440 km s^{-1}. For a fair evaluation of other possibilities, we also test models in which the infalling radial velocity exceeds 550 km s^{-1} and is slower than 440 km s^{-1}. The tangential velocity vector at the initial position is described by two parameters specifying its norm and direction. The former parameter (the speed of the tangential velocity) is varied as a ratio of the radial velocity from 0 to 0.47 in $\sim 6.7 \times 10^{-2}$ increments. The latter parameter (direction of the tangential velocity) covers 360° divided into 32 bins.

In this parameter survey, the infalling satellite contains 65,536 particles, and the gravitational softening length is set as 50 pc. The progenitor is assumed as a King sphere with a total mass of $M_{\rm tot} = 3 \times 10^9 M_\odot$, concentration $c = 0.7$ and a tidal radius of $r_{\rm t} = 4.5$ kpc, since this model best matches the progenitor dwarf galaxy when the progenitor follows Fardal's orbit (Model A in Part I). The numerical scheme adopts a second-order leapfrog integrator and a shared fixed time step.

To sweep such a wide parameter space, we perform a vast parameter survey utilizing a graphics processing unit (GPU) cluster, namely, HA-PACS at the University of Tsukuba. For this purpose, a highly optimized N-body simulation code is implemented on the GPU cluster (see Part II). The combination of the state-of-the-art architecture and the highly optimized code enable the parameter study. HA-PACS, which equips over thousand boards of NVIDIA Tesla M2090, is a desirable system to sweep the wide parameter space. Furthermore, the code has a peak performance of 1 TFlop/s in single precision with a single NVIDIA Tesla M2090 board, which is 76% of the theoretical peak performance. Around a thousand runs of low-resolution N-body simulations finish in a day when 128 boards of NVIDIA Tesla M2090 (about one-eighth of HA-PACS) are in use.

14.2 On-the-fly Analysis

Since the number of N-body simulation runs is prohibitively large, we automatically and simultaneously analyze each snapshot of the numerical simulations. A check list of the automatic online evaluation is provided below:

1. The stellar stream and the west shell exist, and each mass exceeds $10^7 M_\odot$. This minimum value is much smaller than $2.4 \times 10^8 M_\odot$ estimated by Fardal et al. (2006), who assumed $M/L_{\rm V} \approx 7$.

2. The stellar stream is the most luminous structure in the southern area. The giant stellar stream is the most luminous object found by the PAndAS project in this region (McConnachie et al. 2009; Martin et al. 2013). This criterion eliminates the event of the undiscovered former satellite surviving the collision with M31.

3. The position of the surface density peak of the stellar stream matches that of the observed peak; the density of the simulated stream must peak within a fan-like region of angular width 15°, containing the observation field of the giant stellar stream (Font et al. 2006). This condition is similar to that described in Fardal et al. (2013).

4. The shapes of the two stellar shells adequately agree with the observed shapes. To quantify how precisely each run reproduces the observed shapes of the two shells, we compute the reduced χ^2 given by

$$\chi^2_\nu \equiv \frac{1}{\nu}\sum_{i=0}^{N-1}\left(\frac{x_{i,\,\mathrm{sim}} - x_{i,\,\mathrm{obs}}}{\sigma_{i,\,\mathrm{obs}}}\right)^2, \quad \nu = N-1, \tag{14.1}$$

   where the number of sampling points N is 48. The edge position of the observed shells is evaluated from the star count map prepared by Irwin et al. (2005). Successful parameter sets must satisfy $\chi^2_\nu \leq 1.7$ (99.7% confidence level according to Press et al. 2007).

5. The sharpness of the edge of the two stellar shells is consistent with the observations. We stipulate that the stellar density inside the edge is more than two times the stellar density outside the edge.

6. The mass-density ratios among the stellar stream, the east shells and the west shells are similar to the observed ratios. Subtracting the noise from the observed star count map by Irwin et al. (2005), we obtain the number density ratios of the east shell over the stream and the east over the west shells as 1.77 ± 1.57 and 2.05 ± 1.80, respectively. We stipulate that the mass density ratio is within 1σ scatter.

To "observe" the simulated snapshots, we must assume that the mass-to-light ratio of the satellite galaxy is evaluated in the visible light range (V-band); i.e., $M_{\mathrm{tot}}/L_{\mathrm{V}}$ where L_{V} is V-band luminosity. On the basis of the Faber-Jackson relation in the nearby universe (Falcón-Barroso et al. 2011; Toloba et al. 2012), the estimated V-band magnitude of the satellite galaxy is -17.73 ± 0.69 (corresponding to $M_{\mathrm{tot}}/L_{\mathrm{V}}$ of $2.84^{+2.52}_{-1.34}$). To eliminate the effects of large uncertainty in the mass-to-light ratio, we assume the bright end of the Faber-Jackson relation ($M_{\mathrm{tot}}/L_{\mathrm{V}} = 1.51$) and "observe" as many faint structures as possible. To mimic the observed star count map (Irwin et al. 2005), we "observe" the numerical results imposing a limiting magnitude of $V = 24.5$ (the detection limit of the Wide Field Camera on the Isaac Newton Telescope; Irwin et al. 2005).

The method for detecting the edge of the stellar shell while "observing" snapshots is optimized to capture all edge-like features. All density peaks and valleys along a radial direction on the sky plane are assigned as edge candidates, and the candidate nearest to the observed edge is tagged as the "observed" edge in the snapshots. If we know the actual mass-to-light ratio of the progenitor satellite galaxy, then the easiest and most plausible way to determine the shell edges is to combine the mass-to-light ratio with the instrument detection limit.

This simple method, however, would miss some edge signatures if the mass-to-light ratio is assumed greater (i.e. the satellite is assumed fainter) than the actual ratio. To avoid this situation, we detect the edge of the shells by the abovementioned method. Since only a small number of the N-body particles are used, spurious density peaks and valleys are introduced by Poisson noise, which artificially decreases the reduced χ^2 value. Later, this effect will be eliminated in the high-resolution N-body simulations (see Chapter 15).

Here, we compare the above "observing" criteria with those of earlier studies. In all of the earlier studies, the structures formed after a galactic merger had reproduced (all or some of) the observed global shapes (minor merger scenarios by Fardal et al. (2007, 2012, 2013); Mori & Rich (2008); Sadoun et al. (2013), and major merger scenario by Hammer et al. (2010, 2013)). Most studies have compared the shapes of the simulated and observed structures with the naked eye; exceptions are Fardal et al. (2013) and study presented in Part I.

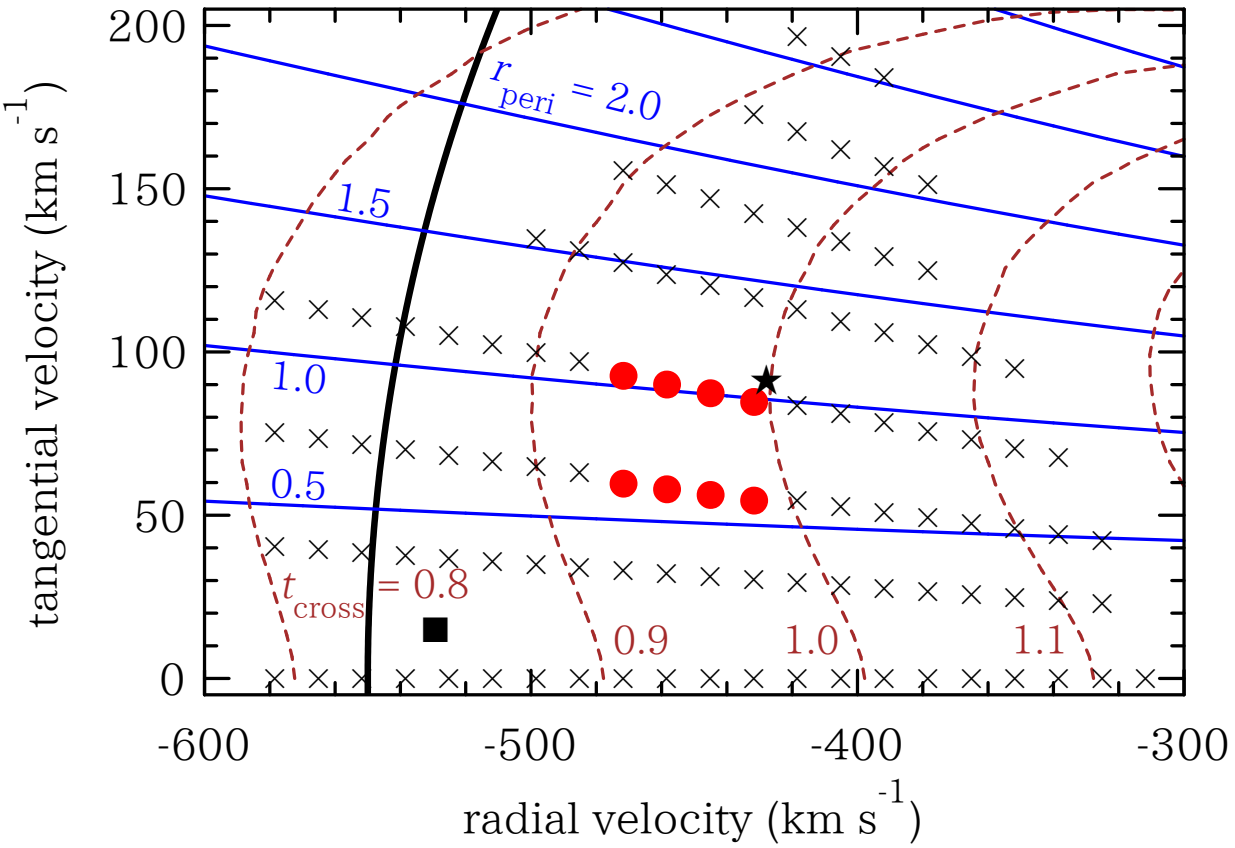


Fig. 14.1: Results of 44,880 runs of the low-resolution parameter study (corresponding to ∼5.7 million orbit models) simulating the infalling orbit of the satellite. The horizontal and vertical axes are the infalling radial and tangential velocities, respectively, of the satellite 7.63 kpc away from the center of M31. Red filled circles indicate that the results accurately reproduce the observed structures. Crosses represent results that failed to reproduce the observed structures. Overlaid curves show contour maps of periapsis, $r_{\rm peri}$ (kpc; blue solid lines), and crossing time $t_{\rm cross}$ (in units of free-fall time of the satellite; brown dashed lines). The filled star and the filled square represent the infalling orbit of the progenitor satellite galaxy in earlier studies Fardal et al. (2007) and Sadoun et al. (2013), respectively. The thick solid curve corresponds to the escape velocity 550 km s^{-1} at 7.63 kpc away from the center of M31.

14.3 Constraints on the Orbit of the Satellite

We have performed 44,880 runs of N-body simulations, corresponding to 5,699,760 models of the infalling orbit of the progenitor satellite. The results of the 5,699,760 orbit models are shown in Fig. 14.1. By automatic "observation" and by analyzing the N-body simulations described in Section 14.2, we have identified 138 orbit models that accurately reproduced the observed structures, hereafter referred to as orbit candidates (filled circles in Fig. 14.1). More specifically, only 138 out of 5,699,760 infalling orbit models passed our tests in this low-resolution parameter study. In other words, the possible parameter space is an extremely narrow region of the phase space. Periapsis $r_{\rm peri}$ (contour map of solid curves) and $t_{\rm cross}$ (contour map of dashed curves) are evaluated by test-particle calculation under a fixed potential in the spherical components of M31 (bulge and halo). The crossing time is defined as the time required to pass the region $r \leq r_{\rm c} = 4.3\,{\rm kpc}$ at the first pericentric passage, where $r_{\rm c}$ is the critical radius from the center of M31. At $r_{\rm c}$, the core radius of the satellite corresponds to its Hill radius against the M31 bulge. Within this radius, the tidal force exerted by M31 largely governs the time evolution of the satellite. At the core radius of the satellite, the free-fall time $t_{\rm ff}$ is 15 Myr.

The distribution of orbit candidates in Fig. 14.1 is concentrated around $r_{\rm peri} \cong 0.6 - 1$ kpc and $t_{\rm cross} \cong 0.95\ t_{\rm ff}$. Tidal forces exerted by the bulge of M31 stretch and disrupt the infalling satellite. Since the strength of the tidal force from the bulge of M31 is proportional to $r_{\rm peri}^3$, the parameters that adequately match the observed structures are restricted to a narrow $r_{\rm peri}$ region. Figure 14.1 shows that $t_{\rm cross}$ is also tightly constrained, implying the importance of the satellite's dynamical response to the tidal force exerted by the bulge of M31. The strong tidal field in the bulge of M31 ensures that even a small difference of the crossing time markedly affects the present-day structures.

One of the most important results in this study is that a maximum infall velocity of the progenitor satellite galaxy exists ($\sim$480 km s^{-1}). Since the escape velocity is 550 km s^{-1} (thick solid curve in Fig. 14.1), the observed structures can only be formed by an M31-bound satellite galaxy. The collision that occurred several hundred megayears ago should have been the first collision of the infalling satellite, because the strong tidal field exerted by the bulge of M31 will destroy the satellite in a single passage. However, as noted by Sadoun et al. (2013), this situation does not naturally arise in the hierarchical CDM context. This controversy and its solution will be described in Chapter 16.

We now compare our results with those of related studies (Fardal et al. 2007; Sadoun et al. 2013). The infalling orbit found by Fardal et al. (2007) (star in Fig. 14.1) locates near the edge of the area occupied by the 138 orbit candidates. This indicates consistency between our study and that of Fardal et al. (2007). Contrarily, the infalling orbit of Sadoun et al. (2013) is located outside of our area. Sadoun et al. (2013) set the satellite distant from M31 in order to delay its collision with M31. This discrepancy between our study and Sadoun et al. (2013) chiefly arises from the strict criteria adopted in our study, especially the reduced χ^2 analysis imposed on the shapes of the observed two stellar shells.

Chapter 15 Infalling Orbit of the SMBH

Here, we discuss high-resolution N-body simulations of the 138 orbit candidates that survived the low-resolution parameter study described in the previous chapter. First, we explain the numerical differences between the high- and low-resolution models in Section 15.1. The resulting positions of the SMBH derived from the high-resolution N-body simulations are presented in Sections 15.2 and 15.3.

15.1 Numerical Modeling with the SMBH

The purpose of this chapter is to restrict the locality of the wandering SMBH in the host galaxy. To simultaneously reproduce the observed structures and track the orbit of the SMBH, we have performed high-resolution N-body simulations of M31 interacting with a progenitor satellite containing an SMBH. We set the number of particles in the satellite to 524,288 and the gravitational softening length to 13 pc (equivalent to $\sim$10 times increase in the number of particles with $\sim$1/4 softening length, relative to the low-resolution survey). Here the SMBH is represented by an additional particle of mass $3 \times 10^6 M_\odot$ ($\sim$500 times more massive than the other N-body particles), which is placed at the center of the progenitor dwarf galaxy. This mass derivation assumes that the progenitor's stellar mass corresponds to its dynamical mass and that the Magorrian relation ($M_{\rm BH} \sim 10^{-3} M_{\rm sph}$) holds. Specifically, we adopt the maximum mass of the SMBH. All the other parameter setups are the same as those for the low-resolution N-body simulations, and the computation also performs on HA-PACS.

