
**From the mass-concentration relation of
haloes to GPUs and into the web: A
guide on fully utilizing super computers
for the largest, cosmological
hydrodynamic simulations**

Antonio Ragagnin



München 2018

**From the mass-concentration relation of
haloes to GPUs and into the web: A
guide on fully utilizing super computers
for the largest, cosmological
hydrodynamic simulations**

Antonio Ragagnin

Dissertation
an der Fakultät für Physik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Antonio Ragagnin
aus Pordenone (Italien)

München, den 27.10.2018

Erstgutachter: PD Dr. Klaus Dolag

Zweitgutachter: Prof. Dr. Michael Bader

Tag der mündlichen Prüfung: 13 Dezember 2018

Contents

1	Introduction	1
1.1	Historical overview of cosmological simulations	1
1.2	Ingredients of cosmological simulations	3
1.3	Numerical techniques	4
1.3.1	N-body codes	5
1.4	Gravity solvers	7
1.4.1	Direct Sum	8
1.4.2	Tree	9
1.4.3	Particle-Mesh (PM)	10
1.4.4	P ³ M	11
1.4.5	Tree-PM	11
1.5	Domain Decomposition	12
1.6	Hydrodynamics	15
1.6.1	Grid codes	15
1.6.2	Lagrangian codes (SPH)	16
1.7	Cosmological Simulations	19
1.7.1	Galaxy Clusters	21
1.7.2	Simulations vs. Observations	23
1.8	The Gadget3 code	26
1.8.1	Gadget's tree walk	27
1.8.2	SPH	30
1.8.3	Cooling	30
1.8.4	Star Formation and stellar feedback	31
1.8.5	Thermal Conduction	32
1.8.6	Black holes growth and AGN feedback	32
1.8.7	Halo Finder	33
1.8.8	Gadget3 bottlenecks	34
1.9	Modern large cosmological hydrodynamic simulations	35
1.9.1	Magneticum Simulations	35
1.9.2	Delivering large simulation data	36
1.10	Overview of this thesis	37

2	Exploiting the Space Filling Curve Ordering of Particles in the Neighbour Search of Gadget3	39
2.1	Neighbour Search in Gadget3	40
2.1.1	Impact of the Neighbour Search in the Performance	40
2.1.2	Implementation of Neighbour Recycling using a Space Filling Curve	42
2.1.3	Adaptive Neighbour Recycling	42
2.1.4	Side Effects of Neighbour Recycling	43
2.2	Speedup of the Recycling Neighbours Approach on Gadget3	43
2.2.1	The Test Case	43
2.2.2	Results	44
2.3	Conclusions	47
3	The dependency of halo concentration on mass, redshift and fossilness in Magneticum hydrodynamic simulations	49
3.1	Numerical Simulations	50
3.2	The dependency of concentration on mass and redshift	52
3.3	Concentration and fossil parameter	57
3.3.1	Concentration evolution in time	58
3.3.2	Concentration as a function of the fossil parameter	60
3.4	Virial ratio and concentration	62
3.5	Conclusions	65
4	A web portal for hydrodynamic, cosmological simulations	67
4.1	Introduction	68
4.2	The Simulations	70
4.2.1	The Magneticum Project	70
4.2.2	The Metadata	71
4.2.3	Raw Simulation Data Layout	71
4.3	Structure of the web portal	73
4.3.1	Overview of the multi-layer architecture	73
4.3.2	The Web Portal	73
4.3.3	Job Control Layer	77
4.3.4	Compute Layer and HPC storage	78
4.3.5	Implementation	78
4.4	Exploring Simulations	79
4.4.1	Selecting Objects by browsing maps	79
4.4.2	Composing Queries	82
4.5	The Services	83
4.5.1	ClusterInspect	83
4.5.2	SimCut	84
4.5.3	SMAC	84
4.5.4	PHOX	86
4.6	Conclusions	90

4.7	Credits	92
4.8	Appendix A. Indexing Files	93
4.9	Appendix B. Preparing Simulations for the data center	95
4.9.1	Describing the Simulations	96
4.10	Appendix C. Instrument Configurations	96
5	Gadget3 on GPUs	103
5.1	Gadget3 Parallelisation Infrastructure	104
5.2	GPU porting	105
5.2.1	Memory Transfer	106
5.2.2	Adaptive Timesteps	107
5.2.3	MPI Communication	108
5.2.4	Barnes-Hut, SPH and Conduction porting	108
5.3	Results	108
5.4	Conclusions	111
6	Conclusions	113

Zusammenfassung

Aufgrund ihrer Größe birgt die Durchführung von modernen, hydrodynamischen, kosmologischen Simulationen eine ganze Reihe von Herausforderungen in sich, umso mehr, da hier das Ziel ist, derartige Simulationen für noch größere und besser aufgelöste Volumina durchzuführen. Dabei helfen die Analysen derartiger Simulationen die so verbesserten numerischen Modelle mit Beobachtungsdaten zu vergleichen. Zu aller erst ist es hier unbedingt notwendig in den Anwendungsprogrammen SIMT (Single Instruction Multiple Thread) und/oder SIMD (Single Instruction Multiple Data) Instruktionen zu nutzen, um die Rechenleistung der aktuellen und kommenden Generationen von Supercomputern so gut wie möglich auszunutzen zu können. Darüber hinaus stellt die Datenmenge, welche derartige hydrodynamischen, kosmologischen Simulationen der jüngsten Generation ($> 0.5PB$) generieren, eine große Herausforderung dar, wenn es darum geht wissenschaftliche Ergebnisse und die Simulationsdaten selbst öffentlich zugänglich zu machen und mit anderen Wissenschaftlern zu teilen: Abfragen, Durchsuchen und Filtern einer so großen Datenmenge erfordern Kenntnisse aus dem Bereich des High Performance Computing (HPC). Diese gehören häufig nicht zum Ausbildungshintergrund eines auf diesem Gebiet tätigen Wissenschaftlers. Da für solche Datenmenge HPC Speichersysteme zum Einsatz kommen, wird der öffentliche Zugriff auf diese wissenschaftlichen Daten und Ergebnisse zusätzlich stark erschwert, da der Zugriff üblicherweise für Wissenschaftler nur mit einem Account für die jeweilige HPC-Einrichtungen möglich ist.

In dieser Arbeit werden diese verschiedenen Aspekte miteinander verknüpft. Eine ausführliche Optimierung des zugrundeliegenden Simulationsprogramm (Gadget3) erlaubte es, die bisher größte derartige Simulation durchzuführen. Diese erlaubt durch ihren großen dynamischen Bereich die Wechselwirkung der sichtbaren und der dunklen Materie auf die zugrundeliegende Form der sich bildenden Halos sehr genau und über einen großen Bereich, von Galaxien bis zu den massivsten Galaxienhaufen, zu studieren. Des Weiteren wurde ein Webinterface entwickelt, mit denen die Ergebnisse derartiger Simulationen mit anderen Wissenschaftlern einfach und umfangreich geteilt werden können.

Im ersten Schritt ergab die detaillierte Performance-Analyse, dass der Prozess des Suchens von Nachbarpartikeln, welche mit einem Teilchen in direkter Wechselwirkung stehen (im Folgenden Neighbour-Search genannt), die Zeit, welche für die Berechnung der eigentlichen physikalischen Prozesse zur Verfügung steht, stark limitiert. Dieser Neighbour-Search besteht darin, einem Baum-Graphen zu folgen und die in ihm angeordneten, benachbarten Teilchen entweder mithilfe eines Öffnungskriteriums des Baumes (in der Regel

ein geometrisches Kriterium, welches den erlaubten Fehler in der Gravitationsberechnung beschreibt) oder mit Hilfe einer vordefinierten Suchlänge (z. B. die in der Hydrodynamik verwendete Kernelgröße) zu sammeln. Wir haben das oben skizzierte Problem mit Hilfe eines Optimierungsschemas, welches wir als Neighbour-Recycling bezeichnen, auf ein akzeptables Maß vermindert. Dafür nutzen wir die raumfüllende Peano-Kurve, welche in der Gadget-Implementierung für den Lastbalance-Algorithmus verwendet wird. Diese sorgt dabei dafür, dass im simulierten Volumen benachbarte Teilchen auch im Hauptspeicher des Rechners benachbart abgelegt werden. Wir minimierten den oben genannten Engpass, indem wir die Teilchen in Blöcken gruppieren, welche fortlaufend im Hauptspeicher liegen und die normale Neighbour-Search für die einzelnen Teilchen durch einen Neighbour-Search mit entsprechend leicht größerer Suchlängen ersetzen.

Dieses oben beschriebene Code-Refactoring eröffnete die Möglichkeit, die Anwendung Gadget3 auf das gesamte SuperMUC-System zu skalieren und so die bis dato größte, hydrodynamische, kosmologische Simulation (Box0/mr, Magneticum-Projekt, ¹) durchzuführen. Wir analysieren die Ergebnisse dieser Simulation und zweier weiterer Magneticum Simulationen und vergleichen den Konzentrationsindex (Verhältnis Cluster-Radius zu Skalen-Radius) von Galaxien und Galaxienhaufen mit früheren Theorie- und Beobachtungsarbeiten zu diesem Thema in einem bis dahin für hydrodynamische Simulationen unerreichten Massenbereich. Unter Berücksichtigung von baryonischer Materie in den Magneticum-Simulationen konnten wir zeigen, dass sogenannte fossile Gruppen (d. H. Objekte, die den größten Teil ihrer Lebensdauer existiert haben ohne Störungen ausgesetzt gewesen zu sein) den höchsten Wert von Konzentrationsindex haben. Dies ist in klarer Übereinstimmung mit aktuellen Beobachtungen.

Um diese größten Simulationen öffentlich zugänglich zu machen, haben wir ein Webportal entwickelt und implementiert. In diesem kann ein Benutzer zunächst eine Simulation auswählen und interessante Galaxienhaufen suchen und auswählen. Dazu stehen einerseits sehr hoch aufgelöste 2D-Karten zum Navigieren zur Verfügung, andererseits erlaubt ein Interface komplexe Abfragen von Detailinformationen der Galaxie und Galaxienhaufen. Diese können sogar grafisch zusammengestellt werden und als Elemente in benutzerdefinierten Grafen dargestellt werden. Nach der Auswahl eines Objekts (z.B. Galaxienhaufen) kann der Benutzer weitere, über das Portal zur Verfügung stehende Post-Processing Jobs starten. Durch die komplexe Architektur des Portals ist dies möglich, ohne dass der Nutzer einen eigenen Nutzer-Account auf dem HPC-System besitzt. Es reicht eine simple Registrierung im Portal aus. Inzwischen hat das Portal mehr als hundert registrierte Nutzer und es werden jeden Monat tausende von Post-Processing Jobs über das Portal ausgeführt.

Da in der aktuellen und kommenden Generationen von Supercomputern Beschleuniger (z. B. Grafikkarten) eine zunehmend wichtigere Rolle spielen, haben wir des weiteren daran gearbeitet, Gadget3 auch auf GPU-basierten Systemen effizient nutzen zu können. Diese Neuentwicklung in Gadget3 ermöglicht es, rechenintensive Teile der Applikation auch auf den GPUs der entsprechenden Supercomputer auszuführen. Damit eröffnen wir erstmals

¹<http://www.magneticum.org>

die Möglichkeit, die kommende Generation von Supercomputern, welche SIMT-/SIMD freundliche Programme benötigen, hoch effizient nutzen zu können. Tests der neuen Implementierung ergaben eine Beschleunigung von $> 2x$ bei der Verwendung der Beschleuniger für ein realistische Setup, entsprechend einer der hochauflösenden Magneticum-Simulationen.

Abstract

The current generation of hydrodynamic cosmological simulations is facing a number of challenges when it comes to running and analysing larger and more resolved volumes in order to compare numerical studies with observational data. First, simulation codes need to take advantage of Single Instruction Multiple Thread (SIMT) and Single Instruction Multiple Data (SIMD) paradigms in order to fully exploit the computational capabilities of current and future generations of super computers. Additionally, the amount of data generated by the most recent hydrodynamic simulations ($> 0.5PB$) poses a physical barrier when it comes to opening the access and sharing results with the scientific community: queries, look ups and filtering over such high amount of data need High Performance Computing (HPC) skills that are often not part of the background knowledge of a scientist working in such field; additionally, since only HPC storages are capable of containing this amounts of data, this prevents a public access of the scientific results, as scientists are required to apply and register to HPC facilities.

In this work, these different aspects are linked together. A tailored optimisation and a refactoring of a widely used code for simulations (Gadget3) made it possible to perform the largest hydrodynamic simulation so far. As the various simulated matter contents (dark matter, gas, stars and black holes) cluster together to form haloes, by combining three cosmological hydrodynamic simulations it was possible to study in detail the shape of simulated objects with masses from the ones of single galaxies to masses comparable to the largest galaxy clusters of our universe. Furthermore, a web interface has been developed, where the results of simulations can be easily and extensively shared with other scientists. And finally, there has been an effort in shaping and preparing the code for the new and current generation of super computers, by porting it to the GPUs.

The first part of this work shows how the process of neighbour search prevents large Gadget simulations from spending most of its execution time in actual physical computations. The neighbour search consists of walking the tree and collecting neighbouring particles over a tree opening criteria or over a pre-defined searching length. To solve the issue an approach to recycle neighbours is used. This approach takes advantage of the space-filling curve ordering of Gadget particles, which itself ensures that particles that are nearby in memory are also nearby in the simulated volume. This algorithm solves the aforementioned bottleneck by grouping particles in memory-consecutive chunks and by performing less neighbour searches over larger searching length.

This refactoring opens the possibility to scale the code over the whole SuperMUC ma-

chine and run one of the largest cosmological hydrodynamic simulations (Box0/mr and Box2b/hr from the Magneticum project). I analyse the results of this simulation and other two Magneticum simulations and compare the concentration parameter (ratio between cluster radius and its scale radius) of galaxy clusters with previous theoretical and observational works over an unprecedented mass range for hydrodynamic simulations. Thanks to the presence of baryons in Magneticum simulations, it is possible to show how Magneticum simulations predict that fossil groups (i.e. objects that have lived unperturbed for most of their life) have the highest value of concentration values, in agreement with observations.

In order to share and make the largest simulations accessible, a web portal ² is built to let users find interesting clusters by (i) navigating over zoomable 2D maps, (ii) by graphically composing complex queries over galaxy and galaxy cluster properties and (iii) by selecting elements in user-defined plots. Users can then select a galaxy cluster and submit some pre-defined post-processing jobs that will run in a computing cluster, without any actual registration at the computing facility.

As the current and new generations of super computers may employ accelerators (e.g. graphic cards), I present a way to port Gadget3 to the GPUs. The reason of this porting is twofold: besides making it possible to run on the current GPU based super computers, to fully exploit the new generation of super computer it is anyway necessary to have SIMT and SIMD friendly code. A test of the new implementation of the code shows a total speedup greater than 2 in one of the high-resolution Magneticum cosmological boxes.

²<http://c2papcosmosim.uc.lrz.de>

Chapter 1

Introduction

1.1 Historical overview of cosmological simulations

Holmberg (1941) is the very first numerical simulation described in literature of an N-body system under its own gravitational force. Without using computers, light bulbs were used to represent the N bodies. The author computed the gravitational force acting in each object by placing photocells near every light bulb. And since light intensity follows the inverse square-law, the measurements were capable of estimating the Newtonian force.

After few decades, with the help of computers, cosmological numerical simulations were able to sample up to few thousands particles under the effect of their own gravitational force. Peebles (1970) is one of the first works in this field: it explored the formation time of a coma-cluster-like object by sampling their matter content using around 300 particles. It quantifies the formation time of the cluster by evaluating the time it takes for the system to adhere to the virial theorem. Such kind of estimations were already computed with the help of analytical models (see e.g. van Albada, 1961), however numerical approaches were able to test it on a model without over simplifications (e.g. spherical symmetry).

The first simulations had two main problems: they weren't able to recover the long survival time of galaxies inside the dense environments of clusters (see e.g. White and Rees, 1978), and were unable to reproduce the slope of the observed matter density profile. Klypin (2000) points that the first problem was solved by drastically increasing the resolution of simulations, which was possible only with the drastic increase of computing performance over the years; while the second problem was caused by the lack of description of a larger scale structure (clusters were simulated as initially isolated top-hat over-densities), and the solution relied on making use of simulations on cosmological volumes.

Up to the first half of the 1980's, simulations were computing gravitational forces between all pairs of objects. This approach has a computational time that asymptotically scales with the square of the number of particles N , and thus a complexity of $O(N^2)$. To produce relatively precise simulations, works like Aarseth (1985) computed the gravitational force with a fourth-order interpolation scheme in order to maximise the time-step of the integration.

One of the first fully cosmological simulations started around the 1980's, one example is the work from Klypin and Shandarin (1983) that covers more than $100Mpc^3$ with $N = 32^3 = 32768$ particles. The very high number of particles is justified by the integration scheme that samples the gravitational potential on a $3D$ grid. At each time-step, a loop over all particles computes the amount of mass inside each cell of the $3D$ grid. Then a second loop updates the acceleration of each particle by looping over each grid cell. This approach makes the time of total computation of the gravity force to scale with $O(N \cdot N_{grid\ cells})$, where $N_{grid\ cells}$ is the number of grid cells. The initial conditions were obtained by first producing a random distribution of particles and then evolving it analytically as long as the gravitational interaction was in its linear regime, using an approximate theory for large density perturbations by Zel'dovich (1970).

As time passed, a number of improvements have been made both in the direction of making faster computation at the expense of a slightly lower precision and in improving the performance of the codes by exploiting the power of the state of the art of computers. First approaches organised particles in a tree structure where each node of the tree was storing the total mass of the particles inside it (e.g. Jernigan, 1985). In this kind of code, the force contribution from distant tree cells is computed by grouping particles together and collapsing the mass of the particles in the centre of their cells. The tree also helps to lower the time that is necessary to find nearby particles. Those approaches gave a complexity of $O(N \log N)$.

There have been several tree codes presented in the literature, and the so called Barnes-Hut algorithm (Barnes and Hut, 1986a) was the first of this kind, that didn't need to reconstruct the tree at each time step and at the same time, gave well defined upper limits for the error estimation. However, the algorithm still required update of the tree at every time-step. Its main feature is that the volume is recursively subdivided into 8 octants until the sub-volume contains only one particle (such structure is also named an "octree"). After each time-step, particles may move away from the cell of their parent tree cell, as a consequence, the tree cell gets larger. If one does not rebuild the Barnes Hut tree at every timestep, one has to update the cell sizes to take into account particles that left the original volume, and, as a result of which, force computation gets slower because some cells will intersect.

Adaptive time-steps are used in cosmological simulations because particles span a wide range of acceleration values, from void regions (where particles experience smaller accelerations) to clustered regions (where particles experience higher accelerations). The idea is to bin particles for different minimum time-steps, and then to loop over time-bins and eventually update the acceleration of active particles only. All other particles have their velocity and position drifted using the previously cached acceleration (see Quinn et al., 1997, for more details).

In simulating larger and larger volumes, people started rewriting these integration algorithms in a parallel fashion. For instance, (Barnes, 1986) proposes a parallel version of the above mentioned Barnes Hut algorithm. Supercomputers with thousands of CPUs were produced after the nineties (e.g. the Intel Paragon with 4000 CPUs Esser and Knecht, 1993), and thereby starting the production of codes that scale on thousands of CPUs (see

e.g. Warren and Salmon, 1995b) that can exploit such architectures.

Parallelisation happens at different levels:

- Vectorising: a single CPU is capable to run the same instruction on multiple data (also called SIMD, single instruction multiple data).
- Multi-threading: each CPU has multiple threads that share the same memory and can run different instructions in parallel.
- Accelerators: most modern machines have so called accelerators (e.g. graphic cards) where the CPU can "offload" data and code to execute it in the accelerator, and possibly, the CPU is now free to overlap additional computations.
- Multi-node: multiple computers (also called nodes) can pass messages between each other to communicate data. The most common interface for this kind of parallelism is the Message Passing Interface (MPI)¹ that defines a standard of routines to pass data between CPUs.

1.2 Ingredients of cosmological simulations

Cosmological simulations need to take into account the expansion of the universe due to its own matter and energy content. The so called Friedmann-Robertson-Walker (FRW) equations (Friedmann, 1922) are the solution of the general relativity equations (Einstein, 1916) for a homogeneous, expanding universe. To obtain such solution one starts from a generic metric for a homogeneous and isotropic space-time, described by the invariant distance between infinitesimal line elements ds ,

$$ds^2 = c^2 dt^2 - a(t)(dx^2 + f_K(x)^2 x^2 d\omega^2),$$

where K is the Gaussian curvature and it has units of lengths⁻², so that $K = 0$ describes a flat universe, t is the time and $a(t)$, is the so called scale factor that depends only on time and describes the expansion of the universe.

By imposing this homogeneous and isotropic space-time metric on the general relativity equations, one obtains the following equations for the evolution of the scale factor $a(t)$:

$$\frac{\dot{a}(t)^2 + Kc^2}{a(t)^2} = \frac{8\pi G\rho(t) + \Lambda c^2}{3} \quad (1.1)$$

$$\frac{\ddot{a}(t)}{a(t)} = -\frac{4\pi G}{3} \left(\rho(t) + \frac{3p(t)}{c^2} \right) + \frac{\Lambda c^2}{3}, \quad (1.2)$$

where $\rho(t)$ is the density of the universe at time t , $p(t)$ the pressure, G the Newton's gravity constant, Λ a possible vacuum energy density term, and c is the speed of light.

¹<https://www.mpi-forum.org>

The Hubble parameter $H(t)$ is defined as

$$H \equiv \frac{\dot{a}}{a}, \quad (1.3)$$

where its present day value is H_0 , and combining Equations 1.1, 1.2 and 1.3 together, there is a density threshold (called critical density, ρ_c) so that the expansion critically halts (as for a flat space-time with $K = 0$), where

$$\rho_c = \frac{3H^2}{8\pi G}.$$

A universe with a density higher than ρ_c will eventually collapse, while a universe with a density lower than ρ_c will expand indefinitely. The ratio between the density of the universe and the critical density is the so called density parameter Ω , so that

$$\Omega = \frac{\rho}{\rho_c}.$$

The parameter Ω is then typically expressed as the sum of densities of the various components of the universe: matter (with density ρ_m), radiation (with density ρ_r) and the density term from the cosmological constant and their respective density parameters $\Omega_m = \rho_m/\rho_c$, $\Omega_r = \rho_r/\rho_c$ so that $\Omega = \Omega_m + \Omega_r + \Omega_\Lambda$. Cosmological simulations use these as input parameters. They are usually inferred from observations of the temperature fluctuations of the cosmic microwave background (CMB).

1.3 Numerical techniques

While the FRW equations provide the time evolution of the scale factor, the evolution of the matter content of non-homogeneous regions must be computed numerically. In cosmological simulations, the collisionless matter content is described as a fluid interacting only through gravity that obey the Boltzmann equations

$$\nabla\Phi \frac{\partial f}{\partial \vec{p}} = \frac{\partial f}{\partial t} + \frac{\vec{p}}{m \cdot a^2},$$

where f is the phase-space distribution function of the fluid, which is a function of position \vec{x} , momentum $\vec{p} = ma^2\vec{x}$ and the time t .

Gravitational force is then computed using the Poisson equation

$$\nabla^2\Phi = 4\pi Ga^2 (\rho(\vec{x}, t) - \rho_{bg}(t)),$$

where $\rho_{bg}(t)$ is a background density and ρ is the density of the fluid, so that

$$\rho(\vec{x}, t) = \int f(\vec{x}, \vec{p}, t) d^3p.$$

A widely used approach in the so called grid-code is to solve fluid equations in a 3D grid. There, differential operators are solved by replacing it by finite difference approximations. But most codes sample the dark matter content by means of particles.

1.3.1 N-body codes

In general, an N-body system will have a set of properties $y(t)$, where t is the time, and the code will solve differential equations of the kind

$$\frac{dy(t)}{dt} = f(y). \quad (1.4)$$

There are several ways to integrate the variable y from time t to time $t + \Delta t$. The implicit Euler method integrates Equation 1.4 by a discrete version of the equation 1.4 and computing f in $t + \Delta t$:

$$\frac{y(t) - y(t + \Delta t)}{\Delta t} = f(y(t + \Delta t)),$$

which needs the inversion of the function f in order to find an explicit value for $y(t + \Delta t)$. The explicit Euler method, on the other hand approximates the value of f in Eq. 1.4 to $f(y(t))$ as following

$$\frac{y(t) - y(t + \Delta t)}{\Delta t} = f(y(t)), \quad (1.5)$$

this gives the possibility to easily write $y(t + \Delta t)$ as a function of $y(t)$. This method, is numerically simpler, but introduces the assumptions that f does not vary too much in the time interval and it is accurate only to the first order. To be able to reach the second order of accuracy, it is possible to write the differential equation in its integral form as

$$dy = f(y)dt \implies y(t + \Delta t) - y(t) = \int_t^{t+\Delta t} f(y(t))dt. \quad (1.6)$$

The first order Taylor expansion of this equation gives the Euler explicit method, while the second order expansions gives the following more precise evaluation

$$y(t + \Delta t) - y(t) = \int_t^{t+\Delta t} f(y(t))dt \simeq f(t)\Delta t + f'(t)\frac{\Delta t^2}{2}. \quad (1.7)$$

The problem of solving Equation 1.7 is that one needs to compute the derivatives of f in order to find $y(t + \Delta t)$. To approximately solve Equation 1.7 it is possible to rewrite the derivative $f'(t)$ in its discrete form $((f(t + \Delta t) - f(t)) / \Delta t)$, so that

$$y(t + \Delta t) - y(t) dt \simeq \frac{f(y(t))\Delta t + f(y(t + \Delta t))}{2}, \quad (1.8)$$

which still have the problem of implicitly containing $y(t + \Delta t)$ in f . A widely used approach to find y from 1.8 and 1.7 is to first solve Eq. 1.5 and then to substitute the new $f(y(t + \Delta t))$ in Eq. 1.8.

This chain of substitution can be rewritten as

$$y(t + \Delta t) - y(t) dt \simeq \frac{f(y(t))\Delta t + f(y(t) + f(t)\Delta t)}{2}. \quad (1.9)$$

A more generic approach to reach higher orders of accuracy is to use the Gauss quadrature rule. It states that for every function g that can be approximated to a polynomial of degree $2n - 1$, its integral form can be written as

$$\int_a^b g(t)dt \simeq \frac{b-a}{2} \sum_{i=1}^N w_i f\left(\frac{b-a}{2}t_i + \frac{a+b}{2}\right),$$

where w_i are some coefficient and t_i are some intermediate evaluation points between a and b . Once the w_i coefficients are known it is possible to prove that the accuracy of such formula is of order N . For instance, Equation 1.5 is the special for $N = 1$ where the integral boundaries are $a = t, b = t + \Delta t$ and $w_1 = 1, t_1 = y(t)$. And 1.8 is the special case for $N = 2$ where $w_1 = 1/2, t_1 = y(t), w_2 = 1/2$ and $t_2 = y(t) + f(t)\Delta$. t_2 is $y(t)$ computed with an accuracy of $N = 1$.

In general, solving n th Gauss quadrature technique to numerically solve equations at the n th order of accuracy falls into the so called Runge Kutta methods. Runge Kutta intermediate points depend always on the function computed in the previous intermediate points.

A widely adopted Runge-Kutta approach is its fourth-order formulation for the Runge-Kutta, where for a function $f(y, t)$ that depends both on y and t , the value of $y(t + \Delta t)$ is found by the following computations:

$$\begin{aligned} y(t + \Delta t) &= y(t) + \left(\frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}\right) \\ k_4 &= f\left(y(t) + k_3 \frac{\Delta t}{2}, t + \Delta t\right) \\ k_3 &= f\left(y(t) + k_2 \frac{\Delta t}{2}, t + \frac{\Delta t}{2}\right) \\ k_2 &= f\left(y(t) + k_1 \frac{\Delta t}{2}, t + \frac{\Delta t}{2}\right) \\ k_1 &= f(y(t), t) \end{aligned} \tag{1.10}$$

A list of coefficients of the Runge-Kutta higher order solvers can be found in Ascher and Petzold (1998). A main advantage of Runge-Kutta is the possibility of having adaptive time steps, in fact, by evolving y with a higher order Runge Kutta solver it is possible to estimate the error and thus adapt the time-step accordingly.

Another class of solvers are the "leap frog" solvers. They rely on the fact that equations of motion depend at most on the second order derivative. In this technique the velocity is evolved with a half time step shift with respect to the position. This technique falls in the category of the second order accuracy solver and goes as following: (1) define the position x and velocity $v = x'$, and $v' \equiv f$ then (2) the leap frog technique evaluates x_i at t , x_{i+1} at $t + \Delta t$. (and, in general, x_{i+k} is evaluated at $t + k\Delta t$), while (3) the velocity

$v_{i-1/2}$ is evaluated at $t - \Delta t/2$, $v_{i+1/2}$ at $t + \Delta t/2$, and so on. Given these assumptions, the numerical integration for x and v can be written as

$$\begin{aligned} x_{i+1} &= x_i + v_{i-1/2}\Delta t \\ v_{i+1/2} &= v_{i-1/2}\Delta t + f(x_i, t). \end{aligned} \quad (1.11)$$

A drawback of this particular version of the leap frog is that this version is not suitable for an approach with what is called variable (or local) time stepping. Another variant is the so called kick-drift-kick (KDK) scheme where there is a first integration (kick) of the velocity, followed by a drift in the position and followed by a kick in the velocities:

$$\begin{aligned} v_{i+1/2} &= v_i + f(x_i, t)\frac{\Delta t}{2}, \\ x_{i+1} &= x_i + v_{i+1/2}\Delta t, \\ v_{i+1} &= v_{i+1/2} + f(x_{i+1}, t + \Delta t)\frac{\Delta t}{2}, \end{aligned} \quad (1.12)$$

which opens the possibility for variable time-steps. This technique is widely used in cosmological simulations since the gravitational force f does not depend on the velocities. Finally, a commonly used criteria for the time-step size for cosmological simulations is to choose a time-step smaller $t_{j+1} - t_j$ smaller than $\alpha\sqrt{\frac{\epsilon}{|\vec{r}_i|}}$ (see Dolag et al., 2008a, for a review), where ϵ is the error upper limit and α is a proportionality constant.

Springel (2005a) (see Figure 4 of their paper) show how leap frog and KDK produce stable and more precise keplerian orbits than the single-timestep solver (e.g. Runge-Kutta with fourth order accuracy).

1.4 Gravity solvers

Cosmological simulations integrate equation of motions over positions and momenta(see Peebles, 1980, for more details):

$$\begin{aligned} \dot{\vec{p}}_i &= -m_i\nabla\Phi \\ \dot{\vec{x}}_i &= -\frac{\vec{p}}{m_i a^2} \end{aligned}$$

where the index i runs over the particles, and Φ is the gravitational potential at the particle position. In an expanding universe is the velocity is proportional to the scale factor $v_i = a(t)\dot{x}_i$. If the gravitational potential is given by the contribution of the N bodies only, then the acceleration is given by the Newtonian force:

$$\dot{\vec{p}}_i = -G \sum_j \frac{m_i m_j}{|\vec{r}_i - \vec{r}_j|}$$

This approach however gives unwanted results when two particles approach each other, producing a too high and unrealistic acceleration. This happens because particles sample

an ensemble of real dark matter particles within a finite volume, thus the force between two colliding simulated particles shouldn't diverge. To overcome this problem, the so called softening length ϵ is added to the distance evaluation of the potential to prevent it from vanishing (see Bouchet and Hernquist, 1988, for more details). The softening ϵ reduces a spurious two-body relaxation effect, and the force between two particles is computed as

$$\dot{\vec{p}}_i = -G \sum_j \frac{m_i m_j}{\sqrt{(\vec{r}_i - \vec{r}_j)^2 + \epsilon^2}}. \quad (1.13)$$

Power et al. (2003) points that a value of ϵ from 1/20-1/50 of the mean inter-particle separation in the simulation is the minimum value to avoid spurious relaxation effect. Simulations of gravitational interactions deal with potentials having, ideally, an infinite range. This becomes problematic in cosmological simulations with an always higher number of particles in order to compete with surveys that observe more and more well resolved galaxies and clusters. There are several techniques that compromise precision and performance. The direct sum approach (interaction between all pairs of particles) is the most precise and computationally expensive. Other techniques presented below are able to speed up the computation by paying the price of a slightly lower precision.

1.4.1 Direct Sum

The direct sum approach computes the gravitational interaction between all pairs of particles in the simulation (as in Eq. 1.13). It is the most precise approach, but if used alone, it has a complexity $O(N^2)$. For this reason, it is often used in combination with other approximation. Such approximations store the large scale mass distributions over grids or tree structures, and use the direct sum only over short range interactions. When the direct sum is used over short range interactions, the complexity becomes $O(N * N_{ngb})$, where N_{ngb} is the typical number of neighbouring particles of a short range interaction.

The architectures that benefit more from direct sum are the ones with a large number of threads, as GPUs. Some modern super computers have Graphic Processing Units (GPU) in their computing nodes (see for instance Piz Daint in Lugano²). The recent Pascal GPU³ has 32 threads in each of its 64 cores. Graphic cards execute single instruction on multiple threads (SIMT) and they fit extremely well computations as the gravitational force, and cut the computation time of a factor proportional to the number of threads.

