The Astrophysical Journal Supplement Series, 231:5 (20pp), 2017 July
https://doi.org/10.3847/1538-4365/aa799c
© 2017. The American Astronomical Society. All rights reserved.

The Splashback Radius of Halos from Particle Dynamics. I. The SPARTA Algorithm

Benedikt Diemer
Institute for Theory and Computation, Harvard-Smithsonian Center for Astrophysics, 60 Garden St.,
Cambridge, MA 02138, USA; benedikt.diemer@cfa.harvard.edu
Received 2017 March 28; revised 2017 June 2; accepted 2017 June 12; published 2017 July 14

Abstract
Motivated by the recent proposal of the splashback radius as a physical boundary of dark-matter halos, we present a parallel computer code for Subhalo and P AR tic le Trajectory Analysis (SPARTA). The code analyzes the orbits of all simulation particles in all host halos, billions of orbits in the case of typical cosmological N-body simulations. Within this general framework, we develop an algorithm that accurately extracts the location of the first apocenter of particles after infall into a halo, or splashback. We define the splashback radius of a halo as the smoothed average of the apocenter radii of individual particles. This definition allows us to reliably measure the splashback radii of 95% of host halos above a resolution limit of 1000 particles. We show that, on average, the splashback radius and mass are converged to better than 5% accuracy with respect to mass resolution, snapshot spacing, and all free parameters of the method.

Key words: cosmology: theory – dark matter – methods: numerical

1. Introduction

According to the widely accepted ΛCDM paradigm, dark matter (DM) collapses into a cosmic web of walls, filaments, and halos (Zel’dovich 1970; Bond et al. 1996). The collapse becomes highly nonlinear, making numerical simulations the predominant tool for studying the formation and structure of DM halos (Klypin & Shandarin 1983; Davis et al. 1985; Efstathiou et al. 1985). The DM component in such simulations is almost always represented by virtual particles, whether the simulations include hydrodynamics or not (e.g., Hockney & Eastwood 1981; Kravtsov et al. 1997; Springel et al. 2001; Springel 2010; Hopkins 2015). As the densities and gravitational accelerations are highest at the centers of halos, a significant part of the computing time is typically spent on the orbits of particles around halo centers.

Despite this computational effort, the orbits of individual halo particles in cosmological volumes are practically never investigated in detail. There are several reasons for neglecting the orbital trajectories, namely that they have no directly observable impact on galaxies and that they make for a large, cumbersome data set that is difficult to interpret. Some effort was invested in understanding how particle orbits are connected to the observed universal density profile of halos, such as through the radial orbit instability (Merritt & Aguilar 1985; Huss et al. 1999; MacMillan et al. 2006; Gajda et al. 2015), but most studies directly investigate integrated quantities such as density and angular momentum profiles. Subhalo orbits have received somewhat more attention, largely because they can be observed as the distribution of satellite galaxies around a large host (Knebe et al. 2004; Kravtsov et al. 2004; Benson 2005; Reed et al. 2005; Kuhlen et al. 2007; Sales et al. 2007; Ludlow et al. 2009; Lovell et al. 2011; Jiang et al. 2015). However, modeling subhalo orbits correctly is difficult, due to physical effects such as dynamical friction (van den Bosch et al. 1999; Boylan-Kolchin et al. 2008) and numerical effects such as over merging (Moore et al. 1996; van den Bosch 2017).

Given that particle orbits in cosmological simulations have not been investigated in detail, they could open a new window into the processes by which halos form and evolve. For example, the splashback radius, $R_{\text{sp}}$, has recently been proposed as a physically motivated definition of the halo boundary (Adhikari et al. 2014; Diemer & Kravtsov 2014; More et al. 2015). Conventionally, halo radii such as $R_{\text{vir}}$ are defined to contain a particular overdensity contrast inspired by the spherical top-hat collapse model (Gunn & Gott 1972), but such definitions do not generally correspond to any feature in the dynamics of particles or in the density profiles (see Diemer et al. 2017, hereafter Paper II, for a more detailed discussion). The splashback radius, however, is directly connected to particle dynamics: it is the radius where particles reach the apocenter of their first orbit after infall. By analogy to the spherical collapse model, this radius is a well-motivated halo boundary, as it separates infalling matter from matter that is orbiting in the halo potential (Fillmore & Goldreich 1984; Bertschinger 1985; Adhikari et al. 2014; Shi 2016). Moreover, the splashback radius has recently been detected observationally in the stacked member galaxy density profiles of massive galaxy clusters (More et al. 2016; Baxter et al. 2017; see also Zu et al. 2016 and Busch & White 2017, as well as the related observational studies of Tully 2015; Adhikari et al. 2016; Patej & Loeb 2016; Umetsu & Diemer 2017).

Despite these successes, some fundamental theoretical and observational challenges remain. First, both the calibration of More et al. (2015) and the observational detections have been based on stacked density profiles from large samples of halos. Instead, we would like to determine $R_{\text{sp}}$ for individual simulated halos, for example in order to measure the halo-to-halo scatter in $R_{\text{sp}}$. Mansfield et al. (2017) successfully measured nonspherical splashback shells in individual halos, but their algorithm demands a relatively high resolution of 50,000 particles per halo and encounters difficulties at the lowest mass accretion rates. Thus, we have yet to measure $R_{\text{sp}}$ for the majority of halos in a cosmological simulation volume. Second, we have yet to understand the relation between the apocenter of particles’ first orbits and the density profile. Particles enter halos with a distribution of energies and angular momenta, presumably causing a distribution of apocenters that we wish to relate to the steepening in the density profile. While Mansfield et al. (2017) confirmed that their splashback shells are consistent with the motion of particles near the shell, we do...
not know what percentile of splashbacks (if any) their measurement corresponds to.

In this work, we present the first computational framework for the systematic analysis of particle orbits in N-body simulations, including cosmological volumes containing a large number of halos. This code is called Subhalo and PARticle Trajectory Analysis (SPARTA) and will be described in detail in a future publication. Here, we focus on those parts of the algorithm that are relevant for measuring the splashback radius. Although SPARTA tracks subhalos as well as particles, we refrain from using subhalo orbits in this investigation because they suffer from dynamical friction and numerical resolution issues. We carefully test the convergence of the algorithm with mass resolution and snapshot spacing, and we show that it is capable of reliably measuring \( R_{\text{sp}} \) in the vast majority of halos with more than 1000 particles. In Paper II, we will present the statistics of the measured splashback radii as a function of mass accretion rate, halo mass, redshift, and cosmology.

We begin by defining the various symbols used throughout the paper in Section 2 and describing our numerical simulations in Section 3. In Section 4, we describe the algorithm to measure splashback radii. We present convergence tests of the algorithm in Section 5. We discuss the physical interpretation of our results in Section 6, and we summarize our conclusions in Section 7.

2. Definitions

Throughout the paper, we assume a flat \( \Lambda \)CDM cosmology with a mean matter (cold dark matter and baryon) density \( \rho_m \) and a critical density \( \rho_c \).

2.1. Halo Radii and Masses

We denote three-dimensional radii measured from the halo center as \( r \) and reserve capital \( R \) for specific radii used to define the halo boundary. The spherical overdensity mass of a halo is defined as the mass within the radius enclosing a density \( \rho_\Delta \) where \( \Delta \) is an overdensity with respect to \( \rho_m \) or \( \rho_c \) at a particular redshift, such that

\[
M_{\Delta m} = M(< R_{\Delta m}) = \frac{4\pi}{3} \Delta \rho_m(z) R_{\Delta m}^3
\]

for example \( R_{200m} \) and \( M_{200m} \), or

\[
M_{\Delta c} = M(< R_{\Delta c}) = \frac{4\pi}{3} \Delta \rho_c(z) R_{\Delta c}^3
\]

for example \( R_{200c} \) and \( M_{200c} \). The labels \( M_{\Delta} \) and \( R_{\Delta} \) indicate a varying overdensity \( \Delta_{\Delta} \), which we compute using the approximation of Bryan & Norman (1998).

In keeping with this scheme, we use \( R_p \) for the splashback radius (i.e., the apocenter of the first orbit) of individual tracers such as particles and \( R_{\text{sp}} \) for the overall splashback radius of a halo. The corresponding masses \( m_p \) and \( M_{\text{sp}} \) are defined as the masses enclosed by those radii. We will derive \( R_p \) from a distribution of the \( R_p \) of the particles, and we investigate a number of definitions with respect to this distribution. We use superscripts to denote their relation to the particle distribution, such as \( R_p^{\text{20m}} \) for the mean, \( R_p^{50\%} \) for the median, and \( R_p^{75\%} \) for the 75th percentile of the distribution.

The splashback radius, and the outer profiles of halos in general, are most universal in units of any radius defined with respect to the mean density of the universe, particularly \( R_{200m} \) (Diemer & Kravtsov 2014; Lau et al. 2015). Thus, our method for determining \( R_{sp} \) is largely based on \( R_{200m} \) and \( M_{200m} \), and we use those quantities throughout the paper unless otherwise noted. We sometimes express halo mass as peak height, \( \nu \), which is defined as

\[
\nu \equiv \nu_{200m} = \frac{\delta_c}{\sigma(M_{200m}, z)} = \frac{\sigma(M_{200m}, z = 0) \times D_i(z)}{\delta_c},
\]

where \( \delta_c = 1.686 \) is the critical overdensity for top-hat collapse (Gunn & Gott 1972, ignoring a weak dependence on cosmology and redshift), \( D_i(z) \) is the linear growth factor normalized to unity at \( z = 0 \) (e.g., Eisenstein & Hu 1999), and \( \sigma(M) \) denotes the rms density fluctuation in a sphere whose radius is the Lagrangian radius corresponding to mass \( M \). We use the fitting function of Eisenstein & Hu (1998) to compute the linear power spectrum on which the variance is based.