15.2 Hermitage of the SMBH

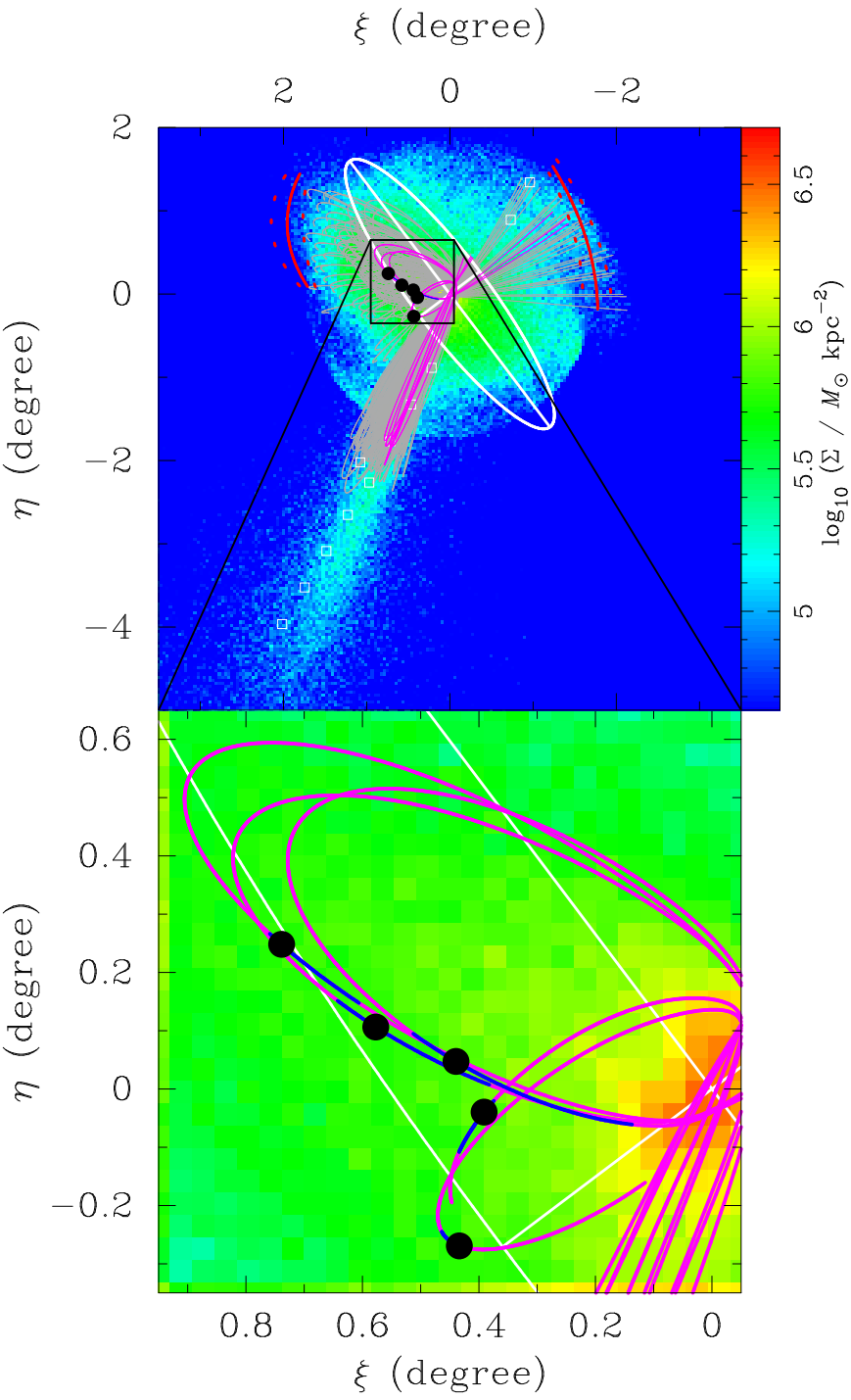


Fig. 15.1: Mass distribution (column density) maps of the debris of the dwarf galaxy in standard coordinates centered on M31. The color scale is shown along the right vertical axis of the upper panel. In the upper panel, global distribution of N-body particles (the best-fit epoch of the orbit model corresponding to SMBH ID 3 in Tab. 15.1) is shown as a color image while white curves and lines show the M31 disk. White squares show observed points of the giant stellar stream Font et al. (2006), and red curves show the edge and the width of the observed shells. The lower panel is a $1^\circ \times 1^\circ$ enlarged view of the black square in the upper panel. Black circles show the most probable current position of the SMBH. Gray and magenta curves show the orbits of the SMBH particles for the 138 orbit candidates and the 5 successful candidates, respectively. Blue curves show the SMBH positions when the observed shells are reproduced at the 99.7% confidence level.

Table. 15.1: Summarized information of five SMBH particles at the best-fit epoch

ID	χ^2_ν [a]	ξ (degree) [b]	η (degree) [b]	$r_{\rm M31}$ (kpc) [c]	R.A. (J2000.0)	Decl. (J2000.0)	D (kpc) [d]	$v_{\rm los}$ (km s^{-1}) [e]
1	1.42	0.74	0.25	27.3	00 45 41.60	+41 31 02.16	754.8	−131.0
2	1.51	0.39	−0.04	48.9	00 44 18.16	+41 13 45.71	731.4	−355.4
3	1.56	0.43	−0.27	36.8	00 44 28.43	+40 59 59.54	743.8	−178.1
4	1.59	0.58	0.11	22.0	00 45 02.82	+41 22 31.47	759.4	−77.7
5	1.61	0.44	0.05	17.9	00 44 29.79	+41 18 58.52	763.2	−33.1

(a) Reduced χ^2 when matching observed and simulated shapes of the two stellar shells.

(b) Position in M31-standard coordinates.

(c) Distance from the center of M31.

(d) Distance from the Local Standard of Rest.

(e) Heliocentric line-of-sight velocity.

Table. 15.2: Phase space location of five SMBH particles at the initial condition

ID	x (kpc)[(a)]	y (kpc)[(a)]	z (kpc)[(a)]	v_x (km s^{-1})	v_y (km s^{-1})	v_z (km s^{-1})
1	−3.60	6.03	−2.98	235.55	−362.46	121.18
2	−2.33	6.08	−3.97	142.56	−410.83	193.19
3	−2.33	6.08	−3.97	138.53	−399.21	187.73
4	−3.76	5.87	−3.10	243.22	−354.93	128.10
5	−3.45	6.18	−2.85	218.27	−376.19	111.29

(a) Position in standard coordinates centered on M31.

In the high-resolution N-body simulations, 5 orbit models out of the 138 orbit candidates accurately reproduced the observed structures. The increased number of particles yields a smoother mass distribution than in the low-resolution calculations, since Poisson noise is reduced. This effect eliminates orbit models whose reduced χ^2 values were underestimated when matching the shapes of the observed shells, as discussed in Section 14.2. Figure 15.1 shows the mass distribution map of the debris of the satellite galaxy, obtained by the best-fit epoch of the orbit model corresponding to SMBH ID 3 in Tab. 15.1. As shown in the upper panel of Fig. 15.1, N-body simulations accurately reproduce the observed structures, and the SMBH exerts no significant effect on the formation of the global structures. The top panel of Fig. 15.1 is overlaid with the orbits of the SMBH particles for the 138 orbit candidates (gray curves on the mass distribution map). Magenta curves show the orbits of the SMBH particles for the five successful candidates. The five black circles show the most probable current positions of the SMBH in the corresponding orbit models. Positional and velocity information of the five SMBH particles at the best-fit epoch and initial condition of simulations are summarized in Tables 15.1 and 15.2, respectively. The candidates are listed in ascending order of reduced χ^2 at the best-fit epoch.

The lower panel of Fig. 15.1 is an enlargement of the black hatched region of the upper panel, covering a region of $1^\circ \times 1^\circ$ ($\sim$15 kpc $\times$15 kpc). The blue curves trace the orbits of the SMBH particles when the observed shells are reproduced at the 99.7% confidence level. These curves indicate the possible regions currently occupied by the SMBH. Clearly, the blue curves are confined to a small region ($\sim 0^\circ.6 \times 0^\circ.7$), so the candidate field in which the SMBH must exist is within $1^\circ \times 1^\circ$. The above tight constraint for the current position of the wandering SMBH is imposed by strong constraints on the following two factors: (1) the infalling orbit of the progenitor galaxy and (2) the period in which the global structures are reproduced. The SMBH resides close to its apoapsis, implying that the SMBH moves relatively slowly, and the uncertainty in the current position is smaller than in other positions, such as the near periapsis.

15.3 Locus of the SMBH

Figure 15.2 shows the resultant χ^2 map of the 138 runs of the high-resolution N-body simulations, together with the spatial distribution of satellite galaxies around M31 in the M31-centric spherical coordinate system defined by McConnachie & Irwin (2006b). The M31-centric galactic longitude $l_{\rm M31}$ ranges from $-180^\circ \leq l_{\rm M31} \leq 180^\circ$, where $l_{\rm M31} = 0^\circ$ points toward the Milky Way. The M31-centric galactic latitude $b_{\rm M31}$ ranges from $-90^\circ \leq b_{\rm M31} \leq 90^\circ$, where $b_{\rm M31} = 0^\circ$ indicates the plane of the M31 disk. M31 and the Milky Way locate at the center and at $(l_{\rm M31}, b_{\rm M31}) = (0^\circ, -13^\circ)$, respectively (not plotted in the figure).

To compare the infalling orbits tested in this study with the distribution of satellite galaxies, the parameter space of low-resolution N-body simulations (light-brown region) and the results of high-resolution N-body simulations (color map) are also shown in Fig. 15.2. The light-brown region covers the northern hemisphere of the M31 halo, while the small area occupied by the color map indicates that the satellite must have

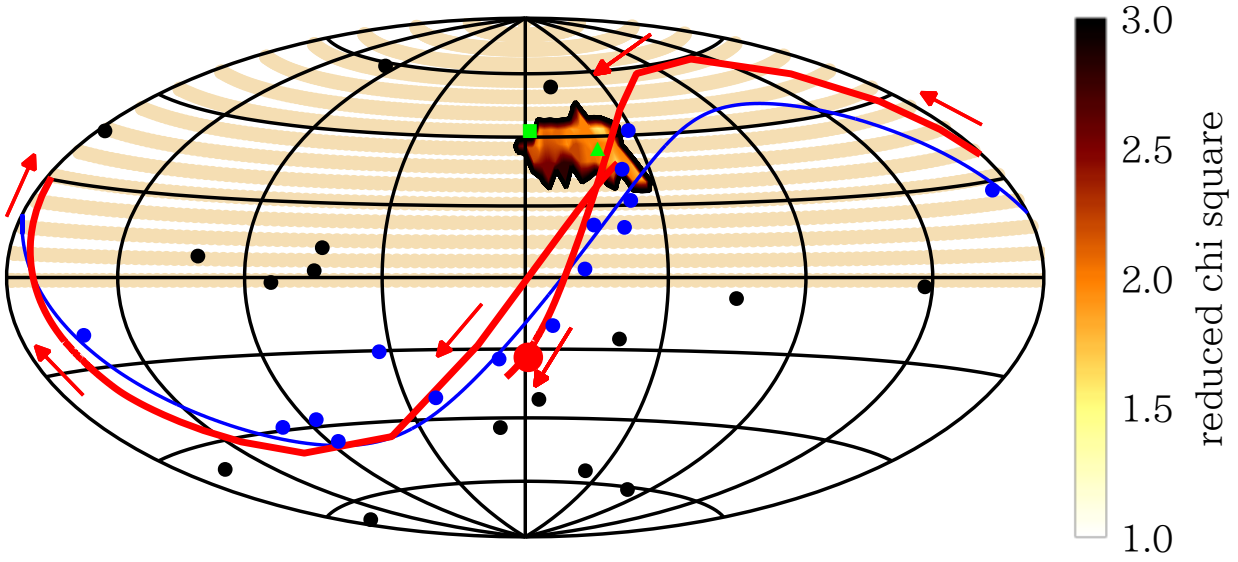


Fig. 15.2: Aitoff-Hammer equal-area projection of satellite galaxy distribution around M31 and infalling orbits of the progenitor. Horizontal and vertical axes indicate the M31-centric galactic longitude and latitude, respectively. Blue circles indicate satellite galaxies distributed in a vast thin disk with a pole at $(l_{\rm M31}, b_{\rm M31}) = (-78^\circ.4, 38^\circ.3)$ (blue curve: Ibata et al. 2013; Conn et al. 2013). Black circles indicate other satellite galaxies listed in Ibata et al. (2013) and Collins et al. (2013). Light-brown points show infalling satellite orbit models investigated in our low-resolution parameter study; these appear as light-brown bands or zones in the northern hemisphere. The overlaid color map shows the results of the high-resolution parameter study in terms of reduced χ^2 analysis of the shapes of the observed and simulated stellar shells. The green square and triangle show the infalling orbit models of Fardal et al. (2007) and Sadoun et al. (2013), respectively. All quantities related to the infalling orbit were evaluated at 7.63 kpc from the center of M31 (initial separation of N-body simulations in this study). The bold red curve with arrows shows the orbit of an SMBH particle (SMBH ID 3 in Tab. 15.1). The SMBH moves along the blue curve (disk plane of the satellites) from the orange-colored region in the bottom left direction. The SMBH progresses along the blue curve and reaches the filled red circle, indicating its current position.

infallen within a very narrow directional range in the northern hemisphere to reproduce the observational structures. The bold red curve with arrows shows the orbit of an SMBH particle (SMBH ID 3 in Tab. 15.1). The SMBH moves along the blue curve (disk plane of the satellites) from the orange-colored region in the bottom left direction. The SMBH progresses along the blue curve and reaches the filled red circle, indicating its current position.

We compare our results with those of related studies (Fardal et al. 2007; Sadoun et al. 2013). For this purpose, Fig. 15.2 also plots the infalling orbits at $r = 7.63$ kpc adopted in earlier studies (Fardal et al. 2007; green square; Sadoun et al. 2013; green triangle). The infalling orbit of Fardal et al. (2007) locates at the edge of the parameter space in the high-resolution N-body simulations. Earlier, we established that it also locates near the edge of the possible parameter region in terms of the infalling velocity (see Fig. 14.1). Thus, this orbit almost matches the orbits of our parameter study. Although the infalling orbit of Sadoun et al. (2013) occupies the high-resolution parameter region in Fig. 15.2, their orbit is inconsistent with our results. This discrepancy arises because the infalling radial velocity is very high, and the orbital angular momentum very low, in their model (as mentioned in Section 14.3).

Here we briefly consider the possibility that the progenitor satellite galaxy entered from the southern hemisphere of M31. As mentioned in Section 14.1, to produce the giant stellar stream occupying the southern hemisphere, the progenitor satellite galaxy should fall into the central region of M31 from the

northern hemisphere. In Section 14.3, we also demonstrated that the progenitor satellite passed at 1 kpc distance from the center of M31 during its free-fall time. This indicates that the tidal interaction generated by the bulge of M31 through the pericentric passage is sufficiently strong to destroy the satellite. In other words, the merger investigated in this study was the first merger between the satellite and M31. Therefore, we conclude that the satellite galaxy entered from the northern hemisphere of M31, as earlier assumed in this study.

Chapter 16 Discussion

In Section 16.1, we discuss and confirm the validity of assumptions employed in this study. Then, we discuss the origin of the possible orbits (Section 16.2) and relations with other components observed in M31 (Section 16.3). Kawaguchi et al. (submitted to ApJ) estimated the broadband spectrum from the SMBH based on the SMBH orbit presented in Chapter 15. In Section 16.4, we introduce their feasibility study on detecting the wandering SMBH.

16.1 Validity of the Assumptions

In this paper, we assume a fixed potential model for M31. Therefore, the simulations exclude the possible effects of dynamical friction introduced by the bulge, the disk, and the halo of M31. However, dynamical friction plays a key role in sinking the SMBHs from the halos to the central regions of their host galaxies. To estimate how dynamical friction changes the orbits of SMBHs, we now evaluate the amount of kinetic energy lost by the SMBH through dynamical friction. The Chandrasekhar formula (Binney & Tremaine 2008) gives the energy dissipation rate due to dynamical friction $W_{\rm fric}$ as

$$W_{\rm fric} = -\frac{4\pi G^2 {M_{\rm BH}}^2 \rho\left(\boldsymbol{r}_{\rm BH}\right) \ln\Lambda}{v_{\rm BH}} \left[\mathrm{erf}(X) - \frac{2X}{\sqrt{\pi}} e^{-X^2}\right]. \tag{16.1}$$

Here, $\rho(\boldsymbol{r})$ is the mass density profile, and X is defined as $v_{\rm BH}/\sqrt{2\hat{\sigma}^2}$, where $\hat{\sigma}$ is the velocity dispersion. The quantities $\ln\Lambda$, $\boldsymbol{r}_{\rm BH}$, and $\boldsymbol{v}_{\rm BH}$ denote the coulomb logarithm, position, and velocity of the SMBH, respectively. To estimate the effects of dynamical friction, we must estimate the mass distribution, the velocity dispersion of field particles, and the coulomb logarithm. For this purpose, we adopt the Hernquist bulge, the exponential disk, and the NFW halo assumed in the N-body simulations. The velocity dispersion of the bulge is 260 km s^{-1} (Geehan et al. 2006). From the equation of motion and Poisson's equation (see Mori & Rich 2008), the velocity dispersion of the disk along its rotation axis is obtained as 60 km s^{-1}. We assumed a thin and axisymmetric disk, with a flat rotation curve and constant velocity dispersion. If the distribution function is further assumed as isotropic, the velocity dispersion of the halo is 233 km s^{-1} at its scale radius (Widrow 2000). The coulomb logarithm is approximately given by

$$\ln\Lambda = \ln\left(\frac{b_{\rm max} {v_{\rm typ}}^2}{G M_{\rm BH}}\right), \tag{16.2}$$

where $b_{\rm max}$ is the maximum impact parameter (here assumed as 40 kpc, sufficiently greater than the size of the bulge and the disk of M31), and $v_{\rm typ}$ is the typical velocity (velocity dispersion in the bulge and halo; maximum rotation velocity of 260 km s^{-1} in the disk).