The main limitations to the highest speedup are both because of the slow CPU-CPU and CPU-GPU communication that occur when some part of the code must be executed on the CPU or when the problem can't fit a single computing node. In the case of CPU-CPU communication, the simulation communicates over multiple nodes that need to communicate in order to exchange the interaction between pair of particles that lie in different GPUs. Also, because of the low amount of memory of each GPU thread, such commu-

²<https://www.cscs.ch/computers/piz-daint/>

³<https://devblogs.nvidia.com/inside-pascal/>

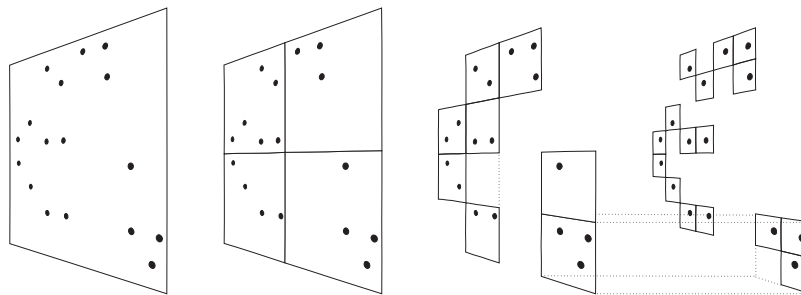


Figure 1.1: 2D representation of a tree structure of particles. Each square is subdivided in 4 sub squares until it contains a single particle. During the force computation the tree walk will stop when the angle subtended by a square is less than a pre-set threshold, or when a leaf contains a single particle. Figure taken by Springel et al. (2001a).

nication usually employ shared buffer for communication , that further compromises the parallelism.

Additionally, a limitation of systems with GPUs is their single precision floating point precision, while most recent simulations, due to the very high number of particles, need double precision floating point numbers in order to avoid round off errors when summing up the contribution of the acceleration from a high number of interacting bodies.

1.4.2 Tree

The idea behind tree codes is to use the direct sum approach only for nearby particles, and to collapse distant particles into point like pseudo-particle with the same mass as the total mass of the particles group. The implementation require a tree structure that recursively subdivides the volume until it contains only one or more particles. This technique needs a threshold aperture angle that is set a priori before the simulation. When one computes the force acting on a given particle, the cells of the tree that subtend an angle lower than the threshold are opened until one of two conditions are met: 1) if the tree cell subtend a angle lower than the threshold, then the cell is considered as a pseudo-particle with the total mass stored inside the cell, or 2) the cell of the tree is a leaf and contains one or more particles and a direct sum is performed over these particles.

The benefit of this approach is that the force computation of each particle requires a tree walk that has a complexity of $O(\log N)$, which gives a total complexity of $O(N \log N)$ for each tree walk.

There are several implementations of this technique. In the Barnes-Hut algorithm (Barnes and Hut, 1986b), the 3D volume is recursively subdivided in 8 sub-volumes until they contain only a single particle. Additionally, every cell of the tree stores the information of the size of the cell and the total mass inside it. Figure 1.1 shows a representation of the Barnes Hut structure taken from Springel et al. (2001a). After a timestep it can happen that some particles leave the subvolume of the tree. To avoids a tree rebuild, the size of the

cell is artificially increased in order to enclose particles that are no more in their original cube. This produces an overhead in the tree walk but avoid a complete tree rebuild.

Another type of tree code is the k-dimensional tree (KD-tree), where the volume is recursively split in half, first in the x-axis, then in the y-axis, then in the z-axis, then in the x-axis again and so on. This kind of tree is a lot easier to implement and used in the code pkdgrav3 (Stadel, 2001; Potter et al., 2017; Ballouz, 2017). With respect to Barnes-Hut, KD-tree has non-cubic cells and requires a higher order multipole expansion of the matter content.

1.4.3 Particle-Mesh (PM)

In this approach the gravitational potential is sampled on a grid, which is then used to compute the force on every particle. The main advantage of this approach is that the number of operations scales with the number of particles times the grid size, while the drawback is that the precision depends on the grid size. The matter sampled inside the grid is then converted to the gravitational potential and then differentiated in order to obtain the acceleration on a given grid.

The simplest way to assign particles to a mesh is by assigning at each particle the nearest grid point (also called, Nearest Grid Point or NGP). If the mesh is not fine enough, then discontinuities in the density field will generate discontinuities in the potential, and consequently a too strong force.

In general, it is preferred to distribute particles in the n nearest grid elements. The most common choice consists in distributing them in the 8 or 27 nearest cells.

Once the density grid is built, the Poisson equation is solved in the Fourier space. In fact the gravitational potential

$$\Phi(\vec{x}) = -G \int_V \frac{\rho(\vec{y})}{|\vec{x} - \vec{y}|} d\vec{y},$$

is transformed in the Fourier space

$$\hat{\Phi}(\vec{k}) = \hat{g}(\vec{k})\hat{\rho}(\vec{k}),$$

where $\hat{\Phi}(\vec{k})$, $\hat{\rho}(\vec{k})$ and $\hat{g}(\vec{k})$ represents the Fourier transform respectively of the potential, the density matter and the Green function $g(\vec{x}) = |\vec{x}|^{-1}$.

The values of \hat{g} can be computed only once in the simulation and re-used during all time steps. The PM code will then evaluate the potential in the Fourier space and transform back the potential in the real space $\Phi(\vec{x})$. Once the field Φ has been computed, the particle-mesh approach differentiates between the nearby grid cells to obtain the contribution of the acceleration acting on the particles inside the given cell. The first order scheme to compute this derivative is to compute difference between the 2 neighbouring grid cells on a given direction. The second order scheme uses the 4 nearby cells on a given direction.

The last step of the PM algorithm is to apply the acceleration to the particles inside the cells. The acceleration is smoother with the same smoothing function used to distribute the mass of particles in the neighbouring cells.

1.4.4 P³M

Being the particle-particle (PP, or direct sum) approach more precise but slower compared to the faster and less precise PM, Hockney et al. (1974) presented a hybrid approach that uses the PP approach at small scales and the PM approach at larger scale. Efstathiou and Eastwood (1981); Hockney and Eastwood (1988); Eastwood (1988); Efstathiou et al. (1985) are the first works in literature that present this approach in detail in both plasma physics and cosmological simulations.

The idea is to split the Fourier transform of the potential $\hat{\Phi}(\vec{k})$ in a short range component $\hat{\Phi}^{short}(\vec{k})$ and long range component $\hat{\Phi}^{long}(\vec{k})$. The division is not sharp and it is applied at a threshold length k_s with an exponential decay of the kind $e^{-\frac{k^2}{k_s^2}}$ (rather than a top hat cut). The value of k_s is chosen so that $k_s \cdot L \gg 1$, where L is the length of the simulated box. The lower k_s the lower is the area where the Barnes Hut approach holds, making the computation faster but dominated by the lower-precision PM approach. The direct sum computation is then damped with the inverse-transform of the decay function, that in this case is the error-function $erf(x)$. Given these premises, the following relations hold

$$\begin{aligned}\hat{\Phi}(\vec{k}) &= \hat{\Phi}^{short}(\vec{k}) + \hat{\Phi}^{long}(\vec{k}) \\ \hat{\Phi}^{long}(\vec{k}) &= \hat{\Phi}(\vec{k}) e^{-\frac{k^2}{k_s^2}} \\ \Phi^{short}(\vec{x}) &= -G \sum_i \frac{m_i}{|\vec{x}_i - \vec{x}|} erf(|\vec{x}_i - \vec{x}| k_s).\end{aligned}$$

At later stages of a cosmological simulation, when clustering becomes strong, clusters will form on scales that are smaller or of the same order of magnitude of the cell sizes. In this regime, the execution time of high-resolution cosmological simulations, is dominated by the cost of the direct summation. Couchman (1991) present an adaptive version of the P^3M , called Adaptive Particle Mesh (APM), where over dense region are sampled by sub-grids. Additionally, the method adopted in Couchman (1991) uses multiple levels of grid refinement to further reduce the cost of direct summation in clustered regions.

Ferrell and Bertschinger (1994) present a parallel scheme for the P^3M algorithm. Their approach (see section 4 of their paper) is to assign a virtual processor at each grid point. First, the local acceleration is evaluated at each grid point (based on its particles) and then send its particles to queues associated to other grid points. As virtual processors receive particles, they will add their contribution to the local particles. As the authors point out, this technique is efficient only in regions with uniform density. This issue is solved in Theuns (1994) where they present a parallel version of the adaptive P^3M code.

1.4.5 Tree-PM

The Tree-PM approach is very similar to the P^3M approach, but the short distance interaction is computed using a Tree algorithm as Barnes-Hut. This approach is used in the family of cosmological simulations codes Gadget (Springel et al., 2001a, where the most

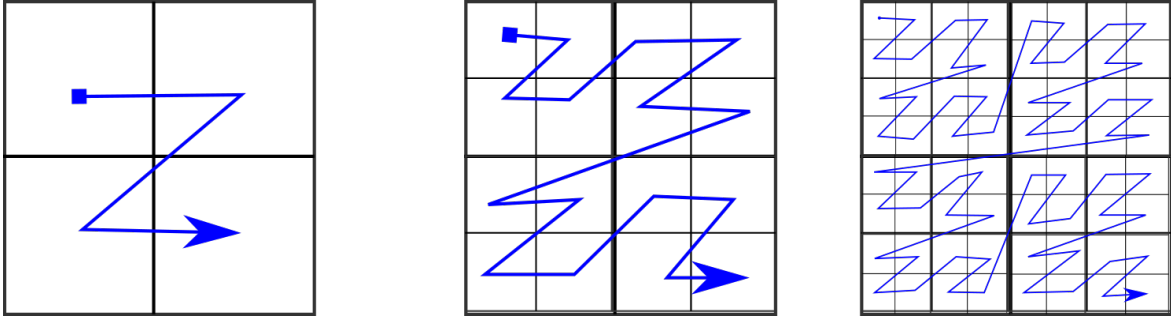


Figure 1.2: A representation of the 2D Morton curve. Each sub volume is recursively linked to the subvolume of the same level with a “Z” shaped curve .

used version GADGET-2 is presented). Dubinski (1996) present one of the first approach on the parallelisation of tree codes. The main idea is to use the so called Orthogonal Recursive Bisection (ORB). ORB employs the use of a volume subdivision similar to the Barnes Hut algorithm, but the main volume is cut along an arbitrary dimension and a split in a position chosen to keep the same number of particles on both sides. The slicing continues by changing axis at every subdivision until it reaches a tree of n levels, that therefore results in 2^n sub-volumes. At this point different sub-volumes are sent to different CPUs and each should have the same amount of particles, which in principle should imply that each CPU has roughly the same amount of interactions to compute. As a consequence, each CPU should spend roughly the same time for each time step.

The force computation algorithm of Dubinski (1996) is difficult to parallelise over multiple CPUs. An important obstacle in this parallel code is the impossibility to practically store a copy of the ORB and Barnes Hut tree in each computing node. Dubinski (1996) solved this problem with the algorithm proposed in Salmon et al. (1990), where nodes store only the part of the ORB tree that contains only the domain subdivision (which is a lot smaller than the whole Barnes Hut tree). During the force computation, when the parallel Barnes Hut finds that a branch of the tree has to interact with a ORB cell in a different node, it sends its own “pruned” branch to the other CPUs. In the same way every CPU receive a “pruned” version of a foreign Barnes Hut tree and computes the interaction of this “guest” branches to their local Barnes Hut tree. After such computation, they send the result back to the original CPU.

1.5 Domain Decomposition

The process of dividing the computational work between nodes is called domain decomposition. The approach used by Dubinski (1996) (see our discussion in Section 1.4.5) to equally split simulated volumes between nodes is not optimal in cosmological simulations with high resolution. In fact, high resolution cosmological simulations have both clustered regions and low dense regions, regions with different densities having different computing

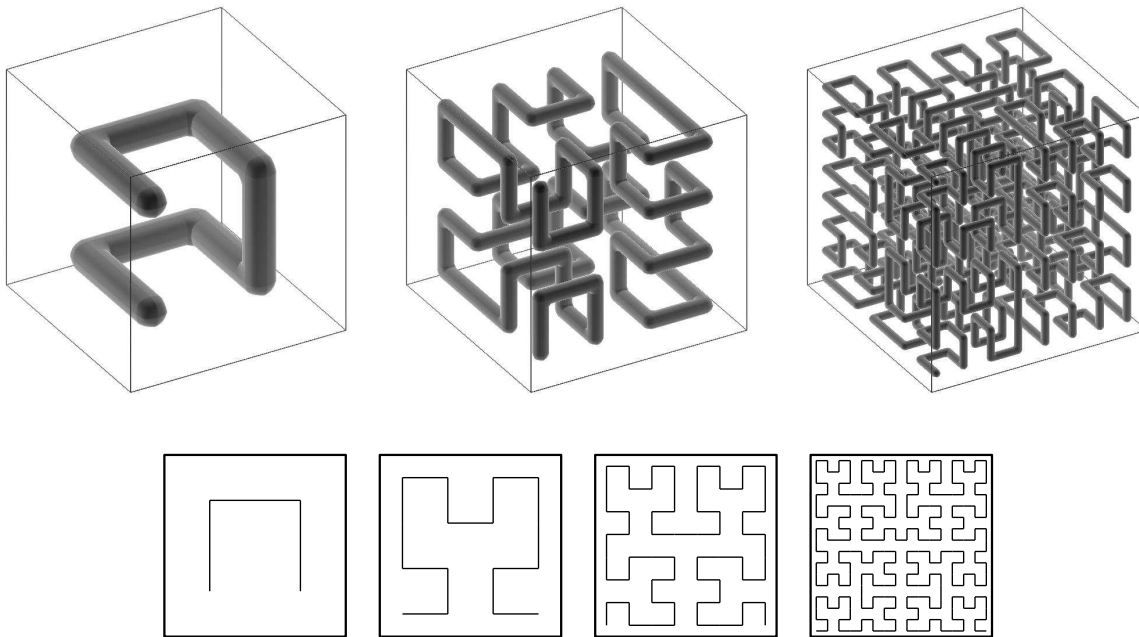


Figure 1.3: $3D$ (top) and $2D$ (bottom) representation of the Space-filling Peano-Hilbert curve . Taken by (Springel, 2005a).

times.

A naive approach to solve this problem would be to split the domain in a large number of cubes so that each CPU will get approximately the same amount of under-dense and over dense regions. A draw back of this approach is an increased communication between CPUs.

A good domain decomposition must minimise the computing time on each CPU and minimise the communication between CPUs. One of the first approaches to solve this problem is presented in Warren and Salmon (1995a). In that work the authors experiment different domain decompositions for a Barnes Hut code. Besides the ORB approach they order the particles according to the so called Morton curve. A representation of this ordering is shown in Figure 1.2. The ordering is very similar to the ORB split, but the subvolume is divided at half of the size, instead of the point that contains divide half particles. Particles are then re-organised in memory so that chunks that stay in one side of the volume are contiguous. As a result, each sub volume is recursively linked to the subvolume of the same level with a “Z” shaped curve . After this step, the array of particles is divided in chunks of equal size. Since dark matter particles automatically sample the most dense regions, this approach gives both compact domains (minimised the communication) that are also equally balanced.

Most modern domain decomposition techniques, as the most recent versions of Gadget (Springel, 2005a) employs the Peano-Hilbert space-filling curve. Figure 1.3 show a sketch of the Peano curve taken by Springel (2005a). The peculiarity of this curve is that it forms

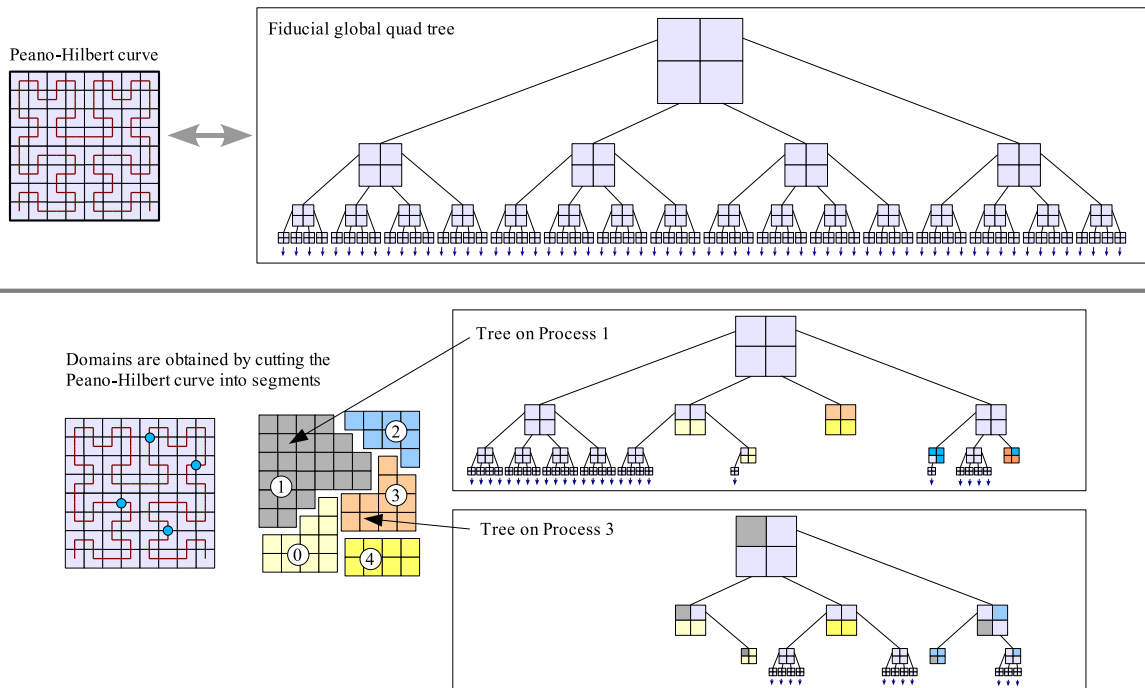


Figure 1.4: 2D Representation of the Gadget2 domain decomposition. Top panel shows how the particles are ordered and set up in a Barnes Hut tree, bottom panel shows how the Hilbert a space filling curve is divided in chunks and the corresponding top-nodes of three in each CPU, where they store both the Barnes Hut data and the information of which top-node belongs to which CPU (represented with different colours). Figure taken by (Springel, 2005a).

a “U” shape (in 2D) that fills the cells of the same level. In the limit of infinite recursion the curve continuously fills the whole volume.

In three dimensions, a basic curve is extended in an analogous way. One interesting property of these space-filling curves is their self similarity. A cut of the space-filling curve gives spatial domains that are compact and are simply connected. The benefits are that (i) it minimises the surface and thus the communication between CPUs and (ii) that each CPU gets the same amount of clustered particles. Additionally, Springel (2005a) orders the Barnes Hut cells with the same space filling curve ordering, and as a consequence it is possible to store a so called top-nodes tree, with the domain decomposition information (what node correspond to what CPUS) and the Barnes Hut tree in the same data structure.

Every CPU will then have the same top-nodes, and they are used to know what cell corresponds to which CPU. Additionally they also store their own local copy of their Barnes Hut tree. During the computation of a force, every CPU is able to know if it has to communicate with another CPU and will send its particles according to this scheme.

1.6 Hydrodynamics

While gravity follows the Newtonian force between particles, to describe gas physics it is necessary to include a solver of the the comoving Euler equations for an inviscid gas. Such equations are

$$0 = \frac{d\rho}{dt} + \rho \nabla \cdot \vec{v}, \quad (1.14)$$

$$0 = \frac{d\vec{v}}{dt} + \frac{\nabla P}{\rho} + \nabla \Phi, \quad (1.15)$$

$$0 = \frac{du}{dt} + \frac{P}{\rho} \nabla \cdot \vec{v} + \frac{\Lambda(u, \rho)}{\rho}, \quad (1.16)$$

where ρ is the density of the fluid, \vec{v} is the velocity, P the pressure, u the internal energy, Φ is the gravitational potential and the total derivative d/dt is $\partial/\partial t + \vec{v} \cdot \nabla$. These equations are closed by including also the equation of state, that for a mono-atomic ideal gas (and thus is described by an adiabatic index $\gamma = 5/3$) is

$$P = (\gamma - 1) \rho u.$$

1.6.1 Grid codes

Grid codes use - for example - finite-difference methods to approximate numerically the evolution of Eulerian fluids. Cen (1990) is one of the first grid codes for hydrodynamic cosmological simulations, while these codes performed reasonably well in simple test cases, they failed to correctly resolve shocks.

This is very important in cosmological simulations, where gas that falls into clusters of galaxies is shock-heated from 10^4 to 10^8 Kelvin, and thus simulations need to resolve temperature discontinuities between grid cells (see e.g. Bertschinger, 1998, section 2.2.2.

of their paper for a review). More modern approaches use an approximated solution of the Riemann problem between cell boundaries and so to evolve fluids with discontinuities. The total-variation diminishing (TVD) technique, compute fluxes using the Riemann solver and multi cell interpolation to compute derivatives. Plus, codes as Quilis et al. (1996) employ slope limiters to avoid nonphysical post-shock oscillations.

Anninos and Norman (1994) present the first grid code where more dense regions were described with a fixed mesh refinement and it was used in a cosmological simulation, while Bryan and Norman (1997) present the first adaptive mesh refinement (AMR) code for cosmological simulations. There, using 64^3 grid cells then are able to reach a resolution equivalent to 8192^3 fixed-resolution grid cells (see e.g. Bertschinger, 1998, for more details).

A widely used approach to incorporate dark matter in grid codes is to describe dark matter as an N -body system (see e.g. the ENZO code presented in Bryan and Norman, 1995).

1.6.2 Lagrangian codes (SPH)

On the other hand, a technique called Smoothed Particle hydrodynamics (SPH) is used to sample the Eulerian version of the equations on particles. SPH samples the properties of a fluid (e.g. density, pressure, temperature) in particles that can exchange momentum and thus move around the simulated volume in order to oversample more dense regions.

The main difference between the two techniques in the integration scheme of hydrodynamics is that a grid code computes fluxes between grid elements (as would be originally implied by the Euler equation) while the N-body scheme SPH smoothes the fields and then approximates the equation as sum over particles (see Price, 2012, for a review).

In the basic version of SPH from Gingold and Monaghan (1977) and Lucy (1977) (see also Gingold and Monaghan (1982)), for any field $F(\vec{r})$, one defines its smoothed version $F_s(\vec{r})$ as

$$F_s(\vec{r}) = \int F(\vec{s})W(\vec{r} - \vec{s}, h)d\vec{s},$$

where the integral runs over the volume covered by the fluid and sampled by the SPH particles, W is the so called smoothing kernel that weights all contributions from the particles as a function of their distances from \vec{r} .

It is required by the theory that

$$\lim_{h \rightarrow 0} W(\vec{r}, h) = \delta(\vec{r}),$$

where δ is the Dirac delta, so that $F_s \rightarrow F$ for $h \rightarrow 0$.

The above integral thus, can be approximated to a sum over particle properties as:

$$F_s(\vec{r}) \simeq \sum_j \frac{m_j}{\rho_j} F_j W(\vec{r} - \vec{r}_j, h),$$

where j runs over the particles index, ρ_i, m_i, \vec{r}_i are respectively the density, the mass and the position of the i -th, particle and F_j is the value of the field associated with the particle j .

If W has compact support (i.e. $W(|\vec{x}| > h, h) = 0$) then the summation is performed over on a finite number of neighbouring particles.

To exploit SPH in cosmological simulation, one need to chose the smoothing length so that the kernel will automatically adapt to change in the density environment. Since h gives information on the radius of a kernel, Steinmetz and Müller (1993) proposed to evaluate h so that

$$\rho_i \cdot h_i^3 = \text{const.}$$

SPH will perform multiple iterations over the particles in order to adjust the smoothing length so that it includes approximately a pre-fixed number of neighbours. This is, by the way, a very expensive process because it requires to find all neighbours within a certain sphere for each gas particle. Multiple iterations of neighbours finding must be executed for particles with an exceeding or underestimated smoothing length. A symmetric version of the window function W is defined as W_{ij} . There are two main choices for a symmetric W , one that is to compute W in the average h ,

$$W_{ij} = W(|\vec{x}_i - \vec{x}_j|, h_{ij}),$$

where h_{ij} is the mean between h_i and h_j . A second option is to define the symmetric version as the mean W as

$$W_{ij} = \frac{W(|\vec{x}_i - \vec{x}_j|, h_i) + W(|\vec{x}_i - \vec{x}_j|, h_j)}{2}$$

Note that in the next computations, the derivative of W should have terms like $\partial W / \partial h$, that is ignored in most SPH formalism. A widely used SPH kernel W is the so called cubic-spline kernel, defined as

$$W_{3D}(x, h) = \frac{\sigma}{h^\nu} \begin{cases} 1 - 6 \left(\frac{x}{h}\right)^2 + 6 \left(\frac{x}{h}\right)^3 & 0 \leq \frac{x}{h} < 0.5 \\ 2 \left(1 - \frac{x}{h}\right)^3 & 0.5 \leq \frac{x}{h} < 1 \\ 0 & 1 \leq \frac{x}{h} \end{cases}, \quad (1.17)$$

where ν is the number of dimensions and σ is a normalisation constant that varies with the number of dimensions of the problem; for a 1D, 2D or 3D problem the value of σ is respectively $16/3, 80/7\pi, 8/\pi$. This kernel has the advantage of being both compact and continuous in the second derivative. The advantage of having a continuous second order derivative is a lower error in interpolating the integration by summation interpolants, see Section 7 in Monaghan (1992) for more details.

Given a kernel that has smooth derivatives, it is possible to find a SPH description for field derivatives. The derivative of a vector field \vec{A} can be naively defined as

$$\left(\nabla \cdot \vec{A}\right)_i = \sum_j \frac{m_j}{\rho_j} \vec{A}_j \frac{\partial W(\vec{r}_i - \vec{r}_j)}{\partial x_i}. \quad (1.18)$$

Noteworthy, if A is the velocity field, then the divergence of a velocity field does not go to zero if all velocities are equal. There are other, more numerically stable ways to compute the derivative of a field. Concerning the identity

$$\nabla \left(\rho \vec{A} \right) = \rho \nabla \cdot \vec{A} + \vec{A} \cdot \nabla \rho,$$

one can rewrite Equation 1.18 as

$$\left(\nabla \cdot \vec{A} \right)_i = \frac{1}{\rho_i} \sum_j m_j \left(\vec{A}_j - \vec{A}_i \right) \cdot \frac{\partial W(\vec{r}_i - \vec{r}_j, h_i)}{\partial x_i}$$

Such approximation of $\nabla \cdot \vec{A}$ has the advantage of providing a vanishing velocity in case the field A is evaluated on the velocity field \vec{v} . Another widely used formulation for field derivatives is by employing the identity

$$\nabla \cdot \left(\frac{\vec{A}}{\rho} \right) = \frac{\nabla \cdot \vec{A}}{\rho} - \vec{A} \cdot \left(\frac{1}{\rho^2} \nabla \rho \right)$$

so one obtains a symmetric form for the field derivative:

$$\left(\nabla \cdot \vec{A} \right)_i = \rho_i \sum_j m_j \left(\frac{\vec{v}_j}{\rho_j^2} + \frac{\vec{v}_i}{\rho_i^2} \right) \frac{\partial W(\vec{r}_i - \vec{r}_j, h_i)}{\partial x_i}.$$

It is also possible to derive those equations by using a Lagrangian for a fluid $L[\rho, \vec{v}] = \int_V \frac{1}{2} \rho(\vec{r}) \vec{v}^2 + \rho(\vec{r}) m(\vec{r}) d\vec{r}$ and applying the variational principles on it (Eckart, 1960).

Theoretically the kernel function W that smoothes the field can cover the whole volume, this would lead to a summation over all particles for each particle to process, and give the algorithm a complexity of $O(N^2)$ which will disrupt all efforts made (for instance Barnes Hut approach) to lower the complexity to $O(N \log(N))$ simulations with gravity. For this reason people use kernels that vanish after a certain distance.

In the classical SPH formulation energy, linear momentum, angular momentum and entropy are all simultaneously conserved (Springel, 2011). This has a drawback when gas forms shock waves. Shock waves and contact discontinuities produced by gas dynamics open the necessity to use the integral form of Euler equation, as opposed to the above use of their differential form (Landau and Lifshitz, 1959). One of the main consequences of shock fronts is that the inviscid description breaks down and increases the entropy. The consequence for SPH codes is that they cannot naturally describe shocks, as they keep the entropy constant. For this reason people add an artificial term that dissipates the kinetic energy into heat, and increases the total entropy accordingly. Such term is effectively a friction force that has the effect of dumping relative differences in particle velocities. The drawback of this approach is, of course, that one must add a mechanism so that it activates only on shocked regions, or, as a drawback one would break the conservative properties of SPH (Springel, 2011).

The viscous term that is most often added in SPH codes is

$$\left(\frac{d\vec{v}_i}{dt}\right)^{viscosity} = - \sum_j \frac{1}{2} m_j \Pi_{ij} \nabla (W(\vec{r}_i - \vec{r}_j, h_i) + W(\vec{r}_i - \vec{r}_j, h_j)), \quad (1.19)$$

where Π_{ij} is chosen to be a symmetric matrix that provides the viscosity properties so that linear and angular momentum are still conserved. To still conserve energy, an additional energy loss Π_{ij} is added in the internal energy equation

$$\left(\frac{du_i}{dt}\right)^{viscosity} = - \sum_j \frac{1}{2} m_j \Pi_{ij} (\vec{v}_i - \vec{v}_j) \cdot \nabla (W(\vec{r}_i - \vec{r}_j, h_i) + W(\vec{r}_i - \vec{r}_j, h_j)). \quad (1.20)$$

Monaghan and Gingold (1983) introduced what is still the most used candidate for Π_{ij}

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\rho_{ij}} & \text{if } (\vec{v}_i - \vec{v}_j) \cdot (\vec{r}_i - \vec{r}_j) < 0 \\ 0 & \text{otherwise} \end{cases}. \quad (1.21)$$

where

$$\mu_{ij} = \frac{h_{ij} (\vec{v}_i - \vec{v}_j) \cdot (\vec{r}_i - \vec{r}_j)}{|\vec{r}_i - \vec{r}_j|^2 + \epsilon h_{ij}},$$

and h_{ij} denotes the average smoothing length between a particle i , j and c_{ij} is the average sound speed, and the parameters α have typical values from 0.5 – 1 and $\beta = 2\alpha$ and $\epsilon = 0.01$ (see Springel, 2011).

Springel (2010) proposes a novel scheme, implemented in the code AREPO, that eliminates both the suppression of fluid instabilities in SPH and the Galilean invariance of AMR. The approach uses an unstructured mesh that is represented by a Voronoi tessellation generated by points that can move arbitrarily inside the simulated volume. Although employing particles, the mesh is used to solve the ideal hydrodynamic equations with hyperbolic conservation laws based on a Riemann solver. For a stationary choice of the point positions the code falls back to the ordinary Eulerian method. The code however moves the points according to the velocity of the local flow, thus producing a Galilean-invariant solver that is also able to treat shocks. Additionally, because of the moving of points, the code automatically oversample regions that are over-dense, as in the SPH approach.

1.7 Cosmological Simulations

Nowadays cosmological simulations come in two main categories: Eulerian and Lagrangian. The first category integrates the equations of state of the various matter components (usually dark matter, gas, stars and black holes) on a grid and solves the 3-dimensional discretised equations of on a grid (see e.g. the code grid code Enzo by Bryan et al. (2014)). Due to the collapse and structure formation in cosmological simulations, the matter content in collapsed regions must be sampled inside finer grid cells than matter on voids where the

evolution of perturbations due to gravity is in its linear regime. People use Adaptive Mesh Refinement (AMR) codes that subdivide grid cells on regions that need higher resolutions. Those smaller cells can be merged together when higher resolution is not needed anymore.

N-body codes, on the other hand, begin with an initial set of particles sampling the simulated fluids, each with a certain mass, position and volume (for instance, a dark matter particle will sample an ensemble of real dark matter particles within the volume sampled by the particle in simulation, described by their smoothing length ϵ discussed in the above paragraphs). One advantage of N-body simulations with respect to grid codes is that particles move with the velocity of the fluid and thus automatically over-sample clustered regions.