2.2. Dynamical Time and Mass Accretion Rate

The dynamical time of halos, \( t_{\text{dyn}} \), will be useful as a fundamental time unit for various purposes. However, numerous different definitions are used throughout the literature and are often not carefully distinguished. Generally, \( t_{\text{dy}n} \) is defined as the ratio of a characteristic size and a characteristic velocity, where the size can either be the halo radius \( R_{\Delta} \) (time to pericenter), its diameter \( 2R_{\Delta} \) (crossing time), or its circumference \( 2\pi R_{\Delta} \) (orbital time), and the velocity is

\[
v_{\Delta} = \sqrt{\frac{GM_{\Delta}}{R_{\Delta}}}. \]

Here, we are interested in the time it takes a particle to reach the apocenter of its first orbit after infall, that is, the time until splashback. Thus, we define the dynamical time as

\[
t_{\text{dyn}}(z) = t_{\text{cross}}(z) = \frac{2R_{\Delta}}{v_{\Delta}} = 2^{3/2} t_{H}(z) \left( \frac{\rho_s(z)}{\rho_\Delta(z)} \right)^{-1/2},
\]

where \( t_H \) is the Hubble time,

\[
t_H(z) = \frac{1}{H(z)} = \frac{3}{8\pi G \rho_\Delta(z)}.
\]

We note that at high redshift \( \rho_m \approx \rho_c \), so that for \( \Delta = 200 \) m the dynamical time is a fixed fraction of the Hubble time, \( t_{\text{dy}n} \approx (2^{3/2}/\sqrt{200}) t_H \approx t_H / 5 \).

The dynamical time is the basis for our definition of the mass accretion rate. While theoretical models often use an instantaneous (or constant) accretion rate \( s = d \log(M)/d \log(a) \), such a definition is of little practical use when dealing with simulation data because the noisy nature of the mass accretion histories means that the scatter in \( s \) increases without upper bound as the time interval decreases. Thus, Diemer & Kravtsov (2014) defined the mass accretion rate over a finite range of time:

\[
\Gamma_{\text{DK14}} \equiv \frac{\Delta \log(M)}{\Delta \log(a)} = \frac{\log(M_f) - \log(M_i)}{\log(a_f) - \log(a_i)},
\]

where the \( a_i \rightarrow a_f \) pairs were chosen manually to correspond to roughly a crossing time (see also Lau et al. 2015; More et al. 2015; Mansfield et al. 2017). As we expect \( R_{sp} \) to be sensitive
to the accretion history during a particle’s orbit, we adjust their definition by setting the time interval to the dynamical time:

$$ \Gamma_{\text{dyn}}(t) = \frac{\log[M(t)] - \log[M(t - t_{\text{dyn}})]}{\log[a(t)] - \log[a(t - t_{\text{dyn}})]}, $$ (8)

where \( M = M_{200\text{hm}} \).

### 3. N-body Simulations

We apply our algorithm to a suite of dissipationless N-body simulations of different box sizes and resolutions (Table 1). Our fiducial cosmology is the same as that of the Bolshoi simulation (Klypin et al. 2011) and is consistent with WMAP7 (Komatsu et al. 2011), namely a flat ΛCDM cosmology with \( \Omega_m = 0.27, \, \Omega_b = 0.0469, \, \sigma_8 = 0.82, \) and \( n_s = 0.95 \). The initial power spectrum was generated using the CAMB code (Lewis et al. 2000) and translated into initial conditions using the second-order Lagrangian perturbation theory code 2LPTIC (Crocce et al. 2006). The simulations were started at redshift \( z = 49 \), sufficiently high to avoid transient effects (Crocce et al. 2006). The simulations were run with the publicly available code GADGET2 (Springel 2005).

We use the phase–space halo finder ROCKSTAR (Behroozi et al. 2013a) to extract halos and subhalos from each simulation and the CONSISTENT-TREES code (Behroozi et al. 2013b) to establish subhalo relations and assemble merger trees. The halo catalogs and merger trees used in this paper differ from those used in the papers listed in Table 1, as they were run with the most recent versions of ROCKSTAR and CONSISTENT-TREES and, most importantly, used \( R_{200\text{m}} \) as the halo radius. This definition matters because a halo is defined to be a host halo if it is not within \( R_{200\text{m}} \) of another, larger halo. The radius used for these host and subhalo definitions was computed using only bound particles because subhalo masses can otherwise include large, spurious contributions from their hosts, leading to errors in the merger trees. For the remainder of the paper, however, we generally use \( R_{200\text{m}} \) as computed from all particles, bound and unbound, and explicitly state when using bound-only masses and radii. For the vast majority of host halos, the difference between the two masses is small.

Typically, one worries about two types of resolution issues when dealing with N-body simulations: mass resolution (the number of particles in a halo) and force resolution (the length scale below which forces are non-Newtonian). With the algorithm presented in this paper, we need to consider a third resolution scale, namely the time spacing of the snapshots. In general, only about 100–200 snapshots of a simulation are saved, resulting in relatively poor time sampling. Figure 1 gives an overview of the snapshot spacing for the simulations used in this paper. In units of the dynamical time, our main simulation suite exhibits a snapshot spacing of between 0.15\( t_{\text{dyn}} \) at early times and a little less than 0.1\( t_{\text{dyn}} \) at \( z = 0 \). In order to investigate the convergence properties of our algorithm with time spacing, we ran a smaller test simulation (see Table 1) with roughly twice the number of snapshots. We then subsampled the snapshots of this TestSim200 by factors of 2 and 4, creating test cases with realistic (TestSim100) and extremely poor snapshot resolution (TestSim50). We note that the CONSISTENT-TREES merger tree algorithm is not expected

### Table 1

**N-body Simulations**

| Name          | \( L \)  | \( N^3 \) | \( m_p \) | \( \epsilon \) | \( \epsilon/(L/N) \) | \( z_{\text{initial}} \) | \( z_{\text{final}} \) | \( N_{\text{snap}} \) | \( \Delta t_{\text{snap}} \) | Cosmology       | Reference     |
|---------------|---------|-----------|-----------|---------------|-----------------------|--------------------------|--------------------------|----------------|----------------|----------------|--------------|
| L2000         | 2000    | 1024\(^3\) | 5.6 \times 10^{11} | 65       | 1/30                  | 49                       | 0                        | 100             | 20            | WMAP (Bolshoi) | DK15         |
| L1000         | 1000    | 1024\(^3\) | 7.0 \times 10^{10} | 33       | 1/30                  | 49                       | 0                        | 100             | 20            | WMAP (Bolshoi) | DK13         |
| L0500         | 500     | 1024\(^3\) | 8.7 \times 10^{9}  | 14       | 1/35                  | 49                       | 0                        | 100             | 20            | WMAP (Bolshoi) | DK14         |
| L0250         | 250     | 1024\(^3\) | 1.1 \times 10^{8}  | 5.8      | 1/42                  | 49                       | 0                        | 100             | 20            | WMAP (Bolshoi) | DK14         |
| L0125         | 125     | 1024\(^3\) | 1.4 \times 10^{7}  | 2.4      | 1/51                  | 49                       | 0                        | 100             | 20            | WMAP (Bolshoi) | DK14         |
| L0063         | 62.5    | 1024\(^3\) | 1.7 \times 10^{6}  | 1.0      | 1/60                  | 49                       | 0                        | 100             | 20            | WMAP (Bolshoi) | DK14         |
| L0031         | 31.25   | 1024\(^3\) | 2.1 \times 10^{5}  | 0.25     | 1/122                 | 49                       | 2                        | 64              | 20            | WMAP (Bolshoi) | DK15         |
| TestSim200    | 62.5    | 256\(^3\)  | 1.1 \times 10^{4}  | 5.8      | 1/42                  | 49                       | -0.1                     | 193             | 9             | WMAP (Bolshoi) | This work    |
| TestSim100    | 62.5    | 256\(^3\)  | 1.1 \times 10^{4}  | 5.8      | 1/42                  | 49                       | -0.1                     | 96              | 9             | WMAP (Bolshoi) | This work    |
| TestSim50     | 62.5    | 256\(^3\)  | 1.1 \times 10^{4}  | 5.8      | 1/42                  | 49                       | -0.1                     | 48              | 9             | WMAP (Bolshoi) | This work    |

**Note.** The N-body simulations used in this paper, where \( L \) denotes the box size in comoving \( h^{-1} \) Mpc, \( N \) the number of particles, \( m_p \) the particle mass in \( h^{-1} \) M⊙, \( \epsilon \) the force softening length in physical \( h^{-1} \) kpc, \( z_{\text{initial}} \) and \( z_{\text{final}} \) the redshift range of the simulation, \( N_{\text{snap}} \) the number of snapshots written to disk, and \( \Delta t_{\text{snap}} \) the redshift of the first snapshot. The references correspond to Diemer et al. (2013, DKM13), Diemer & Kravtsov (2014, DK14), and Diemer & Kravtsov (2015, DK15).

---

**Figure 1.** Snapshot spacing in the simulations used in this paper, expressed as the number of snapshots per dynamical time. The solid blue line shows the spacing of the main simulations (L0031–L2000, see Table 1), which all share the same snapshot spacing, leading to a resolution of about 6.5 snapshots per dynamical time at the earliest times and about 12 at \( z = 0 \). The dashed lines show the spacing for TestSim200 and its subsampled variants.
to work reliably below about 100 snapshots, depending on how the snapshots are spaced (Behroozi et al. 2013b). We study the convergence of our algorithm with snapshot spacing in Section 5.2.

4. Algorithm

In this section, we describe our algorithm to measure the splashback radii of simulated halos. Many of the figures used to illustrate our method and its convergence are based on our test simulations to save computational expense, and because the large number of outputs allows us to study the convergence of the algorithm with snapshot spacing. We have verified that the results (e.g., the $\Gamma - R_{sp}$ relation) from the test simulation do not, in any important or systematic way, differ from the full-size simulations.

4.1. General Overview

We have implemented our splashback algorithm within the somewhat more general code framework SPARTA, which will be presented in a separate paper (B. Diemer 2017, in preparation). Here, we restrict ourselves to its most basic components and the algorithms that are relevant for determining the splashback radius and mass of halos.