Figure 16.1 shows the effects of dynamical friction on the orbital evolution of SMBH particles. The rate of energy loss, and the energy loss over 1 Myr (normalized by kinetic energy) evolves as shown in the top and bottom panels, respectively. Dynamical friction is relatively large when the SMBH passes its periapsis (a region of high mass density), and exerts negligible effects up to the present day ($t = 0$). Since the stream was formed by a near head-on collision (see Section 14.3), the progenitor resided in the central region of M31 for very short periods. This explains why our results were essentially unaffected by dynamical friction. As shown in the bottom panel of Fig. 16.1, about 10^{-3} of the SMBH's kinetic energy was dissipated in a single

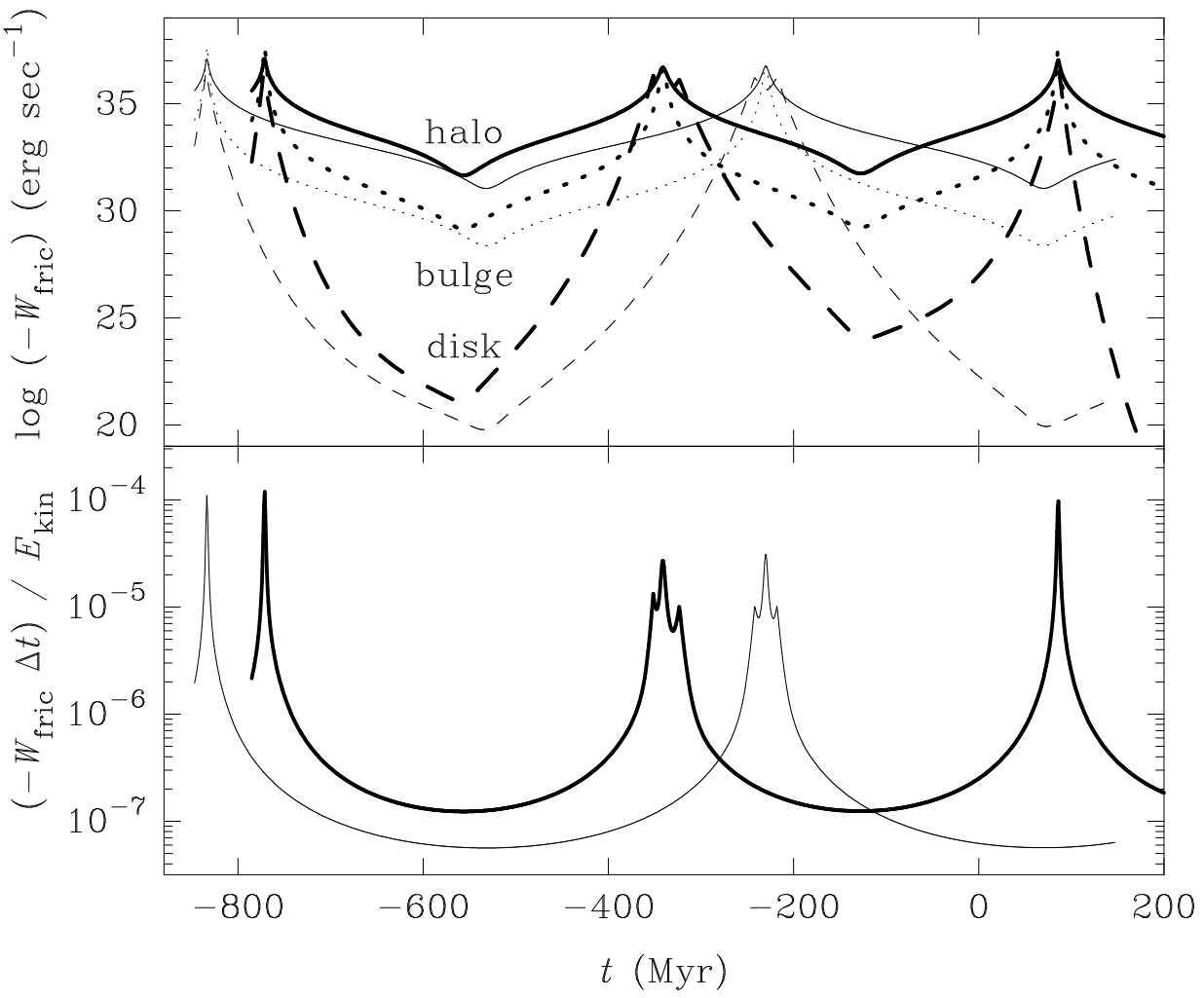


Fig. 16.1: Estimating the effects of dynamical friction on the orbital evolution of SMBH particles. In the upper panel, the rate at which kinetic energy is lost through dynamical friction is plotted as a function of time. Solid, dotted, and dashed curves show the contributions by M31's halo, bulge, and disk component, respectively. Thin and thick curves show the time evolution of each contribution for two SMBH particles selected from five successful candidates (SMBH ID 1 and 2 in Tab. 15.1, respectively). Peaks and troughs in these curves correspond to the passing of SMBHs through their periapsis and apoapsis, respectively. The lower panel plots the energy lost through dynamical friction as a function of time. In these plots, the energy loss during Δt of 1 Myr is normalized by the kinetic energy of the particles throughout the same period. Both panels clearly show that the effects of dynamical friction amplify only as the SMBH passes the central region of M31; thus, dynamical friction exerts a negligible effect on the motion of SMBHs at the present time.

pericentric passage as the SMBH traversed the central region of M31 over ∼10 Myr (see Section 14.3). Thus, dynamical friction encountered during a few crossings exerts no influence on the motion of the SMBH in this study. Consequently, our N-body simulations, which ignore dynamical friction, are sufficiently realistic to predict the current position of the SMBH.

In the next place, SMBH shifts from its initial position by gravitational Brownian motion, a random walk in momentum space perturbed by gravitational encounters with nearby stars (Merritt 2001, 2005).

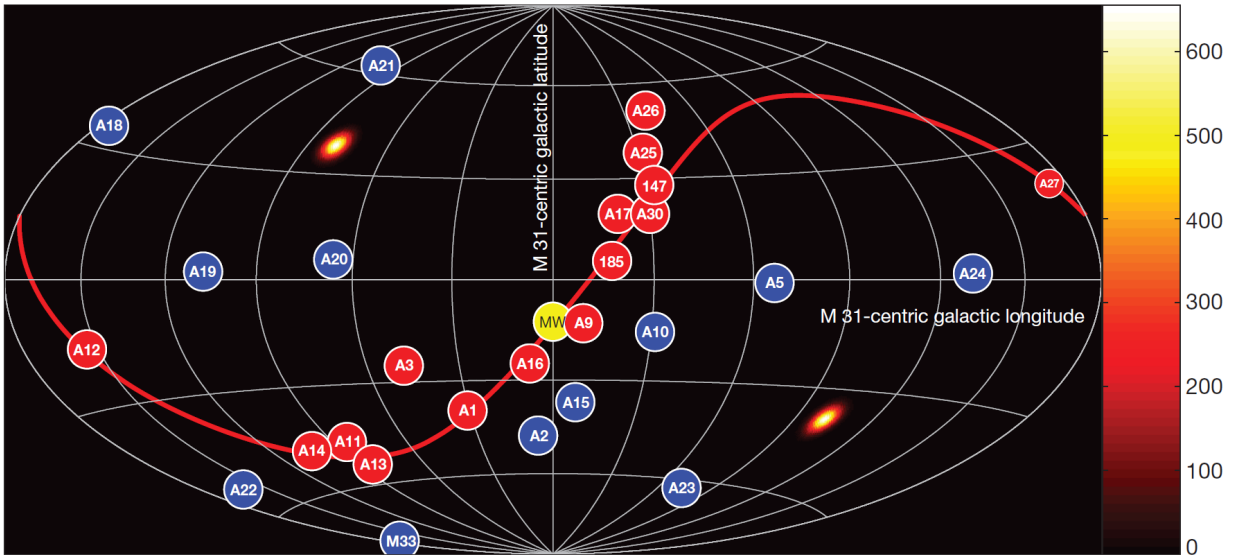


Fig. 16.2: The distribution of 27 satellite galaxies around M31, taken from Ibata et al. (2013). A red curve shows a plane that 15 red circles distribute along it, and the background color image exhibits the probability density function about the poles of the plane.

According to Merritt et al. (2007), the expected velocity dispersion of the SMBH is

$$5.0 \times 10^{-2}\,\mathrm{km\,s^{-1}} \times \left(\frac{\tilde{m}}{1M_{\odot}}\right)^{1/2} \times \left(\frac{\tilde{\sigma}}{50\,\mathrm{km\,s^{-1}}}\right), \tag{16.3}$$

where $\tilde{m}$ and $\tilde{\sigma}$ are the effective stellar mass and the one-dimensional velocity dispersion of nearby stars in the satellite galaxy, respectively. In this paper, we assume that the velocity dispersion of the satellite galaxy is $\tilde{\sigma} = 49.1$ km s^{-1} at the center, and $\tilde{\sigma} = 39.3$ km s^{-1} at the core radius. This value is negligibly small and will not visibly alter the SMBH orbit; hence, we conclude that excluding these effects does not alter our predictions of the current SMBH position.

16.2 Origin of the Progenitor Satellite

In the recent observational studies of the satellite galaxy distribution around M31, Ibata et al. (2013) and Conn et al. (2013) concluded that 15 of the satellite galaxies are arranged in a disk-like structure. Figure 16.2 distinguishes the satellite galaxies forming a vast thin disk around M31 (Ibata et al. 2013; Conn et al. 2013, red circles) from other satellite galaxies (Ibata et al. 2013; Collins et al. 2013, blue circles), which are randomly scattered. As noted by Ibata et al. (2013) and Conn et al. (2013), approximately a half of the M31 satellite galaxies locates near the disk plane.

The origin of the satellite distribution is an open question (Hammer et al. 2013; Goerdt & Burkert 2013; Bahl & Baumgardt 2013). Fouquet et al. (2012) and Hammer et al. (2013) proposed a scenario to explain the observed distribution. Hammer et al. (2013) compared the observed satellite distribution to the M31 formation model as a gas-rich major merger (Hammer et al. 2010), and found the induced tidal tails are lying in the plane (see Fig. 16.3). If a sufficient number of tidal dwarf galaxies exist in the tidal tails, then the observed satellite plane would be explained. Goerdt & Burkert (2013) proposed an alternative scenario. They argued that the satellite accretion via multiple cold streams naturally explains the thin disk of satellites. On the other hand, Bahl & Baumgardt (2013) reported that the distribution can be reproduced within the standard cosmological framework based on their analysis of Millennium II simulation (Boylan-Kolchin et al. 2009). Bahl & Baumgardt suggested that the observed distribution might be a statistical

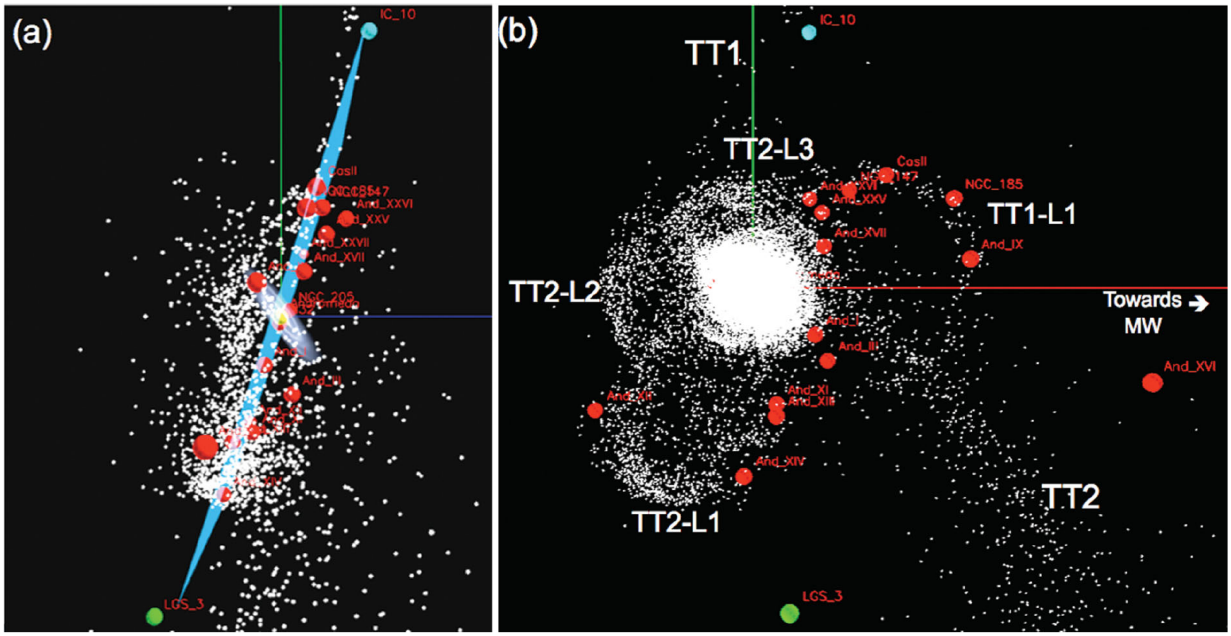


Fig. 16.3: The distribution of M31 satellite galaxies and stellar particles in the numerical simulation, taken from Hammer et al. (2013). The blue array represents the disk plane found by Ibata et al. (2013); Conn et al. (2013).

fluctuation surviving ~ 1 Gyr of an underlying more spherical satellite distribution. Future observations, especially proper motions of satellites, and higher resolution simulations are necessary to understand the origin of the satellite distribution.

Figures 16.4-16.6 illustrate a three-dimensional distribution of satellite galaxies and the SMBH orbits. A difference between Fig. 16.5 and Fig. 16.6 is the coordinate system: the M31 standard coordinate system and the M31 disk orthogonal coordinate system are adopted in Figures 16.5 and 16.6, respectively. On the latter coordinate system, the M31 disk lies on the XY plane and the Milky Way locates at $(X, Y, Z) = (760\,\mathrm{kpc}, 0\,\mathrm{kpc}, -175\,\mathrm{kpc})$.

Figure 16.4 reveals a surprising result: a part of the possible orbit region overlaps the disk plane of the satellite galaxies around M31. This infers a close connection between the formation process of the giant stellar stream and the disk plane of the satellite galaxies. Sadoun et al. (2013) considered that several hundred megayears ago is very recent to admit first-time collision between a bound satellite galaxy and its host galaxy. However, this unnatural situation can be understood from the disk-like distribution of satellite galaxies. If the satellites are distributed in this way, the probability of satellite-satellite interactions is higher than when the satellites are randomly distributed since the number density of satellite galaxies is greater in disk geometry. A satellite galaxy within the disk plane can lose most of its angular momentum through inter-satellite interactions. Once this occurs, the satellite is expected to follow a highly radial infall orbit toward M31, generating the observed structures such as the giant stellar stream. In future studies, we will evaluate the cross-section of these interactions and discuss the possibility of this working hypothesis.

One of candidate objects potentially perturbs the disk of satellites is M33. Since the estimated virial mass of M33 ($1.5 \times 10^{11}\,M_\odot$ by van der Marel et al. 2012) is much greater than that of the progenitor satellite of the stream and the shells, M33 may effectively pull the orbital angular momentum away from the satellite. An M31-M33 interaction model proposed by McConnachie et al. (2009) indicates that M33 passed only 53 kpc away from M31 about 2 Gyr ago (Fig. 16.7). After their proposal, Sohn et al. (2012) and van der Marel et al. (2012) presented results of the proper motion of M31 based on *Hubble Space Telescope* measurements. Based on this new observation with the observed proper motion of M33 (Brunthaler et al. 2005), we here

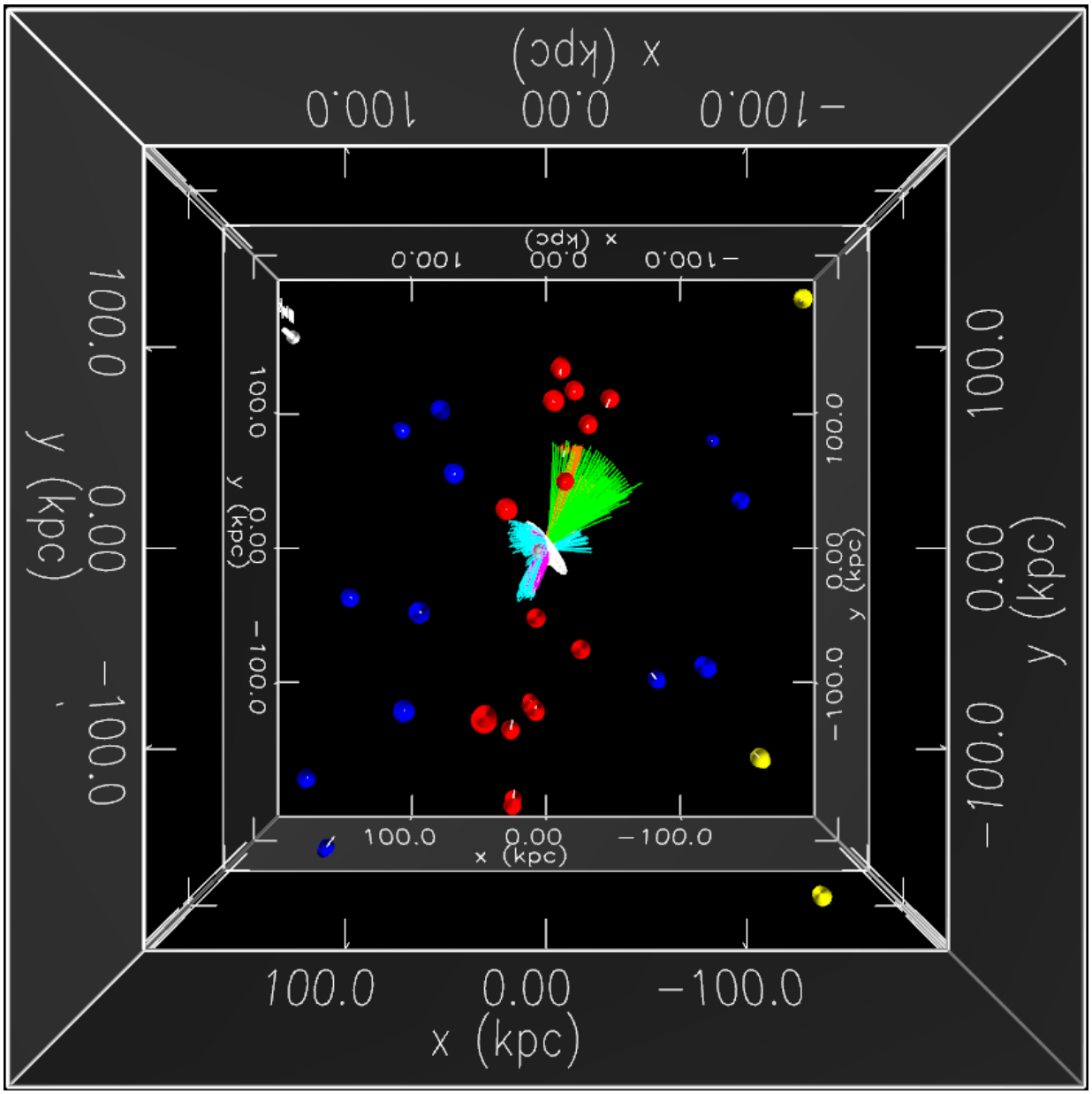
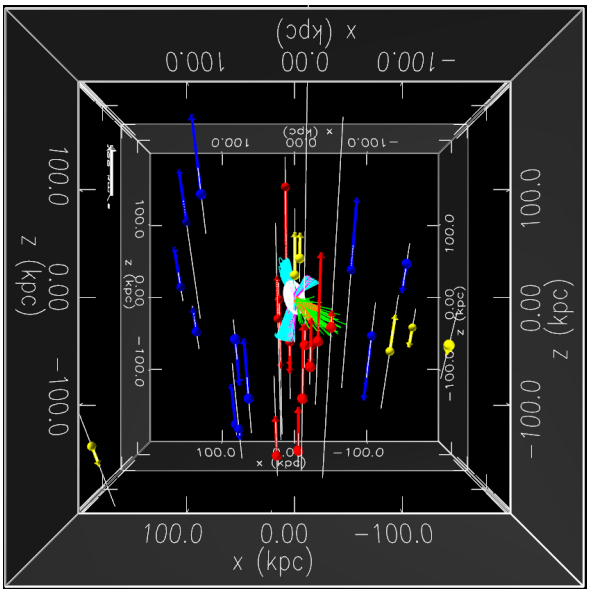