Inflation theory predicts fluctuations δ that are Gaussian fluctuations (Bertschinger, 1995). These perturbations $\delta(\vec{x})$ can then be expressed as

$$\delta(\vec{x}) = \frac{\rho(\vec{x})}{\langle \rho \rangle}$$

and their evolution is described both by the continuity equation

$$0 = \frac{\partial \delta}{\partial t} + \nabla \cdot [(1 + \delta)\vec{u}] \quad (1.22)$$

and the Euler equation for a fluid under the Hubble flux of $H(t)$ and expanding with a scale factor $a(t)$,

$$\frac{\partial \vec{u}}{\partial t} + 2H(t)\vec{u} + (\vec{u} \cdot \nabla)\vec{u} = -\frac{\nabla \phi}{a^2}. \quad (1.23)$$

Gravitational potential is related to the density field with the Poisson equation

$$\nabla^2 \phi = 4\pi G \langle \rho \rangle a^2 \delta, \quad (1.24)$$

where the gradients (∇) are computed with respect to the comoving coordinates and H is the so called Hubble flow \dot{a}/a , whose time evolution for a given matter. Its time evolution is often expressed as $H(t) = E(t)H_0$, and where H_0 is the present day value and $E(t)$, for a universe with a dark energy density Ω_Λ and a matter density Ω_m , on a flat universe

$$E(z) = \sqrt{\frac{\Omega_m}{a^3} + \omega_\Lambda}.$$

Equations 1.22 1.23 and 1.24 can be combined and approximated to a second order differential equation over the variable δ (see Borgani, 2008, for more details)

$$0 = \frac{\partial^2 \delta}{\partial t^2} + 2H \frac{\partial \delta}{\partial t} - 4\pi G \langle \rho \rangle \delta, \quad (1.25)$$

which represent the Jeans instability equations for a pressureless fluid with an additional expansion due to the Hubble flow. This additional expansion is able to contrast the growth of perturbations that would otherwise grow exponentially as

$$\delta(t) = \exp(\sqrt{4\pi G \langle \rho \rangle} t). \quad (1.26)$$

Fluctuations whose initial values have been computed in the Fourier spaces $\hat{\delta}(\vec{k})$ are then transformed to the real space potential $\phi(\vec{q})$ via a Fourier transform so that

$$\phi(\vec{q}) = \sum_{\vec{k}} \frac{\hat{\delta}(\vec{k})}{k^2} e^{i\vec{q}\cdot\vec{k}}.$$

Fluctuations are further evolved using the Zel'dovich (1970) approximation. This approximation starts from an un-perturbed lattice \vec{q} , and gives the new distribution \vec{x} at time t as

$$\begin{aligned} \vec{x} &= \vec{q} + D^+(t)\Phi(\vec{q}) \\ \vec{v} &= a \frac{dD^+(t)}{dt} \nabla\psi(\vec{q}), \end{aligned}$$

where D^+ is the growth factor (Efstathiou et al., 1985),

$$D^+(z) = \frac{5}{2} \Omega_m E(z) \int_z^{\text{inf}} \frac{1+z}{E(-z)} d - z.$$

As perturbation grows and, when the overdensity is large enough, the perturbations reach a maximum expansion and then detach from the Hubble flow and collapse. Most density profiles of dark matter haloes from both simulations and observations can be described by a Navarro Frank and White (NFW) profile (Navarro et al. (1996, 1997), see Borgani and Kravtsov (2011) for a review). These density profiles ρ are modelled as a function of the radial distance r as:

$$\rho(r) = \frac{\rho_0}{\frac{r}{r_s} \left(1 + \frac{r}{r_s}\right)^2},$$

where r_s is a scale radius separating the internal region and the external region, and ρ_0 is twice the density at $\rho(r = r_s)$.

As haloes do not have clear boundaries, the virial radius R_{vir} is defined as the radius at which the density of the halo is the one of a theoretical virialised homogeneous spherical over density in an expanding universe. This density threshold is $\Delta_{vir}\rho_{crit}$. Here ρ_{crit} is the critical density $\rho_{crit} = 3H^2/4\pi G$ and Δ_{vir} is a parameter that depends on the cosmology. For instance $\Delta_{vir} \approx 178$ in an Einstein de Sitter cosmology (see Naderi et al. (2015) for a review). More generally, people use radii definitions that are independent from the cosmology and refer to R_Δ as the radius that includes a over density of $\Delta \cdot \rho_{crit}$. For the following analysis, both $\Delta = 200$ and $\Delta = 500$ and the corresponding radii of R_{200} and R_{500} are used.

1.7.1 Galaxy Clusters

The concentration as $c_\Delta \equiv R_\Delta/r_s$ and quantifies the size of the internal region of a cluster, compared to its radius. Bullock et al. (2001) is a pioneering theoretical work in the study

of the concentration in c_Δ a Λ CDM universe. Their toy model based on a isolated spherical over density whose scale factor a at the collapse time is a_c , predicts a concentration $c \propto a/a_c$, where the proportionality constant is universal for all haloes.

Various works in the literature fit the concentration as a power law of the halo mass and redshift. They mainly find a very low dependency of concentration from redshift and a slow but steady decrease of the concentration with mass (see e.g. Dutton and Macciò, 2014; Merten et al., 2015). In comparing various works one must first consider carefully how the concentration is computed. Some theoretical works, as Ludlow et al. (2012); Prada et al. (2012), derive the dark matter concentration from the peak in the velocity dispersion instead of constraining r_s from a NFW fit of the dark matter density profile. The concentration computed with this technique is systematically higher than the concentration computed with a NFW fit, usually with a relative error of 1 – 10%, and Meneghetti and Rasia (2013) show that the two values can diverge significantly.

On top of this, Lin et al. (2006) found that introducing non-radiative gas physics in numerical simulations increases the concentration, while Duffy et al. (2010) show how the additional inclusion of AGN feedback decreases the halo concentration the lower is the halo mass, up to a difference of $\approx 15\%$ for haloes of $M \approx 10^{11}M_\odot$. Simulations with various dark energy models, as in De Boni (2013), showed that in the $c - M$ relation the normalisation is sensitive to the cosmologies and Duffy et al. (2008) showed that the predicted concentrations of dark matter only runs are much lower than inferred from X-ray observations of groups and clusters.

Most recent high resolution dark matter only simulations showed an up turn trend of the mass concentration relation of simulations in the highest mass regime at very high redshift (see Zhao et al., 2009; Klypin et al., 2011; Prada et al., 2012). The cause of such upturn is still unclear.

Additionally, the mass-concentration plane of various theoretical and observational studies has a scatter that can span over an order of magnitude. Macciò et al. (2007) proposed that the scatter is partially due the non spherical symmetry of the initial fluctuation, while Neto et al. (2007) (see Fig. 10 in their paper) shows how this scatter can be partially disentangled by describing the concentration as a function of the formation time of the halo. Observational studies found that fossil objects (i.e. objects remained unperturbed for a long period of time) are the objects with the highest concentration (Pratt et al., 2016; Kundert et al., 2015; Khosroshahi et al., 2006; Humphrey et al., 2012, 2011; Buote, 2017). Fossil objects are supposed to be relaxed systems and, as defined in Voevodkin et al. (2010), have a difference in magnitude in the R band $\Delta m_R \geq 1.7$ between the most luminous object and the second most luminous object within a distance of $\frac{1}{2}R_{200}$. This is in agreement with theoretical studies on unperturbed haloes in dark matter only simulations, where dynamically relaxed systems have higher concentration than average (Cui et al., 2017; Klypin et al., 2016; Davis et al., 2011).

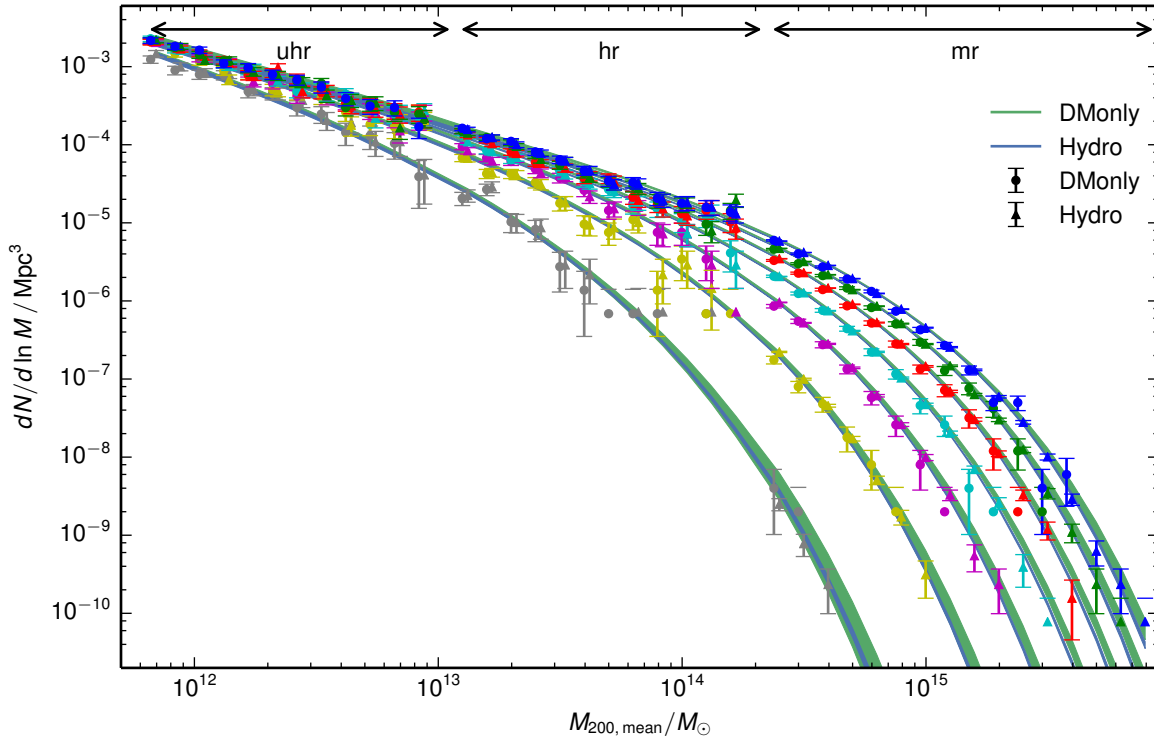


Figure 1.5: An example of Halo Mass Function (HMF) from the Magneticum simulations (Box0/mr, Box2/hr and Box4/uhr), taken from Bocquet et al. (2016).

1.7.2 Simulations vs. Observations

Nowadays cosmological simulations incorporate a lot more physics than just gravity. They simulate the intergalactic medium (IGM) that follows the laws of hydrodynamics and they include additional processes that take place below the resolution limit (e.g. star formation). Those processes are called sub-grid models and are recipes to model the outcome of processes that happen at the lowest scales. For instance, modern high resolution simulations on cosmological volumes follow gas particles with a resolution in mass of $\approx 10^6 M_\odot$. It is clear that they cannot follow single cloud collapsing objects that forms stars (with a mass range of $< 10^{3-4} M_\odot$).

In order to overcome the problem of implementing unresolved physics, the simulations model the star formation due to the gas conditions as pressure due to gravity, turbulence, magnetic fields (see e.g. Widrow et al. (2012); Kravtsov (2003)), and implement schemes that produce simulated star particles with a mass resolution of $\approx 10^6 M_\odot$. This means that the simulated star particles don't represent single stars, they represent a whole population of stars with their own initial mass function (IMF). The current state of the art simulations include effects of Supernovae explosions, black holes seeding and accretion, and many phenomena.

To compare the number of well resolved objects with surveys, there is the need to

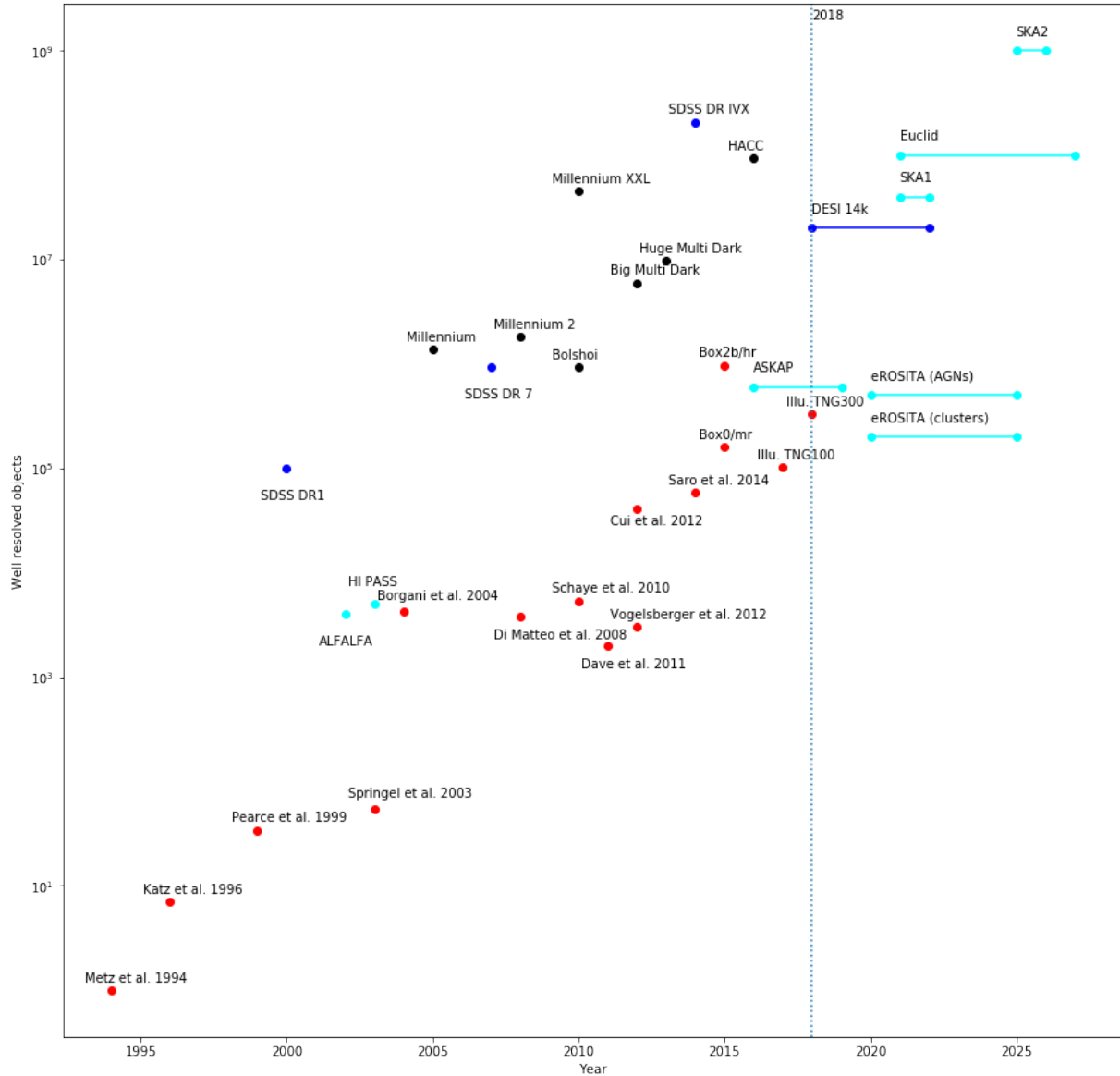


Figure 1.6: Well resolved objects in simulations and surveys over time. Grey points represent dark matter only simulations, red points represent hydrodynamics simulations, blue colour represents radio surveys and light blue colour represent optical surveys. Segments represent non finished/planned surveys. This plot uses 10^3 particles for well resolved clusters in dark matter only simulations and 10^4 particles for hydrodynamic simulations.

estimate the number of galaxy clusters produced in a simulation. To do such estimation, one first identifies the corresponding amount of particles N_p that a simulated galaxy clusters must have in order to be considered well defined. Then, one computes the minimum mass for a well resolved object in a given simulation, and finally, the total number of predicted objects is the integral of dn/dM above this mass threshold. For this purpose a minimum number of 10^3 particles in a cluster for dark matter only simulations is chosen and 10^4 particles for simulations with hydrodynamic. Then the minimum mass of a well resolved galaxy cluster M_{min} is then

$$M_{min} = N_p \frac{M_{tot}}{N_{tot}},$$

where M_{tot} is the total mass inside the simulation (so $M_{tot} = \sum_1^N m_i$, where m_i is the mass of the i -th particle and N is the total number of particles). In case of a dark matter only cosmological simulation, where all particles always have the same mass m_{DM} , one obtains a minimum mass of

$$M_{min}^{\text{dark matter only}} = N_p m_{DM}.$$

To compute the total amount of objects that with a mass higher than a certain threshold, one has to integrate the so called halo mass function dn/dM . The halo mass function is the number density of objects dn per mass interval dM (Press and Schechter, 1974) and has been constrained numerically by various works in the literature (see e.g. Bocquet et al., 2016, for one of the most recent constrains from simulations) and is usually expressed as

$$\frac{dn}{dM} = f(\sigma) \frac{\langle \rho_m \rangle}{M} \frac{d \ln \sigma^{-1}}{dM},$$

where $\langle \rho_m \rangle$ is the mean matter density and $\sigma^2(M, z)$ defined as

$$\sigma^2(M, z) \equiv \frac{1}{2\pi^2} \int P(k, z) \hat{W}^2(kR) k^2 dk,$$

that represents the variance of the matter density ρ smoothed using the Fourier transform of a top hat of size R that span a volume of $4\pi R^3/3 = \langle \rho_m \rangle / M$ (Bocquet et al., 2016). And $f(\sigma)$ is a fit function usually chosen to be

$$f(\sigma) = A \left(\left(\frac{\sigma}{b} \right)^{-a} + 1 \right) \exp \left(-\frac{c}{\sigma^2} \right),$$

that depends on the fit parameters A, a, b, c . Figure 1.5 shows a representation of a halo mass function for the WMAP7 cosmology (Komatsu et al., 2011, with total matter density parameter $\Omega_0 = 0.272$ and a baryon fraction of 16.8%) as given by (Bocquet et al., 2016). The number of objects $N(> M_{min})$ that are more massive than a certain threshold is given by the integral

$$N(> M_{min}) = \int_{M_{min}} \frac{dn}{dM} dM.$$

Figure 1.6 shows the list of computed number of objects of the largest simulations over time, and the number of well resolved objects in previous and future surveys. First collisionless simulations from Peebles (1970); White (1976) resolved single clusters with less than 1000 particles, while Metzler and Evrard (1994) is the first hydrodynamic simulation of a galaxy cluster, with 1000 particles. And Katz et al. (1996); Pearce et al. (1999); Murali et al. (2002); Springel and Hernquist (2003a) are the first hydrodynamic simulation considering multiple clusters interacting. Borgani et al. (2004) remained the hydrodynamic simulation with most clusters for 5 years. Schaye et al. (2010); Di Matteo et al. (2008); Davé et al. (2011); Vogelsberger et al. (2012); Cui et al. (2012); Saro et al. (2014) are the first simulations over cosmological volumes ($> 40Mpc$). And Dolag et al. (2015) present Magneticum Box0/mr and Box2b/hr simulations, while Nelson et al. (2018); Pillepich et al. (2018) present Illustris and Illustris TNG simulations. York et al. (2000); Frieman et al. (2008); Blanton et al. (2017) present the data release from the multi-spectral imaging and spectroscopic redshift survey Sloan Digital Sky Survey (SDSS), where they consider only SDSS objects with an associated spectroscopic data. ALFALFA is a 300meter radio telescope based in Puerto Rico (Martin et al., 2010) devoted to neutral atomic hydrogen surveys. The HI Parkes All Sky Survey (HI PASS Meyer et al., 2004) was a pioneering HI survey that covers the entire southern sky. The Australian Square Kilometre Array Pathfinder (ASKAP) is a pioneering array of radio telescopes (Johnston et al., 2008), that led the route for the Square Kilometre Array (SKA) (Duffy, 2014). eROSITA (extended ROentgen Survey with an Imaging Telescope Array Kolodzig et al., 2013) is a future X-ray telescope built in Germany that will be launched in 2019. Amendola et al. (2013) present a future visible and near-infrared space telescope Euclid, while Abdalla et al. (2012) present the Dark Energy Spectroscopic Instrument (DESI), which will provide spectrographic data of distant galaxies. Such new instrument will probe the expansion history of the Universe and possibly shed light on the physics of dark energy. For a review on surveys and their number of observed objects, see Duffy (2014).

It is clear that simulations of the next decade will need to resolve a larger number of objects, and for this purpose it is necessary to study and solve the bottlenecks of current codes for cosmological simulations. In the next section there will be a discussion on the technical aspects of Gadget, a widely used code for hydrodynamic cosmological, and what are its bottlenecks and how they were solved.

1.8 The Gadget3 code

Gadget3 (GALaxies with Dark matter and Gas intEracT) simulates the evolution of interacting Dark Matter, gas and stars in cosmological volumes (Springel et al., 2001a; Springel, 2005b). While Dark Matter is simulated so it interacts only through gravity, gas obeys the laws of hydrodynamics. Both Dark Matter and gas are simulated by a particle approach. Gadget3 uses a Tree-PM (see, e.g. Xu (1995)) algorithm for the gravitational interactions between both Dark Matter and gas particles. Smoothed Particle hydrodynamics (SPH) is used for the hydrodynamic interaction, as described in Price (2012).

Gadget3 uses Smoothed Particle hydrodynamics (SPH) for the hydrodynamics evolution of gas particles. The formulation used in the Magneticum simulations is an improved SPH scheme described in Beck et al. (2016). Springel et al. (2005a); Tornatore et al. (2003, 2007) describe the treatment of radiative cooling, heating, ultraviolet (UV) back-ground, star formation and stellar feedback processes. Cooling follows 11 chemical elements ($H, He, C, N, O, Ne, Mg, Si, S, Ca, Fe$) using the publicly available CLOUDY photo-ionisation code (Ferland et al., 1998). Teklu et al. (2015); Remus et al. (2017) have a more detailed description of many of these physical processes.

The description of the prescription for black hole (BH) growth and for feedback from AGNs can be found in Di Matteo et al. (2005, 2008); Fabjan et al. (2010); Hirschmann et al. (2014).

Galaxy halos are identified using a friend-of-friend (FoF) algorithm and sub-halos are identified using a version of SUBFIND (Springel et al., 2001a), adapted by Dolag et al. (2009) to include the baryon component.

Gadget3 employs a variety of physical processes, e.g. gravitational interactions, density calculation, hydrodynamic forces, sub-grid models for star formation and black hole seeding and accretion. All these algorithms need to process a list of active particles and find the list of nearby particles (“neighbours”). These neighbours are typically selected within a given searching sphere, defined by a given searching radius, defined by local conditions of the active particles (see, e.g. Hernquist and Katz (1989)). This problem is called neighbour search and is one of the most important algorithms to compute the physics implemented in Gadget3.

1.8.1 Gadget’s tree walk

All the main Gadget modules (gravity, SPH, stellar feedback, thermal conduction, and black hole feedback) share the same implementation and communication pattern. In the first phase of each module, for all active particles in the current time-step, Gadget needs to perform a tree walk (over the Barnes Hut tree data structure) in order to both find neighbouring particles and neighbouring regions that belong to a different MPI task. Gadget will then exchange guest particles between MPI tasks and in the second phase of a module, it searches for neighbours (via a tree walk) for the received guest particles.

In the standard implementation of Gadget, the mere tree walk and communication parts (as will be described in detail in Chapter 2) consume more times than the actual physical computations. For this reason in Chapter 2 I present a new neighbour search algorithm which drastically lowers the execution time. Implementing such a new neighbour search was necessary in order to run Magneticum Box0/mr and Box2b/hr (see Section 1.9.1)

When a MPI rank has computed the interactions over the received guest particles, it will send the results back to the original MPI rank. This MPI rank will then merge the received contribution with the contribution from the local neighbours. Gadget uses a relatively small exchange buffer (compared to the number of particles in a large simulation), and for this reason, if it is not possible to exchange all particles at one time, the first and second phases must be repeated until all active particles have been processed.

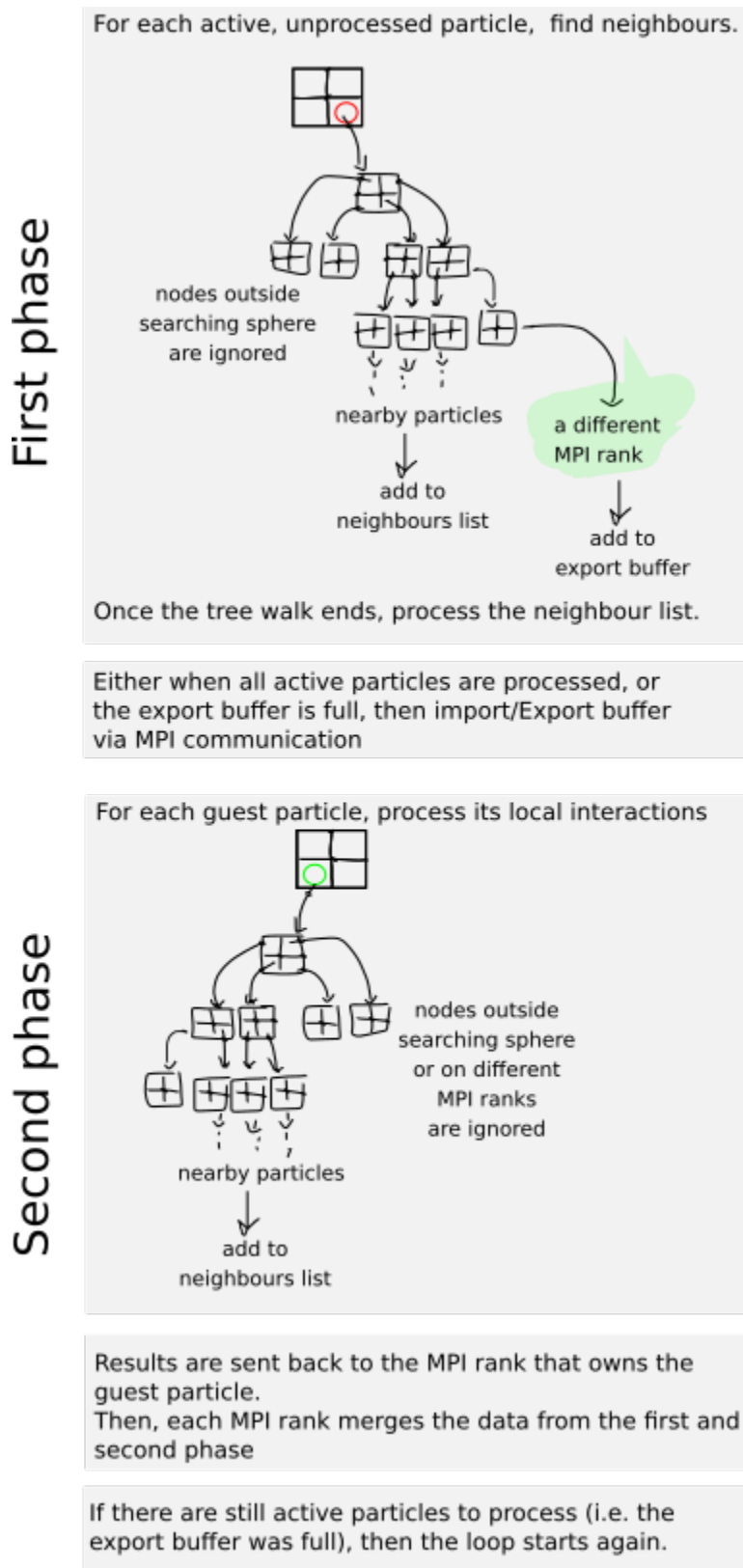


Figure 1.7: Graphical representation of the typical loops performed in the Gadget modules.

Figure 1.7 shows in more detail the two phases of the neighbour search, that can be summarised in the following steps:

- First phase:
 - for each internal active particle P_i : walk the tree and find all neighboring particles closer than the searching distance h_i ;
 - when walking the tree: for every node belonging to a different MPI process, particle and external node are added to an export buffer;
 - if the export buffer is too small to fit a single particle and its external nodes: interrupt simulation.
 - if the export buffer is full: end of first phase.
 - physical quantities of P_i are updated according to the list of neighbours obtained above.
- Particles are exported.
- Second phase:
 - for each guest particle P_i : walk the tree and search its neighbours;
 - update properties of P_i according to the neighbours list;
 - send updated guest particles back to the original MPI process.
- The current MPI process receives back the particles previously exported and updates the physical quantities merging all results from the various MPI processes.
- Particles that have been updated are removed from the list of active particles.
- If there are still active particles: start from the beginning.

The definition of neighbouring particles is slightly different between the Gadget3 modules. In the Density module, neighbours of the particle P_i are all the particles closer than its searching radius h_i . In the hydrodynamics module, neighbours are all particles P_j closer than $\max(h_i, h_j)$ to P_i .

To perform the Neighbour Search in Gadget3, an octree is used to divide the three dimensional space. Further optimisation is obtained by ordering the particles according to a space-filling curve. In particular, Gadget3 uses the Hilbert space-filling curve to perform the domain decomposition and to distribute the work among the different processors.

1.8.2 SPH

Gas hydrodynamic is implemented in Gadget using the SPH formulation (see Section 1.6.2) as described in Beck et al. (2016), with the possibility of including a time dependent scheme for artificial viscosity and artificial conduction as presented in Beck et al. (2016). The SPH implementation is divided in two stages, first Gadget updates the smoothing length of all active particles, when all active particles have been updated, it then computes the hydro-force acting on said particles.

The update of the smoothing lengths is a very time consuming operation because it is itself an iterative process. At each iteration on the smoothing length computation, Gadget3 must perform tree walks for each active particles as described in Section 1.8.1. This poses a large slow down in large cosmological simulations, since they can perform up to 10 - 15 iterations, each with its slow neighbour search.

After the update of the smoothing lengths, Gadget must perform an additional tree walk for each active particle (as in Section 1.8.1) to update the hydrodynamic force over all active particles. This tree walk is computational expensive because of the large number of computations.

1.8.3 Cooling

While gravity is the main driver of the collapse of the large-scale overdensity peaks in the initial conditions, thermodynamic properties can affect galaxy formation. Radiative cooling is supposed to be the main cause for condensation and long time life time of galaxies inside clusters (White and Rees, 1978). Such process is included in Gadget with phenomenological treatment of star formation and its associated energy feedback, and approximate radiative transfer.

Radiative loss is included in Equations 1.14, 1.15 and 1.16 by modelling the so called cooling function $\Lambda(u, \rho)$.

A first naive approach could describe the cooling function $\Lambda(T)$ chosen so the cooling rate per unit volume is $n_e n_p \Lambda(T)$ (where n_e and n_p denote respectively the free electron and proton densities). Cen (1992) is the first cosmological simulation to follow the non-equilibrium abundances of electrons, hydrogen, helium and molecular species. Ferland et al. (1998) presents the code CLOUDY that take into account heavier elements (“metals”). As shown in Peebles and Dicke (1968), the largest cooling of gas under 10^4K is given by H^- , H_2^+ and H_2 .

Gadget stores the cooling tables (Λ vs. T) data in pre-computed tables in a number of redshift slices (Tornatore et al., 2003, 2007). At a given time in the simulation it interpolates the values between redshift and temperature bins to compute Λ . The effects that are included in Λ act on a plasma composed by H and He that can produce, H_I , H_{II} , He_I , He_{II} , He_{III} , as computed by Maio et al. (2007), in the assumptions of a optically thin gas in ionisation equilibrium where three-body cooling is considered negligible.

1.8.4 Star Formation and stellar feedback

The process of star formation happens at scales that are lower than the spatial resolution of cosmological simulations ($\approx kpc$). In order to avoid a cooling catastrophe, one has to include the heating effects of energy feedback from Supernovae explosions.

The implementation presented in Springel and Hernquist (2003b) (and currently used in most Gadget simulations) requires that the process of star formation occurs only in Jeans unstable regions (where density perturbations grow, see Equations 1.25 and 1.26):

$$\frac{h_i}{c_i} > \frac{1}{\sqrt{4\pi G \rho_i}}, \quad (1.27)$$

where c_i is the local speed of sound. Additionally, it asks for a convergent velocity flow of the SPH particles

$$\nabla \vec{v}_i < 0$$

and for its density ρ_i to lay in an environment that is denser than a given fixed threshold, so that

$$\rho_i > 0.1 \frac{\text{atom}}{\text{cm}^3}.$$

If a gas particle passes the previous three criterion, it is then selected as being star forming, and its star formation rate (SFR) \dot{M}_\star is defined as

$$\dot{M}_\star = \frac{d\rho_\star}{dt} = -\frac{d\rho_i}{dt} = \frac{c_\star \rho_i}{t_\star},$$

where c_\star is a dimensionless star formation rate parameter and t_\star is a characteristic star formation timescale. Gadget assumes t_\star as being the maximum between the Jeans-equation dynamical timescale (Equation 1.27) and the cooling time

$$t_{\text{cool}} = u_i \left(\frac{du_i}{dt} \right)^{-1}.$$

To avoid to over-spawn star particles (that will fill the memory and/or ruin the performance of the computing node), Gadget spawns a new collisionless star particle only (with the accumulated star mass M_\star) with a probability p that is decided for each particle at each timestep as

$$p = 1 - e^{-\frac{c_\star \Delta t}{t_g}},$$

where Δt is the timestep size. After a stellar particle is spawned, the gas particle mass is decreased according to the mass of the new star particle. When a gas particle reaches a mass of zero it is removed from the simulation. The model also takes into account that each star particle will be spawned with a well defined initial mass function. It will estimate how many stars are massive enough to produce Supernovae (SNe), and the model can estimate the amount of energy released from each star particle at each timestep due to the supernovae explosions (typically 10^{51} erg/supernova). Such energy is distributed

around the surrounding gas particles. This is done under the approximation that life time of a supernovae is below the time resolution limit of the simulation.

Noteworthy SNe feedback is implemented together with the chemical enrichment, which again needs an additional tree walk for each active particle as described in Section 1.8.1. Such tree walk must be performed for all active gas particles.

1.8.5 Thermal Conduction

Some of the first theoretical and numerical studies on X-ray emission from gas in galaxy clusters (Fabian et al., 1994; Allen et al., 2001) predicted that gas should continuously cool and produce a mass deposition rate of $\approx 100M_{\odot}/yr$. Since observations as Fabian et al. (2001); Böhringer et al. (2001) failed to find the expected properties, Narayan and Medvedev (2001) propose that galaxy clusters must have a source of heating in order to replace the lost through X-ray emission. Fabian (1994) ruled out the presence of material with star formation and temperatures below $1/3rd-1/4th$ of the cluster virial temperature. This result was in contrast with hydrodynamic simulations and Narayan and Medvedev (2001) proposed the addition of thermal conduction to solve this problem.