SPARTA is an MPI-parallelized C code designed to follow the trajectories of dynamical tracers in particle-based simulations. It tracks halos (as defined by some halo finder and merger tree code) in a time-forward manner, starting at the first snapshot. The domain can be decomposed in two ways: either into slabs of adjustable size in each dimension or using a space-filling curve. Each process is concerned only with the halos and subhalos within its volume, with overlapping regions due to the spatial extent of halos. The volumes are rectilinear in the case of slabs and arbitrarily shaped in the case of a space-filling curve. The most important concepts in SPARTA are as follows:

1. A halo provides the largest unit of memory, and each halo is uniquely assigned to one process. Subhalos are not treated as halos, but as dynamical tracers within a host halo (Section 4.2).
2. Dynamical tracers include particles and subhalos and are assigned to a host halo. A tracer can exist in multiple halos at the same time (Sections 4.3 and 4.4). At each snapshot, the positions of the tracers in each halo are connected to their previous trajectories. Only four time bins of each trajectory are kept in memory.
3. The trajectories snippets are analyzed for particular “events” (also referred to as “results”) such as infall into a halo or apocentric passage (Sections 4.5 and 4.6). The algorithm presented here works on one-dimensional trajectories (i.e., radius and radial velocity).
4. When a halo ceases to exist (e.g., because it merges or because the end of the simulation is reached), the events connected to its tracers are analyzed, for example to compute the splashback radius (Section 4.7).

The following sections describe these concepts in detail. SPARTA writes all results to an hdf5 file\(^1\) and contains a python module to aid with the reading and analysis of such output files.

---

*HDF5 stands for “hierarchical data format” and is a popular file format that can be read using most commonly used programming languages. See [hdfgroup.org](http://www.hdfgroup.org) for more information.

---

4.2. Halos and Subhalos

SPARTA tracks all halos and subhalos in a catalog, where each halo is uniquely assigned to a process. At each snapshot, SPARTA connects halos to their descendants by matching their unique IDs, saving certain halo properties and updating others. Whenever a record with no progenitor is found in the halo catalog, a new halo is created and sent to the process that is responsible for its location in space. Thus, SPARTA does not have to be started at the first snapshot, but its intended mode of operation is to follow entire halo histories. Furthermore, SPARTA explicitly tracks the relation between host and subhalos as defined by the merger trees. Subhalos are forced to live on the same process as their host and keep a pointer\(^2\) to their host, while hosts keep a list of pointers to all their subhalos. Inconsistencies in these relations cause the code to abort, ensuring that no subhalo relations are omitted by accident. The host or sub status of a halo is recorded for its full history.

At each snapshot, each process computes the boundaries of the rectilinear, potentially periodic volume that contains all its halos, including a particular search radius around the halo centers. This radius depends on the tracers in each halo and various settings (see Section 4.3 for details). All particles contained within the rectilinear volume are loaded from snapshot files, and a tree is constructed from their positions (we use the tree implementation of the ROCKSTAR halo finder, Behroozi et al. 2013a). For each halo, the particles within its search radius are found using a tree search, and the mass profile is constructed from those particles.

While the halo catalog may already specify $R_{200m}$, SPARTA computes it directly from the mass profile. If the search radius is too small to find $R_{200m}$, it is increased iteratively until a solution has been found. We emphasize that the values of $R_{200m}$ are computed from the full particle distribution, including bound and unbound particles, in order to avoid the ambiguities inherent in any unbinding procedure. For host halos, the difference between the bound and full mass profiles is generally small. For subhalos, however, the difference can be large because the density around a subhalo may never reach sufficiently low values, meaning that material from its host is included in the spherical overdensity $R_{200m}$, which leads to an unphysically high radius and mass. Thus, whenever we need to consider subhalo masses and radii, we use the bound-only values from the halo catalog. Specifically, when computing the mass ratio of a subhalo to its host halo, we divide the subhalo’s bound-only peak mass (the highest mass it ever attained) by the host halo’s bound-only mass.

4.3. Particle Tracers

A dynamical tracer is simply defined as an object (e.g., a dark-matter particle or subhalo) that follows some orbit with respect to a halo center (a “trajectory”; see Figure 2 for examples). Particle tracers can be part of multiple halos at the same time because each halo keeps a separate, dynamically allocated array of tracers of each type. Particle tracers are created whenever a particle first comes within $r_{create} = 2R_{200m}$ of a halo’s center, and SPARTA follows its trajectory from that point onward. Due to the enormous number of particles in a

---

\(^{2}\) In the context of SPARTA, a “pointer” refers to an array index rather than a memory pointer, as memory allocation is dynamic and the actual locations of objects can change at any time.
simulation, storing the full trajectories of all tracers in memory is impossible. Thus, SPARTA keeps only a certain number of snapshots (four for the purposes of this paper). The particle properties that are stored are adjusted depending on the chosen output variables (for example, radius and radial velocity rather than the full three-dimensional position, unless the latter is necessary for a particular purpose).

At each snapshot, all particle trajectories are analyzed for certain events, including first infall into the halo (crossing $R_{200m}$, hereafter “infall”) and reaching the apocenter of its first orbit (hereafter “splashback”). When such an event is detected, a tracer result (hereafter simply “result” or “event”) is recorded and stored in the halo’s result arrays. In principle, a tracer can create any number of results, though infall and splashback, by definition, only occur at most once for each tracer in a given halo. Events are stored in separate arrays but carry the ID of their generating tracer so that they can be reconnected to each other later.

A particle tracer’s life can end for one of three reasons, namely (1) when the analysis of its trajectory has finished (i.e., when the search for both an infall and a splashback event has succeeded or failed), (2) when it strays too far from the halo to be considered a dynamical tracer of the halo potential ($r > r_{\text{delete}} = 3R_{200m}$), or (3) when the halo itself ceases to exist according to the catalog. At this point, the tracer object is deleted from memory, and only its results remain. If the deleted tracer had entered within $R_{200m}$ of the halo, we add its ID to a list of tracers to be ignored in the future. Such a list is kept by each halo for each tracer type and checked before creating new tracers to avoid accidentally treating a returning tracer as infalling for the first time.

Our choice of the maximum radius for a particle tracer, $r_{\text{delete}}$, matters somewhat, as it sets an upper limit for the size of the splashback radius. On the other hand, we cannot set $r_{\text{delete}}$ to arbitrarily high values because particle tracers dominate SPARTA’s memory consumption, and because a large search radius would force each process to consider a large fraction of the simulation box. We discuss our choice of $r_{\text{delete}} = 3R_{200m}$ in Section 5.7 and show that splashbacks at larger radii typically belong to halos that are being disrupted.

This complication also highlights why tracers are kept only for host halos. In subhalos, we cannot tell whether a particle belongs to the subhalo or its host. If the latter, the particle’s motion about the subhalo center will make no sense dynamically, as the particle does not execute a trajectory in the subhalo’s potential. Thus, when a halo falls into a larger host and becomes a subhalo, we remove all its tracers and add them to its “ignore” list. The halo keeps all its previous tracer results, but no new tracers are added while it is a subhalo. If it becomes a host halo again (i.e., if it is a “splashback halo”), we begin adding new tracers once again.

A special case occurs for particles that are already in the halo when it is first found in the catalog, for particles in subhalos that become host halos again, and (more generally) any particle

![Figure 2. Characteristic particle and subhalo trajectories in radius (top row) and radial velocity (bottom row). The radial velocities are offset from each other for clarity, and the dashed lines mark zero velocity for each orbit. The columns show well-resolved particle orbits in a large halo (left, based on TestSim200), subhalo orbits in the same halo (center), and particle orbits in a much smaller halo that are more poorly resolved in time (right, based on TestSim100). The points in the top panels mark $r_{sp}$ as determined by the algorithm described in Section 4.6, and the corresponding times are marked with vertical gray lines in the bottom panels. For one of the trajectories shown in the right panels, the splashback radius was not determined because of noise in the trajectory near the orbit’s pericenter.](image-url)
that appears within a halo’s $R_{200m}$ without previously having been tracked. Without going back in time, we cannot determine whether such particles are on their first or not. Thus, we ignore all particles within $R_{200m}$ of a newly created halo or halos that are “reborn” as host halos. We note that this algorithm still allows a small fraction of erroneously identified first orbits: if a particle is outside $R_{200m}$ when a halo is first created or reborn, but had orbited the halo before, we classify its next orbit as being its first infall. In practice, this issue (as well as the entire algorithm dealing with preexisting particles) affects only a small fraction of particles.

4.4. Subhalo Tracers

Subhalo tracers are treated in much the same way as particles, except that their positions and velocities are determined by the halo finder. Subhalo tracers are created whenever a new subhalo is added to a host, but are not deleted if the subhalo leaves the host. Instead, the trajectory of the (former) subhalo is traced until the halo merges away or the simulation ends. Thus, subhalo tracers are allowed to stray far away from their previous host, which poses no performance problem as it does not change the simulation volume for which particles have to be loaded (the halo catalog is loaded in its entirety anyway).

Infall events for subhalos are recorded whenever they cross $R_{200m}$. As the halo merger trees are based on $R_{200m}$ as computed only from bound host halo particles, some subhalos may already lie within the $R_{200m}$ of all particles when they first become subhalos. However, infall events can be constructed from their saved trajectories in most cases. One exception occurs when halos are newly created as subhalos: in this case, the subhalo never technically crossed into its host, and no infall result is recorded. Subhalo trajectories are analyzed for splashback events in exactly the same way as particle trajectories. Figure 2 shows some examples of subhalo orbits.

In our computation of the splashback radius of halos, we will entirely ignore subhalo trajectories because they suffer from dynamical friction and resolution effects (Section 4.7.1). However, subhalos host satellites and are thus an important observational tracer of the density profile and $R_{200m}$ (More et al. 2016). We leave a more detailed investigation of subhalo dynamics for future work.

4.5. Infall Results and Subhalo Tagging

The first type of result we record about a tracer corresponds to its first infall into the host halo, specifically the time when the tracer crosses $R_{200m}$ (i.e., has a radius $r(t_1) > R_{200m}(t_1)$ and $r(t_2) < R_{200m}(t_2)$ at a later snapshot). The exact infall time $t_{inf}$ is determined by linear interpolation. At first sight, the information contained in such an infall result may seem trivial: the time of infall, which, by definition, determines the radius where this infall happened as $R_{200m}(t_{inf})$. However, these events also contain more advanced information, such as the radial and tangential velocities at infall, $v_r/V_{200m}$ and $v_t/V_{200m}$.