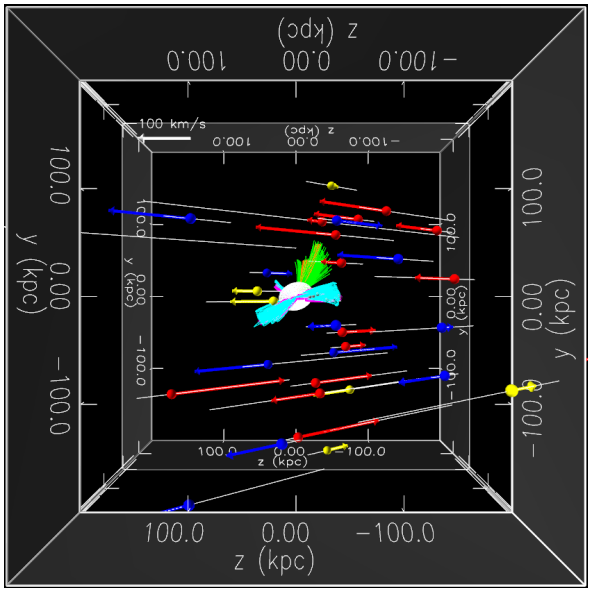
Fig. 16.4: A three-dimensional view of the SMBH orbits and the distribution of satellite galaxies around M31. Colored spheres with bars, vectors, and labels indicate satellite galaxies distributed around M31 with errors of distance measurements, heliocentric line-of-sight velocity, and object name: red spheres denote satellites forming the thin disk, blue spheres denote other satellites listed in Ibata et al. (2013), and yellow spheres indicate satellites listed in Collins et al. (2013) as well as M32 and NGC 205. Cyan and green curves show the 138 candidate orbits; cyan curves result from the N-body simulations, and green curves are orbits of the SMBH falling from the corresponding apoapsis into $r = 7.63$ kpc (evaluated by test-particle calculations). Magenta and orange curves highlight the five successful orbits and dark gray spheres show the current position of SMBHs. Three-dimensional visualization was conducted with the S2PLOT programming library (Barnes et al. 2006; Barnes & Fluke 2008). The online-version enables readers to change the position of the observer interactively (click the above figure to activate). Figures 16.5 and 16.6 exhibit snapshots of this figure from various viewing angles.

shortly examine the feasibility of the past M31-M33 interaction using test-particle calculations.

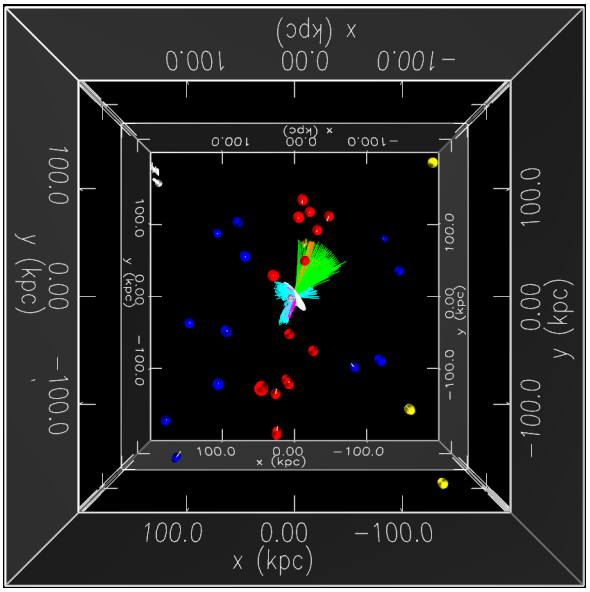
The initial position of M33 is (146, −140, −12) kpc on the M31 standard coordinate system (McConnachie



(a) Top view.



(b) Left view.



(c) Front view (click to activate).

Fig. 16.5: Snapshots of Fig. 16.4 from various viewing angles. The online-version of the front view (corresponds to a view from the LSR) is a three-dimensional plot.

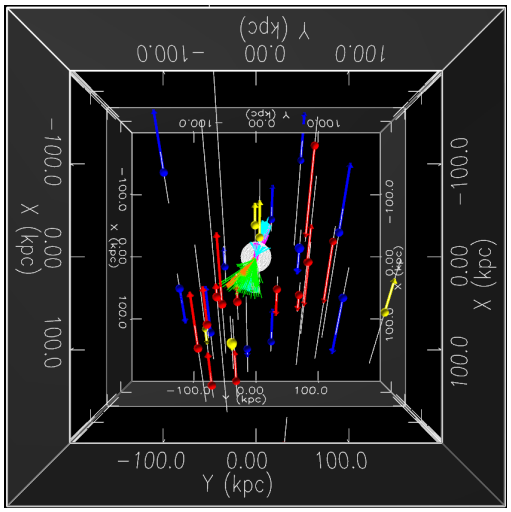
et al. 2004). According to Brunthaler et al. (2005) and van der Marel et al. (2012), the observed velocity of M33 in the M31 rest frame is $(-72 \pm 45,\ 147 \pm 39,\ 118 \pm 7)$ km s^{-1}. Then, the initial velocities of test particles are given by the Maxwell-Boltzmann distribution:

$$f(\boldsymbol{v}) = \prod_{i=1}^{3} \left[\frac{1}{(2\pi {\sigma_i}^2)^{3/2}} \exp\left\{ -\frac{(v_i - \langle v_i \rangle)^2}{2{\sigma_i}^2} \right\} \right]. \tag{16.4}$$

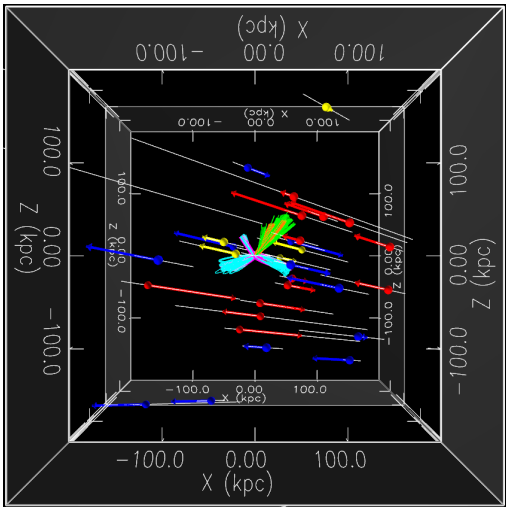
We have distributed 4,194,304 particles in the velocity space and have calculated the orbital evolution in the past. Figure 16.8 shows the result from 5 Gyr ago to the present day. The derived probability distribution of the distance between M33 and M31 (Fig. 16.8a) implies that M33 is most likely approaching M31 for upcoming the first interaction with M31 in this 5 Gyr. About 0.8% of test-particles have experienced an

encounter with M31 at the same period (Fig. 16.8b).

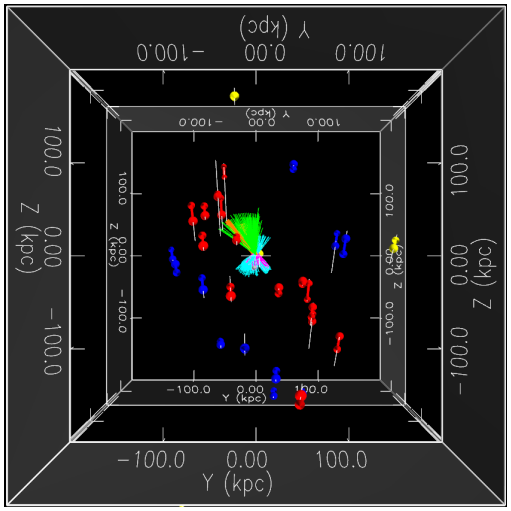
The remarkable correspondence between the orbit and the disk plane becomes apparent when the orbit of an SMBH particle (SMBH ID 3 in Tab. 15.1) is superimposed on the galaxy distribution (see Fig. 15.2 and 16.4). Two of the five successful orbits listed in Table 15.1 (SMBH ID 2 and 3) lie on the disk plane. Therefore, if future observations establish that the SMBH is wandering in the M31 halo, we will be equipped with highly suggestive clues regarding the formation and evolutionary history of the M31 halo. These data will enable a connection of the observed stellar structures and the current and ancient distributions of satellite galaxies. Furthermore, since two of the SMBH orbits coincide with the disk plane of the satellite galaxies, more wandering SMBHs might reside in the M31 halo. Figure 15.2 suggests that an SMBH initially moving along the disk plane remains on the plane. The observed disk-like satellite distribution is expected to trigger satellite-M31 interactions by extracting orbital angular momentum via unknown process, for example, satellite-satellite interactions. Consequently, remnants of ancient merger events should be concentrated in the halo region near the disk plane of the satellites. The higher merger rate suggests that SMBHs will be similarly concentrated in this field of the halo. Therefore, a group of wandering SMBHs might locate along the disk plane formed by the satellite galaxies.



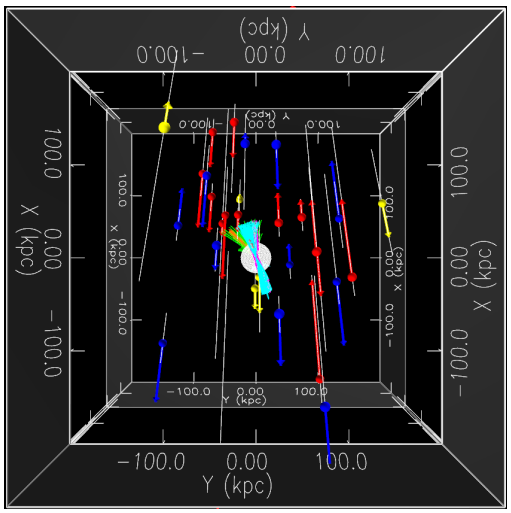
(a) Top view.



(b) Left view.



(c) Front view (click to activate).



(d) Bottom view.

Fig. 16.6: Same as in Figure 16.5 but on the M31 disk orthogonal coordinate system.

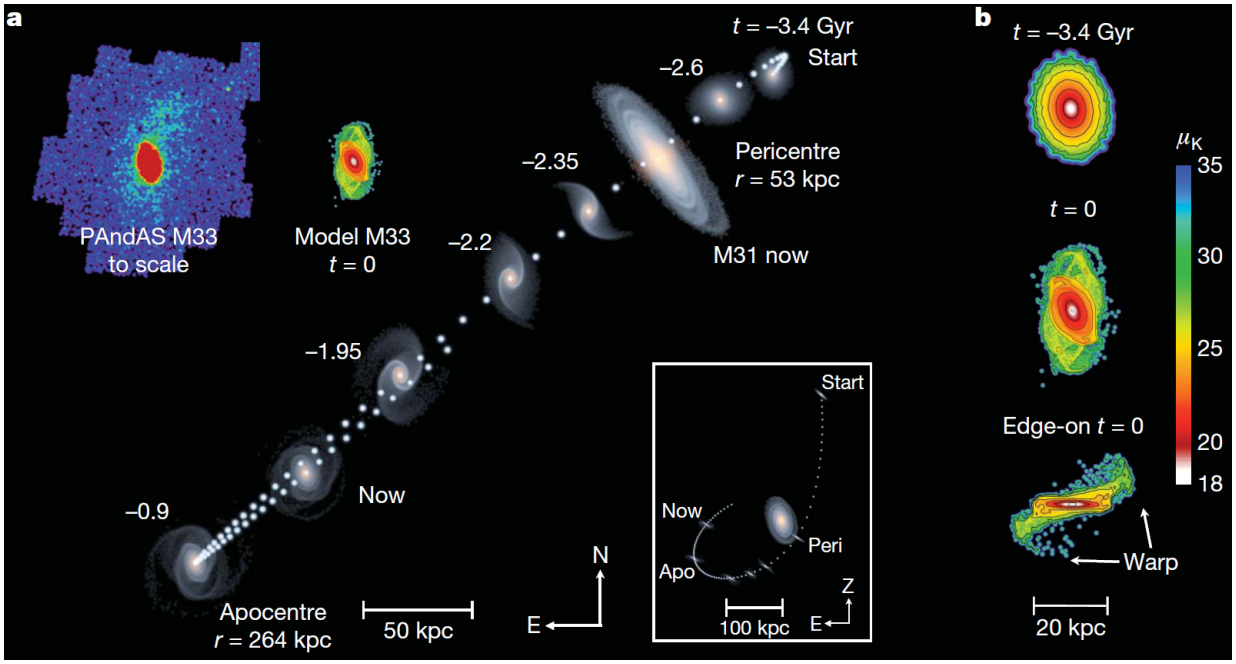
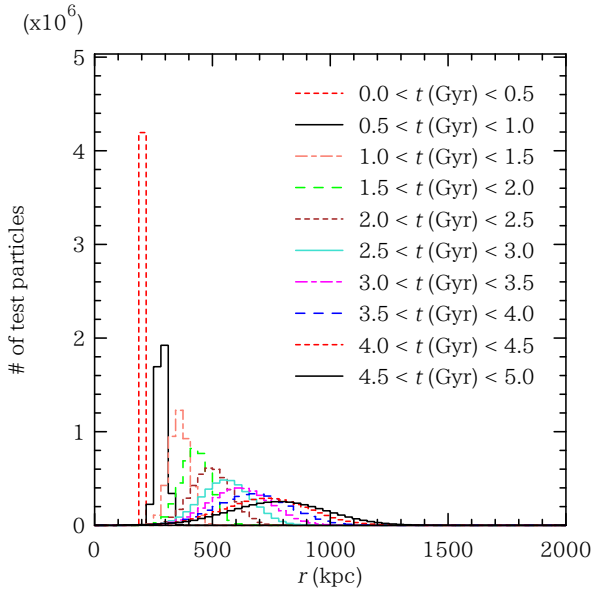
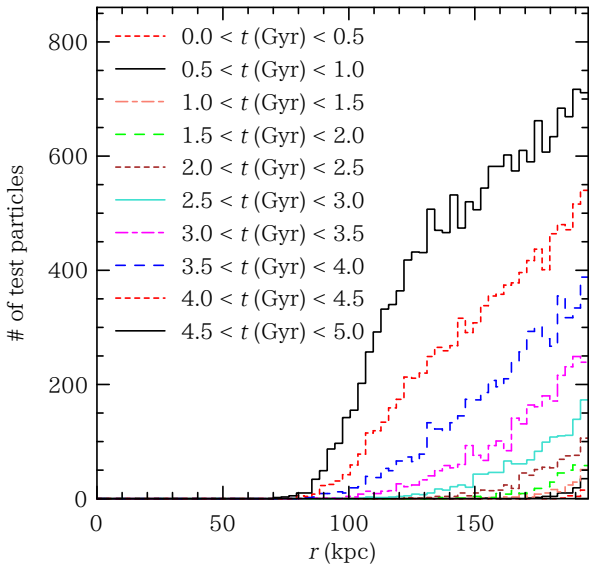


Fig. 16.7: An M31-M33 interaction model, taken from McConnachie et al. (2009).



(a) From left to right, lines represent recent to past distribution.



(b) Within the virial radius of M31. From bottom to top, lines represent recent to past distribution.

Fig. 16.8: Histogram of the distance between test-particles and M31 in the past ($t = 0$ and 1 Gyr are the present-day and 1 Gyr ago, respectively). Each line shows the minimum distance distribution for a given period of 500 Myr.

16.3 Impacts on Components of M31

Here, the study by Gordon et al. (2006) is worthy of note. They concluded that a 10 kpc ring observed in the M31 disk is a remnant of an offset merger. The ring structure of radius ∼10 kpc has been extensively reported in infrared or Hα images as shown in Fig. 16.9 (Habing et al. 1984; Rice 1993; Devereux et al. 1994; Haas et al. 1998; Gordon et al. 2006). Gordon et al. (2006) argued that the ring formed at an offset distance of 1.4 kpc from the center of M31 (Fig. 16.10), close to $r_{\rm peri}$ determined in this study (see Section 14.3). This correspondence between the two studies suggests that the 10 kpc ring is another fossil record of the minor merger investigated in this study.