Such theory was verified by one of the first Gadget simulations with thermal conduction by Dolag et al. (2004). As proposed by Zakamska and Narayan (2003), the thermal conduction \vec{Q} is proportional to the temperature gradient

$$\vec{Q} = -k\nabla T,$$

where k is the so called conduction coefficient. This leads to a change in internal energy because of the relation

$$\nabla \cdot \vec{Q} = -\rho \frac{du}{dt},$$

which relates an internal energy loss ($-\rho du/dt$), with a heat flux.

The thermal conduction is effectively an algorithm to solve a matrix inversion problem, where the matrix has a number of rows as the number of particles. However, given the computational cost of the matrix inversion $> O(N^2)$, a perfect matrix inversion is not feasible. Gadget implementation uses an iterative approximation method with very good convergence properties (see Arth et al., 2014, for more details). Each thermal conduction iteration contains a gadget tree walk (with its neighbour searched and communications, as described in Section 1.8.1) for each inverse matrix iteration.

1.8.6 Black holes growth and AGN feedback

Di Matteo et al. (2008) proposes a model where black holes are mapped in collisionless particles that accrete mass from the surrounding mass from the immediate nearby gas particles. When a Gadget FoF group is identified and the mass is larger than a pre-defined threshold, a black hole particle is added in the centre of the halo. The centre for the seeding is defined by the minimum of the gravitational potential generated by stellar particles.

McConnell and Ma (2013) found a relationship between black hole mass M_{bh} and the stellar mass of the hosting bulge M_* of the form

$$\frac{M_{bh}}{M_\odot} = 10^{8.46} \left(\frac{M_*}{10^{11} M_\odot} \right)^{1.05}$$

Hirschmann et al. (2013) proposes a model currently used in a version of Gadget3 used in the Magneticum simulations where the seeding mass is lower than the one proposed in the relationship of McConnell and Ma (2013). The reason of this mismatch is that the simulated black hole is seeded instantaneously, and they reach the relation after a period of accretion. To compensate for a light black hole mass, a black hole particle has two masses, a “real mass” used to keep track of the accretion and a “dynamical mass” used to compute gravitational interactions.

Since the region where a black hole accrete is under the spatial resolution of nowadays simulations (that is of $\approx 0.5 kpc$), people use accretion models based on the Bondi-Hoyle-Lyttleton model (Hoyle and Lyttleton, 1939; Bondi and Hoyle, 1944; Bondi, 1952). This model assumes a spherically symmetric accretion of all particles that crosses the so called Bondi-radius r_A , defined as the radius of a test particle whose velocity v is the escape velocity itself:

$$r_A = \frac{2GM}{v^2},$$

and with the assumption of a spherical symmetric system, Hoyle and Lyttleton (1939) computes an accretion rate \dot{M} of

$$\dot{M} = \rho v \pi r_A^2 = \frac{4\pi G^2 M^2 \rho}{v^3}.$$

Bondi and Hoyle (1944) scale such this with a constant $\alpha_{HL} = 1.25$ that takes into account the gas pressure that contrasts the gravitational collapse.

The AGN feedback is implemented so that the black hole release energy proportional to its mass as prescribed by (Booth and Schaye, 2009), where the proportionality constant is a feedback efficiency ϵ_f , and a proportionality constant $\epsilon_r \approx 0.1$, *sothat*

$$\dot{E}_{feedback} = \epsilon_f \epsilon_r \dot{M} c^2.$$

The black hole module contains gadget tree walks (with its neighbour searched and communications, as described in Section 1.8.1), performed over the active black hole particles.

1.8.7 Halo Finder

A widespread algorithm that finds halos in a cosmological simulation is the so called Friend of Friend (FoF) algorithm (Davis et al., 1985). FoF is a percolation algorithm which links particles together when their distance is less than a certain threshold, called linking length.

Particles that can be reached walking the links are considered to be part of the same halo. As a consequence particles are associated to only one halo. This algorithm has the advantage of not requiring a prior hypothesis on halo shapes.

When studying galaxy formation and evolution, it is not only useful to save all particles in a halo, but additionally some post-processed quantities as its virial mass and virial radius.

A linking length of 0.2 times the mean inter-particle distance produce FoF halos that, on average, have an over-density comparable to a overdensity of a virialised object in a Λ CDM universe(see e.g. Jenkins et al., 2001).

The galaxy that lies at the centre of a given halo is called central galaxy. The central galaxy of a halo is also usually the most massive galaxy inside the halo. Halos host more galaxies and these are called satellite galaxy.

Satellite galaxies are hosted in sub halos. There exist various algorithms to find satellite galaxies. A widely used algorithm is the so called SUBFIND (Springel et al., 2001b). SUBFIND uses the results of FoF and search density peaks inside each halo previously found by FoF. For each peak it searches the iso-densities surfaces containing saddle points. SUBFIND defines the sub halo as the set of gravitationally bound particles inside the volume delimited by these surfaces. Gadget3 used an improved version of the original SUBFIND that takes into account baryons (Dolag et al., 2009).

SUBFIND density computation is very similar to the SPH smoothing length computation. Since this operation needs several iterations too, SUBFIND is a very computationally expensive module with all the slow down due to the neighbour search as described in Section 1.8.1.

1.8.8 Gadget3 bottlenecks

One of the main bottlenecks that uses most of the computing time in the standard Gadget3 implementation is the so called neighbour search. This process occurs when Gadget walks its Barnes Hut tree in order to find neighbours of a particle (e.g. in the SPH computations).

I analysed the code with the profiling tool `Scalasca` (see Geimer et al., 2010, for details about Scalasca). In Figure 2.1 (left table) I show the profiling results for the hydrodynamics module, which is the most expensive in terms of execution time. The most time consuming modules in Gadget3 are hydrodynamics, density and gravity and thermal conduction, as they are all executed at each time step and they undergo the loop explained in Section 1.8.1. There, neighbour recycling is not implemented in the gravity loop. The reason behind this choice is that the gravity module implements a Tree-PM algorithm Bagla (2002). Unlike in density and hydrodynamics, particles do not have a defined searching radius. In fact the criterion whether or not a node of the tree must be opened takes into account the subtended angle of this node by the particle. Also, for the way it is implemented in Gadget3, the gravity module does not makes a clear distinction between the Neighbour Search and the physics computations, making it difficult to modify the Neighbour Search without a major rewriting of the module.

Name	Size [Mpc/h]	n. part	m_{dm} [M_{\odot}/h]	m_{gas} [M_{\odot}/h]	ϵ [kpc/h]	ϵ_{\star} [kpc/h]
Box4/uhr	48	$2 \cdot 576^3$	$3.6 \cdot 10^7$	$7.3 \cdot 10^6$	1.4	0.7
Box2b/hr	640	$2 \cdot 2880^3$	$6.9 \cdot 10^8$	$1.4 \cdot 10^8$	3.75	2
Box0/mr	2688	$2 \cdot 45363^3$	$1.3 \cdot 10^{10}$	$2.6 \cdot 10^9$	10	5

Table 1.1: Individual setup of the three Magneticum simulations used in this work. The columns contains respectively, the name, the linear size of the simulated volume, the total number of particles, the mass of each dark matter particle, the initial mass of gas particles, the gravitational softening length of both dark matter and gas, and the gravitational softening length of star particles.

	redshift	0	0.5	1	1.5	2
	Min halos					
	$M_{200}[M_{\odot}]$					
Box4/uhr	$3 \cdot 10^{11}$	5257	5539	5717	5799	5716
Box2b/hr	$2 \cdot 10^{13}$	38355	33221	15219	6353	3416
Box0/mr	$5 \cdot 10^{14}$	18914	4790	116	13	1

Table 1.2: Number of minimum critical mass of extracted halos for all analysed simulations, at every redshift with the number of extracted halos of the three Magneticum boxes used in this work.

I devote Chapter 2 in showing a novel approach to speed up the neighbour search. In this approach I group close-by particles together and perform a larger single neighbour search for each set of particles.

1.9 Modern large cosmological hydrodynamic simulations

Thanks to our effort to solve the Gadget bottlenecks (as described in detail in Chapter 2), it was possible to scale Gadget up to the whole SuperMUC machine and complete the executing of Box0/mr and Box2b/hr (presented in Table 3.1).

1.9.1 Magneticum Simulations

The Magneticum simulations, (www.magneticum.org Biffi et al., 2013; Saro et al., 2014; Steinborn et al., 2015; Dolag et al., 2016, 2015; Teklu et al., 2015; Steinborn et al., 2016; Bocquet et al., 2016; Remus et al., 2017) is a set of simulations that follow the evolution of overall up to $2 \cdot 10^{11}$ particles of dark matter, gas, stars and black holes on cosmological volumes. The simulations were performed with an extended version of the N-body/SPH code P-Gadget3 which itself is the successor of the code P-Gadget2 (Springel et al., 2005b;

Springel, 2005a).

The analyses of *Magneticum* haloes is done using *Box0/mr* (to follow the most massive haloes), *Box2b/hr* (to follow haloes with a range of mass between $5 \cdot 10^{14} - 8 \cdot 10^{15} M_{\odot}$) and *Box4/uhr* (to follow haloes with masses in the galaxy range, see Table 3.1). From each simulation a snapshot at redshift $z \approx 0, 0.5, 1, 1.5$ and 2 is selected. In each snapshot, only haloes with a number of dark matter particles greater than 10^4 are considered and then there is a cut in the critical mass so that all objects within this cut are well resolved. Table 3.2 shows the number of selected haloes per each box per each redshift.

The simulations assume a cosmological model in agreement with the WMAP7 results (Komatsu et al., 2011, with total matter density parameter $\Omega_0 = 0.272$ and a baryon fraction of 16.8%). Hubble constant $H_0 = 70.4 \text{ km/s/Mpc}$, index of the primordial power spectrum $n = 0.963$ and a normalisation of the power spectrum corresponding to $\sigma_8 = 0.809$.

1.9.2 Delivering large simulation data

Running, storing and analysing such simulations is a challenging task, both from a technical as well as from a collaborative point of view. Recent generations of HPC facilities provided within initiatives like GAUSS⁴ or PRACE⁵ belong to the first generation of supercomputers which can perform cosmological, hydrodynamic simulations covering both the required large volume and high resolution requirements. Here, the largest simulation performed so far belongs to the *Magneticum* project⁶ and follows 2×4536^3 resolution elements over the whole, cosmological evolution of the universe (Bocquet et al., 2016). Such simulations model many more physical processes (star formation, cooling, winds, etc.) than the typical dark-matter-only counterparts used currently in computational cosmology. These simulations provide a larger set of complex data and can reach several hundreds of terabytes of raw data. Such simulations are performed within large collaborative efforts and results have to be shared with a broader scientific community. A guarantee for a deep scientific impact means that such data are made easily accessible and ready to use within the individual collaborating groups. It implies that data are stored on the HPC facilities for long periods of time, with the possibility to post-process the full data. In addition, it is important to make such data available to a large astrophysical community and allow the scientists to apply analysis tools via standard interfaces.

In this respect, efforts have been done in the recent years in order to share data sets of various kinds with the community. For instance, the Millennium Simulation Data Archive⁷ (Lemson and Virgo Consortium, 2006) is a pioneering work in this field. With the Millennium Simulation Data Archive, the user is able to compose SQL queries over substructure and merger-tree data in order to extract haloes and galaxies from the Millennium Sim-

⁴<https://gauss-allianz.de/>

⁵<http://www.prace-ri.eu/>

⁶<http://www.magneticum.org>

⁷<http://wwwmpa.mpa-garching.mpg.de/Millennium/>

ulation. Users can also download the raw data files. The Cosmosim.org project⁸ allows users to compose additional queries over the list of particles and various post processed quantities (grid cells of density field). The Illustris Galaxies Observatory⁹ provides an application programming interface (API)

1.10 Overview of this thesis

Improving the performance of numerical algorithms and re-thinking them in a parallel fashion is nowadays fundamental in order to run the next generation of cosmological simulations. Chapter 2 presents the solution to one of the main bottlenecks within the code for cosmological simulations Gadget 3. One of the bottlenecks is in the so-called neighbour search, where the tree algorithm in Gadget was spending more time in walking the tree than in computing physical quantities. Improvements like this were added in the latest Magneticum/Box0/mr simulation(see e.g. Bocquet et al., 2016) that were able to catch next years All-sky X-ray surveys such as eROSITA(see e.g. Hofmann et al., 2017) will find up to 10^5 AGNs. Chapter 3 presents some analyses of properties of three of the Magneticum simulations that combined together, can be used to compare some properties of observed and simulated galaxy clusters over 4 – 5 orders of magnitude in mass. Given such a large amount of simulated objects in modern cosmological simulations, this rised the problem of opening and sharing the data access: simulation results are stored over several hundreds of TB of data, and this gives two major challenges: data must be stored in super-computing center that cannot give public access to the storage, additionally most scientists lack basic knowledge of high-performing computing and it would be impossible for them to efficiently look up data in order to find interesting objects. Chapter 4 presents how to overcome these challenges, with a publicly-open web portal that lets users efficiently query a database of simulation results in order to find interesting objects; additionally, it lets the users of the web portals run pre-defined jobs on simulated objects without the need of credentials in the super computer. Since some modern super computers have GPU systems, in Chapter 5 I discuss what prevents Gadget to fully exploit SIMT and SIMD parallelism and present a way of porting Gadget on such architectures. I finally draw our conclusions and future prospects in Chapter 6.

⁸<https://www.cosmosim.org/>

⁹http://www.illustris-project.org/galaxy_obs/

Chapter 2

Exploiting the Space Filling Curve Ordering of Particles in the Neighbour Search of Gadget3

The content of this chapter has been published in Ragagnin et al. (2016)

Antonio Ragagnin^{1,2,3}, Nikola Tchipev⁴, Michael Bader⁴, Klaus Dolag³, and Nicolay Hammer¹

¹ Leibniz Supercomputing Centre, München, D-85748 Garching, Germany

² Excellence Cluster Universe, Boltzmannstrasse 2, D-85748 Garching, Germany

³ Universitäts-Sternwarte, Fakultät für Physik, Ludwig-Maximilians Universität München, Scheinerstrasse 1, D-81679 München Germany

⁴ Department of Informatics, Technische Universität München, D-85748 Garching, Germany

Gadget3 is nowadays one of the most frequently used high performing parallel codes for cosmological Hydrodynamic simulations. Recent analyses have shown that the Neighbour Search process of Gadget3 is one of the most time-consuming parts. Thus, a considerable speedup can be expected from improvements of the underlying algorithms.

In this work we propose a novel approach for speeding up the Neighbour Search which takes advantage of the space-filling-curve particle ordering. Instead of performing Neighbour Search for all particles individually, nearby active particles can be grouped and one single Neighbour Search can be performed to obtain a common super set of neighbours.

Thus, with this approach we reduce the number of searches. On the other hand, tree walks are performed within a larger searching radius. There is an optimal size of grouping that maximise the speedup, which we found by numerical experiments.

We tested the algorithm within the boxes of the Magneticum large scale simulation project. As a result we obtained a speedup of 1.65 in the Density and of 1.30 in the Hydrodynamics computation, respectively, and a total speedup of 1.34.

2.1 Neighbour Search in Gadget3

Simulations of gravitational or electromagnetic interactions deal with potentials having, ideally, an infinite range. There are several known techniques (e.g. Barnes-Hut Barnes and Hut (1986b), Fast Multipole Expansion Greengard and Rokhlin (1997)) that can deal with this problem. These techniques subdivide the interaction in short-range and long-range interactions. The Long-range interactions are resolved by subdividing the simulated volume in cubes, and assigning to each of them a multipole expansion of the potential. The short-range potential is usually evaluated directly. This leads to the problem of efficiently finding neighbours for a given target particle, within a given searching radius. Finding neighbours by looping over all particles in memory is only suitable when dealing with a limited number of particles. Short-distance neighbour finding can be easily implemented by a Linked-Cell approach. Since long-distance computation is implemented subdividing the volume in a tree (an octree if the space is three-dimensional), this tree structure is commonly used for short-distance computations too. This is also a more generic approach, since Linked-Cell is more suitable for homogeneous particle distributions.

2.1.1 Impact of the Neighbour Search in the Performance

Tree algorithms are suitable for studying a wide range of astrophysical phenomena Hernquist (1987); Warren and Salmon (1995). To perform the Neighbour Search in Gadget3, an octree is used to divide the three dimensional space. Further optimisation is obtained by ordering the particles according to a space-filling curve. In particular, Gadget3 uses the Hilbert space-filling curve to perform the domain decomposition and to distribute the work among the different processors.

We analysed the code with the profiling tool Scalasca Geimer et al. (2010). In Figure 2.1 (left table) we show the profiling results for the Hydrodynamics module, which is the

Hydrodynamics Routines	Time [s]	Summary Hydrodynamics	Time [s]
First Phase	$3.21 \cdot 10^5$	Physics	$1.55 \cdot 10^5$
First Phase Neighbour Search	$1.89 \cdot 10^5$	Neighbour Search	$2.63 \cdot 10^5$
Second Phase	$9.81 \cdot 10^4$	Communication	$7.17 \cdot 10^4$
First Phase Neighbour Search	$7.36 \cdot 10^4$		

Figure 2.1: Left: *Scalasca* timing of the most expensive routines of the Hydrodynamics module in Gadget3. Right: Aggregate timing of the Hydrodynamics parts.

most expensive in terms of time.

The Hydrodynamics module is called once every time step. It calls the First Phase and the Second Phase routines multiple times. While the First Phase updates the physical properties of local particles, the Second Phase deals with external particles with neighbours in the current MPI process. Particles are processed in chunks because of limited exporting buffers, so the number of times those functions are called depends on the buffer and data sizes. Between the First Phase and Second Phase calls there are MPI communications that send guest particles to others MPI processes. First Phase and Second Phase routines are the most expensive calls inside Hydrodynamics, Density and Gravity. Both the First Phase and the Second Phase perform a Neighbour Search for every active particle.

In Figure 2.1 (right table), the Hydrodynamics part has been split into three logical groups: Physics, Neighbour Search and Communication. Communication between MPI processes has been estimated a posteriori as the difference between the time spent in Hydrodynamics and the sum of the time spent in the First Phase and Second Phase. This is well justified because no communications between MPI processes are implemented inside First Phase and Second Phase. The time spent in Physics has been computed as the difference between the First (or Second) Phase and the Neighbour Search CPU time. From this profiling, it turns out that for the Hydrodynamics module, Communication and Physics take less time than the Neighbour Search. This was already suggested by a recent profiling V. Karakasis et al.. Both results highlight the interest in speeding-up the Gadget3 Neighbour Search.

The three most time consuming modules in Gadget3 are Hydrodynamics, Density and Gravity. In this work we only improved Density and Hydrodynamics modules. There are two main reasons for excluding the Gravity module from this improvement. First, Gravity module implements a Tree-PM algorithm (see e.g. Verlet, 1967). In the Verlet-List approach, a super set of neighbours is associated to each particle which is used within multiple time steps. In our approach we associate a super set of neighbours to multiple particles, within a single time step. This technique takes into account that two target particles which are close together will also share part of the neighbours.

Neighbour Recycling groups can be built by using the underlying octree structure. Each group can be defined as the set of leaves inside nodes which are of a certain tree depth. Then, a super set of neighbours is built by searching all particles close to that mentioned node. For each previously grouped particle, this super set of neighbours is finally refined.

An advantage of this technique is that the number of tree walks is reduced, though at the expense of a larger searching radius.

The level of the tree at which the algorithm will group particles determines both the number of tree walks and the size of the super set of neighbours. This super set must be refined to find the true neighbours of a particle. Thus, increasing its size will lead to a more expensive refinement.

2.1.2 Implementation of Neighbour Recycling using a Space Filling Curve

Many parallel codes for numerical simulations order their particles by a space-filling curve, mainly because this supports the domain decomposition (Bungartz et al., 2010; Gibbon et al., 2005; Liu and Bhatt, 2000). In this work we will benefit from the presence of a space-filling curve to implement a Neighbour Recycling algorithm. Due to the nature of space-filling curves, particles processed consecutively are also close by in space. Those particles will then share neighbours.

Given a simulation for N particles, our algorithm proceeds as follows. A new group of particles is created, and the first particle is inserted into it. A while loop over the remaining particles is executed. As long as these particles are closer than a given distance R to the first particle of the group, they are added to the same set of grouped particles. Once a particle is found, which is farther than R , the group is closed and a new group is created with this particle as first element. The loop above mentioned is repeated until there are no more particles. We call N_{group} the number of particles in a given group; h_i the searching radius of the i -th particle of the group. Then, a super set of neighbours is obtained by searching the neighbours of the first particle of the group, within a radius of $R + \max(h_i)$. This radius ensure that all neighbours of all grouped particles are included in the searching sphere. For each grouped particle, the super set of neighbours is refined to its real list of neighbours. The refined list of neighbours is then used to compute the actual physics quantities of the target particle. Thus, the number of tree walks is reduced by a factor equal to the average number of particles in a group, $\langle N_{group} \rangle$.

It is clear that a too low value of R will group too few particles and lead to $\langle N_{group} \rangle \simeq 1$, thus leading to no noticeable performance increase. On the other hand, if R is too large, the super set of neighbours will be too large with respect to the real number of neighbours, producing too much overhead.

2.1.3 Adaptive Neighbour Recycling

In typical cosmological hydrodynamic simulations performed by Gadget3, a fixed R will group less particles in low-density regions and more particles in high density regions. Therefore a more reasonable approach is to reset R before every Neighbour Search and choose it as a fraction of the searching radius h_i , which itself is proportional to the local density. In this way, low density regions will have a larger R than high density regions. This is

obtained by imposing the following relation between R and the searching radius h of the grouped particles:

$$R = f \cdot h_0,$$

where f is a constant defined at the beginning of the simulation.

In a typical Gadget3 simulation, the number of particles N_{ngb} within the searching radius h_i is a fixed quantity. Locally it varies only within a few percent. In the approximation that every particle has the same number of neighbours N_{ngb} , we can write it as $N_{ngb} = 4\pi\rho h_i^3/3$, where ρ is the local number density. Furthermore, if the grouping radius is small enough, the density does not vary too much and we can set $h_i = h$. With those two approximations, the super set of neighbours is $N_{candidates} = 4\pi\rho(R + h)^3/3$ and the number of particles in a group is $N_{group} = 4\pi\rho R^3/3$. Combining those relations we obtain the following relation:

$$f = \left(\frac{N_{candidates}}{N_{ngb}} \right)^{\frac{1}{3}} - 1 = \left(\frac{N_{group}}{N_{ngb}} \right)^{\frac{1}{3}} \quad (2.1)$$

2.1.4 Side Effects of Neighbour Recycling

The Neighbour Recycling algorithm will increase the communication. Because tree walks are performed within a larger radius, the number of opened nodes increases. As a direct consequence, nodes of the tree belonging to other MPI processes will be opened more times than the original version. In the standard approach, the export buffer is filled only with particles whose searching sphere intersect that node. Since the new approach walks the tree for a group of particles, all particles belonging to the group are added to the export buffer. This leads to a greater amount of communications.

2.2 Speedup of the Recycling Neighbours Approach on Gadget3

We now investigate quantitatively how the new algorithm affected the performances of the code with respect to the old version. To show in details the effect of this new algorithm, we gradually implemented it in various parts of Gadget3 seeing the partial speedups. First we added the Neighbour Recycling in the First Phase of the Density computations. Then it has been added on both phases of Density computation, and finally it has been added in both the Hydrodynamics and Density computations.

2.2.1 The Test Case

We test the algorithm in a cosmological hydrodynamic environment. We use initial conditions from the Magneticum project (www.magneticum.org, Biffi et al., 2013; Saro et al., 2014; Steinborn et al., 2015; Dolag et al., 2016, 2015; Teklu et al., 2015; Steinborn et al.,

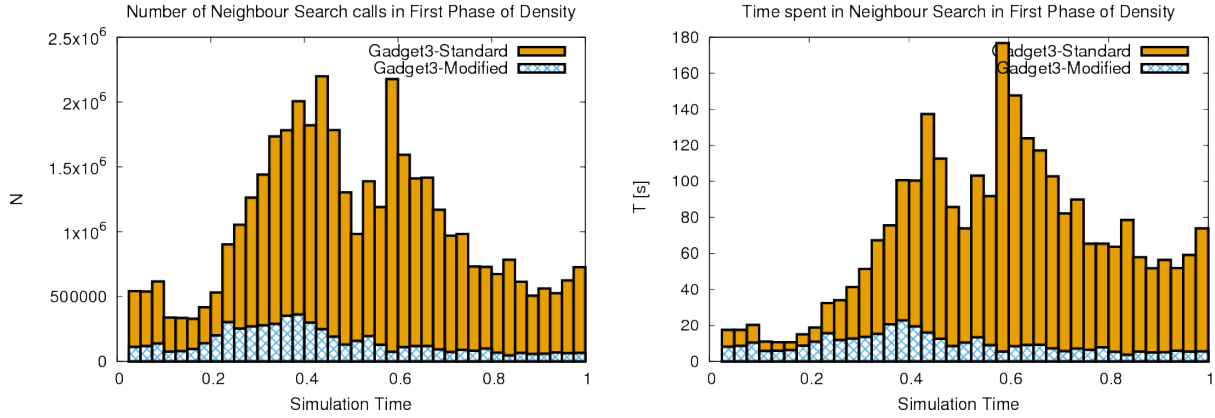


Figure 2.2: Left panel: every bin contains the number of Neighbour Search calls performed in that bin. Right panel: every bin contains the CPU time spent by the Neighbour Search. In both panels the orange (dark) histogram represents the standard version, light blue (light) histogram represents the modified version.

2016; Bocquet et al., 2016; Remus et al., 2017). To test our algorithm we chosen the simulation `box5hr`. This setup has a box size of 18 Mpc/h and $2 \cdot 81^3$ particles. The simulation run on 8 MPI processes, each with 2 threads. The average number of neighbours is set to $\langle N_{ngb} \rangle = 291$.

We have chosen a value of $f = 0.5$. Using Equation 2.1, we obtain $\langle N_{candidates} \rangle = 3.375 \langle N_{ngb} \rangle$. This means that a Tree Walk will now search for 3.375 more particles compared to the old of Gadget3. On the other hand such a high theoretical number of particles in a group will definitively justify the overhead of the Neighbour Recycling. In fact it is inversely proportional to the number of times the tree walk is executed. Still, such a low ratio between the size of superset of neighbours and the true number of neighbours will not produce a noticeable overhead in the refining of the superset of neighbours.

2.2.2 Results

The algorithm has been first implemented in the first phase of Density module computation of Gadget3. Figure 2.2 (left panel) shows the number of Neighbour Search calls performed during the simulation. The Neighbour Recycling version of the code has roughly the same amount of searches throughout the whole simulation, whereas the old version has a huge peak of Neighbour Search calls around a simulation time of 0.4. There, the number of Neighbour Search calls from the standard to the modified version, drops of a factor of 10.

Theoretically, if all particles within the same sphere were put into the same group, the number of Neighbour Search calls should drop by a factor of $\langle N_{group} \rangle \simeq 230$. There may be two main reasons why this value is not reached: some time steps do not involve all particles, thus the density of active particles is lower than the density of all particles (which is used to calibrate the radius of the grouping sphere); moreover, the space-filling-curve

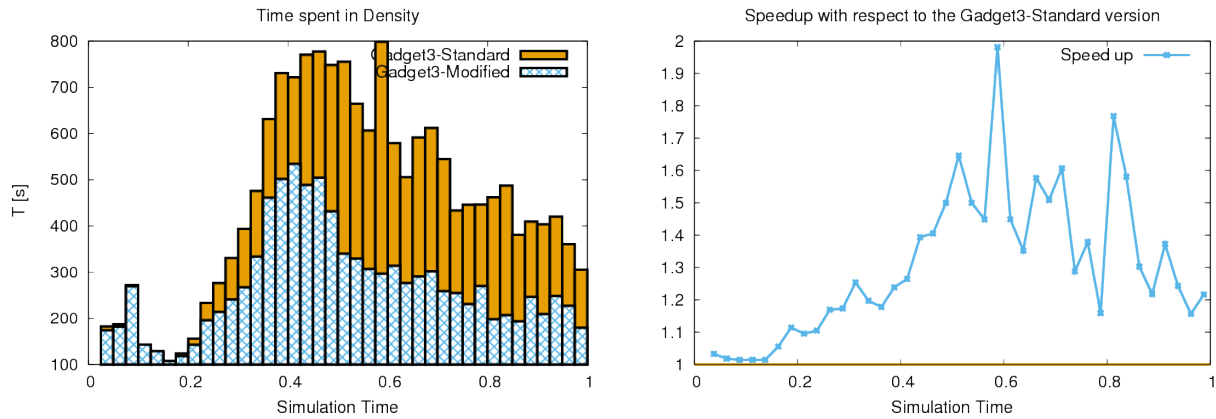


Figure 2.3: Left panel: every bin contains the time spent in executing the Density module. The orange (dark) histogram represents the standard version, light blue (light) histogram represents the modified version. Right panel: speedup of the modified version with respect to the standard version, as a function of the simulation time.

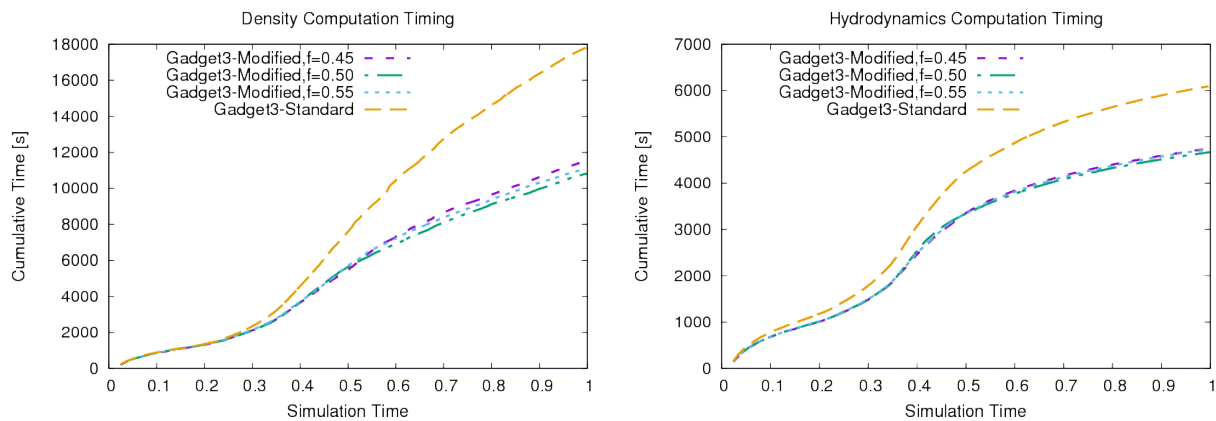


Figure 2.4: Wall time (in seconds) as a function of the simulation time for different runs: standard version and the new version with $f = 0.45, 0.50, 0.55$. Left panel: Density computation timings. Right panel: Hydrodynamics computation timings.

ordering will lead to particles outside the grouping sphere before the sphere is completely filled. Those two effects contribute in reducing the number of particles within a grouping sphere, thus increasing the number of Neighbour Search calls.

Figure 2.2 (right panel) shows the time (in seconds) spent to execute tree walks before and after the modification. Because the simulation runs on a multi core and using multiple threads, the total time corresponds to the sum of CPU times of all threads. This plot shows a speedup that reaches the order of 10 when the simulation time is approximately 0.4. Although the average time of a single Neighbour Search is supposed to be higher, the total time spent for doing the Neighbour Search in the new version is smaller.

The time spent in the density module is shown in Figure 2.3 (left panel). Here the Neighbour Recycling is implemented in both the first and the second phases of the density computation. Unlike previous plots, in this plot the time is the cumulative wall time spent by the code. As already pointed out, this new version increases the communications between MPI processes. The density module also has very expensive physics computations. The maximum speedup on the whole density module is larger than a factor of 2.

Figure 2.5 shows the projected gas distribution in three different phases of the simulation. At the beginning of the simulation gas is distributed homogeneously; this means that the majority of particles are in the same level of the tree. In the middle panel, voids and clusters can be seen. Particles in clusters require smaller time steps, and thus a larger number of Neighbour Search calls. This is in agreement with the peak of Neighbour Search calls around a simulation time of 0.4 in Figure 2.3. This explains why density computations became more intensive for a value of the simulation time greater than 0.4 (see Figure 2.4).

Now we check the impact of the Neighbour Recycling on the whole simulation. Figure 2.3 (right panel) shows the speedup obtained by implementing the Neighbour Recycling in both the Density and Hydrodynamics module (the two numerically most expensive modules). The total speedup reaches a peak of 2.2.

In Figure 2.4 (left panel), using the new approach we see a total cumulative execution time of the Density module of $1.0 \cdot 10^4 s$, while the standard version has $1.7 \cdot 10^4 s$, which correspond to a speedup of 1.64. Figure 2.4 (right panel) shows the same for the Hydrodynamics module. The old version spent a cumulative time of $6.0 \cdot 10^3 s$, whereas the new version has $4.6 \cdot 10^3 s$. Leading to a speedup in the hydrodynamics of 1.30. The Hydrodynamics module achieved a speedup of 1.30. Besides the Density module, a speedup can be seen also at the beginning of the simulation.

Figure 2.4 shows the wall time of the simulation when varying the parameter f . Since we do not knew a priori which value of f will have maximised the speedup, we found it by numerical experiments. We tried several values of it; values of f near zero gives no speedup, while values of f much greater than one slow down the whole simulation. In Figure 2.4 there are the timings for the setups with $f = 0.45, 0.50, 0.55$. The maximum speedup is obtained for $f = 0.50$ in both the Density and Hydrodynamics computations.