Another property of particle infall will turn out to be important in determining $R_{sp}$, namely whether a particle fell in as part of a subhalo, and if so, what the subhalo-to-host mass ratio (SMR) of the subhalo was. Whenever a subhalo falls into a new host, its particles are “tagged” with this SMR, which is in turn passed to the tracer’s infall result. However, the question of which particles should be tagged is somewhat tricky because at infall the subhalo’s $R_{200m}$ already contains a significant fraction of host halo material. One could imagine a number of algorithms to select particles to tag, including the following:

1. All particles within a certain fraction of $R_{200m}$ of the subhalo. However, this selection includes many host particles who happen to be within the subhalo at the time of infall.
2. All bound particles. Given the subhalo’s bound-only mass and concentration, one can estimate its potential based on a Navarro–Frenk–White (NFW) density profile (Navarro et al. 1995, 1996, 1997). Unfortunately, subhalos tend to be already somewhat disrupted at infall (e.g., Behroozi et al. 2014). Thus, this tagging scheme captures particles with low kinetic energy that will fall in with the subhalo and tend to splash back at small radii, but fails to capture particles with low binding energy that tend to be flung out of the halo and splash back at very large radii.
3. All particles that have orbited the subhalo for some time as determined from the age of the corresponding particle tracer objects.
4. All particles that have resided inside the subhalo for some time. We look for an already existing infall result and only tag the particle if this infall result occurred a certain amount of time ago.

After much experimentation, we choose the least selection criterion because it makes the most physical sense and turns out to be relatively robust. In particular, when a subhalo enters a host for the first time, we perform a tree search for all particles within $2R_{200m}$ of the subhalo center (where $R_{200m}$ is computed from only particles bound to the subhalo in order to avoid host halo contributions). For each of the found particles, we check for an existing infall result. If found, we compare the infall time to the current time. In order to avoid erroneously tagging host halo particles, we set the minimum time since the infall of a subhalo particle to $1/2 \tau_{dyn}$. During this time, the infalling subhalo should roughly traverse $R_{200m}$ of the subhalo if it travels at a speed of $\sqrt{\frac{\Omega}{\Omega_{V200m}}}$. Thus, particles that “fall” into the subhalo at separations smaller than $\approx 2R_{200m}$ of the host will not be tagged. There is one final issue we need to address: not all particles that belong to the subhalo have recorded infall times, for example those particles that were already part of a halo when it first appeared in the halo catalog. Thus, if no infall event is found, we check whether the particle ID is on the subhalo’s “ignore” list. If it is, we assume that the particle has belonged to the subhalo for some time and tag it.

If the particle belongs to the subhalo according to these criteria, we find the corresponding tracer object in the host halo and tag it with the new sub-to-host ratio, but only if it has not previously been tagged with a larger ratio. In some cases, the tracer particle can enter the host before the subhalo infall is recorded, leaving behind an infall event without a sub-to-host ratio tag. We look for previously completed infall results of tagged particles and tag those results if found. However, if an infall event is older than $1/2 \tau_{dyn}$ at the time of tagging, we do not tag it because the particle clearly was not part of the infalling subhalo but entered the host independently.

We find that this algorithm converges in the sense that tagging particles more aggressively (e.g., tagging particles that entered the subhalo less than $1/2 \tau_{dyn}$ ago) does not significantly change the averaged $R_{sp}$. This convergence
indicates that the additional particles tagged are not biased in their $r_p$ because they do not truly belong to the subhalo. We further discuss the convergence of our particle tagging algorithm in Section 5.4.

4.6. Tracer Splashback Radii

After a tracer has fallen into a halo, we begin looking for splashback, that is, the apocenter of the tracer’s first orbit in the halo. For simplicity, we ignore all angular information and consider only the tracer’s radius and radial velocity. Ideally, one would record the entire history of a tracer’s orbit, but a significant fraction of the particles in a simulation are being tracked at any given time, meaning that storing many time slices of their trajectories becomes extremely memory-consuming. We thus restrict ourselves to storing only the past four values of the time $t_i$ (in Gyr), the tracer radius $r_i$ and halo radius $R_i$ (in physical $h^{-1}$ kpc), and the radial velocity $v_i$ (in physical $h^{-1}$ kpc Gyr$^{-1}$), where $i$ represents indices running from 0 to 3. We begin our analysis once all four time slices have been set, that is, when the tracer has been in existence for four snapshots.

First, we seek the tracer’s closest approach to the halo center, or pericenter. We detect such minima by finding times when $v_i$ switches from negative to positive values. Due to the fast velocity of the tracer near its pericenter, this switch happens rapidly and can usually be detected robustly. In particular, we look for an upward zero crossing in velocity where $v_1 < 0$, $v_1 < 0$, $v_2 > 0$, and $v_1 > 0$. Demanding two positive and negative time bins on each side of the crossing makes the algorithm robust to noise. However, some trajectories are unresolved in time, that is, have an orbital timescale smaller than four time bins. To detect such orbits, we also record the number of “invalid” upward crossings where only one time bin on each side is positive and negative. If we find more than one such crossing but no valid minimum, we abort the trajectory.

Otherwise, if either $r_1$ or $r_2$ is a minimum (smaller than the adjacent radii), the velocity and radius information are consistent. We could assume a linearly changing velocity and extract the point where $v = 0$, but this method would take into account only velocity but not radial information. Instead, we extrapolate both forward and backward from the two points, writing

$$r_{peri} = r_1 + v_1(t_{peri} - t_1) = r_2 + v_2(t_{peri} - t_2)$$

(9)

and thus

$$t_{peri} = \frac{r_1 - r_2 - v_1 t_1 + v_2 t_2}{v_2 - v_1}. \tag{10}$$

Because of the two equalities in Equation (9), the result tells us whether the radial and velocity information are consistent: if $t_1 < t_{peri} < t_2$, the interpolation has given a valid result, and we adopt $t_{peri}$ and compute $r_{peri}$ from Equation (9). Otherwise, we set $t_{peri}$ to either $t_1$ or $t_2$, whichever has the lower value, and $t_{peri}$ to the corresponding time. We linearly interpolate the halo radius $R(t)$ to $t_{peri}$ and record the ratio $r_{peri}/R_{200m}(t_{peri})$.

Once the pericenter has been established, we begin to look for an apocenter, a maximum in $r$ where $v_r$ changes from positive to negative. As before, we demand that $v_0 > 0$, $v_1 > 0$, $v_2 < 0$, and $v_3 < 0$. If, for example, $v_2$ is negative but $v_3$ positive, we abort the trajectory because it is likely noisy. We apply a similar algorithm as for the pericenter: if either $r_1$ or $r_2$ is a maximum, we use the equivalent of Equation (9) for $r_p$ and find

$$t_p = \frac{r_2 - r_1 + t_1 v_1 - t_2 v_2}{v_1 - v_2} \tag{11}$$

and

$$r_p = r_1 + v_1(t_p - t_1). \tag{12}$$

We accept this solution if $t_1 < t_p < t_2$. This is almost always the case if the trajectory has good time resolution, but in reality, the snapshot spacing can be a significant fraction of the halo’s dynamical time, the halo centers and velocities (as determined by the halo finder) suffer from noise, and $r$ and $v_i$ often seem inconsistent due to accelerations that occurred on timescales smaller than the snapshot spacing. In such cases, we have to weigh multiple options. First, we check whether $r_0$ or $r_3$ are maxima (which happens relatively frequently). However, we also want to consider the information given by the $v$ trajectory, particularly the point where it crosses zero:

$$t_{cross} = t_1 - \frac{v_1}{v_2 - v_1}(t_2 - t_1). \tag{13}$$

Thus, we assign the far-away maximum, say $r_0$, a merit function:

$$f_0 = \frac{r_0}{\max(r_1, r_2)} - 1, \tag{14}$$

This number weighs how much larger the radius of this new maximum is than $r_1$ and $r_2$ and compares the difference to how far in time the maximum strays from the crossing point where $v = 0$. If $r_0$ or $r_3$ is a maximum, and the corresponding merit factor $f_0$ or $f_3$ exceeds 0.2, we accept this maximum. If the merit factor is below 0.2, we set $t_p = t_{cross}$ and interpolate to find the corresponding radius:

$$r_p = r_1 + (r_2 - r_1)\frac{t_{cross} - t_1}{t_2 - t_1}. \tag{15}$$

Once $t_p$ and $r_p$ have been determined, we find the enclosed mass $m_{200}$ by interpolating a two-dimensional grid in log($r/R_{200m}$) vs. time space, which is saved for the last four snapshots in each halo, and for 50 radial bins, spaced logarithmically between 0.01$R_{200m}$ and the maximum tracking radius $R_{200m}$. We have verified that increasing the number of radial bins does not systematically change the inferred splashback masses.

A few examples of the results of our algorithm are shown in Figure 2. The particles shown in the left column fall into their halo at different times and with different velocities. In comparison, the subhalo trajectories in the center column exhibit more noise because they are based on two halo positions identified by the halo finder. Furthermore, we observe that the subhalo orbits gradually shrink due to dynamical friction before the subhalo is eventually disrupted. Finally, the right column shows another set of particle orbits from a significantly smaller halo. These orbits suffer from poorer snapshot time resolution and larger halo finder noise. In one of the four cases, the algorithm aborts because it cannot reliably identify a pericenter, due to multiple adjacent zero crossings of the radial velocity.
The algorithm described in this section identifies a splashback event for about 90% of the particle trajectories in TestSim100 (and for about 85% of the subhalo trajectories). We discuss the convergence of our algorithm with snapshot spacing and mass resolution in Section 5.

4.7. Determining the Halo Splashback Radius

While the infall and splashback events of individual particle and subhalo tracers contain a wealth of information, we are usually interested in a more compact description of the splashback radius, namely the mean, median, or some other percentile of the \( r_{sp} \) events as a function of time. Whenever a halo’s history ends (because it merges or reaches the final snapshot of the simulation), we analyze all particle infall and splashback events to find such averages. This process consists of two main steps: selecting a sample of representative splashback events, and averaging them as a function of time.