Davidge (2012) discovered an overdense region of effective radius 0°.04, at $(\xi, \eta) = (0°.24, 0°.20)$. Our results suggest that the clump found by Davidge (2012) may be related to the giant stellar stream (Fig. 16.11). To check the physical connection between the former minor merger event and the clump, the clump must be discriminated from the M31 disk component in phase space on the basis of spectroscopic observations. Fardal et al. (2013) similarly compared the "location of the progenitor's central material" with the clump position (Fig. 16.12); however, their results were inconclusive because their stellar particles were widely spaced. Since the very central region was resolved by a small number of N-body particles in Fardal et al. (2013), it leads to a sparse wider distribution after the collision. Kirihara et al. (in preparation) investigated a model of M31 colliding with a dwarf spiral galaxy comprising a stellar bulge, disk, and dark matter halo. They reported that part of the bulge component of the progenitor survived the collision. The surviving part is located close to the current position of the wandering SMBH identified in this study. This correspondence strongly suggests that the current SMBH position is independent on the morphology of the progenitor satellite galaxy.

As noted in Section 14.3, the periapsis of the SMBH in the first collision between M31 and the progenitor satellite $r_{\rm peri}$ is around 1 kpc. Through the pericentric passage, a part of stars would be still bound by the SMBH. Here, we simply estimate the mass of the expected star cluster around the SMBH. The M31 bulge strip stars outside the Hill radius $r_{\rm Hill}$ from the SMBH by the strong tidal force. The Hill radius due to the M31 bulge (a Hernquist bulge with the total mass $M_{\rm b}$ of $3.24 \times 10^{10}\ M_\odot$ and the scale radius $r_{\rm b}$ of 0.61 kpc) is represented as

$$r_{\rm Hill} = \left[\frac{M_{\rm BH}}{3M_{\rm b}\left\{r_{\rm peri}/(r_{\rm peri}+r_{\rm b})\right\}^2}\right]^{1/3} r_{\rm peri} = \left(\frac{M_{\rm BH}}{3M_{\rm b}}\right)^{1/3}\left(1+\frac{r_{\rm b}}{r_{\rm peri}}\right)^{2/3} r_{\rm peri}$$
$$= 43\,{\rm pc} \times \left(\frac{M_{\rm BH}}{3\times10^6\,M_\odot}\right)^{1/3}. \quad (16.5)$$

The total mass of stars distributed within the Hill radius in the original King profile is $2.6 \times 10^5\ M_\odot$, about 10% of the SMBH mass. As shown in Figs. 16.13 and 16.14, many globular clusters are detected near the predicted positions of the wandering SMBH (Peacock et al. 2010; Trudolyubov & Priedhorsky 2004). Mackey et al. (2010) in part of the PAndAS project also reported the presence of globular clusters at the same observation area. One of them may be the surviving star cluster which contains the wandering SMBH.

Since the SMBH likely occupies the M31 halo, the SMBH of the progenitor should not be considered as an origin of multiple nuclei in M31* (Lauer et al. 1993; Bender et al. 2005). Furthermore, since the SMBH locates closer to the Milky Way than the M31 disk, the M31 disk will not disturb future observational attempts to detect the SMBH. If the wandering SMBH is experimentally verified in the near future, our understanding of how SMBHs coevolve with their host galaxies will be greatly enhanced. The candidate fields determined in this study will complement future observations. Since the mass of the SMBH is close to the low-mass end of the $M_{\rm BH} - \sigma$ relation, observational discoveries of the SMBH will provide information on the low-mass end of the SMBH-host galaxy associations.

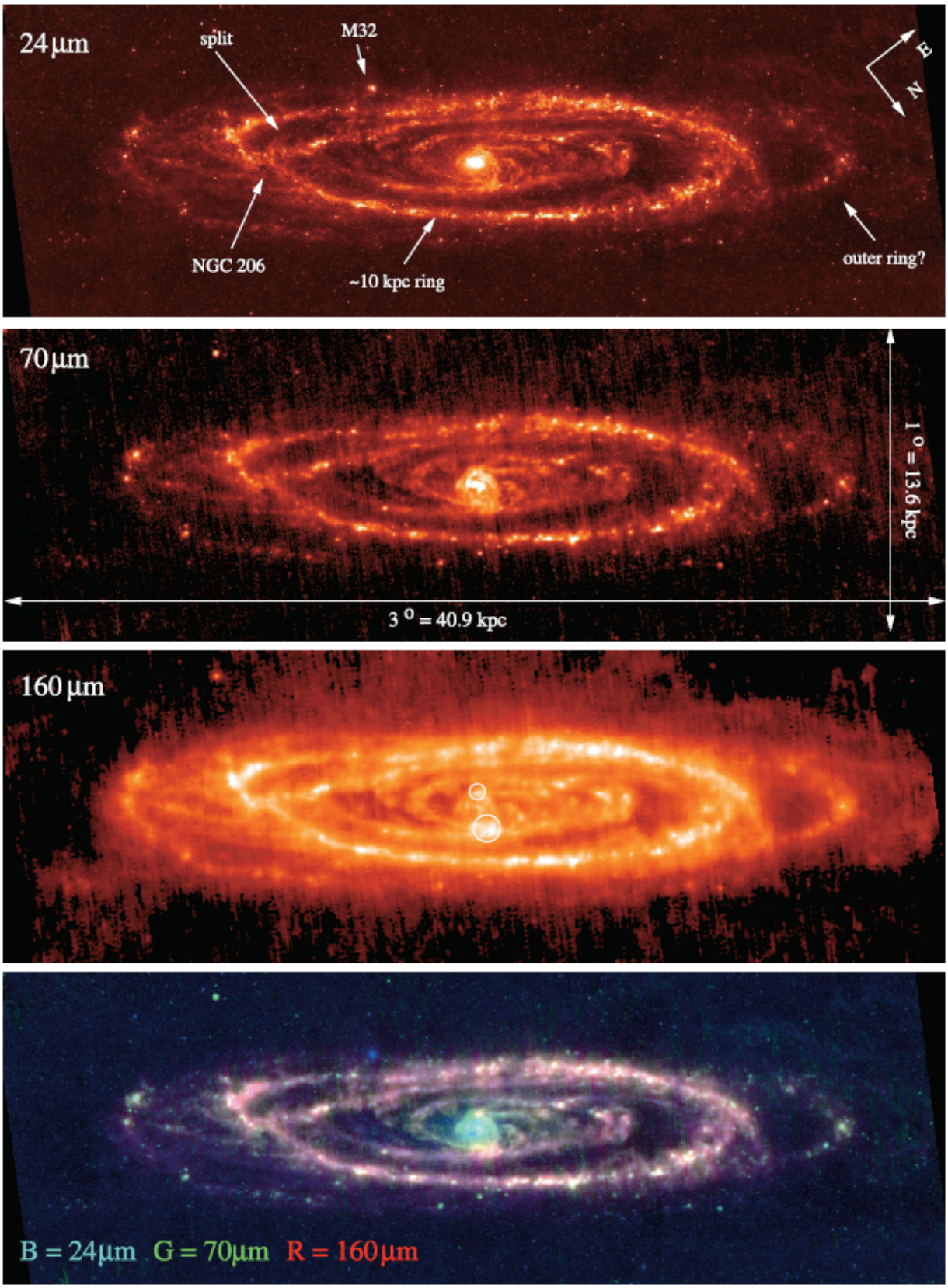


Fig. 16.9: MIPS (Multiband Imaging Photometer for *Spitzer*) images of M31, taken from Gordon et al. (2006).

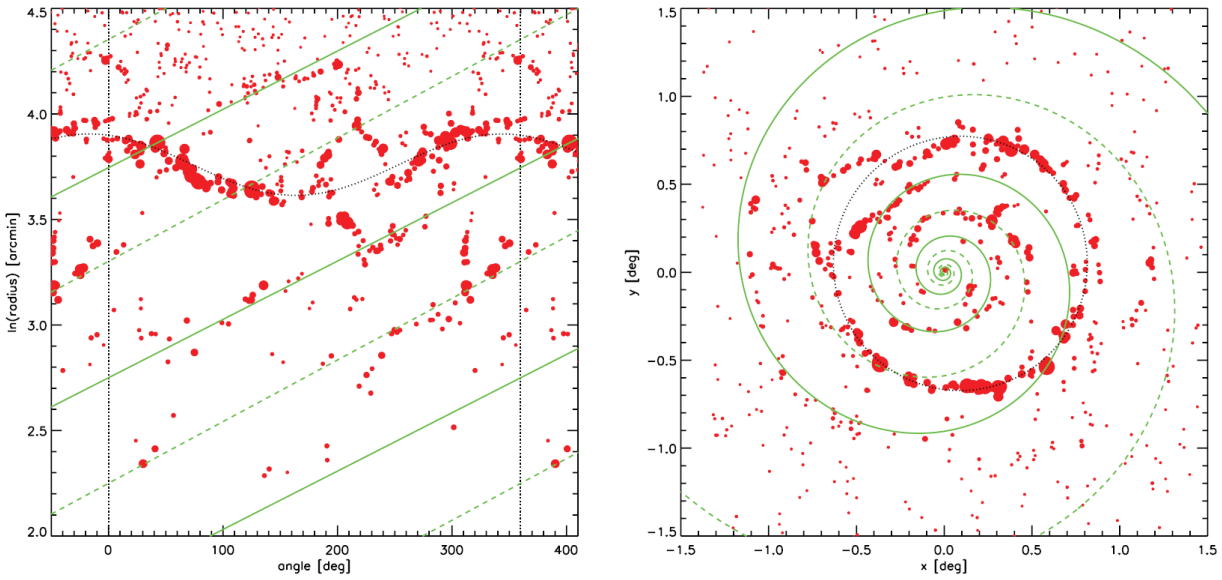


Fig. 16.10: Distribution of point sources detected in all three MIPS bands, taken from Gordon et al. (2006). The left and the right panels are Polar and Cartesian plots, respectively. Overlaid black dotted curve representing an offset circle well fits the point source distribution.

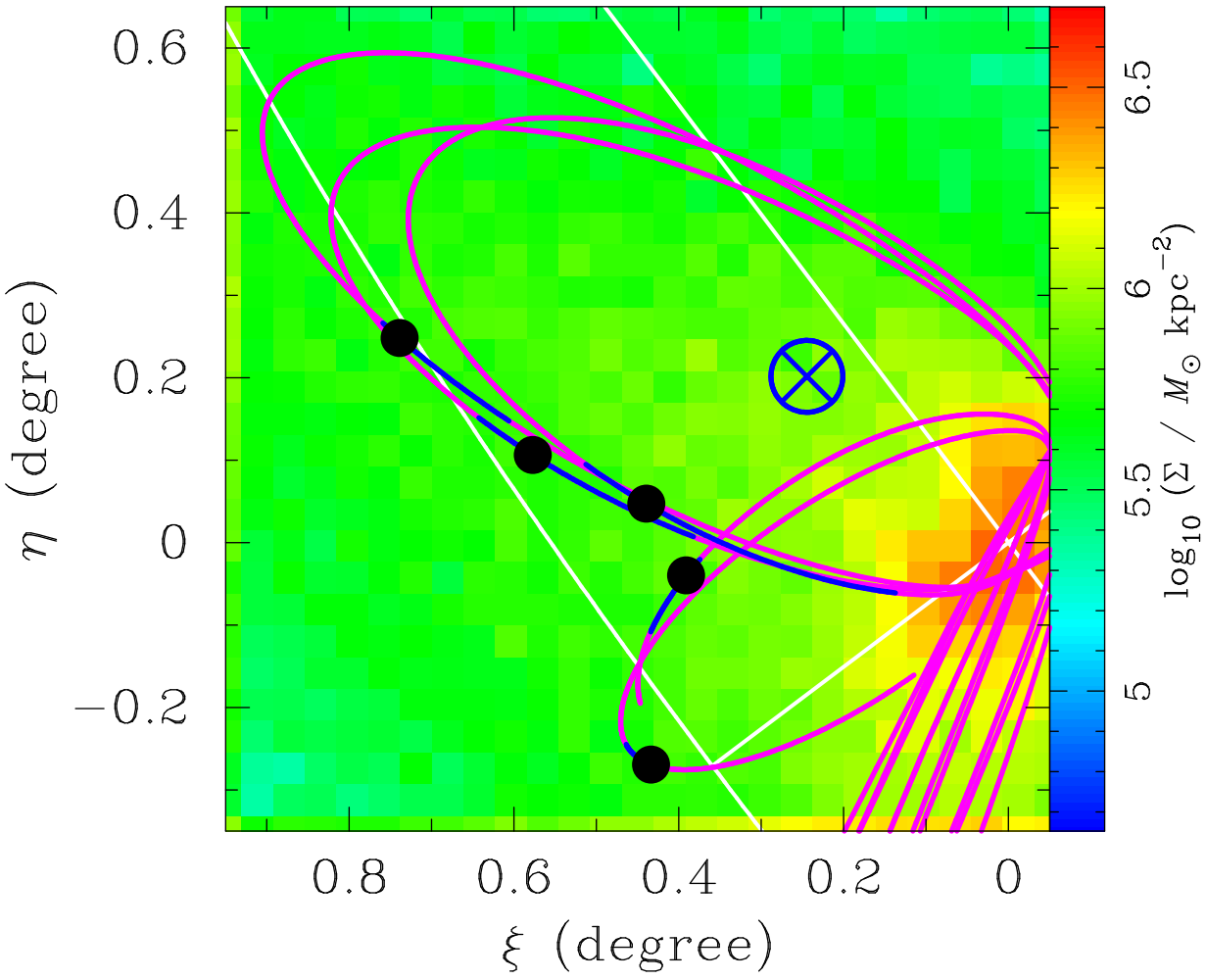


Fig. 16.11: Mass distribution (column density) maps of the debris of the dwarf galaxy in standard coordinates centered on M31 with a stellar clump. A blue cross with a circle shows the position of the clump found by Davidge (2012) at $(\xi, \eta) = (0°.24, 0°.20)$ with its effective radius of $0°.04$. Remains are identical to the bottom panel of Fig. 15.1.

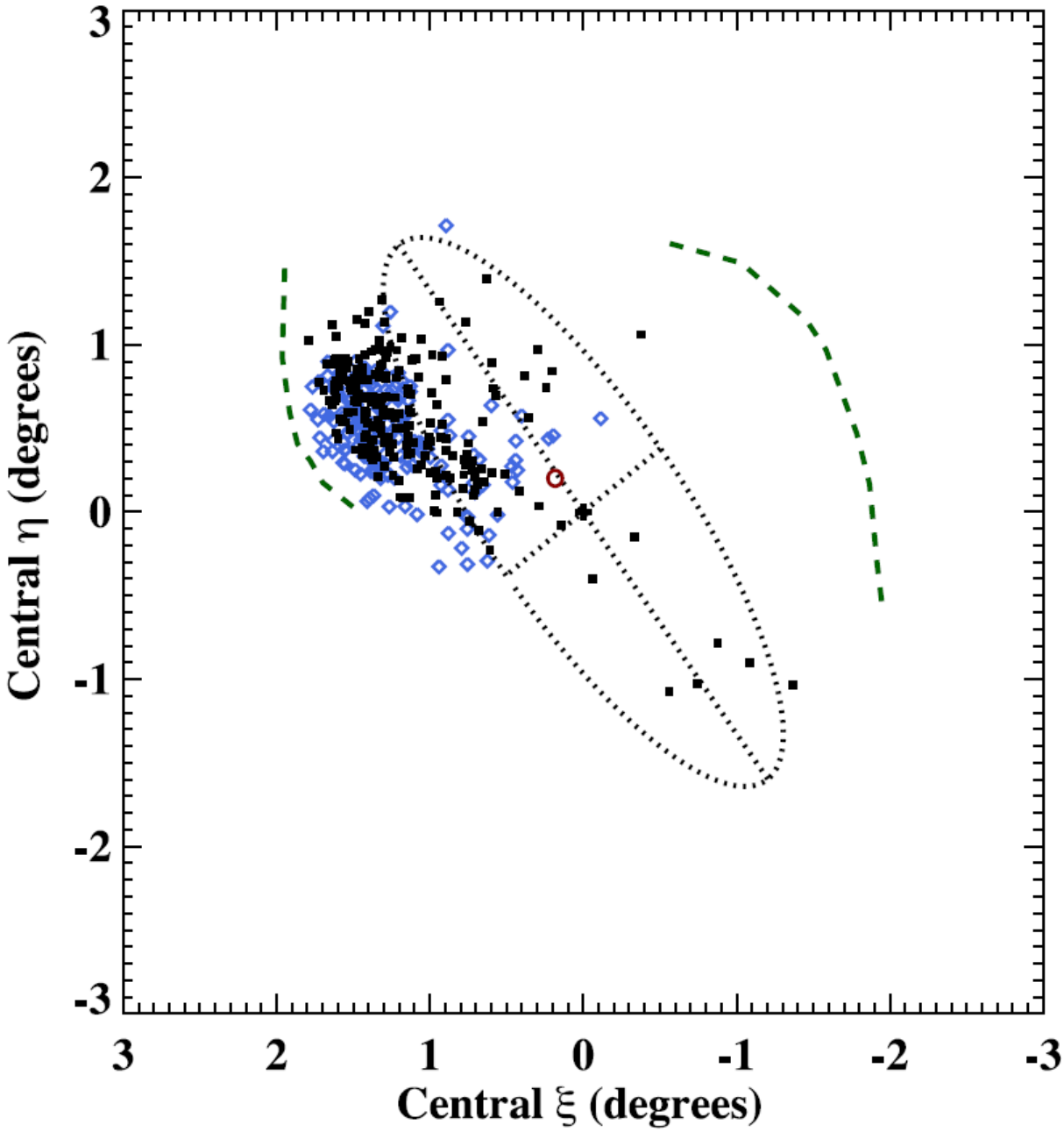


Fig. 16.12: Location of the progenitor's central material, taken from Fardal et al. (2013). Diamonds and squares correspond to stellar and dark matter samples, respectively. A red circle shows the position found by Davidge (2012).