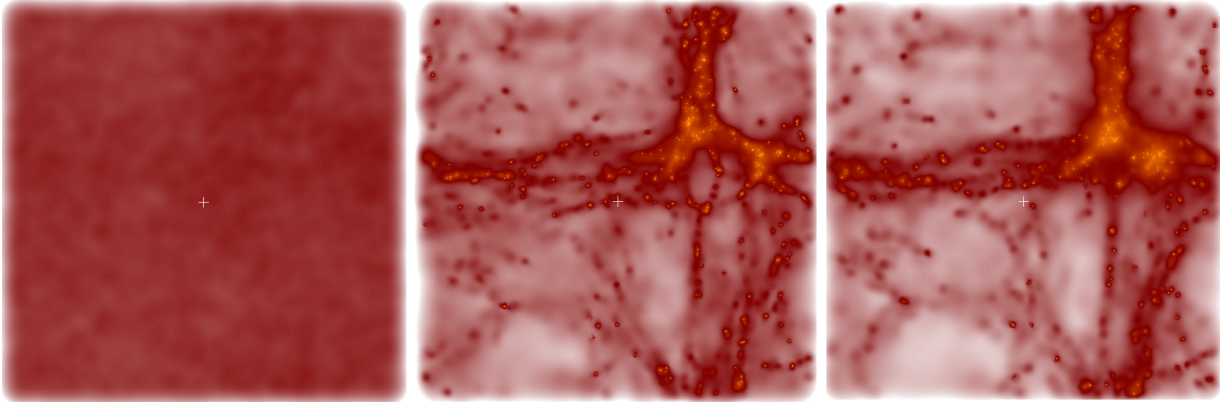


Figure 2.5: Gas distribution of the cosmological simulation `box5hr`. Left panel shows the gas distribution of nearly the initial conditions of the simulation; central panel at the middle of the simulation, where clusters start forming; right panel at the end of the simulation. The simulation contains also Dark Matter and stars that have been removed from those plots.

2.3 Conclusions

We developed and implemented a way to recycle neighbours to accelerate the Neighbour Search in order to fasten Gadget3. Our technique should work, in principle, for any N-Body code with a space-filling-curve ordering of particles.

This technique groups particles that will be processed one after the other and that are close enough, and makes a single neighbour search for them. We presented a version of the algorithm that scales the grouping radius with the local density. This version depends on a constant factor f . We found the value of f that gives the maximum speedup. In case of the simulation `box5hr` of the Magneticum project, corresponds to one half of the searching radius of the single particles. This radius, of course, depends on the way particles are grouped together. In this approach we opted for a grouping that depends on the distance from the first particle of the group. This decision is arbitrary and dictated by the simplicity of the implementation.

This configuration leads to a speedup of the density computation of 1.64, which is known to be one of the most expensive modules in Gadget3. Implementing this technique in the hydro-force computation too gives a speedup of the whole simulation of 1.34.

Chapter 3

The dependency of halo concentration on mass, redshift and fossilness in Magneticum hydrodynamic simulations

The content of this chapter has been submitted to MNRAS, see Ragagnin et al. (2018).

Antonio Ragagnin^{1,2,3}, Klaus Dolag^{3,4}, Lauro Moscardini^{5,6,7}, Andrea Biviano⁸, Mauro D’Onofrio⁹

¹ Leibniz-Rechenzentrum (LRZ), Boltzmannstrasse 1, D-85748 Garching, Germany

² Excellence Cluster Universe, Boltzmannstrasse 2r, D-85748 Garching, Germany

³ Universitäts-Sternwarte, Fakultät für Physik, Ludwig-Maximilians Universität München, Scheinerstrasse 1, D-81679 München Germany

⁴ Max-Planck-Institut für Astrophysik, Karl-Schwarzschild Strasse 1, D-85748 Garching bei München, Germany

⁵ Dipartimento di Fisica e Astronomia, Alma Mater Studiorum - Università di Bologna, via Piero Gobetti 93/2, I-40129 Bologna, Italy

⁶ INAF - Osservatorio di Astrofisica e Scienza dello Spazio di Bologna, via Piero Gobetti 93/3, I-40129 Bologna, Italy

⁷ INFN, Sezione di Bologna, viale Berti Pichat 6/2, I-40127 Bologna, Italy

⁸ INAF - Osservatorio Astronomico di Trieste, via G.B. Tiepolo 11, 34143 Trieste, Italy

⁹ Dipartimento di Fisica e Astronomia G. Galilei, Università di Padova, Vicolo Osservatorio3, I-35122 Padova, Italy

In the first part of this paper we study the dependency of the concentration on mass and redshift using three large N-body cosmological hydrodynamic simulations from the Magneticum project. We find a negative trend on the mass-concentration plane and a slightly negative redshift dependency between the concentration and redshift, in agreement with observations and other numerical works. We constrain the slope of the mass concentration quantity with an unprecedented mass range for hydrodynamic simulations. In the second part of this paper we investigate the origin of the large scatter of concentration by including in the fit also the fossil parameter, defined as the stellar mass ratio between the central galaxy and the most massive satellite. We study in detail the correlation between concentration and fossilness by following their evolution in haloes that undergo major merging and in objects without activity. We find that the internal region keeps eating satellites and this causes both an increase of the fossil parameter and a slow but steady decrease of the scale radius, which increases the concentration. Finally, we study the dependency of the concentration on the virial ratio including the energy term from the surface pressure (E_s). We find that the relation between concentration, fossilness and E_s is due to the existing relation between E_s and the halo accretion rate. In Section 3.2 we fit the concentration as a function of mass and redshift and compare our results with other observational and theoretical works. In Section 3.3 we define the fossilness parameter and show how it increases as the central galaxy of the halo accretes mass and how both the concentration and the fossil parameter change during merging events. We then fit the concentration as a function of the fossilness. In Section 3.4 we discuss the connection between the concentration and the virial ratio, the energy term from the surface pressure and the fossil parameter of the Magneticum clusters on both dark-matter only runs and runs with baryon physics.

In this work we analyse the concentration of haloes of the Magneticum project suite of simulations (Dolag et al., 2015, 2016). The Magneticum project produced a number of hydrodynamic simulations with different resolutions and ran over different volumes including also dark matter runs. The selection of haloes is discussed in Section 3.1.

The plan of this paper is as follows. In Section 3.2 we fit the concentration as a function of mass and redshift and compare our results with other observational and theoretical works. In Section 3.3 we define the fossilness parameter and show how it increases as the central galaxy of the halo accretes mass and how both the concentration and the fossil parameter change during merging events. We then fit the concentration as a function of the fossilness. In Section 3.4 we discuss the connection between the concentration and the virial ratio, the energy term from the surface pressure and the fossil parameter. We summarise our conclusions in Section 3.5.

3.1 Numerical Simulations

The Magneticum simulations (www.magneticum.org, Biffi et al., 2013; Saro et al., 2014; Steinborn et al., 2015; Dolag et al., 2016, 2015; Teklu et al., 2015; Steinborn et al., 2016; Bocquet et al., 2016; Remus et al., 2017) is a set of simulations that follow the evolution

Table 3.1: Individual setup of the three Magneticum simulations used in this work. The columns contain the name, the box size, the total number of particles, the mass of each dark matter particle, the initial mass of gas particles, the gravitational softening length of both dark matter and gas ϵ , and the gravitational softening length of star particles ϵ_* respectively.

Simulation Name	Size [Mpc/h]	n. part	m_{dm} [M_\odot/h]	m_{gas} [M_\odot/h]	ϵ [kpc/h]	ϵ_* [kpc/h]
Box4/uhr	48	$2 \cdot 576^3$	$3.6 \cdot 10^7$	$7.3 \cdot 10^6$	1.4	0.7
Box2b/hr	640	$2 \cdot 2880^3$	$6.9 \cdot 10^8$	$1.4 \cdot 10^8$	3.75	2
Box0/mr	2688	$2 \cdot 4536^3$	$1.3 \cdot 10^{10}$	$2.6 \cdot 10^9$	10	5

Table 3.2: Number of haloes in each snapshot, that have M_{200} higher than minimum mass for resolved haloes (corresponding to at least 10^4 particles).

Simulation	redshift	0	0.5	1	1.5	2	
Min M_{200} [M_\odot/h]	Max M_{200} ($z = 0$) [M_\odot/h]	n. haloes					
Box4/uhr	$1.3 \cdot 10^{11}$	$1.3 \cdot 10^{14}$	1845	1775	1934	1839	1782
Box2b/hr	$4 \cdot 10^{12}$	$1.8 \cdot 10^{15}$	156110	146339	99669	63542	48925
Box0/mr	$8 \cdot 10^{13}$	$3.8 \cdot 10^{15}$	329648	140560	21274	7792	1112

of overall up to $2 \cdot 10^{11}$ particles of dark matter, gas, stars and black holes on cosmological volumes. The simulations were performed with an extended version of the N–body/SPH code P-Gadget3 which itself is the successor of the code P-Gadget2 (Springel et al., 2005b; Springel, 2005a). P-Gadget3 uses an improved Smoothed Particle Hydrodynamics (SPH) solver for the hydrodynamics evolution of gas particles presented in Beck et al. (2016). Springel et al. (2005a) describe the treatment of radiative cooling, heating, ultraviolet (UV) back-ground, star formation and stellar feedback processes. Cooling follows 11 chemical elements ($H, He, C, N, O, Ne, Mg, Si, S, Ca, Fe$) using the publicly available CLOUDY photo-ionisation code (Ferland et al., 1998) while Fabjan et al. (2010); Hirschmann et al. (2014) describe prescriptions for black hole growth and for feedback from AGNs .

Galaxy haloes are identified using a friend-of-friend (FoF) algorithm and sub-haloes are identified using a version of SUBFIND (Springel et al., 2001a), adapted by Dolag et al. (2009) to include the baryon component.

The simulations assume a cosmological model in agreement with the WMAP7 results (Komatsu et al., 2011), with total matter density parameter $\Omega_{0,m} = 0.272$, a baryonic fraction of 16.8%, Hubble constant $H_0 = 70.4 \text{ km/s/Mpc}$, index of the primordial power spectrum $n = 0.963$ and a normalisation of the power spectrum corresponding to $\sigma_8 = 0.809$.

In particular, we use three of the Magneticum simulations presented in Table 3.1. We use Box0/mr to follow the most massive haloes, Box2b/hr to follow haloes within an

Table 3.3: Fit parameters of $c_{200}(M_{200})$ as a power law of the halo mass as in Equation 3.1 for each redshift bin.

redshift	A	B
$z = 0$	6.25 ± 0.07	-0.121 ± 0.004
$z = 0.5$	5.79 ± 0.07	-0.122 ± 0.004
$z = 1$	5.26 ± 0.08	-0.123 ± 0.007
$z = 1.5$	5.36 ± 0.07	-0.117 ± 0.006
$z = 2$	5.37 ± 0.07	-0.097 ± 0.006
$z = 0 - 2$	5.74 ± 0.07	-0.104 ± 0.004

intermediate mass range and Box4/uhr to follow haloes with masses in the galaxy range. From each simulation we selected snapshots nearest to redshifts $z \approx 0, 0.5, 1, 1.5$ and 2. In each snapshot we chose only haloes with a number of dark matter particles greater than 10^4 and then apply a cut in the critical mass so that all objects within this cut are well resolved. Table 3.2 lists the number of selected haloes, for each simulation and redshift, that match this mass-cut criterion.

3.2 The dependency of concentration on mass and redshift

For all selected Magneticum haloes in Table 3.2, we fit the concentration as a function of mass, using the following functional form:

$$c_{200} = A \cdot \left(\frac{M_{200}}{10^{13} M_{\odot}} \right)^B. \quad (3.1)$$

We performed the fit for various redshift bins $z = 0, 0.5, 1, 1.5, 2$ and over the whole range $z = 0 - 2$. The fit was performed using the average concentration computed in 20 logarithmic mass bins that span the whole mass range. The pivot mass $10^{13} M_{\odot}$ is the median mass of all selected haloes.

When we extract all haloes in a mass range over different snapshots from a simulation, it happens that most haloes at high redshift will be re-selected at lower redshift. We argue that this does not introduce a bias in the selection: in fact, the time between the two snapshots is longer than the dynamical time of the halo, ensuring that there is no correlation between the dynamical states of the two objects after such a long period of time.

Table 3.3 shows the fit parameters and their errors that are given by the cross-correlation matrix. The concentration at $10^{14} M_{\odot}$ evolves very weakly with redshift. In order to confirm this, for all selected haloes presented in Table 3.2, we also performed a fit of the halo concentration as a power law of mass and redshift using the relation

$$c_{200} = A \cdot \left(\frac{M_{200}}{10^{13} M_{\odot}} \right)^B \left(\frac{1.47}{1+z} \right)^C. \quad (3.2)$$

The fit was made on the average concentration of the haloes binned by the 5 redshift bins on the same mass bins as before and for the redshift dependency we use the median redshift value of 1.47 as pivot. The fit, performed over all objects gives:

$$\begin{aligned} A &= 6.02 \pm 0.04 \\ B &= -0.12 \pm 0.01 \\ C &= 0.16 \pm 0.01 \end{aligned} \quad (3.3)$$

We can see that the redshift dependency, represented by the parameter C, is low although it differs from zero.

Figure 3.1 shows the mass-concentration plane of Magneticum haloes, where different panels display data at different redshifts. Over-plotted are the fit relations for $c_{200} \propto M_{200}^B$ and $c_{200} \propto M_{200}^B \cdot (1+z)^{-C}$.

Table 3.2 reports a review of the slope values of the mass-concentration plane found on both theoretical and observational works. Figure 3.2 shows a plot of the same data. When the slope of the mass-concentration relation had an uncertainty smaller than few percents, we extrapolated the value of the concentration at the mass of $10^{14} M_{\odot}$ using $h = 0.704$.

Bullock et al. (2001) present one of the first analytical and numerical work on concentration in simulations. They predicted the concentration within the virial radius, that in this work has been converted to a concentration over R_{200} . Although their simulations were performed with a relatively low resolution, their concentration extrapolated at $10^{14} M_{\odot}$ is within the scatter of present days studies. Neto et al. (2007) employ the first very large dark matter-only N-body cosmological simulation, the Millennium simulation, see Springel (2005a) where they constrain the mass-concentration dependency accurately over several orders of magnitudes in mass for dark matter only runs.

Pratt and Arnaud (2005) use X-ray data from XMM-Newton, Mandelbaum et al. (2008); Shan et al. (2017) use lensing from SDSS images, while Covone et al. (2014); Mantz et al. (2016); Groener et al. (2016); Covone et al. (2014); Umetsu et al. (2016) combine both lensing and X-ray reconstruction techniques to find the concentration of the dark matter component of haloes. Observations with X-ray data have usually high uncertainties and need to make assumptions on the dependency between the baryon and the dark matter profiles, producing data with large uncertainties. The low mass regime of the plot shows observations of galaxies from the DiskMass survey from Martinsson et al. (2013). Points from the DiskMass survey cover a very large range of concentration values for low massive haloes, in contrast with simulations. Correa et al. (2015) adopted a semi-analytical model (SAM) that predicts concentration over 5 orders of magnitude. Groener et al. (2016) stack all observational mass-concentration data found in literature and made a single fit from it. Klypin et al. (2016) show the results of the MultiDark N-body simulation and produce a lower concentration than Magneticum haloes. Meneghetti et al. (2014) present a numerical

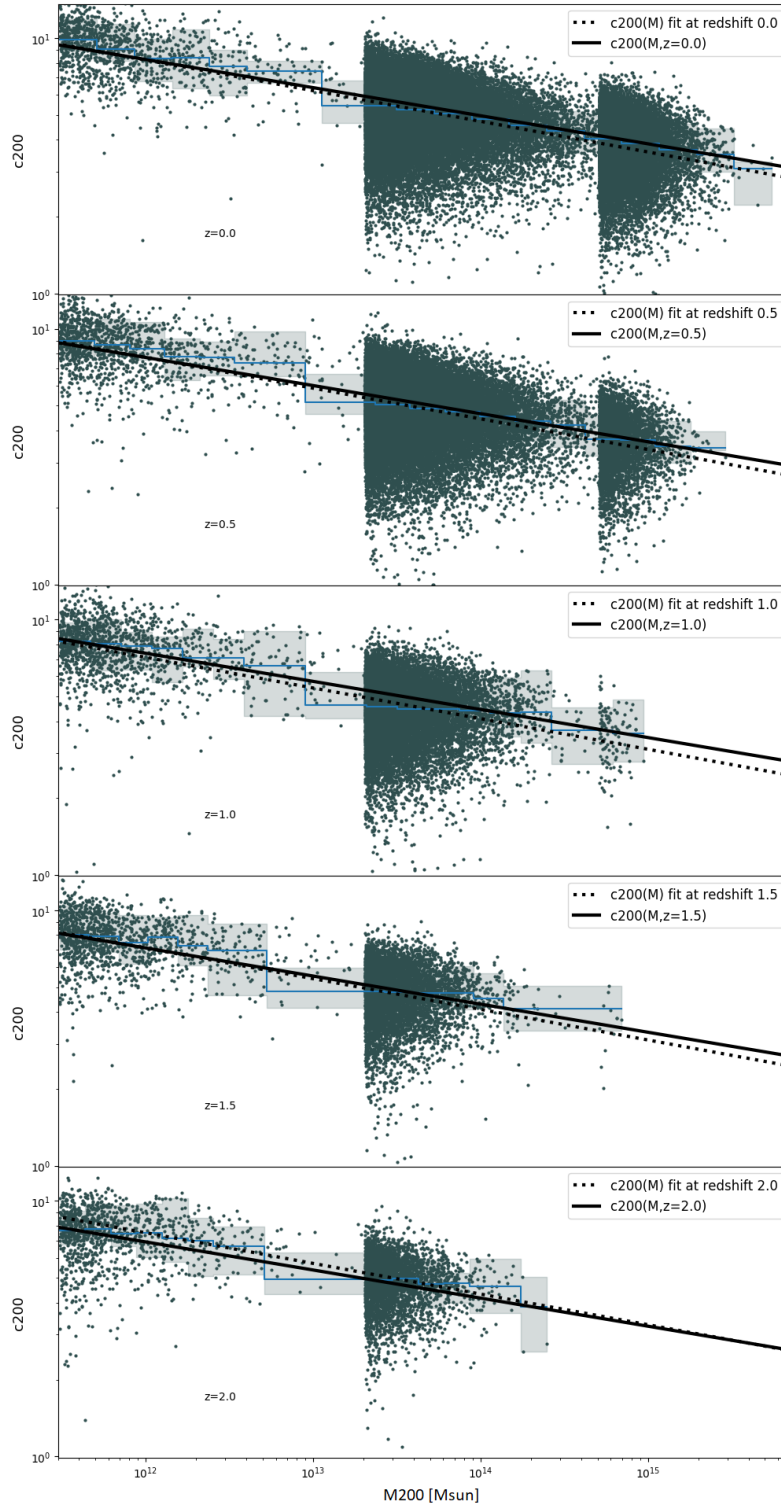


Figure 3.1: Mass-concentration relation for the well resolved haloes in the three Magneticum simulations Box4/ubr, Box2b/hr and Box0/mr (dark points). Each simulation covers three different mass ranges, respectively $M_{200} > 3 \cdot 10^{11} M_{\odot}$, $M_{200} > 2 \cdot 10^{13} M_{\odot}$ and $M_{200} > 5 \cdot 10^{14} M_{\odot}$. In each panel we show haloes of a different redshift bin, the median of the concentration (blue curve), the locus containing 50% of points (shaded area), the fit obtained with a $c_{200}(M_{200})$ fit as in Equation 3.1 and $c_{200}(M_{200}, z)$ as in Equation 3.2 (dotted and solid lines, respectively).

3.2 The dependency of concentration on mass and redshift

Table 3.4: Mass ranges and fit parameters of the mass-concentration relation in literature. The value of $c_{200}(10^{14}M_{\odot})$ is extrapolated at $z = 0$ when the relative error in the fit parameter is smaller than few percents (when provided). Concentration in Bullock et al. (2001) has been converted from c_{vir} to c_{200} .

authors	mass range [M_{\odot}]	slope	$c_{200}(10^{14}M_{\odot})$	comments
Bullock et al. (2001)	10^{11} – $10^{14}h^{-1}$	≈ -0.3	4.1	N-body
Pratt and Arnaud (2005)	10^{14} – 10^{15}	N/A	4 – 6	X-ray from XMM-Newton
Neto et al. (2007)	10^{12} – $10^{15}h^{-1}$	-0.1	4.8	N-body from Millennium
Mandelbaum et al. (2008)	10^{12} – $10^{15}h^{-1}$	-0.13 ± 0.07	4.8	weak lensing via SDSS
Bhattacharya et al. (2013)	$\sim 3 \cdot 10^{12}$ – $10^{15}h^{-1}$	-0.08	4.7	N-body
Martinsson et al. (2013)	10^{11} – 10^{12}	N/A	N/A	Subset of DiskMass survey
Dutton and Macciò (2014)	$10^{12.5}$ – $10^{14.5}h^{-1}$	-0.905	5.2	N-body
Meneghetti et al. (2014)	$6 \cdot 10^{14}$ – $10^{15}h^{-1}$	-0.058	N/A	CLASH mock observations
Ludlow et al. (2014)	10^{12} – $10^{15}h^{-1}$	-0.1	5.5	N-body from Millennium
Covone et al. (2014)	$3 \cdot 10^{13}$ – $2 \cdot 10^{14}h^{-1}$	0.09	5.4	lensing from CFHTLenS
Correa et al. (2015)	N/A	-0.08	3.8	semi-analytical model
Merten et al. (2015)	$5 \cdot 10^{14}$ – $2 \cdot 10^{15}$	-0.32 ± 0.18	N/A	lensing+X rays on CLASH data
Mantz et al. (2016)	$5 \cdot 10^{14}$ – $2 \cdot 10^{15}$	-0.15	N/A	lensing and X-ray from CHANDRA and ROSAT on relaxed clusters
Groener et al. (2016)	$\sim 10^{15}$	-0.16	N/A	comprehensive study (lensing data)
Klypin et al. (2016)	10^{11} – $10^{15}h^{-1}$	-0.12	4.1	N-body from MultiDark
Shan et al. (2017)	$5 \cdot 10^{12}$ – $2 \cdot 10^{14}$	-0.13	3.3	weak lensing on SDSS/BOSS
Biviano et al. (2017)	10^{14} – $2 \cdot 10^{15}$	-0.11 ± 0.1	4.6	dynamics of OmegaWINGS clusters
This work	10^{11} – 10^{15}	-0.1	4.5	Hydro N-body from Magneticum

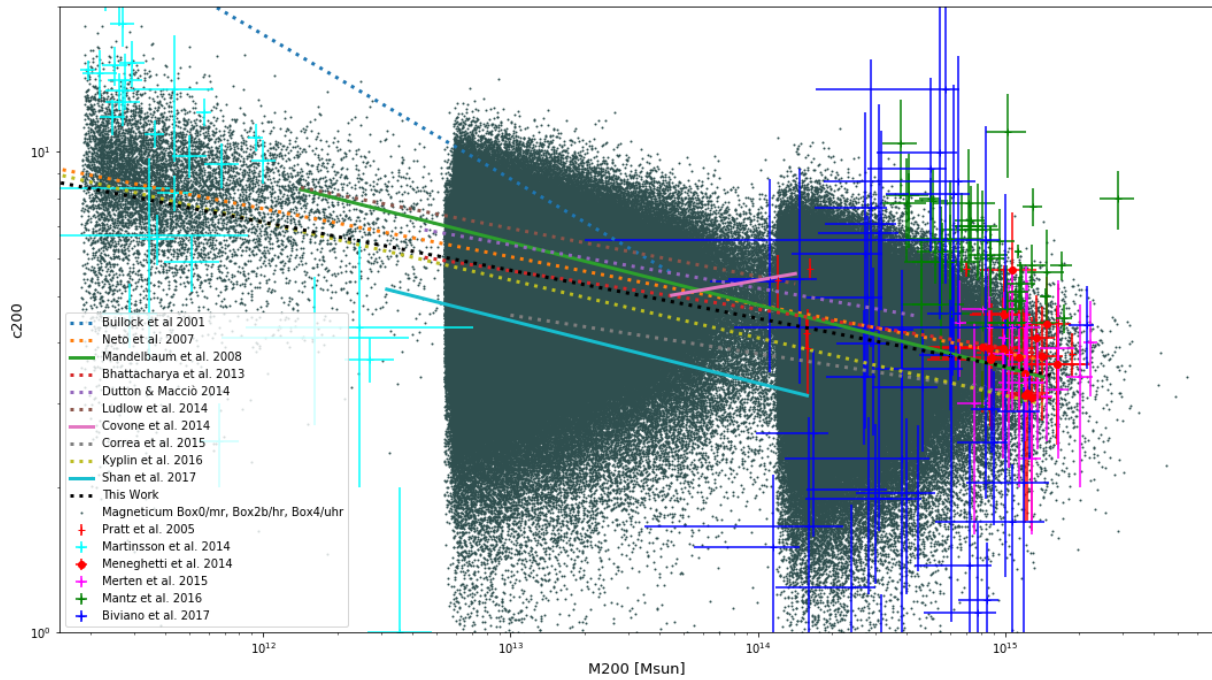


Figure 3.2: The mass-concentration relation. Grey points are haloes from the Magneticum simulations (see Table 3.1) at $z = 0$. Black dashed line is the $c_{200}(M_{200})$ median for Magneticum data points. Dashed lines are predictions from simulations and solid lines are fit from observed concentrations, both at $z = 0$. Error bars are from observations from dynamical mass analyses, with no redshift corrections. All mass conversions are made assuming $h = 0.704$.

work called MUSIC of CLASH where a number of simulated haloes have been chosen to make mock observations for CLASH. Mantz et al. (2016) present results from observations of relaxed haloes. These haloes have a higher concentration in agreement with theoretical studies. The high mass regime of the plot shows results from observations from WINGS (Biviano et al., 2017) and from CLASH (Merten et al., 2015). It must be taken into account that the galaxies from the DiskMass survey are a restricted sub-sample of a very large initial sample. Those galaxies have been chosen so that it is possible to compute the concentration. This may have introduced a significant bias in the concentration estimate. Merten et al. (2015); Biviano et al. (2017); Pratt and Arnaud (2005); Martinsson et al. (2013) compute halo properties using dynamical analyses which have larger uncertainties.

Magneticum low-mass haloes have comparatively lower concentration than dark matter only simulations. This is in agreement with other studies that show a lowering of concentration for low-mass haloes when AGN feedback is active (see Duffy et al., 2010).

3.3 Concentration and fossil parameter

The previous section showed how the concentration can span over an order of magnitude on both observational and theoretical works. In this section we show how the scatter is partially related to “how much” a halo is fossil. We first define a fossilness parameter and then study the evolution over time of both the fossilness and the concentration in some special objects.

Pratt et al. (2016); Kundert et al. (2015); Khosroshahi et al. (2006); Humphrey et al. (2012, 2011); Buote (2017) show how fossil objects have a higher concentration than the average. More generally, simulations found that dynamically relaxed haloes have a higher concentration (see e.g. Klypin et al., 2016).

A fossil object has been defined by Voevodkin et al. (2010) as having a difference in magnitude in the R band $\Delta m_R \geq 1.7$ between the most luminous object and the second most luminous object within a distance of $\frac{1}{2}R_{200}$ from the centre.

In our theoretical work we adapt the definition of the fossil parameter by quantifying it as the stellar mass ratio between central galaxy and most massive satellite:

$$\text{fossilness} = \frac{M_{\star, \text{central}}}{\max \{M_{\star, \text{satellite}}\}}. \quad (3.4)$$

We also extended the search of all satellites to R_{200} (instead of $\frac{1}{2}R_{200}$ proposed by Voevodkin et al. (2010)) because we consider objects outside R_{200} not to contribute to the dynamical state.

We convert the observed magnitude difference to a fossil parameter by assuming a constant ratio between galaxy masses and luminosities,

$$\text{fossilness} = 10^{\Delta m_R / 2.5}. \quad (3.5)$$

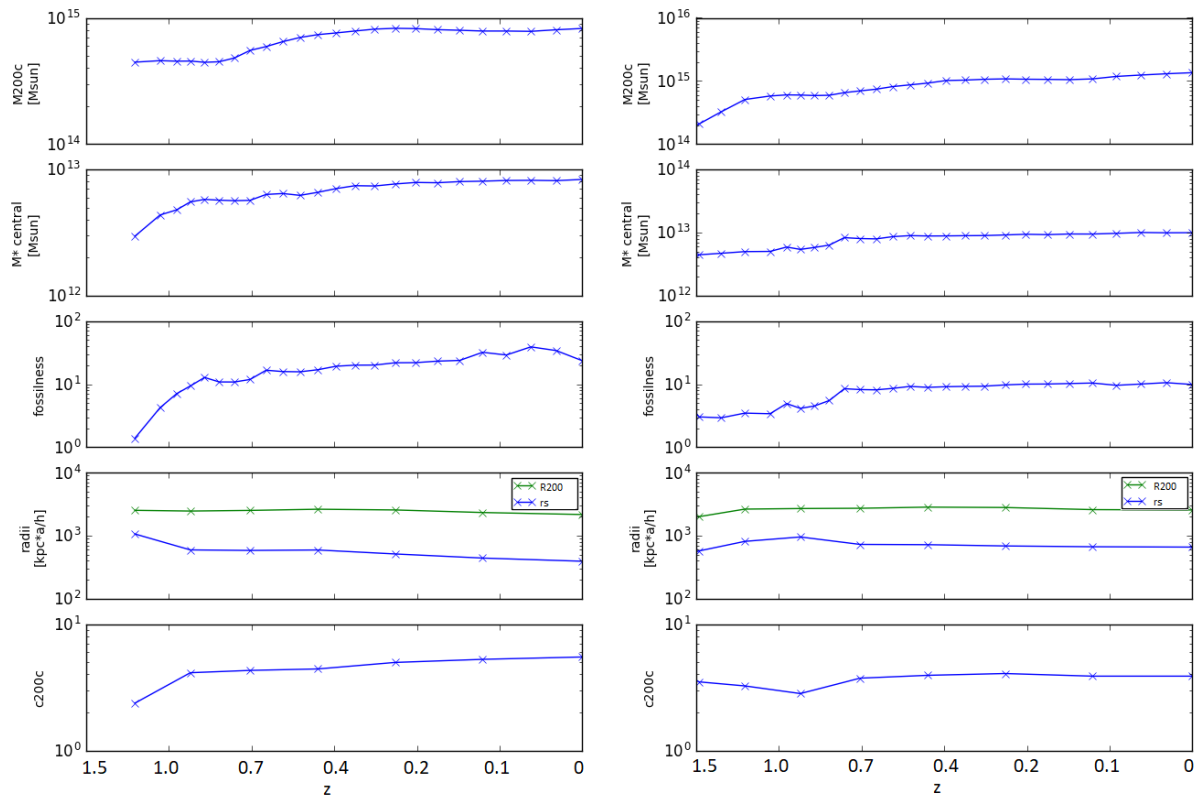


Figure 3.3: The evolution over time of two haloes (left and right panels) from Box0/mr: M_{200} , stellar mass of central galaxy, fossil parameter, R_{200} (in green) and r_s in (blue) and concentration from top to bottom. Both objects have been selected because they lived unperturbed for most of their time and have no major mergers. As long as their central galaxy eats satellites and keep accreting mass, the scale radius decreases and in turn, decreases the concentration to decrease, thus the relationship between concentration and fossilness.

This implies that the $\Delta m_R \geq 1.7$ threshold defined in Voevodkin et al. (2010) corresponds to a fossilness of

$$\text{fossilness} \gtrsim 4.5. \tag{3.6}$$

3.3.1 Concentration evolution in time

In order to understand what brought fossil objects such a high concentration, we followed the evolution of concentration and fossilness for a number of objects in the simulation Box/0mr. We present here two of the few most massive objects at $z = 1.5$ that lived almost un-perturbed until the end of the simulation. They have more than 10^5 particles and a final mass $M_{200} \approx 10^{15} M_{\odot}$. Figure 3.3 shows the evolution of halo mass, the stellar mass of central galaxy, scale radius, halo radius, fossilness and concentration of these haloes.