4.7.1. Selecting Valid Splashback Events

Figure 3 shows the raw data entering this analysis, namely the density of particle splashbacks. Some disturbing structures are apparent: streaks of relatively high event density that migrate outward with time. These features are the signatures of mergers. When a subhalo is tidally disrupted (often near the pericenter of its orbit), the orbits of its particles diverge. Those particles that are most bound to the host halo reach their apocenter soonest and at the smallest \( r_{sp} \), whereas relatively unbound particles are flung out, resulting in a late apocenter and large \( r_{sp} \). As expected, many of the streak features in Figure 3 can be associated with a particular subhalo (round points). One might imagine that the contributions from relatively bound and unbound particles would cancel out, but subhalos suffer from dynamical friction (Chandrasekhar 1943; van den Bosch et al. 1999; Boylan-Kolchin et al. 2008; Adhikari et al. 2016). The more massive the subhalo, the more its contribution is biased toward low \( r_{sp} \). In other words,
subhalos are not faithful dynamical tracers of the host halo potential. Thus, we do not consider subhalo splashback events at all when computing \( R_{sp} \), and furthermore we exclude splashback events from particles that entered as part of a massive subhalo.

In order to visualize the importance of this exclusion, Figure 4 shows the distribution of \( r_{sp} \) and \( m_{sp} \) at a fixed time for an example halo, with the different colors indicating splashbacks that originated from subhalos with different SMRs. The distributions vary hugely from time to time and halo to halo, but those shown in Figure 4 demonstrate a few typical cases. At the earliest times, the distribution exhibits significant shot noise due to the small number of particles. In the second time bin, no major mergers occur, and the distributions in both radius and mass are well fit by a Gaussian, though with a tail toward high radii. In the third time bin, a major merger has occurred, and subhalo particles dominate the splashback events. Due to dynamical friction, they splashback at smaller radii than the nonsubhalo particles. Even the sample with \( SMR < 0.1 \) exhibits a significant dynamical friction bias.

In order to address this issue, we exclude all particle splashbacks that originated from subhalos with \( SMR > 0.01 \). Below this value, the resulting averaged \( R_{sp} \) and \( M_{sp} \) do not change significantly (see Section 5.4 for a more formal discussion). The bottom panels of Figure 3 show the splashback distribution after the SMR exclusion has been applied. Some of the streak features are still apparent (due to particles that were erroneously not tagged as belonging to a subhalo), but the density of such particle events is sufficiently reduced.

### 4.7.2. Averaging over Splashback Events

Given a distribution of splashback events such as that in Figure 3, \( R_{sp}(t) \) and \( M_{sp}(t) \) are, to some degree, a matter of definition. Figure 5 shows the average shape of this distribution as a function of both \( R_{200m} \) and the mean \( r_{sp} \) of each halo. The figure is based on all halos with 10,000 or more particles in TestSim100 at \( z = 0 \pm 0.25 \) Gyr (including lower-mass halos leads to the median dropping to zero closer to the center of the distribution). For comparison, the dotted lines show Gaussian fits to the mean distribution. The mean of \( r_{sp}/(r_{sp}) \) is well described by the Gaussian distribution except for a few-percent tail at \( r_{sp} > 1.5 \langle r_{sp} \rangle \). As a result, the mean and intermediate percentiles of the distribution are very close to those inferred from a Gaussian fit, whereas the highest percentiles exceed the Gaussian expectation.

Figure 5 suggests that the \( r_{sp} \) distribution is relatively regular, and that taking its mean and percentiles should be well-motivated definitions of a halo’s \( R_{sp} \). We wish to avoid binning the \( r_{sp} \) values in time, as that would introduce a number of free parameters such as the bin size. Instead, we smooth each event in time so that the value of \( R_{sp} \) is determined by the \( r_{sp} \) within some time range around it, and we compute the weighted mean of those events. The smoothing function is arbitrary in principle; we use a Gaussian kernel to give each event a weight

\[
w_i = \exp\left(-\frac{(t-t_i)^2}{2\sigma^2}\right),
\]

where \( w_i \) is the weight for event \( i \), and \( \sigma \) is the smoothing radius. The weighted mean is then given by

\[
\langle r_{sp} \rangle = \frac{\sum w_i r_{sp,i}}{\sum w_i}.
\]
the weighted mean of the events at time $t$:

$$R_{\text{sp}}^{\text{mean}} = \frac{1}{W} \sum_{i=0}^{W-1} w_i r_i,$$

where $r_i = r_{\text{sp},i}$ is the radius of the $i$th splashback event and $W$ is the sum of all weights $w_i$. The statistical uncertainty on the weighted mean can be computed from the weighted variance as

$$\sigma_{\text{mean}} = \sqrt{\frac{\sum_{i=0}^{W-1} w_i (r_{\text{sp},i} - R_{\text{sp}}^{\text{mean}})^2}{W^2 - \sum_{i=0}^{W-1} w_i^2}}.$$

We also compute various weighted percentiles of the $r_{\text{sp}}$ distribution. For this purpose, we order the $r_{\text{sp}}$ values ascendingly and define the cumulative percentage at value $i$ as

$$P_i = \frac{100}{N_{\text{BS}}} \left( W_i - \frac{w_i}{2} \right),$$

where $W_i$ is the cumulative weight at value $i$. For a given percentile $p$, we find the first value $k$ with $P_k > p$ and linearly interpolate to obtain $R_{\text{sp}}^p$. If $p < P_0$, we set $R_{\text{sp}}^p = r_0$, and if $p > P_{N-1}$, we set $R_{\text{sp}}^p = r_{N-1}$. The details of this procedure matter only for extreme percentiles and bins with very few contributing splashback events where the statistical uncertainty can be significant. Unfortunately, there is no standard expression for the statistical uncertainty on weighted percentiles of an unknown distribution. Any analytical estimate relies on the derivative of the distribution, which is by definition extremely noisy in the case of small samples. Thus, we perform a bootstrap resampling of the percentile calculation; in other words, we draw $N_{\text{BS}} = 200$ random samples from the distribution with replacement and perform the calculation on each sample. For each percentile, the quoted value is the mean of the samples,

$$R_{\text{sp}}^{p\%} = \frac{1}{N_{\text{BS}}} \sum_{i=0}^{N_{\text{BS}}-1} R_{\text{sp}}^{p\%},$$

and the uncertainty is estimated from the variance of the samples:

$$\sigma^{p\%} = \sqrt{\frac{1}{N_{\text{BS}} - 1} \sum_{i=0}^{N_{\text{BS}}-1} (R_{\text{sp}}^{p\%}, i - R_{\text{sp}}^{p\%})^2}.$$
improves the completeness, but at the cost of extremely uncertain $R_{sp}$ determinations. This minimum does not significantly bias the average $R_{sp}$ of halos.

Finally, we wish to compute $M_{sp}$ as well as $R_{sp}$. We follow the same procedure as for $R_{sp}$, meaning we compute it from the enclosed mass within the splashback radii of the individual particles $m_{sp}$. This choice means that we do not exactly preserve the relation $M_{sp} = M(<R_{sp})$. On the other hand, inaccuracies in $R_{sp}$ (e.g., due to smoothing at times of sharp changes) do not necessarily translate into errors in $M_{sp}$. In other words, we treat $M_{sp}$ as an independent aspect of the distribution of splashback events rather than as a secondary consequence of $R_{sp}$.

4.7.3. Correction for the Final Snapshots

Finally, we need to correct for two biases that occur at the end of the simulation (typically at $z = 0$). First, the number of splashback events in the final time bin (between the second-to-last and last snapshots) is drastically lower than in the previous time bins, which is to be expected given the algorithm described in Section 4.6. Thus, we ignore any splashback events that have occurred after the time of the second-to-last snapshot as they are likely biased in some nontrivial way. Second, the distribution of splashback events considered in the final snapshots becomes asymmetric due to the smoothing discussed above: the Gaussian filter is sensitive to events at earlier times, but there are no events at later times. This asymmetry can lead to a significant and systematic bias because $R_{sp}$ is, in most cases, increasing with time. Thus, ignoring splashback events that would have occurred in the future leads to an underestimate of $R_{sp}$ at late times.

We correct for this asymmetry as follows. In the absence of any information about the “missing” events, we must extrapolate the past evolution of $R_{sp}$ into the future. We recognize that the evolution of $R_{sp}$ (or $M_{sp}$, hereafter summarily called $q(t)$), is given by some true, underlying function $f(t)$, convolved with the Gaussian smoothing kernel such that

$$q(t) = \int_{t-D_{max}}^{t+D_{max}} f(t)w(t)dt.$$  \hspace{1cm} (22)

If the last time at which we have recorded splashback events is $t_f$, we need to make a correction for all time bins where $t + D_{max} > t_f$. In particular, we correct the mean of the distribution in those bins such that

$$q_{corrected} = \frac{w_0q + w^*q^*}{w_0 + w^*},$$  \hspace{1cm} (23)

where $q$ is $R_{sp}$ or $M_{sp}$ determined from the distribution up to time $t_f$, $q^*$ is the contribution we would obtain from the distribution of events at times later than $t_f$, and $w_0 \equiv w(\leq t_f)$ and $w^* \equiv w(\geq t_f)$ are the integrated weights before and after $t_f$. We base our extrapolation on the assumption that $f(t)$ evolves linearly, $f(t) \approx f_0 + f_1t$. We perform a least-squares fit for the free parameters $f_0$ and $f_1$ using the four time bins before $t_f$ and compute the correction term $q^*$:

$$q^* = \int_{t_f}^{t_f + D_{max}} e^{-\frac{t^2}{2\sigma^2}}(f_0 + f_1t)dt^t$$

$$= \left[ \frac{\sigma}{\sqrt{2}} \right] \sigma(f_0 + f_1t)(\text{erf}(x_0) - \text{erf}(x_0))$$

$$+ \sigma^2 f_1(e^{-x_0^2} - e^{-x_1^2}),$$  \hspace{1cm} (24)

where $x_0 \equiv (t_f - t)/(\sqrt{2}\sigma)$ and $x_1 \equiv D_{max}/(2\sigma)$. The function $\text{erf}$ denotes the error function. We perform this correction iteratively, starting with the first time bin that is affected by the asymmetry of $r_{sp}$ events. The next time bin’s correction is then based on the previous corrected time bins, and so on.