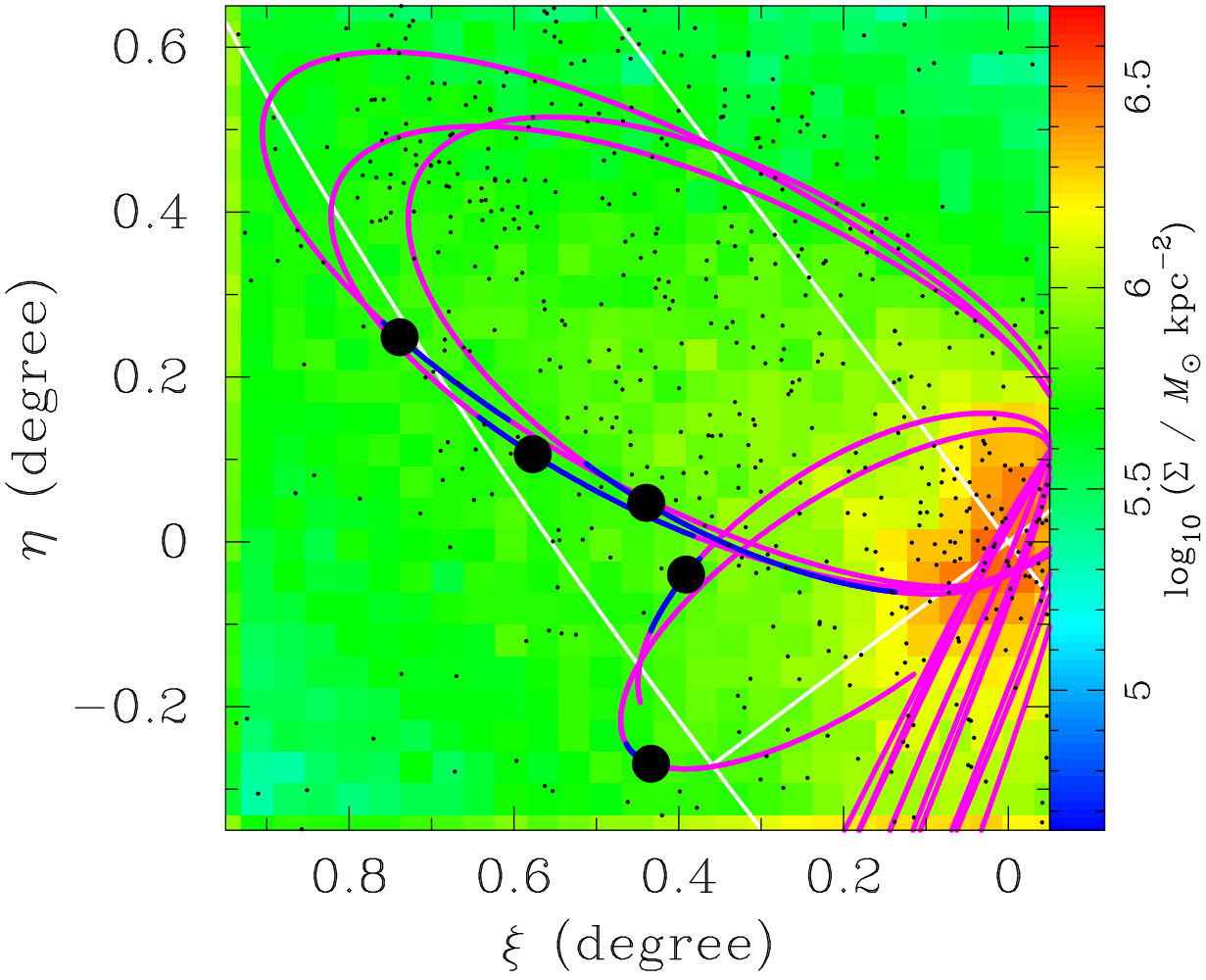


Fig. 16.13: Mass distribution (column density) maps of the debris of the dwarf galaxy in standard coordinates centered on M31 with globular clusters. Small dots show locations of globular clusters listed in a catalogue presented by Peacock et al. (2010). Remains are identical to the bottom panel of Fig. 15.1.

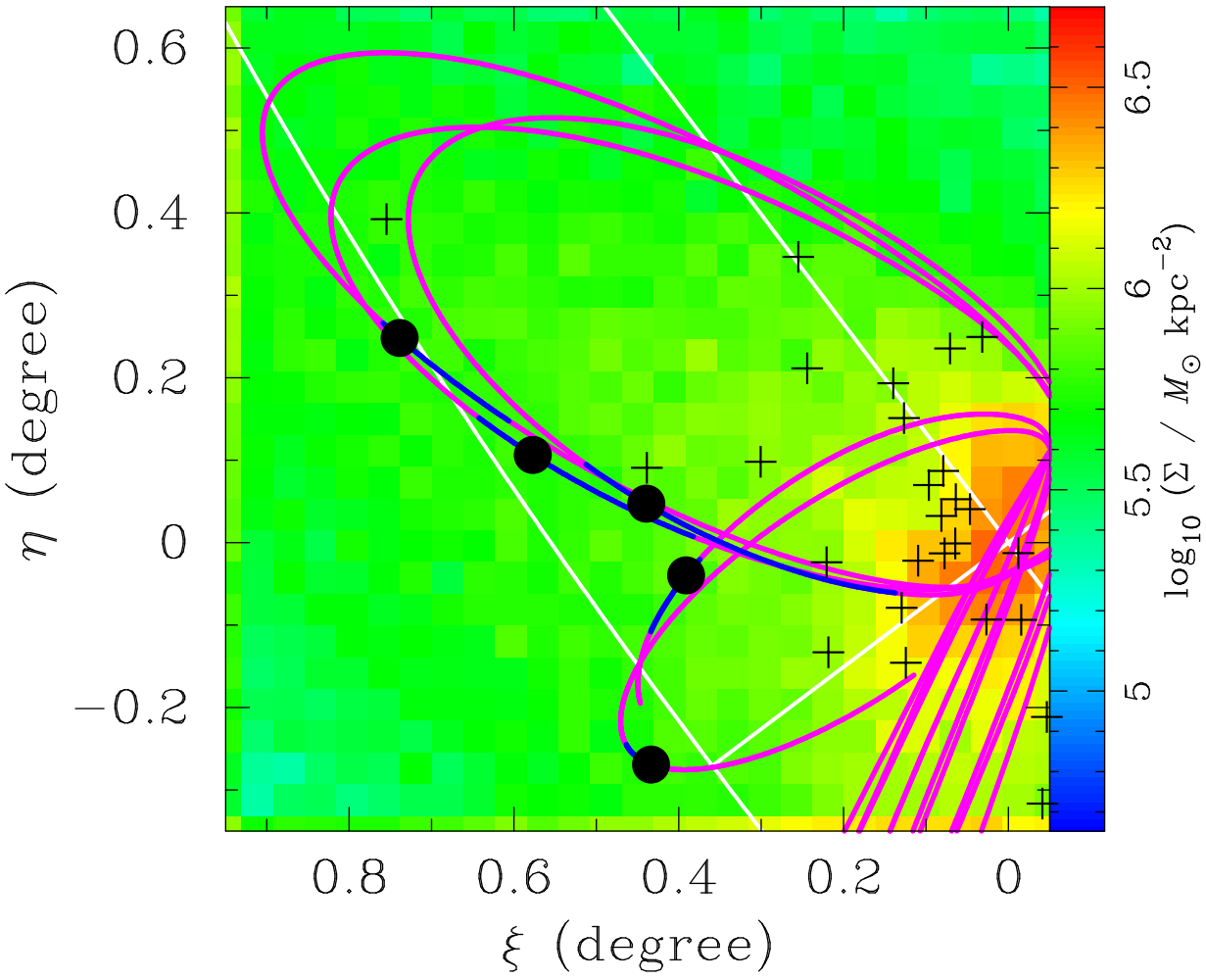


Fig. 16.14: Mass distribution (column density) maps of the debris of the dwarf galaxy in standard coordinates centered on M31 with globular cluster X-ray sources. Crosses represent locations of globular cluster X-ray sources identified by Trudolyubov & Priedhorsky (2004). Remains are identical to the bottom panel of Fig. 15.1.

16.4 Expected Spectrum from the SMBH

Kawaguchi et al. estimated the accretion rate onto the wandering SMBH, whose mass is $M_{\rm BH}$ and velocity is $\boldsymbol{v}_{\rm BH}$, adopting the Hoyle-Lyttleton-Bondi accretion (Hoyle & Lyttleton 1939; Bondi & Hoyle 1944). The mass accretion rate via the Hoyle-Lyttleton-Bondi accretion $\dot{M}$ is described as

$$\dot{M} = 4\pi G^2 \frac{\rho {M_{\rm BH}}^2}{({v_{\rm BH}}^2 + {c_{\rm s}}^2)^{3/2}}, \tag{16.6}$$

where ρ and $c_{\rm s}$ are the mass density and the sound speed of the surrounding inter-stellar medium (ISM), respectively. They assumed that the density profile of the ISM is the same with that of M31 (see Section 2.1), and adopted a constant gas fraction of 0.1 with respect to the whole mass budgets of M31. The Hoyle-Lyttleton radius $R_{\rm HL}$ and the corresponding free-fall time are given as $2GM_{\rm BH}/({v_{\rm BH}}^2 + {c_{\rm s}}^2)$ and $\sqrt{{R_{\rm HL}}^3/(GM_{\rm BH})}$, respectively.

Figure 16.15 shows the time evolution of the above estimated physical quantities. Since the free-fall timescale about 10^5 yr is much shorter than the evolution timescale of the observed structures ($\sim 10^8$ yr),

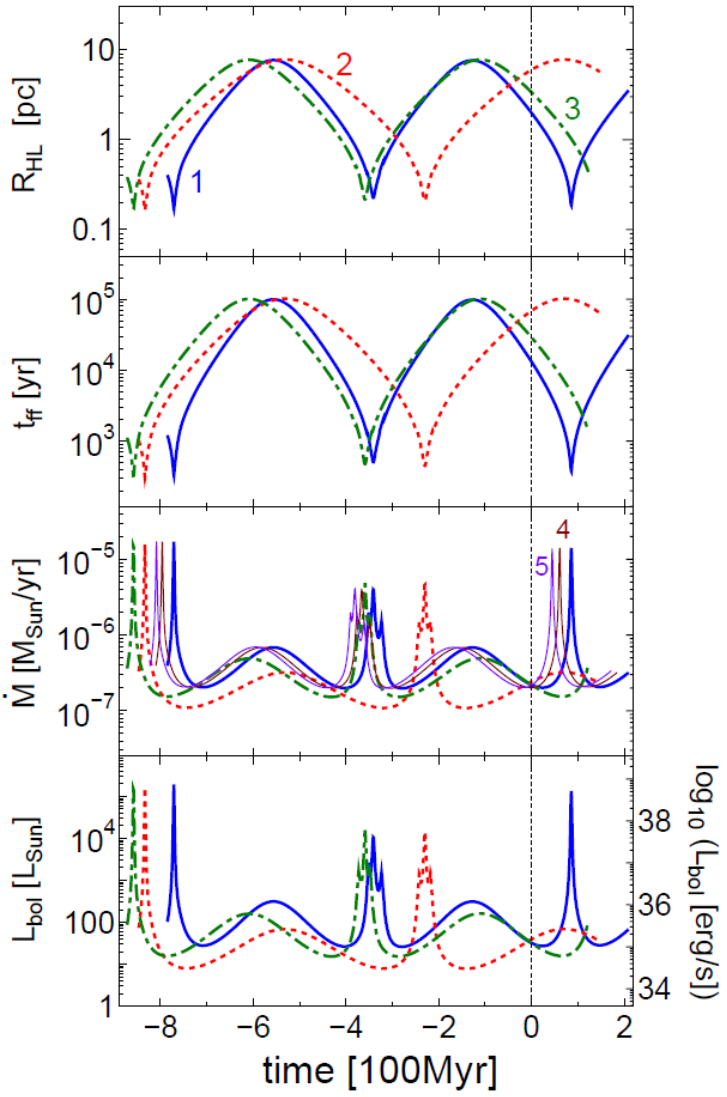


Fig. 16.15: Time evolution of the Hoyle-Lyttleton radius $R_{\rm HL}$, the free-fall timescale at $R_{\rm HL}$, the mass accretion rate $\dot{M}$, and the bolometric luminosity $L_{\rm bol}$ assuming $M_{\rm BH}$ is 10^7 $M_\odot$, taken from Kawaguchi et al. (submitted to ApJ). Each curve corresponds to the SMBH orbit models listed in Tab. 15.1.

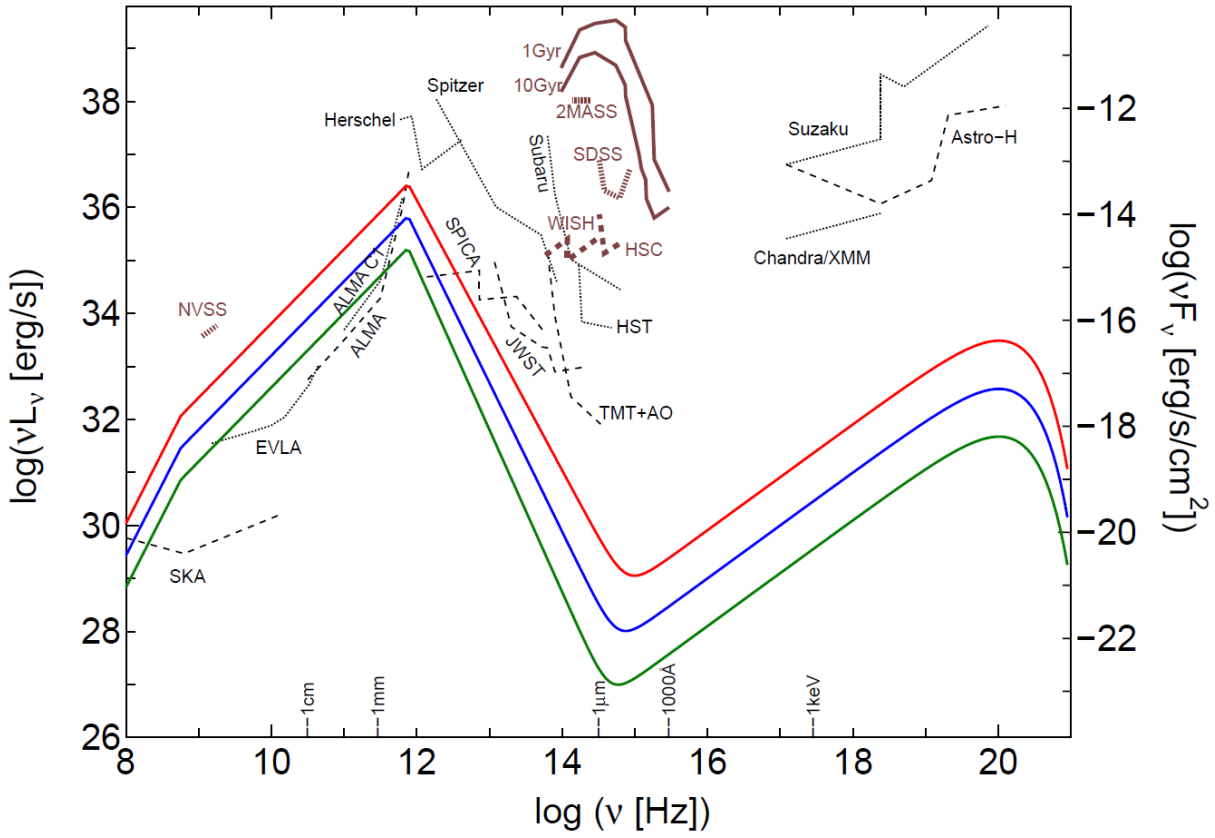


Fig. 16.16: Expected SED of the wandering SMBH, taken from Kawaguchi et al. (submitted to ApJ). Three solid curves show the SED for three different mass of SMBH assumed; green, blue, and red curves represent 5×10^6, 10^7, and 2×10^7 $M_\odot$, respectively. Black lines indicate the detection limits (10σ in 10^4 s integration) of the existing and planned facilities (dotted and dashed lines, respectively).

the post-process treatment is adequate to evaluate the radiation from the orbiting SMBH. The time variation of the mass accretion rate shows several peaks. The high ISM density near the M31 center increases $\dot{M}$ near the periapsis of the orbit (about 800, 300 Myr ago and 100 Myr after). On the other hand, the slow velocity near the apoapsis also increases $\dot{M}$ around 600 and 100 Myr ago. To estimate the radiation from the accretion flow to the SMBH, Kawaguchi et al. adopted the advection-dominated accretion flow (ADAF) model as an accretion model for low mass accretion rates. Since the expected $\dot{M}$ shown in Fig. 16.15 is much smaller than the Eddington rate ($\sim 10 L_{\rm Edd}/c^2 \approx 0.2\ M_\odot\ {\rm yr}^{-1}$ for an SMBH whose mass is $10^7\ M_\odot$), the ADAF model is an appropriate model. They estimate the bolometric luminosity $L_{\rm bol}$ (the bottom panel of Fig. 16.15) by multiplying the radiative efficiency (in other words, the conversion efficiency from the infalling material to the emergent radiation) of $\dot{M}/(L_{\rm Edd}/c^2)$. At the present day, $L_{\rm bol}$ is a few tens $L_\odot$.