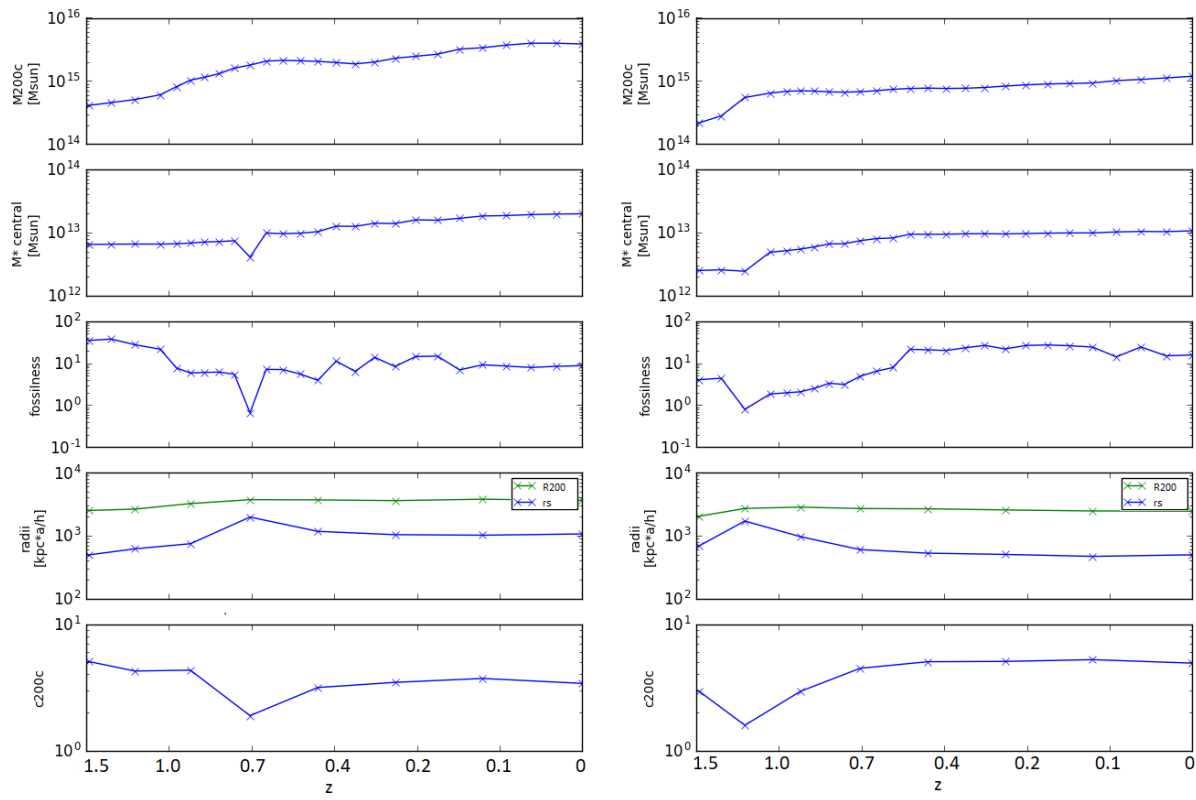


Figure 3.4: As for Figure 3.3 but for objects that have a single major merger in their history.

In these examples it is very easy to see that as long as their central galaxy eats satellites and keeps accreting mass, the scale radius decreases and makes their concentration higher and higher.

Additionally, in Figure 3.4 we show the evolution of two haloes that happen to have only one major merger in their history. When a merger happens then the fossil parameter drops because new massive satellites enter the system and the fossilness value decreases (see Eq. 3.4). As already expected from previous theoretical studies (Neto et al., 2007) we can see that the concentration goes down.

Neto et al. (2007) showed how the scatter in concentration can be partially described by the formation time, in this subsection we showed how a shift in concentration caused by a slow and steady increase of the concentration (led by a decrease of r_s) brings future fossil groups in the top region of the mass-concentration plane.

3.3.2 Concentration as a function of the fossil parameter

Figure 3.5 shows the Magneticum haloes concentration as a function of halo mass, colour coded by fossilness. We also show observational data of fossil groups taken from Khosroshahi et al. (2006); Humphrey et al. (2011, 2012); Pratt et al. (2016); Buote (2017) and haloes from Pratt and Arnaud (2005); Biviano et al. (2017); Bartalucci et al. (2018). Since most observational data were provided in terms of R_{500} and c_{500} , in this plot we show mass and concentration computed using $\Delta = 500$ for all data points. Haloes from Biviano et al. (2017) are colour coded by fossilness by converting the difference in magnitude to ratio of luminosities.

Figure 3.6 shows the concentration distribution for various mass, redshift and colour coded by fossilness bins. We can see that at each mass and redshift bin, the concentration increases with the fossil parameter, while the spread decreases as the fossil parameter increases.

There is a change in slope for very high value of the fossilness parameter so we modelled the dependence of concentration with slopes (see Figure 3.7, with also the fit results):

$$c_{200} = A \cdot \left(\frac{M_{200}}{10^{13} M_{\odot}} \right)^B \left(\frac{1.47}{1+z} \right)^C \cdot \left(\frac{fossilness}{F_0} \right)^D \left(1 + \frac{fossilness}{F_0} \right)^{E-D}. \quad (3.7)$$

The fit was performed with the binning technique as for the previous fits. Additionally, the fossil parameter was binned over 20 logarithmic bins of $fossilness > 1$. In this case, the exponent E maps the asymptotic exponent of c_{200} for high values of fossil parameters, while D is the exponent for low values of the fossil parameter. The value of F_0 in the fit should indicate where the two regimes of the fossilness slope starts to change.

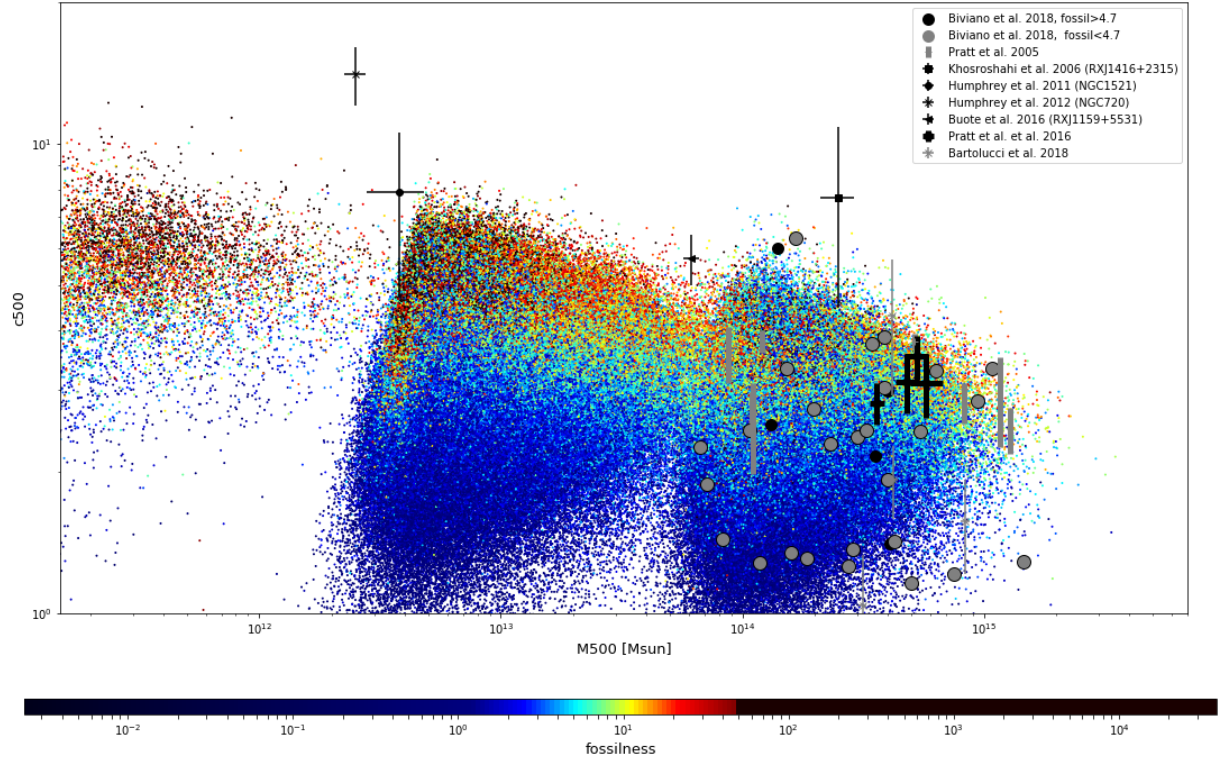


Figure 3.5: Mass-concentration plane. Mass and radius are computed using Δ_{500} . Points are from the Magneticum data and they are colour coded by fossil parameter (defined as the ratio between stellar masses of the central galaxy and the most massive satellite). The colour saturates to black for the 10% outliers in concentration. Fossil objects from Khosroshahi et al. (2006); Humphrey et al. (2011, 2012); Pratt et al. (2016); Buote (2017) are coloured in black, haloes from Pratt and Arnaud (2005); Bartalucci et al. (2018); Biviano et al. (2017) are coloured in grey. Data from Biviano et al. (2017) is divided between high and low fossiliness according to Equation 3.6.

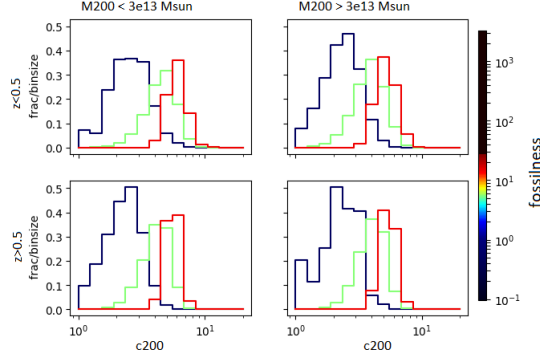


Figure 3.6: Distribution of concentration for various fossilness values. Left panels contain low mass haloes ($M_{200} < 3 \cdot 10^{13} M_{\odot}$) and right panels contain high mass haloes ($M_{200} > 3 \cdot 10^{13} M_{\odot}$), while top row refers to low redshift haloes ($z \leq 0.5$) and bottom row refers to high redshift haloes ($z > 0.5$).

The fit gives the following results:

$$\begin{aligned}
 A &= 7.5 \pm 0.1 \\
 B &= -0.1 \pm 0.1 \\
 C &= 0.13 \pm 0.01 \\
 D &= 0.40 \pm 0.03 \\
 F_0 &= 4.8 \pm 0.7 \\
 E &= -0.015 \pm 0.003
 \end{aligned}
 \tag{3.8}$$

Figure 3.7 shows the fitting relation as well as the data for single haloes and their median. For higher values it is necessary to use a double slope relation.

3.4 Virial ratio and concentration

In this section we study how the virial ratio of Magneticum haloes depend on the concentration and fossilness.

The moment of inertia I of a collisionless fluid under a force given by its gravitational potential Φ , obeys the time evolution equation:

$$\frac{1}{2} \frac{d^2 I}{dt^2} = 2K + W - E_s,$$

where the kinetic energy K includes the internal energy of gas, W is the total potential energy of the system and E_s is the energy from the surface pressure P at the halo boundary:

$$E_s = \int_S P(\vec{r}) r \cdot d\vec{S}.$$

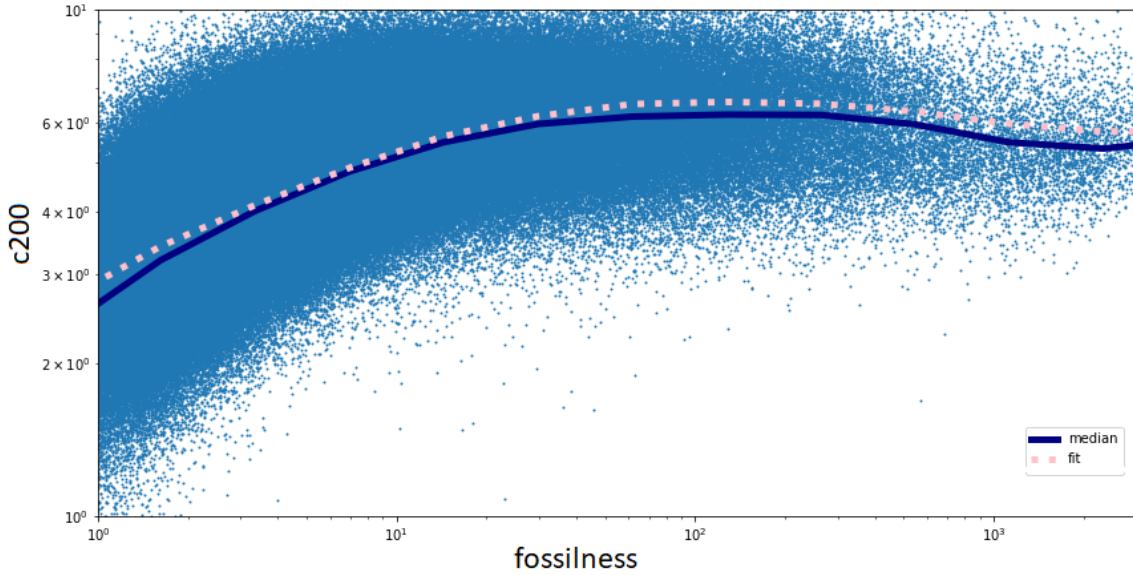


Figure 3.7: Concentration vs. fossilness for Magneticum data. Over plotted are the median, the concentration depending a double power law of the fossilness.

The pressure takes into account the pressure from the gas component.

A system at the equilibrium is supposed to have the so called virial ratio $\eta = 1$, where

$$\eta \equiv -\frac{2K - E_s}{W}.$$

For more details on how to compute these quantities and integrals see Chandrasekhar (1961); Binney and Tremaine (2008); Cui et al. (2017).

Figure 3.8 (left panel) shows the ratio $-2K/W$ versus the concentration for the haloes in the Magneticum Box0/mr run while Figure 3.8 (right panel) shows η versus the concentration. The median η is close to 0.9 and it is generally lower than the median of $-2K/W$. Theoretical works as Klypin et al. (2016) found a lower virial ratio when considering the term E_s . From the figures we can see that there is a correlation between concentration and $-2K/W$, while the correlation is much weaker if we add E_s to the kinetic term.

Figure 3.9 shows the fossil parameter as a function of E_s/W colour coded by the concentration. Fossil objects have lower E_s (accreting less material from outside) than other clusters, thus their more external region has no activity (no in-fall or outfall of material). This is also in agreement with Figure 3.3 where the evolution of fossil concentration is dominated by their internal motions (central galaxy eats satellites).

64 3. The dependency of halo concentration on mass, redshift and fossilness in
Magneticum hydrodynamic simulations

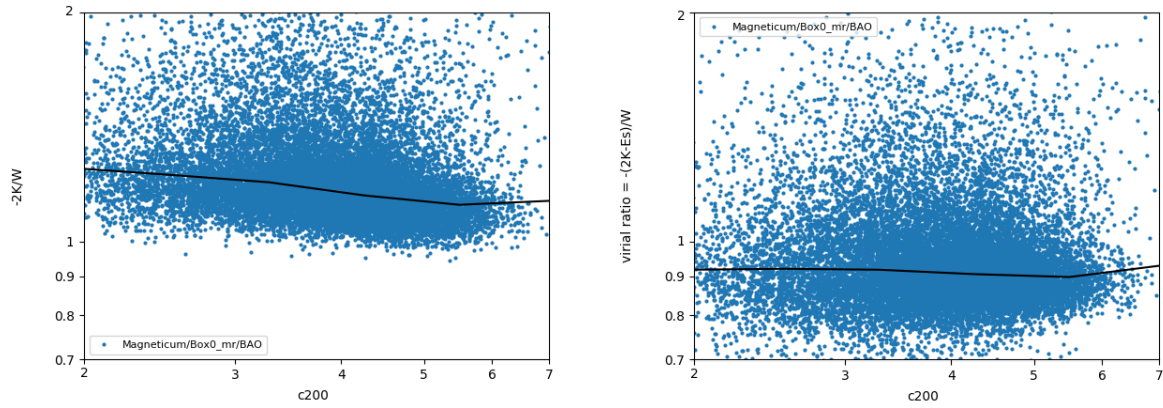


Figure 3.8: Virial ratio without (left panel) and with (right panel) the correction from the pressure term, as a function of the concentration for the simulation Magneticum/Box0/mr.

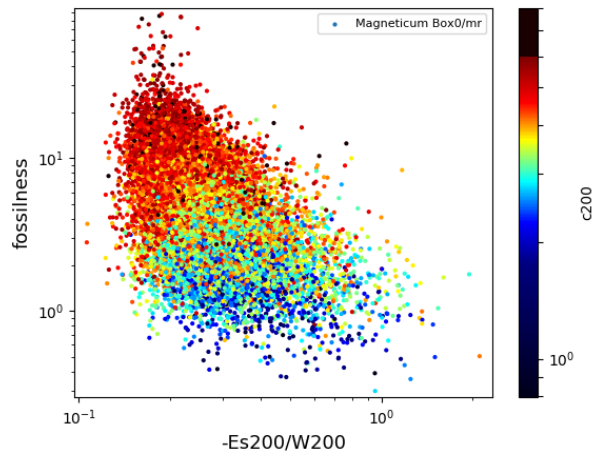


Figure 3.9: Fossilness versus virial ratio for Magneticum/Box0/bao run, colour coded by concentration (colour bar on the right). The colour saturates to black for the 10% outliers in concentration.

3.5 Conclusions

We used three cosmological hydrodynamic simulations from the Magneticum suite to cover a mass range from $3 \cdot 10^{11}$ to $6 \cdot 10^{15} M_{\odot}$ of well resolved clusters from redshift zero to redshift 2 and we computed the concentration for all well resolved haloes and fit it as a power law of mass and redshift.

This is the first study of the mass-concentration relation in hydrodynamic simulations covering several orders of magnitude in mass. For high massive clusters, we found a value of the concentration and its dependency on mass and redshift is in agreement within the large scatter already present in both observations and simulations.

An exception is made for the low mass regime, wherein the Magneticum simulation concentration is systematically lower than concentration found in studies based on dark matter only simulations. Such different behaviour is in agreement with studies of simulations with AGN feedback. They show how the effect of AGN feedback in low mass haloes is capable of lowering the concentration up to a factor of $\approx 15\%$ (see Figure 8 in Duffy et al., 2010) by removing baryons from the inner region of the halo. Thanks to the high mass regime of the Magneticum simulations we are able to capture this effect and its disappearance as the halo mass increases.

In the second part of this work we discussed the origin of the large scatter of concentration in the mass-concentration plane by studying its dependency on the fossilness. Fossil groups are supposed to have had a long period of inactivity and are known to have a higher concentration (see e.g. Neto et al., 2007; Dutton and Macciò, 2014; Pratt et al., 2016). Since we are working with hydrodynamic simulations, we were able to compute the halo fossilness exploiting the stellar masses of galaxies (see Eq. 3.4) and to compare our values to observations.

We showed that when a halo is left unperturbed, both fossilness and concentration steadily and slowly grow with time. This is in contrast with more naive models where an unperturbed halo keeps its concentration making it a mere function of its collapse time (as in Bullock et al., 2001). Interestingly, we found that this change of concentration is due to a decline of the scale radius. We also showed how the scale radius and fossilness increase or decrease together when a major merger occurs (see Figure 3.4). From these analyses, we found that those two effects drive the correlation between concentration and fossil parameter.

Our findings are not in contrast with the fact that relaxed and fossil objects start with a high concentration because of their early formation times, but we show how an additional steady increase of the concentration pushes these objects in the very high region of the mass-concentration plane.

The correlation between concentration and fossilness saturates after $fossilness \approx 20 - 30$ (see the plateau in Figure 3.7) and we provide a double power law fit parameters.

We then examined the concentration as a function of the virial ratio $\eta = -(2K - E_s)/W$ and as a function of the energy from the surface pressure. We found a weak dependency of the concentration on $-(2K - E_s)/W$ and very weak on the terms $-2K/W$ and E_s . The difference between $-2K/W$ and $-(2K - E_s)/W$ is higher for haloes with

lower concentration. This implies that low concentration objects are accreting material from the outside and it is in agreement with the idea that low-concentration haloes are not relaxed. This is compatible with other theoretical works as Klypin et al. (2016).

In Figure 3.9 we showed how the fossil parameter, the concentration and E_s are interconnected. A large value of E_s means that the cluster has a considerable amount of in-falling material and this translates into a low concentration and low fossil parameter; while a low value of E_s (no in-falling material) can be related to both high and low concentrated clusters. Fossil objects have a lower value of E_s indicating a low accretion rate.

Work has still to be done to study the relation between fossil parameter and other quantities that are well known to be tied with the dynamical state of a system, for instance, the difference between centre of mass and density peak position), or the velocity dispersion deviation between the one inferred from the virial theorem. Additional work is also needed in order to understand the connection between central galaxy eating and the redistribution of the angular momentum within the halo, which in turn may give hints on the weak dependency between concentration and spin parameter (as found by Macciò et al., 2008).

Chapter 4

A web portal for hydrodynamic, cosmological simulations

*The content of this chapter has been published in Ragagnin et al. (2017),
the service is accessible via <https://c2papcosmosim.srv.lrz.de>.*

Antonio Ragagnin^{1,2,3}, Klaus Dolag^{2,4}, Veronica Biffi^{4,6}, Marianne Cadolle Bel^{2,7}, Nicolay J. Hammer¹, Aliaksei Krukau^{1,2}, Margarita Petkova^{2,3}, Daniel Steinborn¹

¹ Leibniz-Rechenzentrum (LRZ), Boltzmannstrasse 1, D-85748 Garching, Germany

² Excellence Cluster Universe, Boltzmannstrasse 2r, D-85748 Garching, Germany

³ Universitäts-Sternwarte, Fakultät für Physik, Ludwig-Maximilians Universität München, Scheinerstrasse 1, D-81679 München, Germany

⁴ Max-Planck-Institut für Astrophysik, Karl-Schwarzschild Strasse 1, D-85748 Garching bei München, Germany

⁵ Astronomy Unit, Department of Physics, University of Trieste, via Tiepolo 11, I-34131 Trieste, Italy

⁶ INAF, Osservatorio Astronomico di Trieste, via Tiepolo 11, I-34131 Trieste, Italy

⁷ Max Planck Computing and Data Facility (MPCDF), Gießenbachstrasse 2, D-85748 Garching, Germany

This article describes a data centre hosting a web portal for accessing and sharing the output of large, cosmological, hydro-dynamical simulations with a broad scientific community. It also allows users to receive related scientific data products by directly processing the raw simulation data on a remote computing cluster.

The data centre has a multi-layer structure: a web portal, a job control layer, a computing cluster and a HPC storage system. The outer layer enables users to choose an object from the simulations. Objects can be selected by visually inspecting 2D maps of the simulation data, by performing highly compounded and elaborated queries or graphically by plotting arbitrary combinations of properties. The user can run analysis tools on a chosen object. These services allow users to run analysis tools on the raw simulation data. The job control layer is responsible for handling and performing the analysis jobs, which are executed on a computing cluster. The innermost layer is formed by a HPC storage system which hosts the large, raw simulation data.

The following services are available for the users: (I) `CLUSTERINSPECT` visualizes properties of member galaxies of a selected galaxy cluster; (II) `SIMCUT` returns the raw data of a sub-volume around a selected object from a simulation, containing all the original, hydro-dynamical quantities; (III) `SMAC` creates idealised 2D maps of various, physical quantities and observables of a selected object; (IV) `PHOX` generates virtual X-ray observations with specifications of various current and upcoming instruments.

4.1 Introduction

Entering the so-called era of “precision cosmology” it becomes more and more clear that a theoretical counterpart in the form of very complex, hydrodynamic cosmological simulations is needed to interpret data from upcoming astronomical surveys and current instruments like PLANCK, South Pole Telescope (SPT), PanStars, Dark Energy Survey (DES), Euclid, LOFAR, eROSITA and many more. Such simulations follow the growth of galaxies and their associated components (like stellar population and central black hole) with their interplay with the large scale environment they are embedded in. Upcoming surveys will map large volumes of the Universe as well as record the birth of the first structures, especially galaxies and even progenitors of massive galaxy clusters at high redshift. In fact, their large potential of determining the nature of dark matter and dark energy comes from being able to map the content and geometry of the Universe over most time in cosmic history. For theoretical models this means that simulations have to cover comparable large volumes, especially to host the rarest, most massive galaxy clusters expected to be the lighthouses of structure formation detectable at high redshift. While the Universe makes its transition from dark matter dominated to dark energy dominated (i.e. accelerated expansion), the objects which form within it make their transition from young, dynamically active and star formation-driven systems to more relaxed and near-equilibrium systems observed at low redshifts. Those simulations study the internal evolution of clusters of galaxies with respect to the evolution of the cosmological background. They will be essential to interpret the outstanding discoveries expected from upcoming surveys.

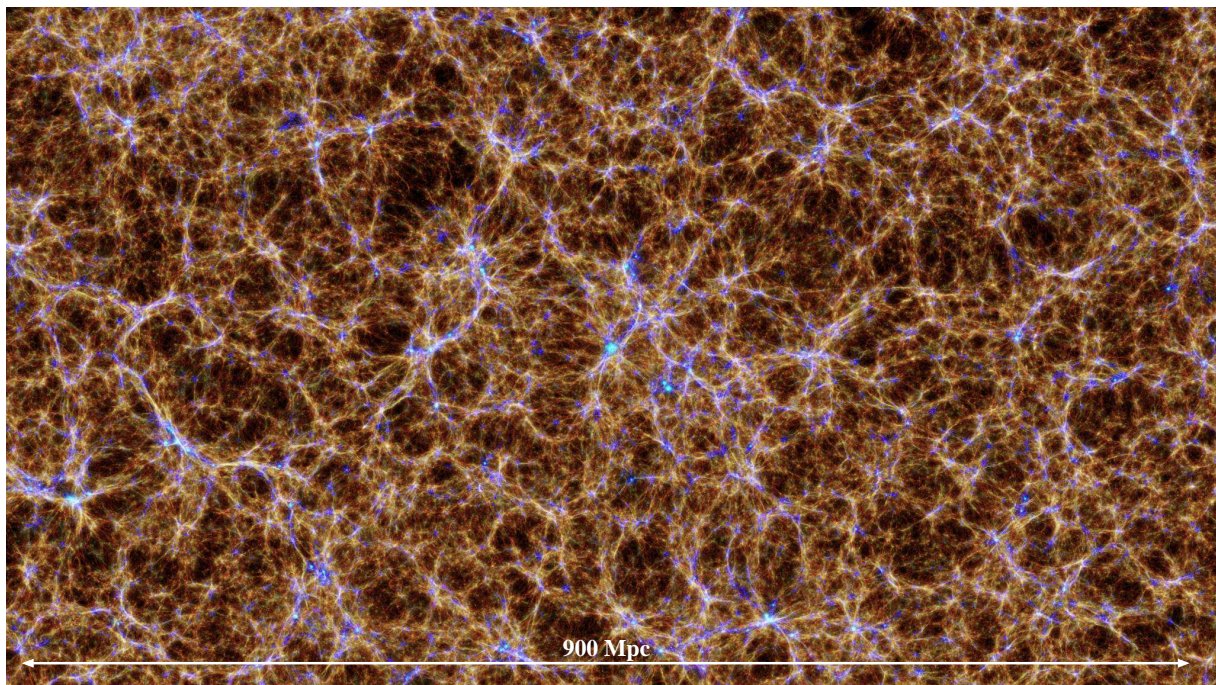


Figure 4.1: A visualisation of a cosmological large scale structure of the *Box2b/hr* simulation from the *Magneticum* project. This map shows diffuse baryons at $z = 0.2$, colour coded according to their temperature. The visualisation is centred on the most massive galaxy cluster in this simulation.

In this respect, efforts have been done in the recent years in order to share data sets of various kinds with the community. For instance, the Millennium Simulation Data Archive <http://wwwmpa.mpa-garching.mpg.de/Millennium/> (Lemson and Virgo Consortium, 2006) is a pioneering work in this field. With the Millennium Simulation Data Archive, the user is able to compose SQL queries over substructure and merger-tree data in order to extract haloes and galaxies from the Millennium Simulation.

Users can also download the raw data files. The Cosmosim.org project <https://www.cosmosim.org/> allows users to compose additional queries over the list of particles and various post processed quantities (grid cells of density field). The Illustris Galaxies Observatory http://www.illustris-project.org/galaxy_obs/ provides an application programming interface (API) where users can filter galaxies and download particle data from the Illustris simulations. The Australian Theoretical Virtual Observatory <https://tao.asvo.org.au/tao/about/> (Bernyk et al., 2016) is an online virtual laboratory where users can compose queries and run services on selected objects in the simulation, for instance producing mock observations or extracting light cones.

Section 4.2 describes data of cosmological simulations and section 4.3 describes the currently available infrastructure. In section 4.4 we describe how users can interact with the web interface and thereby compose science-driven queries to select objects. Section 4.5 describes the services currently implemented in the system.

4.2 The Simulations

In this section we present the simulations made accessible by our data centre.

4.2.1 The Magneticum Project

The Magneticum simulations <http://www.magneticum.org> (see Biffi et al. (2013); Saro et al. (2014); Hirschmann et al. (2014); Steinborn et al. (2015); Dolag et al. (2016, 2015); Teklu et al. (2015); Steinborn et al. (2016); Bocquet et al. (2016); Remus et al. (2017)) follow the evolution of up to 2×10^{11} particles in a series of cosmological boxes ranging in size from $(50\text{Mpc})^3$ to $(4\text{Gpc})^3$. A visualisation of the second largest cosmological simulation can be seen in Figure 4.1. These simulations were used to interpret Sunyaev-Zel'dovich (SZ) data from PLANCK (Planck Collaboration et al., 2013) and SPT (McDonald et al., 2014) as well as to predict cluster properties in X-rays for future missions such as Athena or Astro-H (Biffi et al., 2013). The first mock observations for the eROSITA cluster working group and the Athena+ white book were also produced based on these simulations. Other scientific goals that were achieved with these simulations included studying the properties of the intra cluster medium (ICM) in galaxy clusters (Dolag et al., 2016) as well as predicting the multi wavelength properties of the Active Galactic Nuclei (AGN) (Hirschmann et al., 2014; Steinborn et al., 2015). The large dynamical range probed by the combination of resolution levels and cosmological volumes also allowed us to calibrate the cosmological mass function based on hydro-dynamical simulations to a level required by future cosmological probes

(Bocquet et al., 2016). The detailed treatment of all relevant physical processes allowed us to investigate dynamical properties of galaxies based on morphological classification (Teklu et al., 2015; Remus et al., 2017) for the first time. A small subset of these simulations also follows the evolution of magnetic fields.

The web portal currently allows the user to access a subset of the full *Magneticum* simulation set. The data centre hosts up to 28 outputs of a medium size simulation *Box2/hr*, which utilise 0.8×10^{10} particles, covering a volume of $(500Mpc)^3$ as well as 11 outputs of a larger size simulation; *Box2b/hr*, which utilise 5×10^{10} particles, covering a volume of $(900Mpc)^3$. For each cluster contained in the simulated volumes, the web portal shows to the user a set of pre-computed quantities. The set of pre-computed quantities is chosen to let users select objects in categories (for example, fossils or compact objects) that are widely studied.

4.2.2 The Metadata

Each galaxy cluster object has its metadata content, that is a list of properties associated with it (e.g. mass within a certain radius). The metadata associated with each galaxy cluster contain table which describes its galaxy members.

We extract a reduced subset of halos (made of galaxies, groups or clusters, depending on the size and resolution of the underlying simulation) by applying a lower mass cut and providing the relevant part of the available global properties as metadata for user queries as shown in table 4.3. In addition, for each halo we also store a list of all member galaxies (or satellites). For each of these galaxies/satellites we store some additional metadata as shown in table 4.4, which can be used to further refine user queries.

We use SUBFIND (Springel et al., 2001a; Dolag et al., 2009) to define properties of haloes and their sub-haloes. SUBFIND identifies substructures as locally over dense, gravitationally bound groups of particles. SUBFIND starts with a halo list identified through the Friends-of-Friends algorithm. For each halo and for each of its particles the local density is estimated using adaptive kernel estimation with a prescribed number of smoothing neighbours. Starting from isolated density peaks, additional particles are added in sequence of decreasing density. When a sequence of particles contains a saddle point in the global density field that connects two disjoint over-dense regions, the smaller structure is treated as a substructure candidate, followed by the merging of the two regions. All substructure candidates are subjected to an iterative unbinding procedure with a tree-based calculation of the potential. These structures can then be associated with galaxy clusters, galaxy groups and galaxies and their integrated properties (like gas mass, stellar mass or star-formation rate) can be calculated.

4.2.3 Raw Simulation Data Layout

For the *Magneticum* project, we developed a specific output format for very large scale, n-body, cosmological, hydrodynamic simulations. It employs a spatial space filling curve to produce auxiliary data which allows a fast and direct access to spatial regions within the

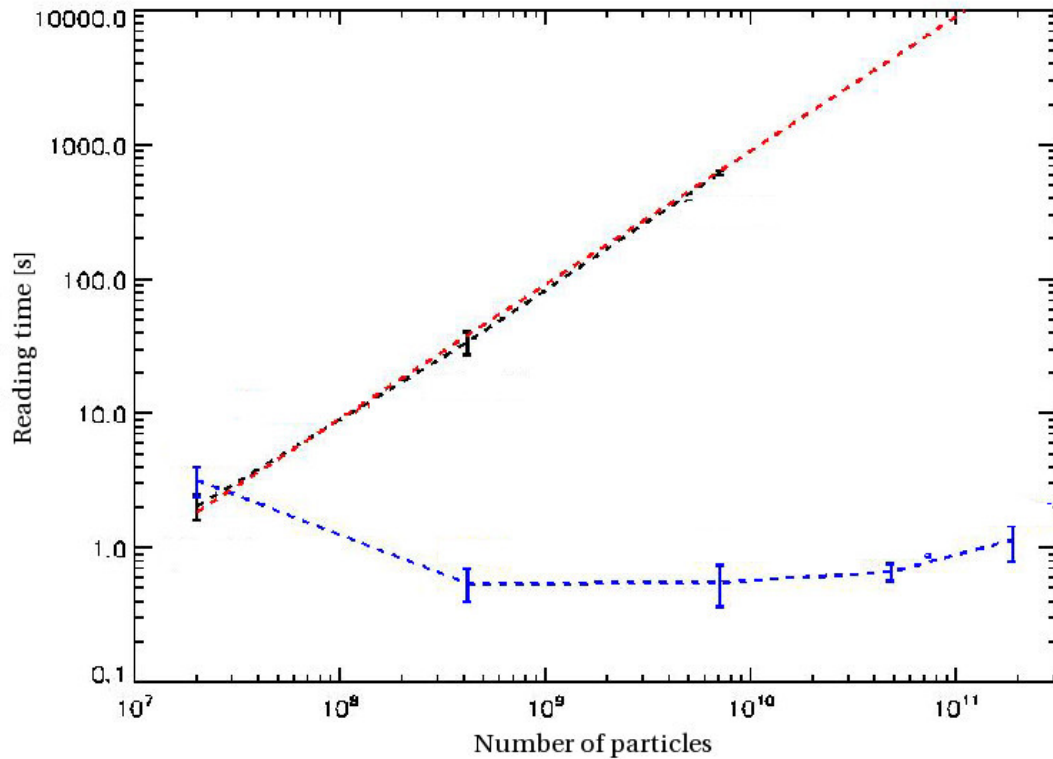


Figure 4.2: Wall clock time spent reading all data for the most massive galaxy cluster from a snapshot as function of the total number of particles for increasing simulation size. In black line there is the brute force approach by reading all data while the blue line is the timing of the improved algorithm. The improved algorithm does a spatial selection of the snapshot by use of key-index files. Those files allow to readout only the relevant part of the snapshot files. The percentage at the individual data points indicate the fraction of particles to be read compared to the overall particle number.

output of smoothed particle hydrodynamics (SPH) simulation. This indexing scheme was extended to work with the multi-file and multi-component output of such SPH simulations. To optimise the access to individual data, some files contains data regarding the content and length of individual data blocks.