The details of how the best-fit slope is determined are unimportant. For example, varying the number of fitted time bins between three and eight has virtually no effect, nor do limits on the $\chi^2/N_{dof}$ of the fit. However, we find that the algorithm described above can over compensate slight and performs better if the weight given to the correction term, $w^*$ in Equation (23), is limited to half of the total weight, that is, $w^* \leq w_0$. Furthermore, the algorithm works best for the most stable estimates of $R_{sp}$ and $M_{sp}$, the mean and median of the distribution of splashback events. The bin-to-bin fluctuations grow for the higher percentiles, leading to additional noise in the extrapolation but no significant improvement in the average bias of the $R_{sp}$ estimates. Thus, we compute the correction factor for the mean, $q_{mean}^{corrected}/q_{mean}$, and apply it to both the mean and all percentile estimates. We discuss the performance of the correction algorithm in Section 5.6.

5. Results

In the previous section, we have described an algorithm to compute a halo’s $R_{sp}$ and $M_{sp}$ throughout its history. In the process, we have introduced a few free parameters that might influence the results (Table 2). In this section, we discuss the algorithm’s performance and convergence with respect to all free parameters as well as mass resolution and snapshot spacing. While we consider the $\Gamma-R_{sp}$ relation to establish convergence, we leave any analysis of the dependence of $R_{sp}$ on halo mass, accretion rate, and cosmology for Paper II.

We will establish a number of cuts on our halo sample, namely a limit of $N_{200m} \geq 1000$ particles and a sub-to-host mass ratio $<0.01$ for all particle $r_{sp}$ events. Wherever the $\Gamma-R_{sp}$ relation is shown, we additionally require that the halo was a host at the beginning of the interval over which $\Gamma$ is measured, namely a dynamical time ago. We do not, however, exclude halos that were a subhalo at some point during that interval. Halos that experienced such a splashback event during the last dynamical time make up for about 2% of the population at $z = 0$, a fraction that drops below a percent at higher redshift where the dynamical time is shorter. Excluding such halos has no measurable impact on the median $\Gamma-R_{sp}$ relation.

5.1. Completeness

As a first test, we consider how successful the algorithm is in assigning splashback radii and masses to halos. The computation can fail either because there are not enough splashback
events close to a given time (i.e., the weight falls below $w_{\text{min}}$; this issue obviously becomes more common for lower-mass halos) or because the halo was recently a subhalo. For such splashback halos, the computation is resumed, but the algorithm needs to accumulate a few snapshots before the first splashback events are recorded.

Figure 6 shows the completeness as a function of the number of particles in a halo for different simulations. We note that only halos that reached $N_{200m} \geq 200$ at some time were written to disk. The incompleteness in poorly resolved halos ($N_{200m} \lesssim 500$) is predominantly caused by a lack of splashback events. In better resolved halos, virtually all failures are due to the halo recently having been a subhalo. This effect is most important for the smallest box sizes because physically less massive halos are more likely to be subhalos.

Most importantly, however, we find that about 95% of halos with at least 1000 particles are assigned an $R_{sp}$ and $M_{sp}$. The only exception are the smallest boxes at low redshift, for example, L0031 at $z = 2$ and L0063 at $z < 2$.

### 5.2. Convergence with Snapshot Spacing

As described in Section 4.6, the SPARTA algorithm uses four time bins to determine the splashback radius of particles, corresponding to about 0.6 $t_{fbp}$ at early times (Figure 1). Given this large fraction of the dynamical time, reducing the time resolution of the trajectories might lead to systematically different estimates of $t_{sp}$ and $R_{sp}$. In this section, however, we demonstrate that this effect is not significant as long as the evolution of a simulation is represented by about 100 or more snapshots.

Figure 7 illustrates a few characteristic ways in which individual particle trajectories are affected by snapshot spacing. The trajectory on the left is well resolved in time even in the low-resolution run, and the algorithm has no problem in reconstructing the correct splashback despite the relatively weak feature in radius and velocity. The second trajectory illustrates the importance of interpolation: even though the snapshot where the radius is largest is missing from the low-resolution version, the splashback event is reconstructed correctly. The third trajectory illustrates a common reason why a splashback event might fail to be found in lower-resolution trajectories: the velocity becomes positive for only one splashback event around splashback, not enough for the algorithm to confidently identify a maximum in radius. Finally, the fourth trajectory illustrates a pathological case: the orbital period of the particle is in resonance with the splashback event frequency, meaning that the true orbit is entirely misrepresented in the lowest-resolution run.

Having understood the effects of snapshot spacing on individual trajectories, we now consider the $R_{sp}$ history of the example halo from Figure 3. The left panel of Figure 8 shows the same data as in the left panel of Figure 3, but using a lower-resolution histogram resolution. The center and right panels show the same plot for the subsampled versions, TestSim100 and TestSim50. The most notable differences between the three panels stem not from the SPARTA results, but from the halo finder output: even between the 200- and 100-snapshot versions, the subhalo splashbacks are distributed somewhat differently. Moreover, in the 50-snapshot run, the merger tree code identified the example halo as a subhalo for a period around 3 Gyr, leading to the gap in splashback events. We note that the merger tree code is not designed to operate with such few snapshots in the first place (Behroozi et al. 2013b). Despite the drastic differences in the merger tree history, the SPARTA results are very similar to the runs with more snapshots.
The right panel of Figure 8 exhibits other visibly different features. For example, some particles splashback at very low radii between 4 and 8 Gyr. We have verified that these events stem from pathological trajectories with orbital times that are in resonance with the snapshot frequency, as shown in Figure 8. While the algorithm will (correctly) fail to identify a splashback in the majority of such trajectories, some randomly mimic a pericenter and apocenter a long time after the particle has entered the halo. Furthermore, the run with the fewest snapshots lacks splashback events at very late times because SPARTA can only identify splashback events that happen within a few snapshots of the end of a trajectory. Given that neither of these issues is observed in the run with 100 snapshots, we need not worry about them.

However, the example halo shown in Figures 3 and 8 is resolved by many particles, and the snapshot spacing could have more severe effects on halos with fewer particles. We investigate a statistical halo sample in Figure 9 that shows the $\Gamma - R_{sp}$ relation for the test simulations with 200, 100, and 50 snapshots. We compare the mean of the halo samples in each simulation because it is more sensitive to outliers than the median. As expected, the run with the lowest number of snapshots shows clear deviations from the other runs, up to 30% at high redshift and mass accretion rate. The run with 100 snapshots, however, has converged to within the statistical uncertainty. We thus conclude that simulations with about 100 snapshots or more, like those used in this paper, are suitable for the dynamical analysis of trajectories.

5.3. Convergence with Mass Resolution

In principle, there is no reason to assume that the splashback radii of individual particles should depend on the number of particles with which a halo is resolved, but one could imagine that the averaging procedure might introduce a bias when only a few $r_{sp}$ events are present. We test the effects of mass resolution statistically by considering the $\Gamma - R_{sp}$ relation in simulations with different particle masses. At fixed mass, each smaller box size corresponds to an eight times higher number of particles (Table 1). However, we need to be careful: the halo mass functions in the different simulation boxes are not the same, which could lead to confusion between resolution effects and a potential mass dependence of $R_{sp}$. We avoid this issue by considering narrow bins in peak height.

Figure 10 shows a comparison between the $\Gamma - R_{sp}$ relation in different simulations for halos with $N_{200m} \geq 1000$. Each row corresponds to a bin in peak height and each column to a redshift. Given the uncertainties on the mean relations, we cannot find any statistically significant mass resolution effects. Moreover, the 68% scatter in the relations (not shown in Figure 10) does not increase appreciably in the lower-resolution simulations, indicating that the lower number of particles in each halo does not introduce a significant random error into the measurements of $R_{sp}$. Lowering the minimum particle number to 500 causes 10% deviations in the mean relations, which persist to particle numbers as low as 200. While Figure 10 shows the relations for the mean $R_{sp}$, the convergence is equally good for $M_{sp}$ and for the higher percentiles. Only the 99th percentile $R_{sp}$ of low-$\nu$ halos shows about 10% deviations between the simulations.

We conclude that the mass resolution limit should depend on the purposes of a given investigation. If 10% errors in the averaged $R_{sp}$ are acceptable, even halos with a few hundred particles can be considered. However, at such low numbers, the completeness decreases significantly (Figure 6). Thus, we stick with a limit of $N_{200m} \geq 1000$ as it guarantees a completeness of about 95% and a convergence of better than 5% with mass resolution.
eliminates unphysical changes to much smaller values is not physically sensible and leads to signifi-
can fluctuations of individual halos computed with different values of \( \sigma_{\text{dyn}} \). However, while this value may seem to work well for individual halos, we need to confirm that the smoothing does not introduce a systematic bias in \( R_{\text{sp}} \). The center column of Figure 11 shows the \( \Gamma - R_{\text{sp}} \) relation for several values of \( \sigma_{\text{dyn}} \). For high values, \( \sigma_{\text{dyn}} > 0.5 \), there is a small but systematic bias toward lower values, but for our fiducial value of \( \sigma_{\text{dyn}} = 0.2 \) this bias disappears.

While the bias is insignificant for the mean and median, there is a positive bias at higher percentiles. In particular, the range up to the 85th percentile experiences a bias of 5% or less at all redshifts, whereas higher percentiles such as the 99th can be biased up to 10%. We have checked that this is not a resolution effect: the bias persists even for highly resolved halos with \( N_{200m} > 10,000 \) and is noticeable when inspecting the \( R_{\text{sp}} \) histories of individual halos computed with different values of \( \sigma_{\text{dyn}} \).

In summary, the fiducial value of \( \sigma_{\text{dyn}} = 0.2 \) leads to unbiased estimates of the mean and median, biases of less than 5% up to the 85th percentile, and increasingly biased measurements of the highest percentiles. However, reducing \( \sigma_{\text{dyn}} \) to much smaller values is not physically sensible and leads to significant fluctuations of \( R_{\text{sp}} \) with time.

5.6. Convergence of the Correction at the Final Snapshots

In Section 4.7.3, we applied a multiplicative correction term to the averaged \( R_{\text{sp}} \) at the last snapshots of a simulation in order
to counteract the effects of missing splashback events that would have occurred in the future. The right column of Figure 11 compares the results with and without this correction term. In particular, the dark blue line shows the $\Gamma - R_{sp}$ relation for the mean as derived from the TestSim100 snapshots, where the simulation was run into the future and thus presents the "correct" solution to compare to. The light blue line shows the results for the last snapshot of the same simulation, but only run to $z = 0.2$. As expected, $R_{sp}$ is underestimated by about 10%. The green line shows the same snapshot but with the correction term applied.