Figure 16.16 shows the broadband spectral energy distribution (SED) of the accretion flow based on the current $\dot{M}$ of SMBH ID 1 and a simplified ADAF model by Mahadevan (1997). Self-absorbed synchrotron emission dominates below $\sim 10^{12}$ Hz. The figure clearly shows that radio observations can detect the emission from the wandering SMBH.

Beck et al. (1978) first detected polarized radio emission from M31. A large part of the radio continuum emission is considered to be synchrotron emission due to cosmic-ray electrons and the inter-stellar magnetic field. Figure 16.17 compares the locations of the polarized radio sources observed by Han et al. (1998) and Gießübel et al. (2013) with the predicted positions of the wandering SMBH. It shows that some known radio sources locate in the observational field proposed in this study. Especially, a radio source named 37W207B (Walterbos et al. 1985) locates near the predicted position of SMBH ID 4 (Tab. 15.1). This discovery

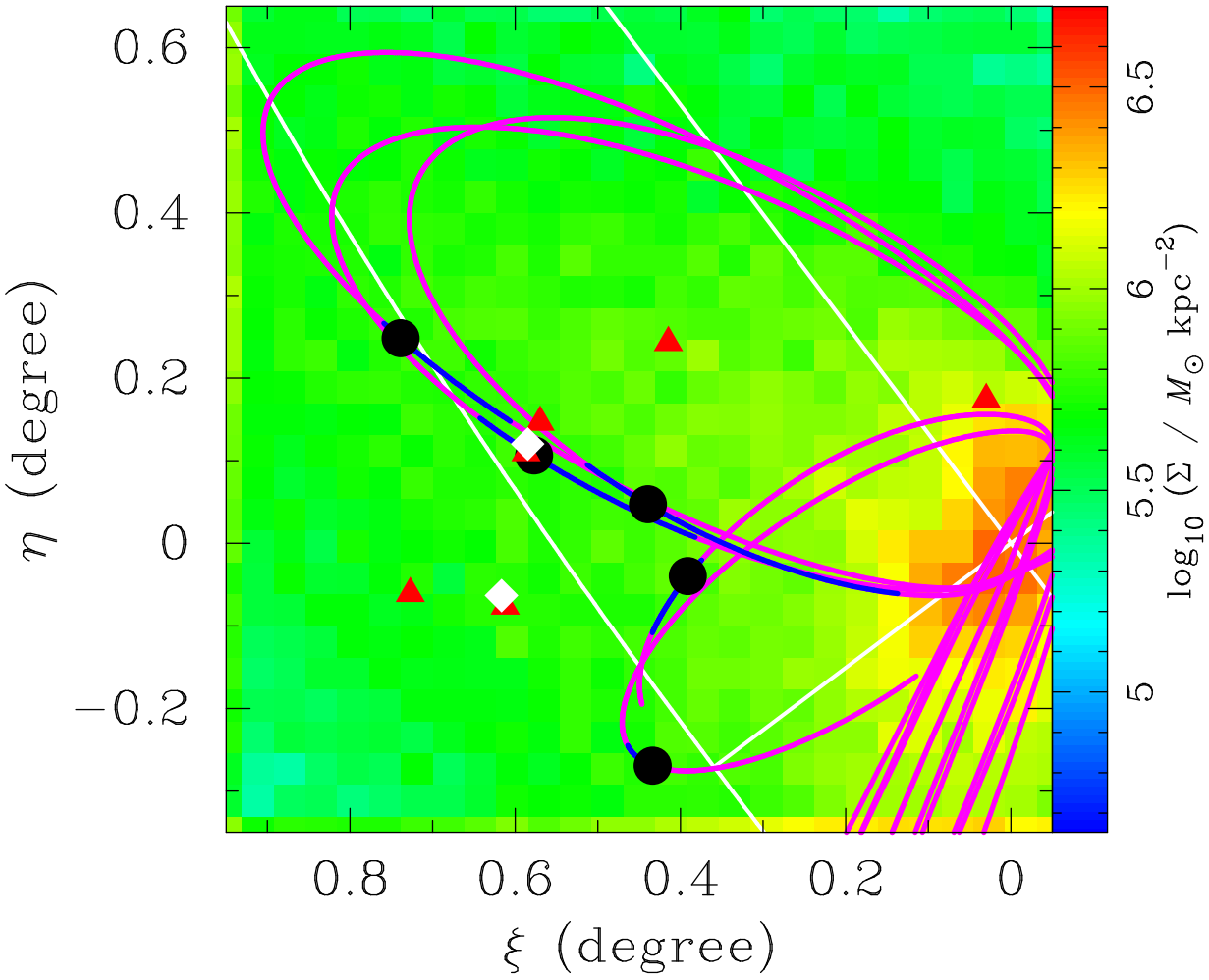


Fig. 16.17: Mass distribution (column density) maps of the debris of the dwarf galaxy in standard coordinates centered on M31 with polarized radio sources. Red triangles and white diamonds correspond the locations of the polarized radio sources observed by Han et al. (1998) and Gießübel et al. (2013), respectively. Remains are identical to the bottom panel of Fig. 15.1.

suggests that we have already detected signals from the wandering SMBHs but do not notice the fact.

Chapter 17 Conclusion

We have investigated an SMBH in an ancient satellite galaxy, whose current position is consistent with the observed structures in the M31 halo. The infalling orbit of the satellite has been first established by conducting numerous low-resolution parameter trials on a high-performance GPU cluster. These preliminary investigations reduce the possible parameter space for the orbit to a manageable size. Next, the orbital evolution of the SMBH has been directly calculated in high-resolution N-body simulations. The hermitage of the SMBH is localized to the northeast stellar shell over an area of $\sim 0^\circ.6 \times 0^\circ.7$. The observational field of $1^\circ \times 1^\circ$ is sufficiently wide to contain all possible positions of the wandering SMBH. Furthermore, we find signatures of the relationships between this particular minor merger and the recently identified thin disk plane formed by M31's satellite galaxies. This discovery may assist in identifying a group of wandering SMBHs in the halo of M31.

Appendix A King Model

In this chapter, we note details of the King model. The distribution function of the King model (or Lowered Maxwellian) is given as

$$f(\mathcal{E}) = \begin{cases} \frac{\rho_1}{(2\pi\sigma^2)^{3/2}} \left(e^{\mathcal{E}/\sigma^2} - 1 \right) & \text{for } \mathcal{E} > 0, \\ 0 & \text{for } \mathcal{E} \leq 0, \end{cases} \tag{A.1}$$

where $\mathcal{E}$ is defined as $\Psi - v^2/2 = -E + \Phi_0$ using a constant Φ_0. Other quantities ρ_1 and σ are parameters that characterize the model.

Integrating (A.1) over the velocity space gives an expression of the mass density ρ as

$$\begin{aligned} \rho &= 4\pi \int_0^\infty dv\, v^2 f(\mathcal{E}) = \frac{4\pi\rho_1}{(2\pi\sigma^2)^{3/2}} \int_0^{\sqrt{2\Psi}} dv\, v^2 \left(e^{\mathcal{E}/\sigma^2} - 1 \right) \\ &= \rho_1 \left\{ e^W \mathrm{erf}\left(\sqrt{W} \right) - \sqrt{\frac{4}{\pi} W} \left(1 + \frac{2}{3} W \right) \right\}, \end{aligned} \tag{A.2}$$

where W is defined as Ψ/σ^2, and the error function is given as

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z dt\, e^{-t^2}. \tag{A.3}$$

Since the mass density profile is given as a functional of W, we must derive W. The Poisson equation for W is expressed as

$$\frac{d}{dr} \left(r^2 \frac{dW}{dr} \right) = -\frac{4\pi G \rho_1 r^2}{\sigma^2} \left\{ e^W \mathrm{erf}\left(\sqrt{W} \right) - \sqrt{\frac{4}{\pi} W} \left(1 + \frac{2}{3} W \right) \right\}. \tag{A.4}$$

To solve the differential equation (A.4), two boundary conditions are necessary. The conditions at the center are $dW/dr = 0$ and $W(r = 0) = W_0$. The dimensionless King parameter at the center W_0 controls the degree of mass concentration.

Figure A.1 shows the volume density profiles for various values of W_0. The vertical and horizontal axes are normalized by the central density ρ_0 and the core radius defined as

$$r_0 = \sqrt{\frac{9\sigma^2}{4\pi G \rho_0}}, \tag{A.5}$$

respectively. Every density profile has a critical radius where the mass density vanishes, named tidal radius r_t. Figures A.3 and A.4 show the profiles of the column density Σ and the radial velocity dispersion σ_r, respectively. The vertical axes of the both figures are normalized by the plotted quantities at the center (the central column density Σ_0 or the central velocity dispersion $\sigma_{\mathrm{r}0}$ in Figs. A.2 or A.3, respectively). The concentration parameter c is defined as

$$c \equiv \log \left(\frac{r_\mathrm{t}}{r_0} \right). \tag{A.6}$$

The relation between W_0 and c is shown in Fig. A.4.

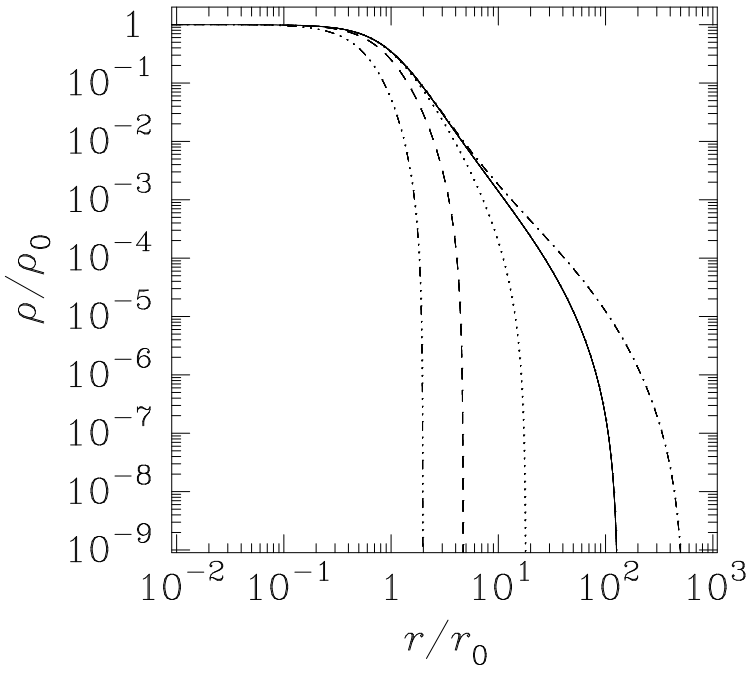


Fig. A.1: Volume density profile of the King model. From left to right, $W_0 = 1$, 3, 6, 9, and 12.

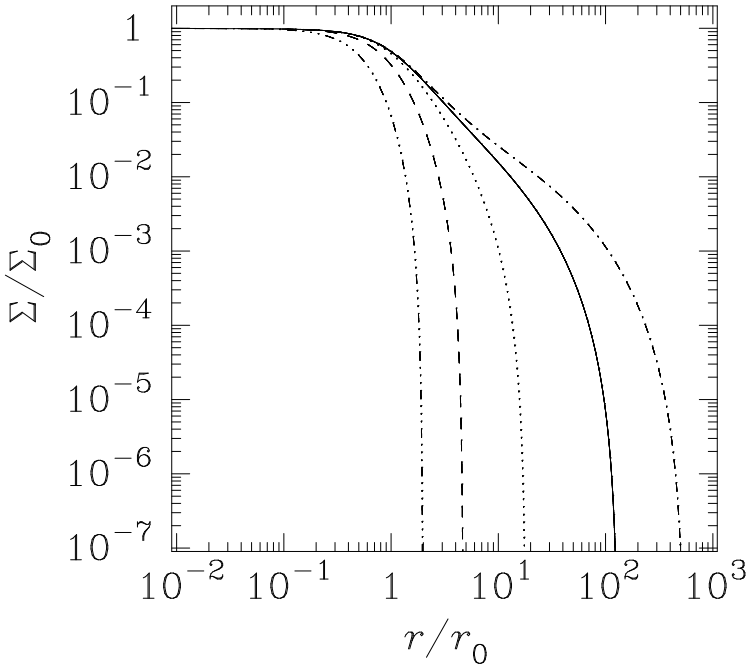


Fig. A.2: Column density profile of the King model. From left to right, $W_0 = 1$, 3, 6, 9, and 12.

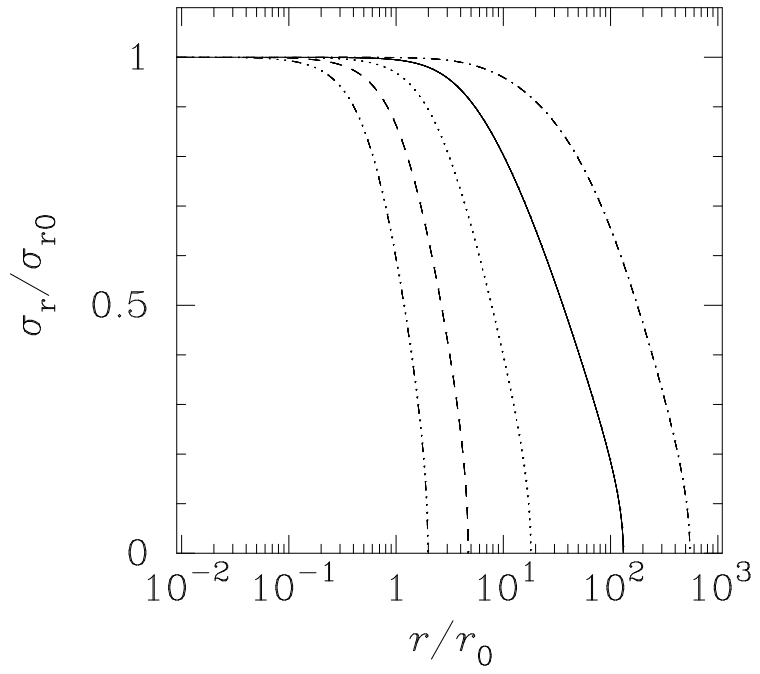


Fig. A.3: Radial velocity dispersion profile of the King model. From left to right, $W_0 = 1$, 3, 6, 9, and 12.

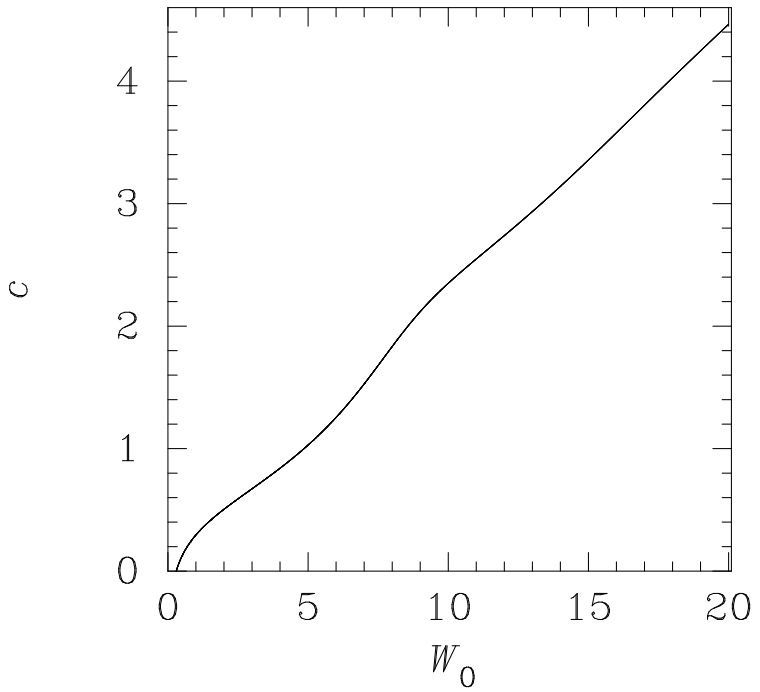


Fig. A.4: Concentration parameter c as a function of the dimensionless King parameter at the center W_0.

Appendix B Computational Cost of Inverse Square Root on GPUs

Here, we shortly discuss estimation of floating-point operation count per interaction required to calculate gravitational acceleration. Historically, a variety number of operation counts have been assumed to evaluate the performance of collisionless N-body simulations. The floating-point operations count of 30 is assumed to evaluate the performance of GRAPE series for collisionless systems (Okumura et al. 1993; Kawai et al. 2000), 20 is GPGPU code by Nyland et al. (2007), and 38 is the most frequently assumed value using various architectures (Kawai et al. 1999; Hamada & Iitaka 2007; Nitadori & Makino 2008; Hamada et al. 2009; Hamada & Nitadori 2010).

Different estimations about computational cost of the inverse square root are the origin of the above difference (e.g. counting of 20 floating-point operations as an inverse square root operation leads to the assumption of 38 floating-point operations per interaction). Since 8 clock cycles are need to perform `rsqrtf()` for GPUs of compute capability 2.0 (Nvidia 2012) whereas addition or multiplication need only 1 clock cycle to perform, the computational cost of `rsqrtf()` corresponds to 8 floating-point operations. Therefore, 26 floating-point operations per interaction, assumed value in this work, is the most plausible estimation for GPUs of compute capability 2.0.