Figure 4.2 shows that the reading of all particles within the virial radius of the most massive galaxy cluster in any of the simulations takes significantly less than 1 second. The overhead to create and read the index list is negligible. The algorithm speeds up also the Magneticum *Box0/mr* simulation post processing. This simulation utilises almost 2×10^{11} particles and the information have to be filtered out of the individual snapshots, which for one time instance occupy 20TB on disk. 4.8 shows in detail how the index list is stored.

4.3 Structure of the web portal

Figure 4.3 illustrates our multi-layer structure (the different layers are separated by a dashed red line). Between those layers, data and processes flow over the web portal, the database, the job control layer, the computing cluster (where the analysis tools are actually executed), and the storage system (where the raw simulation data are stored). The need for a separation between the web interface and the back end, arises from both the necessity of users to run personalised jobs on raw data, managed by a job scheduler of the computing cluster and the protection of the data from unauthorised access.

In 4.9 we show how to prepare data of a simulation in order to add it to the web portal.

4.3.1 Overview of the multi-layer architecture

The user selects a service from the web interface and this information is written into a job database. In our implementation we used PostgreSQL 9.4.6 for all the different databases (which in our case is implemented as a separate independent instance). The backend is triggered by the job database and will configure and submit jobs to a computing cluster which will execute them. Once a job is added, a trigger in the database will make the backend send the job to the computing cluster. Finally, the backend delivers the according data products to the user via a download link which is valid for a limited time. The computing cluster must have access to the HPC file system where the simulation data are stored, however, it does not need to store the data locally.

Almost all parts are based on common packages and available libraries except the core of the backend, which is a customised component tailored for the data flows and job requests to the specific needs of the scientific software performing the data analysis.

4.3.2 The Web Portal

The outer layer of the diagram in Figure 4.3 is the web portal. The web portal consists of two main parts, the web interface. Our web server is built using the Python micro framework FLASK 0.11 and the simulations database. While this database hosts the metadata of all

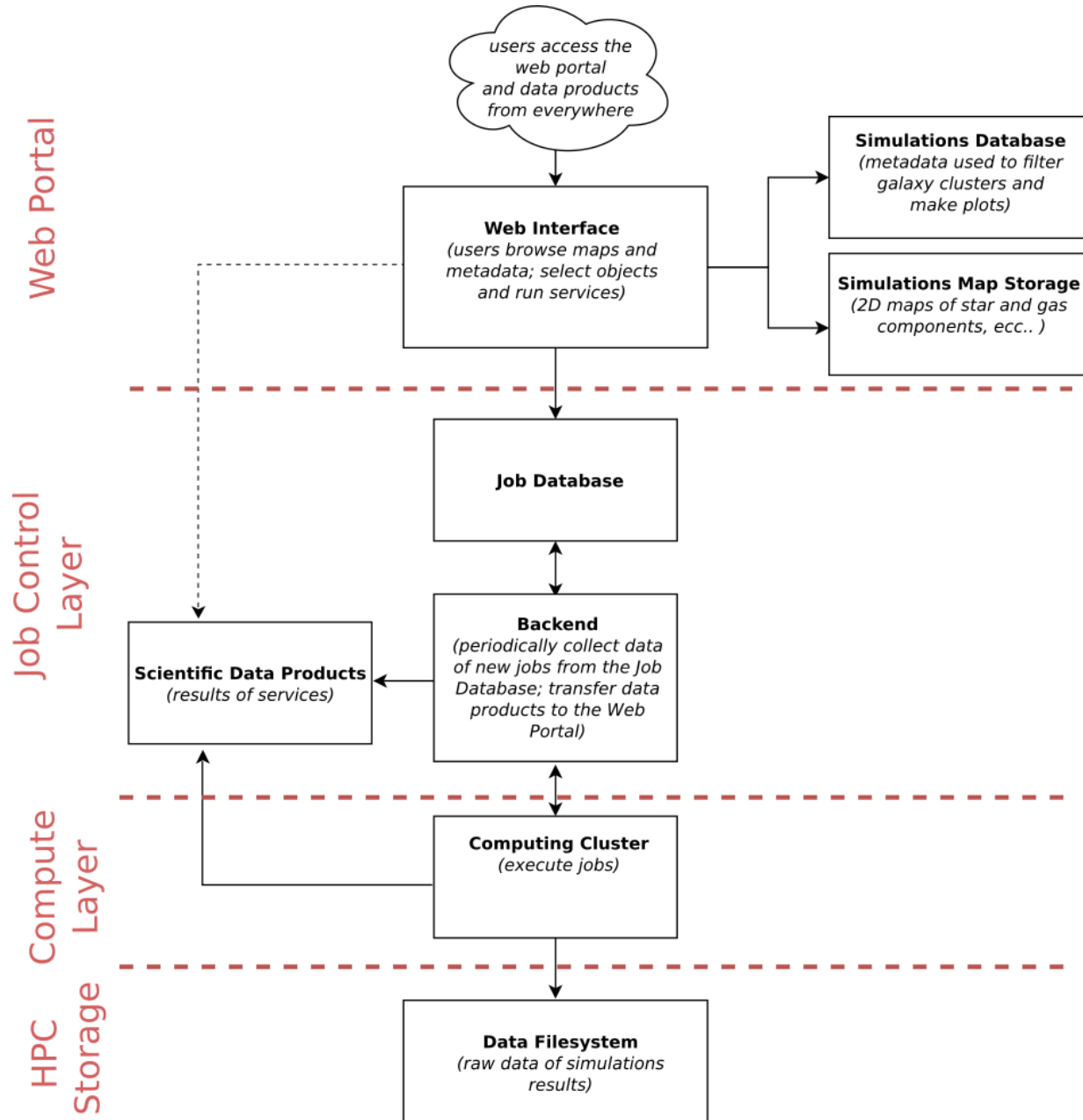


Figure 4.3: Schema summarising how the processes are distributed to different parts of the existing infrastructure, and how the data flows are within the fully operational prototype of the web portal.

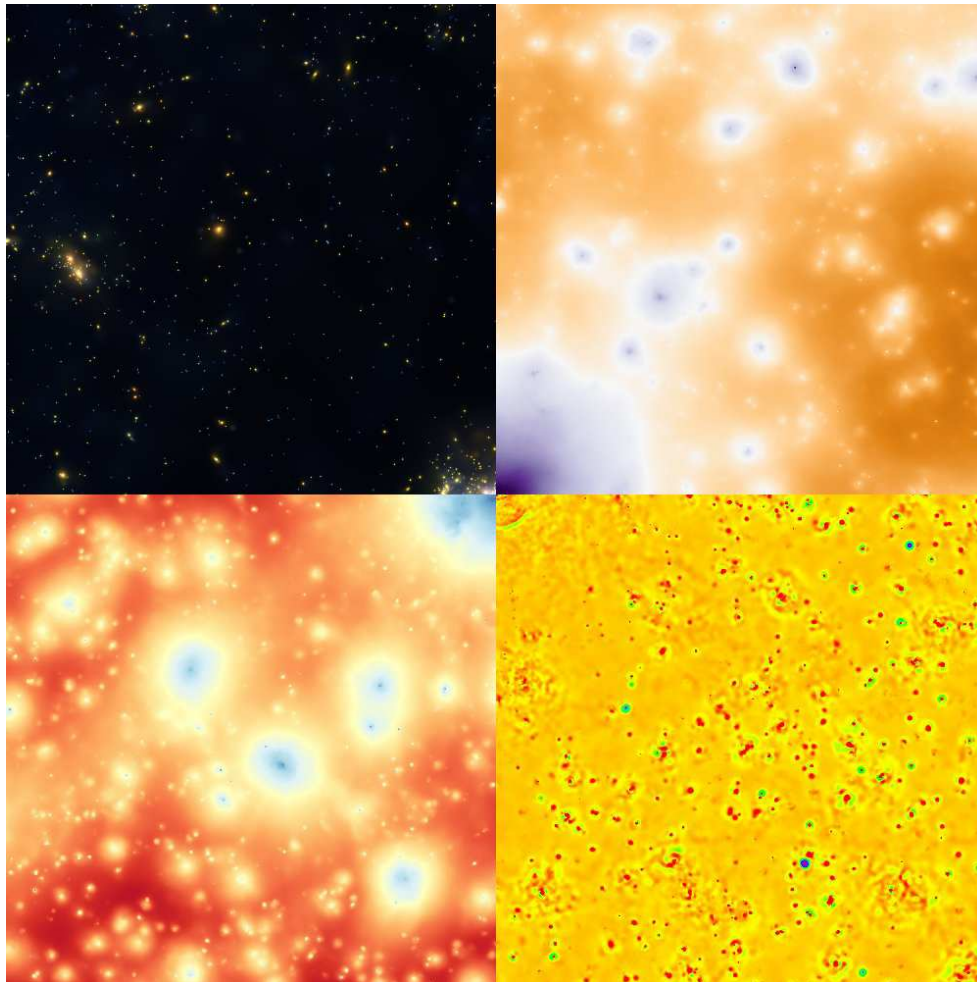


Figure 4.4: The four quadrants, centred on a merging cluster, report: the stellar component (*Stars*, upper left), the ICM pressure (*Compton Y*, upper right), the ICM X-ray emission (*ICM*, lower left) and the pressure fluctuations (*Shocks*, lower right).

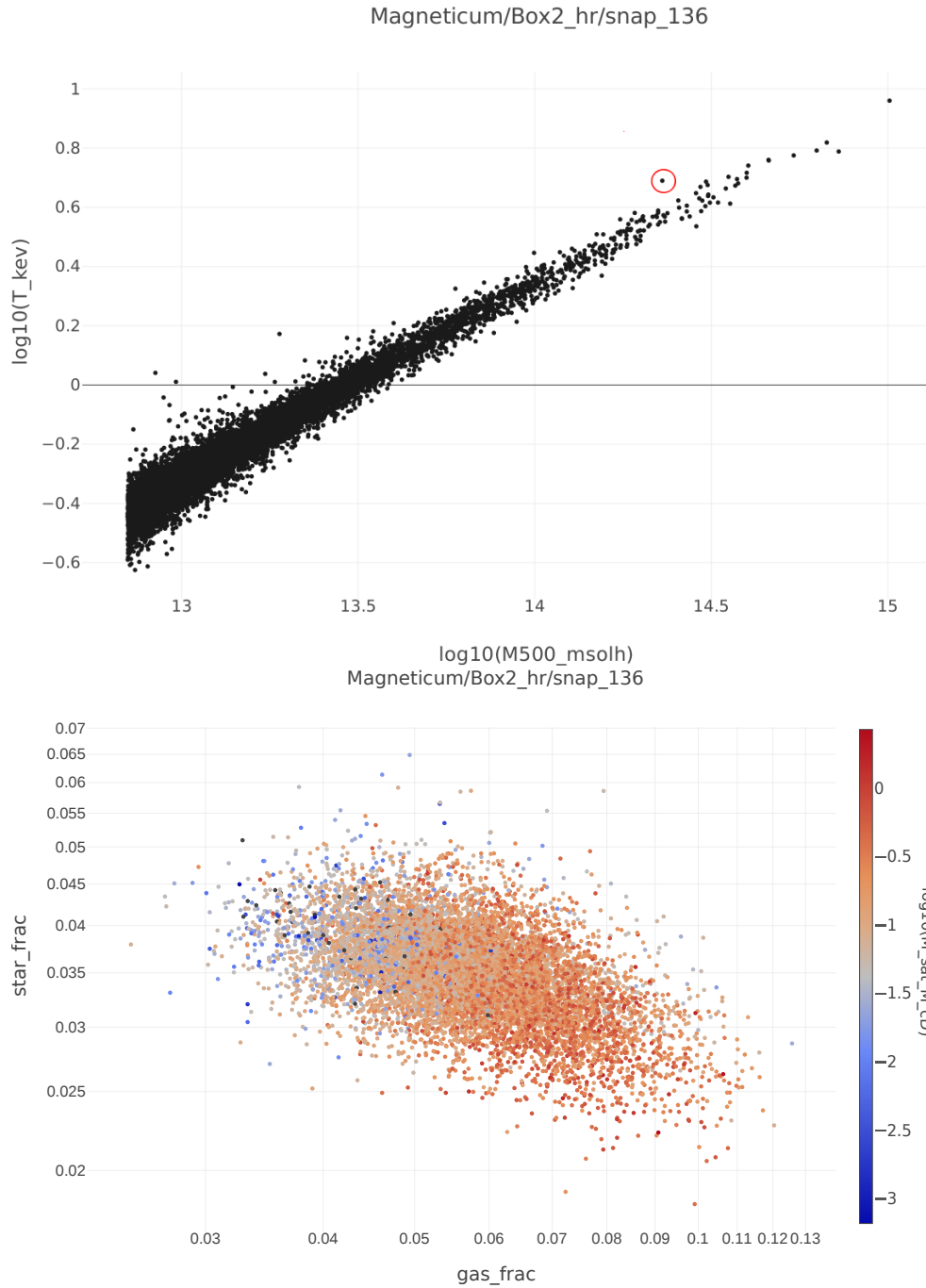


Figure 4.5: Mass-temperature correlation from the metadata of the *Magneticum/Box2/hr* simulation at $z = 0.14$ (left panel), where outliers can be identified by a red circle. The right panel shows the anti correlation of stellar and gas mass fraction, coloured by the stellar mass ratio of satellite galaxies to central galaxy, which often is used as indicator for the dynamical state.

the simulations, the web interface allows users to select objects in various ways through a graphical interface. This interface supports the visualisation of pre-computed 2D maps rendered with Javascript We use OPENLAYERS 3 JavaScript library. Users can navigate through different maps, as shown in figure 4.4 by scrolling and zooming. They can perform complex queries on simulation metadata based on object definitions (in our case based on SUBFIND) which are stored in a database. The CLUSTERFIND tool allows users to obtain all properties of clusters fulfilling the restriction criteria of the query mask. It is then possible to download these results as CSV-tables or to interactively visualise them. The user can thereby make scatter plots, 1D and 2D histograms on arbitrary expressions of the properties within the table. In order to support the selection of interesting cluster, the user can either select an object from the table or directly from the scatter plots by clicking on the corresponding data point. An example of a scatter plot between the mass and temperature is given in figure 4.5 where the user for example could select the cluster that clearly lies outside the mass-temperature relation.

Currently the web portal supports four services: CLUSTERINSPECT, SIMCUT, SMAC and PHOX, which are described in more detail in section 4.5.

The web interface can check the status of submitted jobs for the status of a submitted job and return results to the user.

Once the job is finished, the backend provides the web interface with a web link, where the user can download the results. The user is able to check the status of all jobs and to download results via the web interface.

Metadata from galaxies and galaxy clusters (identified with SUBFIND) are stored in a database on the web portal. Maps can be additionally overlaid. The data centre is able to host and handle large simulations, with order of 10^5 objects, each containing a number of galaxy members. Note that a large galaxy cluster in the simulation can have thousands of individual galaxies as associated data. This results in a huge number of rows in the database. Therefore, to achieve high performance, we used the so called Database Indexing. Database Indexing is a solution provided by databases for speeding up queries on read-only tables. In details, Database Indexing creates a tree structure over the field of a table so that for every indexed field there is a look-up table where data are ordered according to this field. This speeds up every query where data are selected on a specific range of values. We also use a small storage system directly on the web portal in order to store 2D maps of the simulations. The end user can therefore browse quickly through these vast simulation maps and visually seek for objects of interest.

4.3.3 Job Control Layer

The backend is based on Python and is activated when a new job is sent to the database or when the job scheduling system starts/finishes a job. New jobs are set up by preparing an unique job directory and template scripts and parameter files for the different services are used. The job is submitted to the queue of the computing cluster and executed on the cluster. Once the job is completed, the backend will collect the results and communicate a link for downloading the data to the web interface via the shared job database. The job

database is connected with both the web portal and the backend, as can be seen in figure 4.3. As mentioned before, for security reasons, the web interface can not directly send jobs to the computing cluster or access the raw simulation data.

4.3.4 Compute Layer and HPC storage

Large, hydrodynamic cosmological simulations nowadays are performed within large, collaborative efforts. While typically a large number of scientists contributes to the development of the underlying code and its auxiliary functionality needed to perform specific simulations, the actual execution of individual simulations is typically performed by a small number of directly involved scientists. It is practically impossible to grant all scientists of the collaboration (or beyond) direct access to the simulation data on the HPC facility. Therefore we follow a different approach. While the storage file system is typically assigned to the individual scientists within their HPC projects, they give reading permission of the raw simulation data to a single user of the super computing centre. This can be done, as in our case, even on a much smaller, dedicated computing cluster where HPC storage system is available on the computing nodes. The execution of such jobs will be typically done via a batch system. An additional, independent, local storage keeps the data products of the analysis tools. This local storage is represented by the box *Scientific Data Products* in figure 4.3. The data products of the services will be made available for the end user on the local storage, and can be shared with the scientific community without restrictions. Within this concept, neither the web interface nor the actual user ever has (or needs) any direct access to the raw simulation data.

4.3.5 Implementation

Currently, the full outcome of the two Magneticum simulations is stored on a Big Data storage attached to a HPC system (in our case, it is a project space at the Leibniz Rechenzentrum). Analysis jobs run on a separate, much smaller scale computing system. Specifically, we employ to this purpose the computing cluster at the Computational Center for Particle and Astrophysics (C²PAP) ¹.

The web interface is running on virtual machines hosted at LRZ ². The database of metadata properties (cluster and galaxy properties) runs on the same virtual machines of the web interface. Users log in and register by using their e-mail addresses. User registration, encryption and reset of passwords are handled by the Flask-Login python module ³.

Users must register and be approved by administrators of the data center to access its data. Users register to the web portal using their email address as username inside the web interface. The administrator can then grant *roles* to the individual users. Although at the moment all services are opened for all registered users, the implementation allows

¹<http://www.universe-cluster.de/c2pap/>

²<https://www.lrz.de/services/serverbetrieb/>

³<https://flask-login.readthedocs.io/en/latest/>

the administrator to make services only accessible for users with special roles. Moreover, there is the need to share data products with a wider community, where members do not necessarily have access to the system. When a job is created through the web interface by a registered user, a link containing a unique identifier of the job is delivered to the user. As mentioned above, the link can be shared, and its access does not require registration.

To allow a first exploration of the simulations data, there is the possibility to visually explore the simulations without being registered. However, the access to the meta data are limited and all services are disabled in the public browsing mode.

Jobs that are submitted through the web interface are all sent to the computing cluster.

Data products are stored on the additional FTP server which runs on the C²PAP computing cluster. Data products are guaranteed to be stored for 30 days. The data portal and its infra structure is will be available for a minimum of five years.

4.4 Exploring Simulations

Here we describe in detail how the web portal The web portal is a dynamic website and the interface is built using ANGULARJS v1.4.5, JQUERY v1.11.3, JQUERY UI v1.11.3 and BOOTSTRAP v3.3.4 allows users to explore the simulations and how objects of interest can be interactively found by performing complex queries using *Restrict* or plot the metadata quantities using *ClusterFind*.

4.4.1 Selecting Objects by browsing maps

The web interface allows users to explore cosmological structures within the simulations by panning and zooming through high resolution (256 mega pixel) images. Once a simulation is chosen, the output at various points of the cosmological history can be selected. Depending on the underlying simulation, typically between 10 and 40 different output times can be chosen. Generally, four different maps can be selected as the prime visualisation, as shown already in figure 4.4. The diffuse baryonic medium is either visualised colour coded by its X-ray emission (*ICM*) or by its pressure (*ComptonY*) using SMAC (Dolag et al., 2005). The stellar component is visualised by the density of stars and colour coded by the average age of the stellar population using SPLOTCH (Dolag et al., 2008b). Additionally, we computed a filtered visualisation of the ICM pressure (*Shocks*), where shocks and turbulence are visible. In figure 4.4 each of the quadrants shows one of the visuals mentioned before and demonstrates the different appearance of the massive galaxy cluster in the centre. The arc-let (depicted by the green curved line over the yellow background) appearing near the centre in the *Shocks* visual (lower right segment) resembles a shock wave which spans more than 1Mpc in size and indicates a merging cluster. All visualisations are based on 16k×16k pixel sized images which can be explored using the zoomify technique. In addition, the layer-spy can be activated to switch to a different visualisation view within a small region around the cursor, (see figure 4.6). This immediately gives a visual impression of the dynamics and composition of the diffuse gaseous and the stellar component within

Choose Simulation and snapshot

Modify filtering/queries on clusters

Set job parameters and run over the selected object

Choose which map/overlay to show

The screenshot displays the Cosmological Web Portal interface. At the top, there is a navigation bar with links for 'Cosmological Web Portal', 'Applications', 'Query', 'Jobs', 'Admin Area', 'Browse (public access)', 'Credits', and a user profile 'ragagnin@lrz.de' with a 'Logout' button. Below the navigation bar, there is a search and filter area. A green button labeled 'SMAC' is on the left. The main search area contains a dropdown menu with 'Magneticum/Box2_hr', a snapshot selector 'snap_132 (z=0.10)', and a '+' button. To the right, there are filters for 'Stars', 'Overlay ICM', and 'Cluster', with a 'Restrict' button. The main content area shows a dark field of stars with several green circles highlighting filtered objects. A large red and orange circular overlay is labeled 'Layer spy of ICM conten'. A popup window titled 'Cluster details' is open, showing the following data:

Cluster details	
M _{500c} :	1.0543e+15 [M _⊙ /h]
T _{mw} :	9.29729 [KeV]
L _x :	2.94798e+46 [erg/s]
f _{gas} :	12.5085 [%]
f _{star} :	2.84439 [%]
Ca _{hii} :	0.025115
Y _{500c} :	107.34941
M _{500c} /M _{CD} :	0.6368698

Layer spy of ICM conten

Filtered objects are in green circles

Selected cluster has additional information

Figure 4.6: Overview of a given snapshot of a simulation. The layer spy of ICM can be activated within a small region around the cursor.

Choose Simulation and snapshot

Table of properties of selected objects

Filter objects

The screenshot shows the CLUSTERFIND web interface. At the top is a navigation bar with links: Cosmological Web Portal, Applications, Query, Jobs, Admin Area, Browse (public access), Credits, ragagnin@lrz.de, and Logout. Below this is a filter bar containing a green 'PLOT' button and a search area with a dropdown menu set to 'Magneticum/Box2_hr', a snapshot dropdown set to 'snap_132 (z=0.10)', and zoom controls. The main content area is divided into three sections: 1. 'Select Clusters' with a slider for selecting 4741 out of 10324 clusters and sliders for various physical properties (M_{500c}, T_{500c}, L_x, f_{gas}, f_{stars}, C_{shitt}, Y_{500c}, M_{sat}/M_{cD}). 2. A table of object properties with columns: L_x_ergs, M500_msolh, M_sat_M_cD, T_kev, Y500c, cshitt, display_radius, gas_frac, id, mass, r500_kpch, snap_id, and sta. 3. 'Further restrictions based on sub-structures/member galaxies' with checkboxes for 'Compact groups' and 'Merging clusters'. Below the table is a 'Plot type' dropdown set to 'scatter', and input fields for X-axis (log10(M500_msolh)), Y-axis (log10(T_kev)), and Color (log10(M_sat_M_cD)).

Make plots based on expressions on object properties

Apply complex queries (compactness and merging properties) to objects

Figure 4.7: A screenshot of the service CLUSTERFIND. Here there is the selection panel for galaxy clusters and groups, where the objects can be filtered according to their properties.

our universe. It also allows users to instantaneously see various, physical features of the simulation. For example, galaxies in less dense environment appear more often in a bluish colour, indicating a young stellar component while galaxies in more dense environment often appear in yellow and red colours, indicating an older stellar component. This reflects that the underlying simulations correctly reproduce the so called morphological density relation of galaxies, which is one of the most prominent, observed imprints of the large scale structure of galaxy formation. Additionally, the position of galaxy clusters and groups can be overlaid as green circles and an information panel on the cluster properties is visible as soon as a galaxy cluster is selected.

4.4.2 Composing Queries

To select galaxy clusters and groups, the *Restrict* window can be used to perform complex queries among the metadata of clusters and groups, as shown in Figure 4.7. Every time a value is modified by the sliders, a new database query is performed and the selected objects are shown as green circles in the web interface.

Restrict sample by Value

The upper row of sliders allows users to

choose minimum and maximum value for various global quantities, like mass (M_{500c}), temperature (T), bolometric X-ray luminosity (L_x), gas and stellar fractions (f_{gas} and f_{stars} respectively). The results can be displayed or downloaded as ASCII tables. This allows the user to perform simple analyses, for example plotting scaling relations like the well known mass-temperature relation (shown in the left panel of figure 4.5). For example, a prominent outlier in the figure can then be selected in the web interface by restricting $M_{500c} > 14.1$ and $T > 3.05$. The resulting cluster will then be the only marked cluster. By a closer inspection, it can be recognised as a major merger system, already visible in both visualisations, as double emission peak in the diffuse medium as well as second, very large in-falling group of galaxies in the stellar component.

Select by Dynamical State

There are two classical measures (Rasia et al., 2013; Jones et al., 2003; Cui et al., 2011) of the dynamical state of galaxy clusters and groups. One is the ratio of the total stellar mass in satellite galaxies with respect to the mass in the central galaxy. The other is the so called centre shift, which is a measure of the displacement of the “centre of emission” of the diffuse baryons compared to the position of the potential minimum and is typically measured in units of the size (e.g. R_{500c}) of the system. These two measurements can be used to select relaxed and un-relaxed systems. For example, the outlier in the above example shows a very large centre shift of $c_{shift} = 0.13$ and a large stellar satellite fraction of $M_{sat}/M_{cD} = 0.7$, clearly classifying it as major merger. Usage of the metadata tables allows users also to verify other correlations, like the anti correlation between stellar mass

and gas mass fraction as shown in the right panel of figure 4.5. In that case we used one of the merger indicators to colour code the objects across the anti-correlation.

Finding different Classes

As seen in the examples before, the different filters allow users to select objects with different global properties or with different dynamical states. However, such filters can also be used to select different object classes.

Fossil groups have typically a very large, dominant central galaxy and only a very small amount of satellite galaxies. Such objects can be selected via the M_{sat}/M_{cD} parameter, which is the mass ratio between the sum of all satellite galaxies and the central galaxy. In this case, choosing a small value will select fossil groups.

Compact groups are typically characterised by several galaxies of similar mass within a small spacial region. They can be selected by setting an integer N , from 1 to 4, a maximum distance R and the minimum value of the logarithm of the stellar mass ratio between the central galaxy and the N th galaxy (i.e. $\log_{10}(M_{cD}/M_{Nth})$). In figure 4.7) a query with $N = 4$, $R = 100kpc$ and $\log_{10}(M_{cD}/M_{Nth}) < 1$ returns 160 compact group candidates out of 7428 objects fulfilling this criteria at $z = 0.67$ within our example of *Magneticum/Box2/hr*.

Merging clusters can be also selected by marking the corresponding check-box in the *Restrict* window. This query allows to select clusters or groups where at least one sub-structure fulfils the given criteria. The user can select the range of stellar and gas mass content, the relative, tangential or radial velocity and a distance of the sub structure to the centre. To find bullet cluster like systems, one would select a large stellar and gas mass, a large, outgoing velocity (positive v_r) and a large distance from the centre.

4.5 The Services

The results of the different services typically come with different number of files, depending on the configuration of the workflow specified by the user. Therefore, the results are made available in form of a tar ball, which the user can download. Additionally, the web interface provides small iPython examples for every available result. They can be used to have a quick look at the obtained results.

4.5.1 ClusterInspect

CLUSTERINSPECT allows users to browse the member galaxies of the selected cluster. As in CLUSTERFIND, it is then possible to make plots on the properties of the member galaxies. Within the CLUSTERFIND and CLUSTERINSPECT service, interactive plots are made using the JavaScript version of the library PLOTLY.JS v1.9.0.

Block	Type	Size	Blocks
POS	FLOATN	3	0,1,4,5
VEL	FLOATN	3	0,1,4,5
ID	LLONG	1	0,1,4,5
MASS	FLOAT	1	0,1,4,5
U	FLOAT	1	0
RHO	FLOAT	1	0
HSML	FLOAT	1	0
SFR	FLOAT	1	0
AGE	FLOAT	1	4,5
BHMA	FLOAT	1	5
BHMD	FLOAT	1	5
BHPC	LONG	1	5
iM	FLOAT	1	4
Zs	FLOATN	11	0,4
CLDX	FLOAT	1	0
TEMP	FLOAT	1	0

Table 4.1: Data block names, type and sizes currently produced by SIMCUT.

4.5.2 SimCut

The SIMCUT service allows users to create artificial GADGET snapshot files⁴ The file produced by the SIMCUT service contains a list of particles within a region centred on the selected galaxy cluster or the group. For every quantity that is stored (e.g. position, velocity, mass), these files have a so called “block”. Every block has a code name (respectively, POS, VEL) composed of a string of maximum of 4 characters and an array of the size of the number of particles. In addition, the snapshot contains an extension to the P-GADGET2 output files the so called *INFO* block. This block contains information on data types and dimensions of the different values stored in the snapshot, as shown in table 4.1.

4.5.3 SMAC

The SMAC service allows users to construct maps from the simulations using the map making program SMAC (Dolag et al., 2005), which allows to integrate various physical and observational quantities throughout the simulation. Once a galaxy cluster or group is chosen, the service allows the user to select various different map making options:

- *Baryonic density map* [g/cm^2]

⁴They are produced following the same format of P-GADGET2. For more details about the Gadget formats and parameter file configurations, see: <https://wwwmpa.mpa-garching.mpg.de/gadget/users-guide.pdf>.

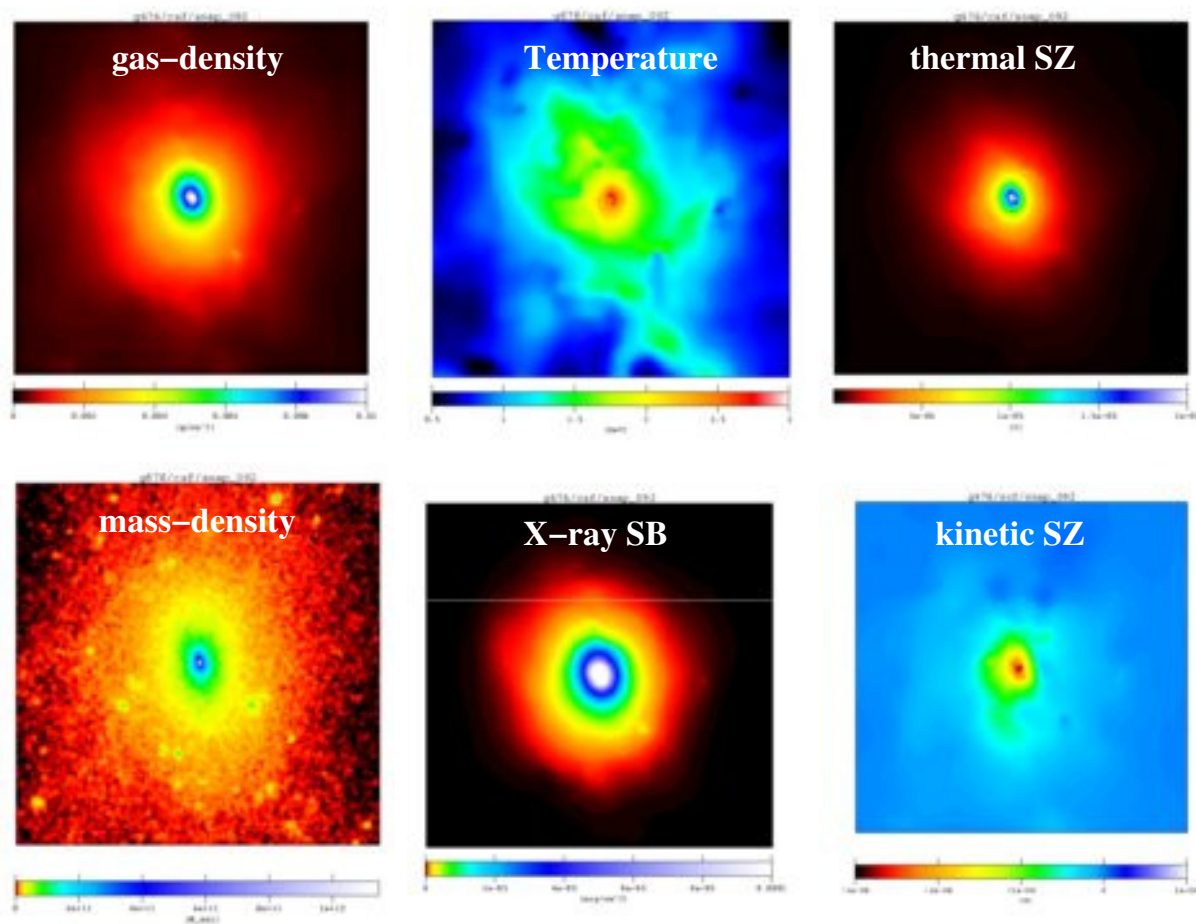


Figure 4.8: A sample of maps currently available within the the SMAC service for a given selected cluster.