Comparisons for mass instead of radius, as well as other statistics such as the median or higher percentiles, look similar, with a slight increase of the bias toward the highest percentiles. On average, the correction term reduces the bias in TestSim100 to less than 1.2% (in both $R_{sp}$ and $M_{sp}$, in the mean and up to the 85th percentile). The highest percentiles, such as the 99th, can still suffer biases of up to 8%. Due to the large bin-to-bin noise in the highest percentiles, it is not easy to remove this bias (see Section 4.7.3). Finally, we note that while the correction term debiases $R_{sp}$ and $M_{sp}$ on average, it leads to slightly increased scatter around the mean $R_{sp}$ and $M_{sp}$. Compared to the uncorrected distribution, the standard deviation increases from between 6% and 9% to between 10% and 12% for the mean and up to the 85th percentile, and from about 11% to about 14% for the highest percentiles.

In summary, the correction recovers the correct $R_{sp}$ and $M_{sp}$ values at the last snapshots on average, but at the cost of adding a random scatter to the estimates for individual halos. However, the correction has a strong effect only at the very latest snapshots. For example, if a simulation with our fiducial cosmology is run to $z = 0$, the correction term has an effect after $z = 0.28$, and a strong effect only within one $\sigma$, after $z = 0.08$. Thus, uncertainties due to the correction can be
avoided by considering a time slightly before the end of the simulation.

5.7. Dependence on the Maximum Tracer Radius

As explained in Section 4.3, we have set the maximum radius to which SPARTA tracks particles to \( r_{\text{delete}} = 3R_{200m} \). This radius sets an effective maximum \( R_{sp} \) because we would miss apocentric passages at larger radii. In a TestSim100 run with an extreme value of \( r_{\text{delete}} = 6R_{200m} \), SPARTA finds additional splashback events in about 4% of halos (depending on redshift and mass resolution), which would indicate a serious bias. The mean \( R_{sp} \) is lower by about 3% in the run with our fiducial value of \( r_{\text{delete}} = 3R_{200m} \), and the highest percentiles can be lower by up to 5%. However, the median of the \( R_{sp} \) distribution is virtually unaffected, with a bias of less than 0.1%.

Moreover, a visual inspection of the affected halos shows that their high-\( R_{sp} \) events are not physically meaningful. Virtually all such events are caused by disruption, that is, particles that have left the halo and caused spurious splashback events far away. In a large fraction of the cases, the disruption is due to the tidal forces from a larger host that a halo is about to merge with. We have experimented with excluding all halos that will become subhalos within one dynamical time, but while such a cut does, indeed, remove a significant fraction of the halos for which \( r_{\text{delete}} \) matters, it also removes many halos that do not suffer from disruption effects whatsoever. Moreover, the cut merely reduces the bias on the mean to 1.5%, meaning that disruption due to host halos is only responsible for part of the effect. The rest is presumably due to fly-by events, close encounters where the center of a halo does not enter \( R_{200m} \) of the larger halo. Such events are common and can easily remove particles from the smaller halo, but are not recorded in the halo catalogs. Another argument against a cut on future subhalos is that it cannot be performed toward the end of a simulation because we do not know which halos would merge in the future.

For these reasons, we refrain from cutting out future subhalos and conclude that a finite value of \( r_{\text{delete}} \) is physically sensible. We caution, however, that the mean \( R_{sp} \) of a sample of halos can be affected by outlier values due to unphysical \( r_{\text{delete}} \) distributions. This is one of the reasons why we focus on the median rather than mean \( R_{sp} \) in Paper II.

6. Discussion

Having demonstrated the numerical convergence of our algorithm, we now discuss topics related to the physical interpretation of our results. We establish a connection with the sharp drop in density associated with \( R_{sp} \) by comparing our results to those from the SHELLFISH code, and we discuss the impact of the orbital parameters of particles, resolution effects, and the significance of extremely low and high accretion rates.

6.1. Comparison with SHELLFISH

All previous approaches to measuring \( R_{sp} \) have relied not on particle dynamics but on the sharp drop in density associated

---

**Figure 11.** Convergence of \( R_{sp} \) with maximum sub-to-host mass ratio (left), smoothing timescale (center), and late-time correction (right). The plots show \( R_{sp}^{\text{num}} \), but the convergence properties are similar for the median and higher percentiles. Left column: When all splashback events are included in the calculation (yellow), \( R_{sp} \) is biased low by up to 20% due to dynamical friction. Excluding particles from subhalos larger than \( \text{SMR}_{\text{max}} = 0.1 \) reduces this bias significantly (green), but a systematic difference of a few percent remains. The light blue line showing the fiducial value of \( \text{SMR}_{\text{max}} = 0.01 \) is indistinguishable from the dark blue reference solution with \( \text{SMR}_{\text{max}} = 0.001 \). Center column: same as the left column, but for different values of the smoothing timescale \( \sigma_{\text{dy}n} \). With aggressive smoothing such as \( \sigma_{\text{dy}n} \gg 0.5 \), a bias compared to the unsmoothed \( R_{sp} \) is apparent. At our fiducial value of \( \sigma_{\text{dy}n} = 0.2 \), this bias has disappeared. Right column: the effect of the correction applied to the last few snapshots of a simulation where the \( R_{sp} \) distribution becomes asymmetric in time. The dark blue line shows the true evolution of \( R_{sp} \), taking future splashback events into account. The light blue line shows the same simulation but stopped at the redshift shown, \( z = 0.2 \), without applying a correction. Due to the asymmetry in the final bins, \( R_{sp} \) is underestimated by about 10%. The green line shows the results after the correction term has been applied, recovering the correct solution on average (the deviations at high \( \Gamma \) are not statistically significant or systematic).
with the splashback caustic (Adhikari et al. 2014; Diemer & Kravtsov 2014; More et al. 2015, 2016; Mansfield et al. 2017). Moreover, the density drop is observable in the real universe, whereas the splashback radii of DM particles are not. Thus, it is paramount that we establish the connection between our results and those based on the density structure of halos. Here, we compare to the only measurement of \( R_s \) in individual halos undertaken so far, namely the SHELLFISH algorithm of Mansfield et al. (2017). We restrict ourselves to a halo-by-halo comparison of \( R_s \) and \( M_s \), and we leave an analysis of the average \( \Gamma - R_s \) relation for Paper II. The SHELLFISH algorithm operates on a fundamentally different principle than SPARTA: it finds sharp density drops in a large number of random sight lines and derives a (not necessarily spherical) \( R_s \) shell that delineates the drop radii. Unlike SPARTA, the algorithm can extract \( R_s \) from a single snapshot alone, but it demands a somewhat higher resolution of 50,000 particles per halo (Mansfield et al. 2017).

The SHELLFISH results are most closely approximated by \( R_s^{50\%} \), a relatively high percentile. Figure 12 shows a comparison of this definition and the SHELLFISH results for halos that fulfill the SHELLFISH resolution requirement. At low \( \Gamma \), the distribution was subsampled in order to achieve more even coverage of all mass accretion rates. The relative difference in the mean or median is, on average, less than 3.3% at all redshifts, and less than 1% when all redshifts are combined. The 68% scatter is largest at \( z = 0 \) (about 18%) and decreases to about 10% at higher \( z \), with an average of 14% when all redshifts are combined. The increased scatter at \( z = 0 \) is partially due to SPARTA’s correction for the final snapshots.

One important difference between the algorithms is that SPARTA (in its current incarnation) works in spherically symmetric coordinates. As expected, SHELLFISH gives smaller \( R_s \) values than SPARTA for the most aspherical halos, which can exhibit major-to-minor axis ratios of up to 2.5. For such objects, SHELLFISH infers \( R_s \) values that are up to \( \approx 30\% \) lower than those of SPARTA. In other words, SHELLFISH computes a volume-weighted spherical \( R_s \), whereas SPARTA’s results are mass-weighted. However, the effects of asphericity can only account for a small fraction of the scatter in Figure 12 because the vast majority of halos are only moderately aspherical, with axis ratios less than 1.5, where the difference between SHELLFISH and SPARTA becomes negligible on average.

Attempting to establish as close a correlation between the two methods as possible, we have also experimented with definitions that combine two of SPARTA’s percentile measurements, such as

$$ R_{s_{\text{combined}}} = \sqrt{R_s^{50\%} \times R_s^{99\%}}. $$

This definition was motivated by the finding that \( R_s^{50\%} \) corresponds to the inner edge of the “steepening region” according to SHELLFISH, that is, the part of the density profile where the slope begins to steepen beyond the values expected from a profile without a splashback feature (Mansfield et al. 2017). However, the outer edge of the steepening region is not approximated to sufficient accuracy by any percentile of the splashback distribution, perhaps because of the relatively unreliable determination of the highest percentiles of the apocenter distribution (Section 5). Thus, we failed to find a definition that agrees with SHELLFISH better than \( R_s^{65\%} \) does, and we will use \( R_s^{87\%} \) for further comparisons in Paper II.
6.2. Dependence on Orbital Parameters

We have treated particles as a set of unbiased dynamical tracers of the halo potential, but in reality they enter the halo with certain initial conditions, namely a radial and tangential velocity. Figure 13 shows how the splashback radius is influenced by the total velocity at infall, the circularity at infall, and the pericentric distance to the halo center. This figure shows all splashback events in TestSim100 that satisfied the bound on the subhalo mass ratio, that occur in halos with $N_{200m} \geq 1000$, and for which $R_{sp}^{mn}$ could be computed. We divide $r_{sp}$ by the halo’s $R_{sp}^{mn}$ at the time of infall in order to scale out the halo’s overall radius.