Acknowledgments

I am particularly grateful to my supervisor Masao Mori for useful discussion, valuable advice, and strong supports throughout my doctoral program in physics. I also appreciate Daisuke Takahashi, my supervisor in computer science under the dual degree program, for useful suggestions and fruitful discussion to develop my codes for numerical simulations. I have greatly benefited from all the other collaborators (Toshihiro Kawaguchi, Go Ogiya, Takanobu Kirihara, Yuriko Saito, Naohito Nakasato, and R. Michael Rich) for valuable discussion, improving my work and providing excellent materials. I thank Mike J. Irwin for allowing the use of their observational data. I also thank Andreas Koch for instructing the usage of the KMM algorithm and discussion related to observations. I would like to express my gratitude to Masashi Chiba and Mikito Tanaka for discussion about the current and the future observations around M31. I have greatly benefited from discussions with Annette M. N. Ferguson and her collaborators during a stay at University of Edinburgh. I also thank Eric Tittley and Olga Degtyareva for their kindness when I stayed at University of Edinburgh.

I have had the support and encouragement of Masayuki Umemura, Kohji Yoshikawa, Takashi Okamoto, Nozomu Kawakatsu, Takuya Akahori, Tomoaki Ishiyama, Ataru Tanikawa, Daisuke Namekata, Kenji Hasegawa, Hidenobu Yajima, and other present and/or past members of astrophysics division. Special thanks also to Taisuke Boku, Daichi Mukunoki, Tetsuya Odajima, Norihisa Fujita, and members of high performance computing system laboratory for their encouragements and providing detailed information on recent high performance architectures. I am very thankful to Masayuki Umemura, Daisuke Takahashi, Masashi Chiba, Naomasa Nakai, and Masao Mori for their constructive comments that have improved my study and doctoral thesis.

The numerical simulations were performed on the FIRST and HA-PACS, at the Center for Computational Sciences, University of Tsukuba. I am grateful to the FIRST and HA-PACS project team for their generous technical assistance. A part of analysis was performed on the hexa cluster, andromeda, and mw.

References

[1] Ashman, K. M., Bird, C. M., & Zepf, S. E. 1994, AJ, 108, 2348

[2] Bahl, H., & Baumgardt, H. 2013, ArXiv e-prints, arXiv:1312.3629

[3] Barnes, D. G., & Fluke, C. J. 2008, New A, 13, 599

[4] Barnes, D. G., Fluke, C. J., Bourke, P. D., & Parry, O. T. 2006, PASA, 23, 82

[5] Barnes, J., & Hut, P. 1986, Nature, 324, 446

[6] Barth, A. J., Greene, J. E., & Ho, L. C. 2005, ApJ, 619, L151

[7] Beck, R., Berkhuijsen, E. M., & Wielebinski, R. 1978, A&A, 68, L27

[8] Bédorf, J., Gaburov, E., & Portegies Zwart, S. 2012, Journal of Computational Physics, 231, 2825

[9] Belleman, R. G., Bédorf, J., & Portegies Zwart, S. F. 2008, New A, 13, 103

[10] Bellovary, J. M., Governato, F., Quinn, T. R., et al. 2010, ApJ, 721, L148

[11] Belokurov, V., Zucker, D. B., Evans, N. W., et al. 2006, ApJ, 642, L137

[12] Bender, R., Kormendy, J., Bower, G., et al. 2005, ApJ, 631, 280

[13] Binney, J., & Tremaine, S. 2008, Galactic Dynamics: Second Edition (Princeton University Press)

[14] Bondi, H., & Hoyle, F. 1944, MNRAS, 104, 273

[15] Boylan-Kolchin, M., Springel, V., White, S. D. M., Jenkins, A., & Lemson, G. 2009, MNRAS, 398, 1150

[16] Brasseur, C. M., Martin, N. F., Macciò, A. V., Rix, H.-W., & Kang, X. 2011, ApJ, 743, 179

[17] Brunthaler, A., Reid, M. J., Falcke, H., Greenhill, L. J., & Henkel, C. 2005, Science, 307, 1440

[18] Bullock, J. S., & Johnston, K. V. 2005, ApJ, 635, 931

[19] Chiba, M., Minezaki, T., Kashikawa, N., Kataza, H., & Inoue, K. T. 2005, ApJ, 627, 53

[20] Collins, M. L. M., Chapman, S. C., Rich, R. M., et al. 2013, ApJ, 768, 172

[21] Conn, A. R., Lewis, G. F., Ibata, R. A., et al. 2013, ApJ, 766, 120

[22] Davidge, T. J. 2012, ApJ, 749, L7

[23] de Blok, W. J. G., van der Hulst, J. M., & Bothun, G. D. 1995, MNRAS, 274, 235

[24] de Vaucouleurs, G., de Vaucouleurs, A., Corwin, Jr., H. G., et al. 1991, Third Reference Catalogue of Bright Galaxies. Volume I: Explanations and references. Volume II: Data for galaxies between 0^h and 12^h. Volume III: Data for galaxies between 12^h and 24^h.

[25] Dekel, A., & Woo, J. 2003, MNRAS, 344, 1131

[26] Devereux, N. A., Price, R., Wells, L. A., & Duric, N. 1994, AJ, 108, 1667

[27] Diemand, J., Kuhlen, M., Madau, P., et al. 2008, Nature, 454, 735

[28] Eskridge, P. B. 1988a, AJ, 96, 1352

[29] —. 1988b, AJ, 95, 1706

[30] Falcón-Barroso, J., van de Ven, G., Peletier, R. F., et al. 2011, MNRAS, 417, 1787

[31] Fardal, M. A., Babul, A., Geehan, J. J., & Guhathakurta, P. 2006, MNRAS, 366, 1012

[32] Fardal, M. A., Babul, A., Guhathakurta, P., Gilbert, K. M., & Dodge, C. 2008, ApJ, 682, L33

[33] Fardal, M. A., Guhathakurta, P., Babul, A., & McConnachie, A. W. 2007, MNRAS, 380, 15

[34] Fardal, M. A., Guhathakurta, P., Gilbert, K. M., et al. 2012, MNRAS, 423, 3134

[35] Fardal, M. A., Weinberg, M. D., Babul, A., et al. 2013, MNRAS, 434, 2779

[36] Farrell, S. A., Webb, N. A., Barret, D., Godet, O., & Rodrigues, J. M. 2009, Nature, 460, 73

[37] Ferguson, A., Chapman, S., Ibata, R., et al. 2004, ArXiv Astrophysics e-prints, astro-ph/0408058

[38] Ferguson, A. M. N., Irwin, M. J., Ibata, R. A., Lewis, G. F., & Tanvir, N. R. 2002, AJ, 124, 1452

[39] Font, A. S., Johnston, K. V., Guhathakurta, P., Majewski, S. R., & Rich, R. M. 2006, AJ, 131, 1436

[40] Fouquet, S., Hammer, F., Yang, Y., Puech, M., & Flores, H. 2012, MNRAS, 427, 1769

[41] Fukushige, T., Makino, J., & Kawai, A. 2005, PASJ, 57, 1009

[42] Gaburov, E., Bédorf, J., & Portegies Zwart, S. 2010, Procedia Computer Science, volume 1, p. 1119-1127, 1, 1119

[43] Gaburov, E., Harfst, S., & Zwart, S. P. 2009, New A, 14, 630

[44] Geehan, J. J., Fardal, M. A., Babul, A., & Guhathakurta, P. 2006, MNRAS, 366, 996

[45] Gießübel, R., Heald, G., Beck, R., & Arshakian, T. G. 2013, A&A, 559, A27

[46] Gilbert, K. M., Fardal, M., Kalirai, J. S., et al. 2007, ApJ, 668, 245

[47] Gilbert, K. M., Guhathakurta, P., Kollipara, P., et al. 2009, ApJ, 705, 1275

[48] Godet, O., Plazolles, B., Kawaguchi, T., et al. 2012, ApJ, 752, 34

[49] Goerdt, T., & Burkert, A. 2013, ArXiv e-prints, arXiv:1307.2102

[50] Gordon, K. D., Bailin, J., Engelbracht, C. W., et al. 2006, ApJ, 638, L87

[51] Guhathakurta, P., Rich, R. M., Reitzel, D. B., et al. 2006, AJ, 131, 2497

[52] Haas, M., Lemke, D., Stickel, M., et al. 1998, A&A, 338, L33

[53] Habing, H. J., Miley, G., Young, E., et al. 1984, ApJ, 278, L59

[54] Hamada, T., & Iitaka, T. 2007, ArXiv Astrophysics e-prints, astro-ph/0703100

[55] Hamada, T., Narumi, T., Yokota, R., et al. 2009, in Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC '09 (New York, NY, USA: ACM), 62:1–62:12

[56] Hamada, T., & Nitadori, K. 2010, in Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC '10 (Washington, DC, USA: IEEE Computer Society), 1–9

[57] Hammer, F., Yang, Y., Fouquet, S., et al. 2013, MNRAS, 431, 3543

[58] Hammer, F., Yang, Y. B., Wang, J. L., et al. 2010, ApJ, 725, 542

[59] Han, J. L., Beck, R., & Berkhuijsen, E. M. 1998, A&A, 335, 1117

[60] Harfst, S., Gualandris, A., Merritt, D., et al. 2007, New A, 12, 357

[61] Heller, A. B., & Brosch, N. 2001, MNRAS, 327, 80

[62] Hernquist, L. 1990, ApJ, 356, 359

[63] Hockney, R. W., & Eastwood, J. W. 1988, Computer simulation using particles

[64] Hoyle, F., & Lyttleton, R. A. 1939, Proceedings of the Cambridge Philosophical Society, 34, 405

[65] Ibata, R., Chapman, S., Ferguson, A. M. N., et al. 2004, MNRAS, 351, 117

[66] —. 2005, ApJ, 634, 287

[67] Ibata, R., Irwin, M., Lewis, G., Ferguson, A. M. N., & Tanvir, N. 2001, Nature, 412, 49

[68] Ibata, R., Martin, N. F., Irwin, M., et al. 2007, ApJ, 671, 1591

[69] Ibata, R. A., Lewis, G. F., Conn, A. R., et al. 2013, Nature, 493, 62

[70] Ichikawa, S., Wakamatsu, K., & Okamura, S. 1986, ApJS, 60, 475

[71] Inoue, K. T., Rashkov, V., Silk, J., & Madau, P. 2013, MNRAS, 435, 2092

[72] Irwin, M., & Hatzidimitriou, D. 1995, MNRAS, 277, 1354

[73] Irwin, M. J., Ferguson, A. M. N., Ibata, R. A., Lewis, G. F., & Tanvir, N. R. 2005, ApJ, 628, L105

[74] Kalirai, J. S., Guhathakurta, P., Gilbert, K. M., et al. 2006a, ApJ, 641, 268

[75] Kalirai, J. S., Gilbert, K. M., Guhathakurta, P., et al. 2006b, ApJ, 648, 389

[76] Kalirai, J. S., Beaton, R. L., Geha, M. C., et al. 2010, ApJ, 711, 671

[77] Kawaguchi, T., Saito, Y., Miki, Y., & Mori, M. submitted to ApJ

[78] Kawai, A., Fukushige, T., & Makino, J. 1999, in Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM), Supercomputing '99 (New York, NY, USA: ACM)

[79] Kawai, A., Fukushige, T., Makino, J., & Taiji, M. 2000, PASJ, 52, 659

[80] Kent, S. M. 1984, ApJS, 56, 105

[81] Khronos. 2011, The OpenCL Specification Version 1.2

[82] —. 2013, The OpenCL Specification Version 2.0

[83] Kirihara, T., Miki, Y., & Mori, M. in preparation

[84] Koch, A., Rich, R. M., Reitzel, D. B., et al. 2008, ApJ, 689, 958

[85] Koleva, M., de Rijcke, S., Prugniel, P., Zeilinger, W. W., & Michielsen, D. 2009a, MNRAS, 396, 2133

[86] Koleva, M., Prugniel, P., De Rijcke, S., Zeilinger, W. W., & Michielsen, D. 2009b, Astronomische Nachrichten, 330, 960

[87] Kormendy, J., & Richstone, D. 1995, ARA&A, 33, 581

[88] Kourkchi, E., Khosroshahi, H. G., Carter, D., & Mobasher, B. 2012a, MNRAS, 420, 2835

[89] Kourkchi, E., Khosroshahi, H. G., Carter, D., et al. 2012b, MNRAS, 420, 2819

[90] Lauer, T. R., Faber, S. M., Groth, E. J., et al. 1993, AJ, 106, 1436

[91] Lewis, G. F., Braun, R., McConnachie, A. W., et al. 2013, ApJ, 763, 4

[92] Mackey, A. D., Huxor, A. P., Ferguson, A. M. N., et al. 2010, ApJ, 717, L11

[93] Magorrian, J., Tremaine, S., Richstone, D., et al. 1998, AJ, 115, 2285

[94] Mahadevan, R. 1997, ApJ, 477, 585

[95] Makarova, L. 1999, A&AS, 139, 491

[96] Makino, J., Fukushige, T., Koga, M., & Namura, K. 2003, PASJ, 55, 1163

[97] Marconi, A., & Hunt, L. K. 2003, ApJ, 589, L21

[98] Martin, N. F., Ibata, R. A., McConnachie, A. W., et al. 2013, ApJ, 776, 80

[99] Matthews, L. D., van Driel, W., & Gallagher, III, J. S. 1998, AJ, 116, 1169

[100] McConnachie, A. W., & Irwin, M. J. 2006a, MNRAS, 365, 1263

[101] —. 2006b, MNRAS, 365, 902

[102] McConnachie, A. W., Irwin, M. J., Ferguson, A. M. N., et al. 2004, MNRAS, 350, 243

[103] McConnachie, A. W., Irwin, M. J., Ibata, R. A., et al. 2003, MNRAS, 343, 1335

[104] —. 2009, Nature, 461, 66

[105] Merritt, D. 2001, ApJ, 556, 245

[106] —. 2005, ApJ, 628, 673

[107] Merritt, D., Berczik, P., & Laun, F. 2007, AJ, 133, 553

[108] Minezaki, T., Chiba, M., Kashikawa, N., Inoue, K. T., & Kataza, H. 2009, ApJ, 697, 610

[109] Mori, M., & Rich, R. M. 2008, ApJ, 674, L77

[110] Mori, M., Yoshii, Y., & Nomoto, K. 1999, ApJ, 511, 585

[111] Mori, M., Yoshii, Y., Tsujimoto, T., & Nomoto, K. 1997, ApJ, 478, L21

[112] Nakasato, N. 2012, Journal of Computational Science, 3, 132 , Scientific Computation Methods and Applications

[113] Navarro, J. F., Frenk, C. S., & White, S. D. M. 1996, ApJ, 462, 563

[114] —. 1997, ApJ, 490, 493

[115] Nitadori, K., & Makino, J. 2008, New A, 13, 498

[116] Nvidia. 2012, NVIDIA CUDA C Programming Guide Version 4.2

[117] —. 2013, CUDA C Programming Guide Version 5.5

[118] Nyland, L., Harris, M., & Prins, J. 2007, Fast N-Body Simulation with CUDA

[119] Ogiya, G., Mori, M., Miki, Y., Boku, T., & Nakasato, N. 2013, Journal of Physics Conference Series, 454, 012014

[120] Okumura, S. K., Makino, J., Ebisuzaki, T., et al. 1993, PASJ, 45, 329

[121] OpenACC. 2011, The OpenACC Application Programming Interface Version 1.0

[122] —. 2013, The OpenACC Application Programming Interface Version 2.0

[123] Peacock, M. B., Maccarone, T. J., Kundu, A., & Zepf, S. E. 2010, MNRAS, 407, 2611

[124] Portegies Zwart, S. F., Belleman, R. G., & Geldof, P. M. 2007, New A, 12, 641

[125] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2007, Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd edn. (Cambridge University Press)

[126] Rice, W. 1993, AJ, 105, 67

[127] Richardson, J. C., Irwin, M. J., McConnachie, A. W., et al. 2011, ApJ, 732, 76

[128] Sadoun, R., Mohayaee, R., & Colin, J. 2013, ArXiv e-prints, arXiv:1307.5044

[129] Sohn, S. T., Anderson, J., & van der Marel, R. P. 2012, ApJ, 753, 7

[130] Spolaor, M., Proctor, R. N., Forbes, D. A., & Couch, W. J. 2009, ApJ, 691, L138

[131] Tanaka, M., Chiba, M., Komiyama, Y., et al. 2010, ApJ, 708, 1168

[132] Tollerud, E. J., Beaton, R. L., Geha, M. C., et al. 2012, ApJ, 752, 45

[133] Toloba, E., Boselli, A., Cenarro, A. J., et al. 2011, A&A, 526, A114

[134] Toloba, E., Boselli, A., Peletier, R. F., et al. 2012, A&A, 548, A78

[135] Trethewey, D., Chapman, S., Irwin, M., et al. in preparation

[136] Trudolyubov, S., & Priedhorsky, W. 2004, ApJ, 616, 821

[137] van der Kruit, P. C. 1987, A&A, 173, 59

[138] van der Marel, R. P., Fardal, M., Besla, G., et al. 2012, ApJ, 753, 8

[139] Walterbos, R. A. M., Brinks, E., & Shane, W. W. 1985, A&AS, 61, 451

[140] Widrow, L. M. 2000, ApJS, 131, 39

[141] Wiersema, K., Farrell, S. A., Webb, N. A., et al. 2010, ApJ, 721, L102

[142] Wolf, J., Martinez, G. D., Bullock, J. S., et al. 2010, MNRAS, 406, 1220

[143] Woo, J., Courteau, S., & Dekel, A. 2008, MNRAS, 390, 1453

[144] Xiao, T., Barth, A. J., Greene, J. E., et al. 2011, ApJ, 739, 28