The first impression is that the scatter in the distributions is large. Clearly, the total velocity at infall (or kinetic energy, left panels) has a significant effect on $r_{sp}$: as expected, particles with higher energy splashback at larger radii. The circularity (quantified as the fraction of the velocity that is in the tangential direction at infall, center panels) has a smaller effect where more radial orbits lead to slightly higher $r_{sp}/R_{sp}$. This difference may seem surprising at first, since in a spherically symmetric halo we would expect only the kinetic energy to matter. However, at fixed kinetic energy and infall time, particles on more circular orbits will splashback later than their radial counterparts. If $R_{sp}$ is growing, they will thus be assigned a smaller relative $r_{sp}$.

Finally, the right column of Figure 13 shows the impact of the pericentric distance of the first orbit, that is, how close the particle came to the halo center after its first infall. This distribution is cut off at $r_{peri}/R_{200m} = 1$ as we do not consider trajectories that never entered within $R_{200m}$. The pericentric radius has a surprisingly small effect on $r_{sp}$. One might imagine that $r_{peri}$ would be a good proxy for circularity and thus show a correlation similar to $v_{tot}$. However, $r_{peri}$ is itself positively correlated with both the kinetic energy and circularity at infall. The two effects appear to more or less cancel out so that $r_{sp}$ is within 5% of the mean at all $r_{peri}$.

The cumulative influence of orbital parameters on a halo’s $R_{sp}$ will, of course, be much smaller because the particles in a halo represent a mixture of radial and tangential orbits. We note that the distribution of pericentric radii depends somewhat on time resolution at small radii, which have to be interpolated if the snapshot spacing is coarse (Section 4.6). Otherwise, the distributions shown in Figure 13 are not significantly different when TestSim200 is used instead of TestSim100.

6.3. Baryonic and Resolution Effects

The results presented in the previous section allow us to assess the potential importance of baryonic effects. After all, the inner density profile and angular momentum distribution can change significantly between DM-only and hydrodynamic simulations (e.g., Velliscig et al. 2014; Zhu et al. 2017). However, the kinetic energy of particles at infall should not be altered much by the presence of baryons, meaning that baryonic effects are unlikely to influence $r_{sp}$ significantly (on average; the orbits of individual particles might change dramatically). We plan to test this conjecture directly by running SPARTA on cosmological simulations with DM-only and hydrodynamic incarnations, such as Illustris (Vogelsberger et al. 2014).

A similar argument applies to some of the well-known resolution effects in $N$-body simulations. The orbits of particles in halos almost invariably suffer from numerical inaccuracies due to the finite mass and force resolutions of simulations, particularly near the halo center, where many particles are clustered within a few force resolution lengths (Moore et al. 1998; Klypin et al. 2001; Power et al. 2003; Diemand et al. 2004a). Thus, the exact orbital trajectories of individual
particles may not be reliable. As long as energy is conserved, however, their splashback radii should be unaffected by such issues. We note that two-body relaxation should only play a role at early times when halos are resolved by few particles (Diemand et al. 2004b).

6.4. Extreme Mass Accretion Rates

We have inspected some halos with extremely low or high mass accretion rates in order to ascertain whether the \( R_{\text{sp}} \) measurements in those regimes are trustworthy. We find that negative accretion rates are virtually always caused by a disruption due to mergers, a situation in which \( R_{200\text{m}} \) may not be well defined in the first place. While the halo finder may suddenly assign a lower \( R_{200\text{m}} \) to a merging pair of halos, the splashback radius does not react in the same way. However, \( R_{\text{sp}} \) does not necessarily capture the effects of a merger either, however, their splashback radii should be unaffected by such issues. We note that two-body relaxation should only play a role at early times when halos are resolved by few particles (Diemand et al. 2004b).

6.4. Extreme Mass Accretion Rates

We have inspected some halos with extremely low or high mass accretion rates in order to ascertain whether the \( R_{\text{sp}} \) measurements in those regimes are trustworthy. We find that negative accretion rates are virtually always caused by a disruption due to mergers, a situation in which \( R_{200\text{m}} \) may not be well defined in the first place. While the halo finder may suddenly assign a lower \( R_{200\text{m}} \) to a merging pair of halos, the splashback radius does not react in the same way. However, \( R_{\text{sp}} \) does not necessarily capture the effects of a merger either, because it would take on the order of a dynamical time for the increased mass to translate into a larger \( R_{\text{sp}} \). The splashback mass, however, does “feel” a merger instantaneously because it is simply defined as the mass within \( R_{\text{sp}} \). Thus, \( M_{\text{sp}}/M_{200\text{m}} \) can reach very large values during mergers, describing a physical reality.

Surprisingly, very high accretion rates also correspond to mergers in most cases, but they indicate a situation where a large subhalo is accreted without disrupting the host sufficiently to lead to a spurious decrease in \( M_{200\text{m}} \). In such situations, \( R_{\text{sp}}/R_{200\text{m}} \) seems to approach a constant, and \( M_{\text{sp}} \) also tracks \( M_{200\text{m}} \) closely. Thus, the flattening of the \( R_{\text{sp}}/R_{200\text{m}} \) and \( M_{\text{sp}}/M_{200\text{m}} \) relations at high \( \Gamma \) appears to be physical.

7. Summary

We have described a code framework and specific algorithm to compute the splashback radius of dark-matter halos in cosmological N-body simulations from particle dynamics. Our main conclusions are as follows:

1. We have introduced SPARTA, a versatile, parallel analysis framework for particle-based simulations that will be described in detail in a future publication.
2. Within this framework, we have presented an algorithm that tracks the orbits of particles and subhalos about their host halo centers. Based on only four stored time bins, SPARTA can reliably determine the time and radius of infall, pericenter, and first apocenter (splashback).
3. We find that particles in large subhalos (mass ratio 0.01 or greater) have lowered splashback radii due to dynamical friction. After excluding such particles, we smooth the splashback distribution in time with a Gaussian filter to obtain the halo’s splashback radius, \( R_{\text{sp}} \).
4. Our algorithm can determine \( R_{\text{sp}} \) for 95% of all host halos with more than 1000 particles within \( R_{200\text{m}} \), while the completeness drops at lower particle numbers and for the smallest simulation boxes at low redshifts. The remaining halos are splashback halos that recently became a host halo again.
5. We test the convergence properties of our algorithm with respect to mass resolution, snapshot spacing, and a number of other parameters. We find that the average splashback radii of halos are converged to better than 5% for halos with at least 1000 particles if the simulation has about 100 or more saved snapshots. This convergence extends to roughly the 85th percentile of the particle splashback distribution and degrades somewhat for the highest percentiles.
6. We compare our measurements of \( R_{\text{sp}} \) to those from the SHELLFISH code of Mansfield et al. (2017), and we find the best agreement if the 87th percentile of the apocenter distribution is used to define \( R_{\text{sp}} \). The algorithms agree to a few percent on average, with about 15% halo-by-halo scatter.

We are planning a number of improvements to the SPARTA code, as well as further investigations into the physical meaning of the apocenter distribution. For example, it remains to be explored how the SPARTA results are related to other methods of measuring caustics in the DM density field, such as the algorithm of Vogelsberger & White (2011) or ORIGAMI (Falck et al. 2012; Neyrinck 2012). Inspired by the results of Mansfield et al. (2017), we aim to extend our analysis to the distribution of splashback events in angular space to measure three-dimensional splashback surfaces. We intend to apply SPARTA to particle-based hydrodynamical simulations such as Illustris (Vogelsberger et al. 2014) in order to investigate how \( R_{\text{sp}} \) correlates with the properties of gas and stars in galaxies and clusters. The SPARTA code will eventually become publicly available.

We thank Peter Behroozi for making his halo finder and merger tree code publicly available, for fixing certain issues related to mass definitions for the purpose of this work, and especially for his tree code, which was adapted for SPARTA. We are grateful to Andrey Kravtsov, Surhud More, Philip Mansfield, Nicola Amorisco, Andi Burkert, Neal Dalal, Lars Hernquist, and Laura Sales for helpful discussions. We are especially indebted to Surhud More and the Kavli IPMU in Tokyo for their hospitality during a visit when key parts of the SPARTA code were designed. B.D. gratefully acknowledges the financial support of an Institute for Theory and Computation Fellowship. This work made extensive use of the MIDWAY computing cluster at the University of Chicago Research Computing Center.

References

Adhikari, S., Dalal, N., & Chamberlain, R. T. 2014, JCAP, 11, 19
Adhikari, S., Dalal, N., & Clampitt, J. 2016, JCAP, 7, 022
Baxter, E., Chang, C., Jain, B., et al. 2017, ApJ, 841, 18
Behroozi, P. S., Wechsler, R. H., Lu, Y., et al. 2014, ApJ, 787, 156
Behroozi, P. S., Wechsler, R. H., & Wu, H.-Y. 2013a, ApJ, 762, 109
Behroozi, P. S., Wechsler, R. H., & Wu, H.-Y., et al. 2013b, ApJ, 763, 18
Benson, A. J. 2005, MNRAS, 358, 551
Bertschinger, E. 1985, ApJS, 58, 39
Bond, J. R., Kofman, L., & Pogosyan, D. 1996, Naturo, 380, 603
Boylan-Kolchin, M., Ma, C.-F., & Quinn, T. 2008, MNRAS, 383, 93
Bryan, G. L., & Norman, M. L. 1998, ApJ, 495, 80
Busch, P., & White, S. D. M. 2017, arXiv:1702.01682
Chandrasekhar, S. 1943, ApJ, 97, 255
Crocce, M., Pueblas, S., & Scoccimarro, R. 2006, MNRAS, 373, 369
Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, ApJ, 292, 371
Diemand, J., Moore, B., & Stadel, J. 2004a, MNRAS, 353, 624
Diemand, J., Moore, B., Stadel, J., & Kazantzidis, S. 2004b, MNRAS, 348, 977
Diemer, B., & Kravtsov, A. V. 2014, ApJ, 789, 1
Diemer, B., & Kravtsov, A. V. 2015, ApJ, 799, 108
Diemer, B., Kravtsov, A. V., & More, S. 2013, ApJ, 779, 159
Diemer, B., Mansfield, P., Kravtsov, A. V., & More, S. 2017, ApJ, 843, 140
Efstathiou, G., Davis, M., White, S. D. M., & Frenk, C. S. 1985, ApJS, 257, 241
Eisenstein, D. J., & Hu, W. 1998, ApJ, 496, 605
Eisenstein, D. J., & Hu, W. 1999, ApJ, 511, 5