- *Total mass density map* [M_{\odot}/cm^2]
- *Mass-weighted Temperature* [keV]
- *Bolometric, X-ray surface brightness* [erg/cm^2]
- *thermal SZ effect* [*Compton Y-parameter*]
- *kinetic SZ effect* [*Compton w-parameter*]

The size of the image as well as the integration depth along the z-axis can be chosen. The data are returned as standard FITS <http://fits.gsfc.nasa.gov/> files containing an image. Figure 4.8 shows some example maps obtained by this service.

4.5.4 PHOX

PHOX is a virtual X-ray telescope that generates X-ray synthetic observations from hydrodynamical, numerical simulation outputs (Biffi et al., 2012, 2013). As a first step, the simulation box is converted into a virtual box of photon packages, generated by the sampling of the X-ray emission spectra calculated for each gas element as well as for each AGN within the simulation.

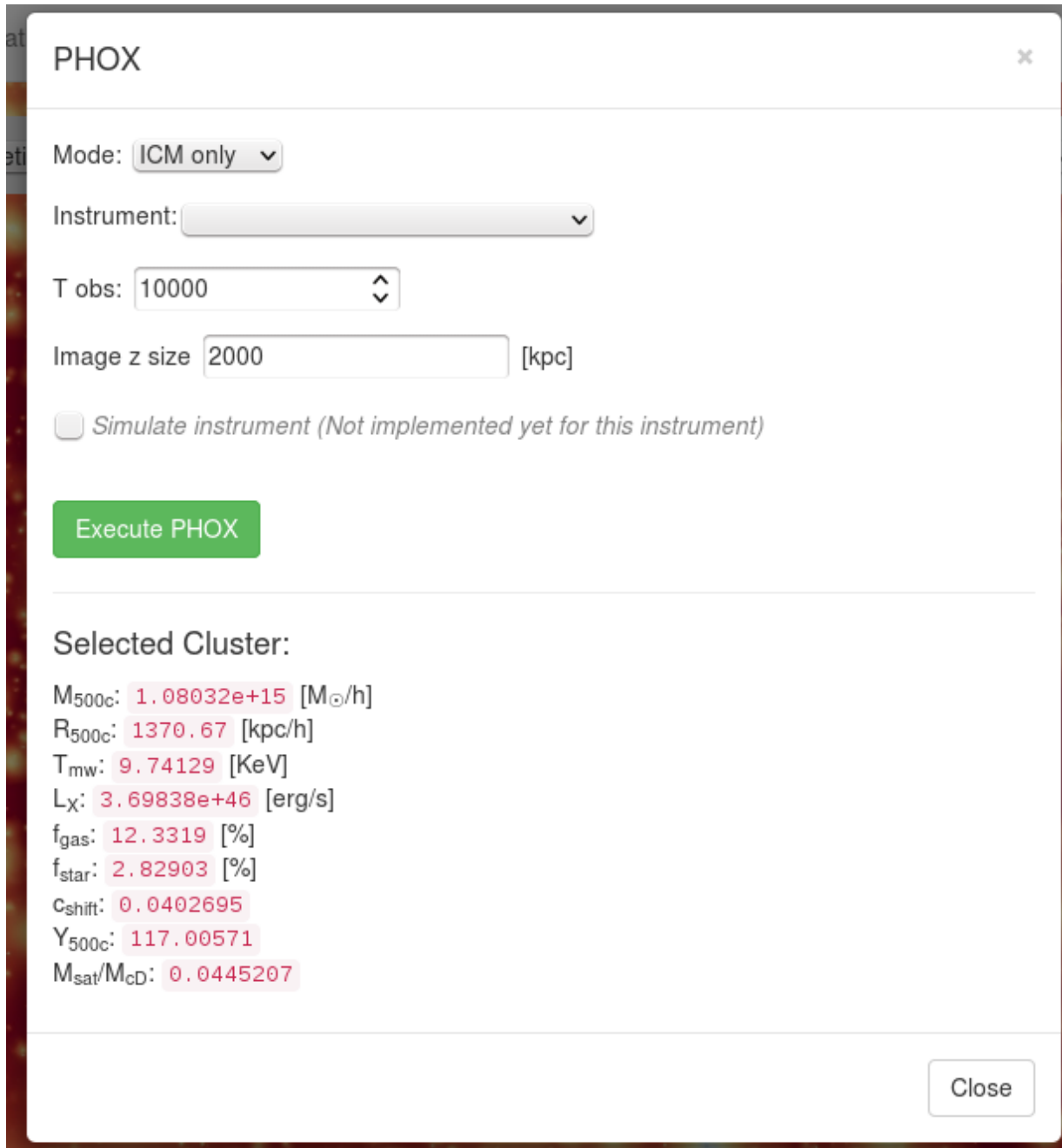
The ICM emission

In order to compute the X-ray emission spectrum of each gas element, the APEC model (Smith et al., 2001) for the emission of a collisionally-ionized, chemically enriched plasma implemented within the external, publicly available package XSPEC⁵ (Arnaud, 1996) is utilised. The properties (namely density, temperature and metallicity, and even chemical composition) of the gas elements are directly used as input for this model. Here, no external libraries of model spectra are constructed, but the emission spectrum generated by XSPEC is processed immediately, gas element per gas element, which allows users to vary the amount of parameters involved to describe the spectral emission without enormously increasing the computational effort and memory requirements. Each spectrum is then populated with a predicted number of photons (according to fiducial, general values for collecting area and exposure time). The photons collected from all the X-ray-emitting gas elements are eventually stored in terms of photon packages, each of them being characterised by the position and velocity of the original emitting element and by the array of energies of the associated photon sample.

The AGN emission

For the AGN component (see Biffi et al. 2016, in prep.), the procedure followed to convert the simulation box into a box of ideal photons emitted by all the AGN sources is similar

⁵<https://heasarc.gsfc.nasa.gov/xanadu/xspec/>



The screenshot shows a web interface for the PHOX service. At the top, the title "PHOX" is displayed with a close button (x) in the top right corner. Below the title, there are several configuration options:

- Mode:** A dropdown menu currently set to "ICM only".
- Instrument:** A dropdown menu that is currently empty.
- T obs:** A numeric input field set to "10000" with a vertical double-headed arrow icon to its right.
- Image z size:** A numeric input field set to "2000" followed by "[kpc]".
- Simulate instrument (Not implemented yet for this instrument)*

A green button labeled "Execute PHOX" is positioned below these settings. A horizontal line separates the settings from the "Selected Cluster:" section, which lists various physical properties of the cluster:

- M_{500c} : 1.08032e+15 [M_{\odot}/h]
- R_{500c} : 1370.67 [kpc/h]
- T_{mw} : 9.74129 [KeV]
- L_X : 3.69838e+46 [erg/s]
- f_{gas} : 12.3319 [%]
- f_{star} : 2.82903 [%]
- C_{shift} : 0.0402695
- Y_{500c} : 117.00571
- M_{sat}/M_{cD} : 0.0445207

A "Close" button is located in the bottom right corner of the panel.

Figure 4.9: Selection panel for executing PHOX, where details on the virtual observation and the instrument properties can be selected

to the one used for the gas, except for the spectral model utilised. Namely, we model the AGN emission with an intrinsically absorbed power law, constrained as follows. We convert the bolometric luminosities L_{bol} of the AGN into rest-frame SXR and HXR luminosities assuming the bolometric corrections proposed by Marconi et al. (2004), which can be approximated by the following third-degree polynomial fits

$$\begin{aligned}\log(L_{hxr}/L_{bol}) &= -1.54 - 0.24\mathcal{L} - 0.012\mathcal{L}^2 + 0.0015\mathcal{L}^3 \\ \log(L_{sxr}/L_{bol}) &= -1.65 - 0.22\mathcal{L} - 0.012\mathcal{L}^2 + 0.0015\mathcal{L}^3\end{aligned}$$

with $\mathcal{L} = \log(L_{bol}/L_{\odot}) - 12$, derived for the range $8.5 < \log(L_{bol}/L_{\odot}) < 13$. (see fig. 3(b) in Marconi et al. (2004)). Here, we mimic the observed scatter in these relations by adding a random scatter of 0.1 to the SXR and HXR luminosities, in logarithmic scale. Then we construct an intrinsic redshifted power law spectrum

$$A(E) = [K(1+z)][E(1+z)]^{-\alpha} \left(\frac{1}{1+z} \right), \quad (4.1)$$

where K is the normalisation constant, α is the photon index and z is the redshift of the source. For every AGN in the simulation, the two parameters K and α can be constrained by integrating the spectrum from the expected values of L_{SXR} and L_{HXR} . In this approach, the obtained distribution of photon indexes α reasonably reflects the observed Gaussian distribution in the range 1.2–2.8, which peaks around ~ 2 (e.g. Zdziarski et al., 2000)

Many observational works suggest that AGN sources also show evidences for the presence of obscuring material (i.e. the torus) in the vicinity of the central BH, which leads to the partial absorption of the emitted radiation. In order to account for this phenomenon, we assign to each AGN in the simulation a value of the obscurer column-density (N_h) by assuming the estimated column-density distribution presented in the study by Buchner et al. (2014) (see top-left panel of fig. 10, in their paper) and based on a sample of 350 X-ray AGN in the 4Ms Chandra Deep Field South. Within the PHOX code, we include this in the construction procedure of the X-ray emission model from AGN sources. The resulting absorbed SXR and HXR luminosity functions are found to be in overall good agreement with those observed (see Biffi et al. 2016, in prep).

Performing an X-ray Observation

With PHOX, the field of the selected photon packages (either only “ICM”/“AGN” or both can be selected) is further processed by taking into account the geometry of the mock study and the *idealised* instrument characteristics. In particular, given the ideal cube of virtual photon packages associated to the simulation output, it is here possible to select the size of the sub-region of interest, centred on the selected galaxy cluster or group, and to choose an instrument, which defines a field of view and a nominal effective area of the X-ray telescope, as shown in Figure 4.9. This also defines the line-of-sight (l.o.s.) direction and accounts for the according Doppler shift in the photon energies due to the peculiar motion of the original emitting particles. The virtual observation then returns the photon list expected for the idealised instrument chosen (assuming the nominal effective area over all energies,

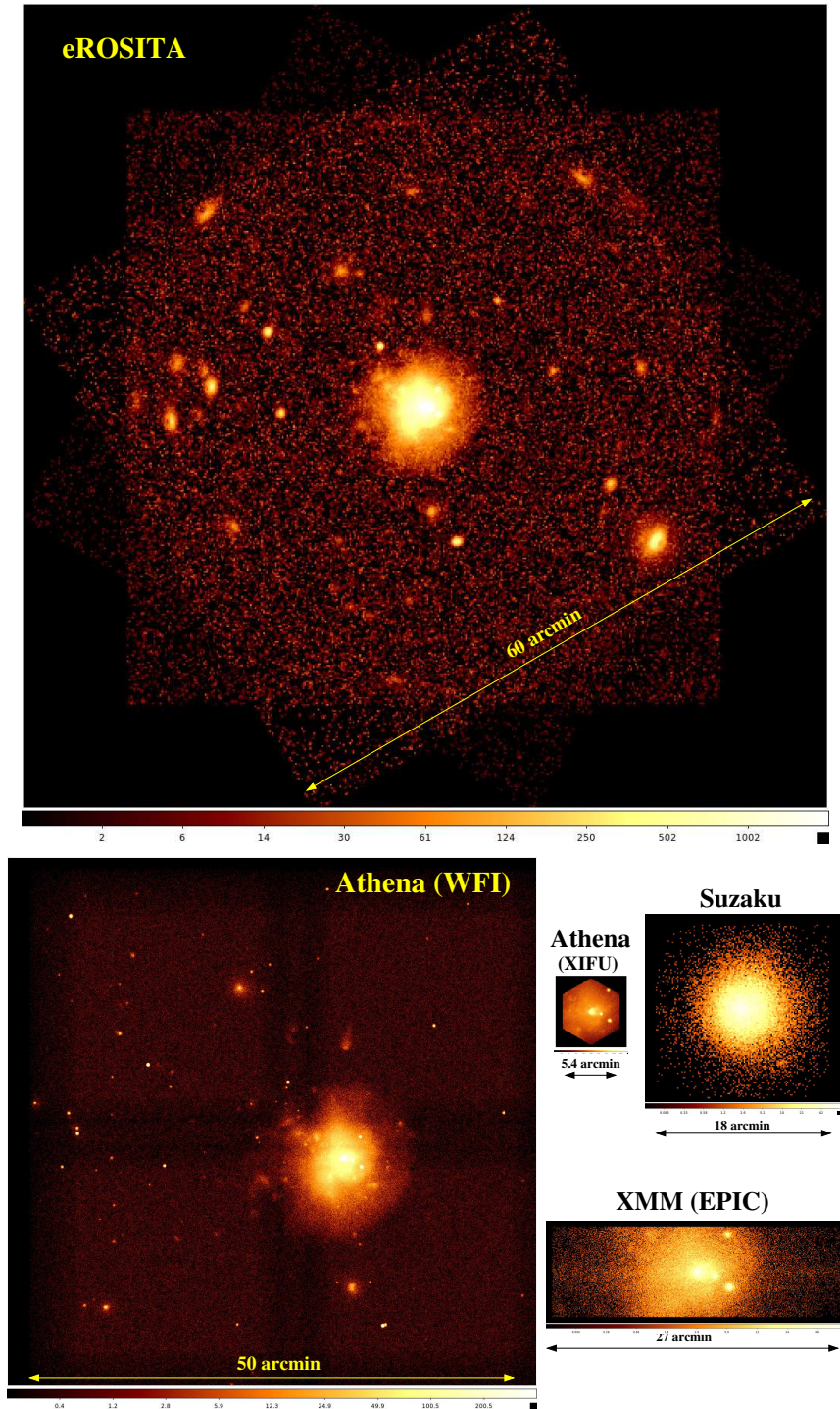


Figure 4.10: Observation of the X-ray emission obtained from the combined ICM and AGN contribution, centred on the most massive cluster at $z = 0.3$, performed with different, current and future X-ray instruments, including the actual instrument simulations.

no beam smearing of the position and keeping the exact energy) and returning the data in the form of a FITS file in SIMPUT <http://hea-www.harvard.edu/heasarc/formats/simput-1.1.0.pdf>. format, which allows users to directly utilise this file for more sophisticated instrument simulations. The time of the virtual observation is restricted, and a limit of $A_{instrument} \times T_{obs} < 10^9$ is enforced. This process typically takes only few seconds to be performed.

Including the mock X-ray observation

We also installed some publicly available instrument simulators, which can be added by clicking on the *Simulate Instrument* check-box, for some of the chosen instruments.

For *XMM(EPIC)*, *eROSITA* and *Athena(XIFU/WFI)* we are using SIXTE ⁶, whereas for *SUZAKU(XIS)* we are using XISSIM, which is part of the HEADAS package (Ishisaki et al., 2007). For *XraySurveyor(HDXI/XCAL)* and *Hitomi (SXS/SXI/HXI)* we are using SIMX ⁷ while for *Chandra (ACIS-S/I)* instruments we are using MARX (Davis et al., 2012). Depending on the instrument and time chosen for the observation, this process can take up to several minutes. It returns an event file which can be then analysed with standard X-ray analysis tools.

Figure 4.10 shows the result for observations centred on the same massive galaxy cluster at $z = 0.3$ while choosing different instruments. The exposure time is set to 100ks (reduced to 50ks for *Athena*) and a slab of 100Mpc along the line of sight is used. Both, the ICM as well as the AGN component are taken into account in this example. For the *eROSITA* case, a simulation of all 7 CCD chips is preformed, delivering 7 event files, which are additionally combined to one single event file, which leads to the appearance of edges from the rotated frames in the combined image as seen in the map. For the *Athena(WFI)* simulation, the 4 individual chips are simulated, as still visible by the gaps in the maps. However, as for real observations, the telescope is made to dither during the integration time to smears out the gaps between the chips, which is controlled by a realistic attitude file which defines the pointing of the instrument.

In 4.10 we show configuration files for SIXTE, XISSIM, SIMX and MARX used for each instrument we included.

4.6 Conclusions

In this work we present a data center based on a multi-layer infrastructure for large cosmological hydro-dynamical simulations. It will give a wide scientific community the possibility to perform analysis tools on data from several large simulations.

The increasing amount of upcoming astronomical surveys makes it necessary to increase resolution and volume of cosmological simulations to interpret the results from such

⁶<http://www.sternwarte.uni-erlangen.de/research/sixte>.

⁷<http://constellation.gsfc.nasa.gov/science/simulatorInstall.html>

observational campaigns. These simulations may include different matter contents and object types (gas, dark matter, stars, black holes) and various physical processes that shapes the formation of objects and structures in our universe. All those factors dramatically increase the output size of current simulations.

As a result, the storage and sharing of the largest simulations that are currently available represents a real challenge.

For example, the size of an individual snapshot of the *Box0/mr* simulation of the *Magneticum* project (which follows almost 2×10^{11} particles), is more than 20TB.

Although we currently only store results from the *Magneticum* project, the data center infrastructure is capable of including other simulations as well.

This data center allows users to configure workflows to run individual analysis tools for specific objects of a given snapshot on the raw simulation data. A web interface helps the user to build compounded queries on metadata of the different simulations, which in our case are galaxies and galaxy clusters as obtained by SUBFIND. This allows users to select a target object in the different simulations with the desired properties. Those queries can be restrictions to global properties like mass or other observables, as well as complex queries on various internal aspects. This gives the possibility to select different general classes of objects (like merging clusters, compact groups or fossil groups). Our initial version of the data center provides so far four services: CLUSTERINSPECT plots and shows data of member galaxies of a given galaxy cluster, SIMCUT gives the possibility of downloading the raw simulation data of all particles belonging to the selected object, SMAC creates 2D maps of different physical or observational quantities, PHOX generates virtual X-ray observations.

A number of considerations about accessibility, security and performance led us to a multi-layer infrastructure: the web portal, the job control layer, a computing cluster and the HPC storage system. The outer layer lets people perform highly compounded and elaborated queries on SUBFIND results; the computing cluster runs the chosen job, reading the raw simulation data directly via access to the HPC storage system and sends the results back to the web portal.

Concerning the cluster selection, there is the possibility of interlocking a number of different sub-queries. Some very basic queries are filtering by mass, temperature, bolometric X-ray luminosity, fraction of stars and gas masses. The user can also select an object by browsing 2D maps of the simulation, or by clicking objects in scatter plots of SUBFIND data.

It is possible to run queries for the selection of clusters in different dynamical states. For instance depending on their satellite fraction and displacements between baryon and potential centre. Other sub-queries allow for the selection of clusters with different degree of compactness by choosing the ratio between masses of the central galaxy and the n-th satellite galaxy within a given distance.

Services themselves come with different parameters that the user is free to choose. SMAC can project onto three different axes and map different matter properties (i.e. density, temperature, X-ray temperature); PHOX can simulate a number of different instruments (i.e. XMM, Suzaku, Chandra, eROSITA, Athena, XraySurveyor, Hitomi) and compute

X-ray emission due to AGN or ICM, separately or together.

Services will be extended in the future to support additional analyses. Possibility to browse observationally motivated realisations of light-cones is foreseen. Finally, the data center will permanently grow in size and amount of simulation data which will be made available for the general scientific community. Although some modifications on the different components would be needed to adapt the concept to other infrastructures and simulations, we are willing to provide the underlying source code on request and to give advice for adapting the package to other institutions. The current infrastructure for the data center is secured for five years.

4.7 Credits

The results obtained via the data center are free to use. Please acknowledge the *Magneticum* Project <http://www.magneticum.org> (Dolag et al. 2016, in prep) and the *Cosmological Web Portal* (this work, Ragagnin et al. 2016). Please also give the references for the individual simulations used (Box2/hr: Hirschmann et al. (2014), Box0/mr: Bocquet et al. (2016)). If results produced with PHOX are used, please cite Biffi et al. (2012), and in case of SMAC please cite Dolag et al. (2005). In the case that results of the X-ray instrument simulators are used, please refer to SIXTE⁸, XISSIM (Ishisaki et al., 2007), SIMX⁹ and MARX (Davis et al., 2012), depending on the simulator used for the simulation.

Acknowledgements

We thank G. Lemson for various discussions and his contribution to various VIO prototypes, which largely inspired this work. We thank T. Dauser, G. Lamer, N. Clerc, A. Rau and E. Pointecouteau for helping with the X-ray data and especially with the instrument simulators, which can be additionally involved within the “PHOX” service. AR thanks Nathan James Turner for carefully checking the grammar of this manuscript. We also thank Ben Hoyle for carefully reading the manuscript. This work has been supported by the DFG Cluster of Excellence “Origin and Structure of the Universe” (www.universe-cluster.de) and the SFB-Tansregio TR33 The Dark Universe. We are especially grateful for the support by the Computational Center for Particle and Astrophysics (C²PAP). Computations have been performed at the Leibniz Supercomputing Centre (LRZ) with CPU time assigned to the Project “pr83li” and “pr86re”. The project received technical support from LRZ as well. Information on the *Magneticum Pathfinder* project is available at

<http://www.magneticum.org>.

⁸<http://www.sternwarte.uni-erlangen.de/research/sixte>

⁹<http://constellation.gsfc.nasa.gov/science/simulatorInstall.html>

4.8 Appendix A. Indexing Files

The following section corresponds to Appendix A published in Ragagnin et al. (2017).

Below we explain how to build the auxiliary data files in order to add third-party simulations to the data center. Figure 4.11 shows a sketch of how the indexing scheme works for the multi-file and multi-component output of our SPH simulations. In such cosmological SPH simulations, the data to be stored is quite complex. The particle data are distributed among many thousands of CPUs. Each CPU holds various particle types, representing different components of the simulated system. For these particles, different properties have to be stored. Some (like *temperature*) are specific to individual particle species (like *gas* particles in this case). Other properties (like *position*) are necessary for all components. To optimise the I/O and to avoid bottlenecks, only a subset of CPUs write in parallel. Data from the individual CPUs are stored in individual sub-files, as illustrated in the left part of figure 4.11. To optimise the access to individual data, files are structured with additional data regarding the content and length of individual data blocks.

We implemented an algorithm which sorts the particles of all CPUs among a coarse-grained space filling curve before writing them. In addition it produces an auxiliary file which allows identification of the sub-data volume elements of any stored property among all particle species associated with each element of the space filling curve. This allows the user to effectively collect all data associated with a given volume in space introducing minimal overhead. In detail, the process works as follows:

- The particles will be ordered along a space filling curve (in our case, a Peano Hilbert curve) with a defined graininess (i.e. *pixel* size) Note that in contrast to the standard domain decomposition in Gadget which is using a 64 bit long key, we are using a shorter, 32 bit long key which strongly reduces the later reading overhead among all CPUs. The space-filling-curve ordering will be used to decompose the spatial region of the simulation among the different CPUs so that all particles within an individual *pixel* are associated to a CPU. During this process, the amount of particles of different species falling within each *pixel* is stored as well.
- The particle data of the various CPUs are written into the sub-files in the same order as above. Therefore particles belonging to the same *pixel* are written in a consecutive order into the files.
- An additional auxiliary file stores the information needed to re-construct the position of each sub-data-block for particles located in the a specific *pixel*. It contains the offsets between different particles species, the list of *pixels* present in the file, the number of particles in the related *pixel* and the offset of each *pixel* within each particle species.
- Finally a super index file is created, which contains the compressed information of *pixel* indexes and the files where they are contained. This super index is later used to recover which sub-files have to be accessed for the reading process.

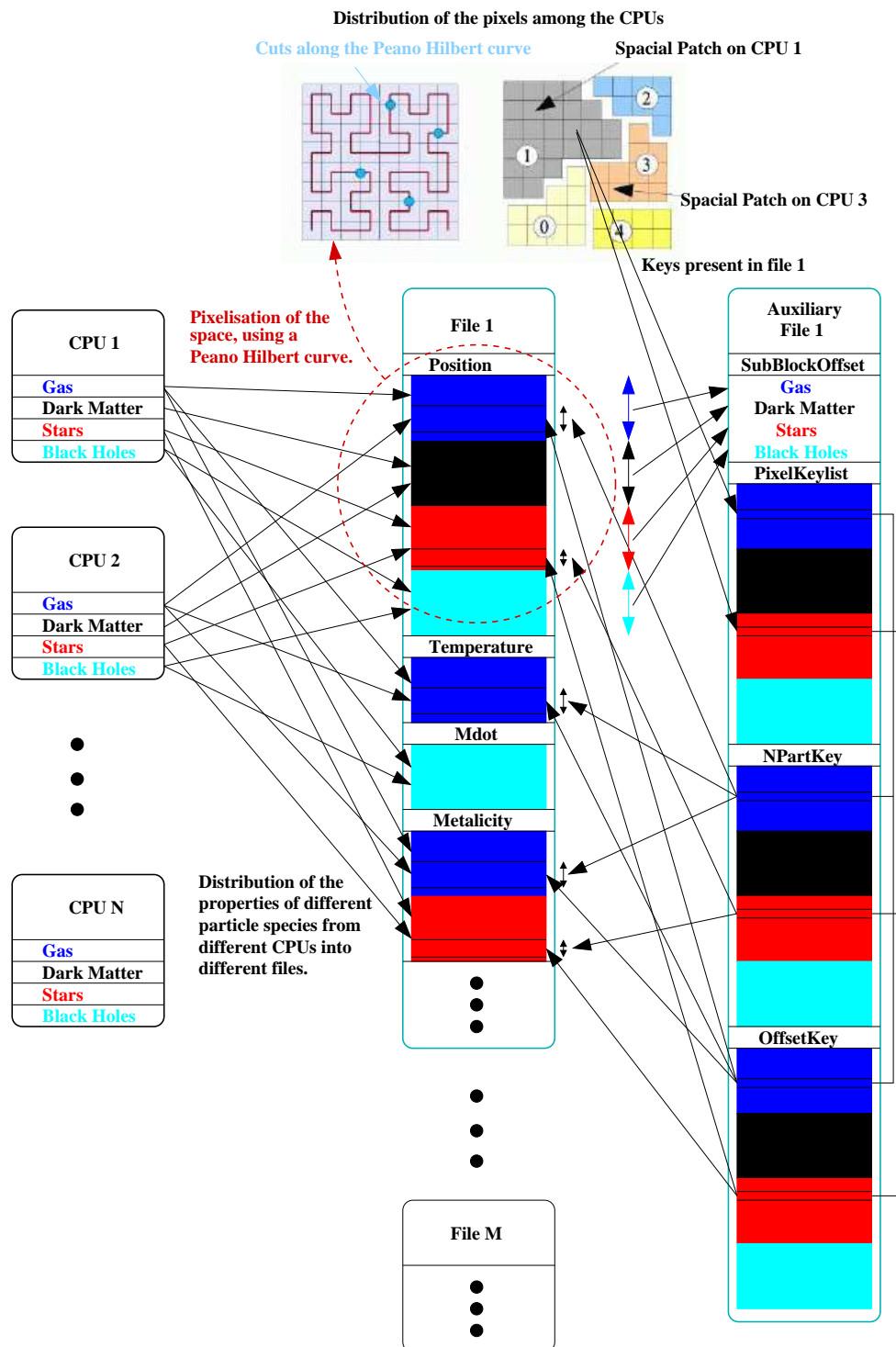


Figure 4.11: This diagram illustrates how data of the different properties, attached to the different particle types are collected from several CPUs and stored into different files (left part). Once the positions of the particles are associated (and sorted) according to their associated elements along a space filling curve, the index of such pixels, together with the number of particles associated with each pixel and the according offset in the files can be stored as auxiliary data.

The additional auxiliary files are produced for each output of the simulations and allow the user to investigate very efficiently individual objects within such cosmological simulation through post-processing tasks. The post-processing software has been adapted to this new output and for reading sub-volumes of the simulation. The following strategy is applied for a given sub-volume of the simulation which should be analysed (e.g. given by a galaxy cluster position and its size):

- First, a list of *pixels* (i.e. the elements of the space filling curve which falls within the region of the space of interest) is prepared.
- Then the super index file is read and a list of the output files which hold the individual subsets of the *pixel* index list is produced.
- For each sub-file in that list, first the *pixel* list is read from the auxiliary file and then it is compared with the *pixels* needed for the current task. For this subset of pixel, the additional information is read from the auxiliary file.
- Now the post-processing can reconstruct the position of the sub-blocks which have to be read from the individual blocks within the data file. This sub-blocks can then be read directly without loading the full data block. As further optimisation, consecutive sub-blocks within individual files, are joined to larger sub-blocks to prevent unnecessary fragmentation of the reading process.

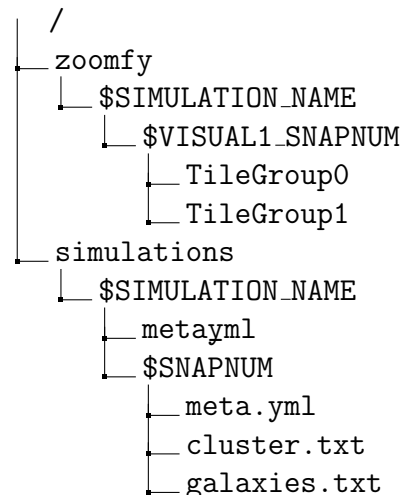
4.9 Appendix B. Preparing Simulations for the data center

The following section corresponds to Appendix B published in Ragagnin et al. (2017).

In this appendix we describe how to add simulations to the data centre. To this scope, several pre-computed data products, meta data and information have to be provided, as we briefly describe in the following.

4.9.1 Describing the Simulations

Every simulation consists of one or more time instances (snapshots), where each one should consist of a set of pre-computed 2D maps in the *zoomify*¹⁰ format, a list of galaxy clusters and their galaxy members. Metadata regarding galaxy cluster properties and member properties are stored in *yaml*¹¹ files. Metadata and maps are stored in files and folder in a way that for every simulation there is a folder for its 2D maps (*zoomify* folder) and one for the metadata (*simulations* folder). Here follows a sample of the folder structure:



The *zoomify* files are stored on the local disk space of the web portal and contain the images which can be browsed by the web interface. Table 4.9.1 shows a list of parameters used to describe a simulation.

Every time instance of a simulation, which is stored in the data center and made available via the web interface, contains its own sub directory which holds the prepared metadata for clusters and galaxies. They contain the information as described in the previous section. In addition, it also contains a general *yaml* metadata file with snapshot details. Table 4.2 contains the parameters that typically describe a snapshot.

Such information and definitions can be extended easily to any other cosmological simulation and will allow the final user to flexibly add new simulations to the system. Also, services can be readily added or disabled for individual simulations.

4.10 Appendix C. Instrument Configurations

The following section corresponds to Appendix C published in Ragagnin et al. (2017).

In this Appendix we report example configuration files for XISSIM, SIXTE, SIM and MARX, used for each instrument currently included within the PHOX service. Note that the exposure time will be replaced by the value chosen in the web interface.

Suzaku(XIS):

¹⁰see <http://www.zoomify.com/free.htm>

¹¹see <http://yaml.org/>

Property	Description
name	Simulation name
code	Description of the Code used
size_mph	Mpc/h
n_particles	number of initial particles
nfiles	number of files the snapshot
mdm_msun	mass dark matter particles [Msun/h]
mgas_msun	mass gas particle [Msun/h]
epsilon_dm	softening of dark matter particles
epsilon_gas	softening of gas particles
epsilon_stars	softening of stars particles
f_bary	cosmological baryon fraction
omega_m	total matter content
omega_lambda	Cosmological constant
hubble	Hubble constant
sigma_8	Normalisation of matter power-spectrum
n_primordial	Primordial power spectrum index
path_exec	file system path to simulation

Property	Description
name	snapshot name
mask_path	not used
width_pixel	total number of pixels
height_pixel	total number of pixels
redshift	Redshift
angular_diameter	Angular diameter distance
phox_max	Maximum GRASP (for Phox)
phox_avail	Phox service available
smac_avail	Smac service available
simcut_avail	SimCut service available
simcut_plot	SimPlot service available

Table 4.2: Properties used to describe a snapshot within a simulation.

quantity	unit	name
ID	[integer]	id
x	[kpc/h]	x
y	[kpc/h]	y
z	[kpc/h]	z
M_{500c}	[M_{\odot}/h]	M500_msolh
R_{500c}	[kpc/h]	r500_kpch
f_{gas}	[fraction]	gas_frac
f_{stars}	[fraction]	star_frac
L_x	[10^{44} erg/s]	Lx_ergs
Y_{500c}	[$\Delta T/T$]	Y500c
M_{sat}/M_{cD}	[fraction]	M_sat_M_cD
c_{center}	[R_{500c}]	c_shift

Table 4.3: Metadata for galaxy clusters and groups, from top to bottom: cluster ID, position (in comoving coordinates), mass and radius in respect to 500 times the critical density, gas and star fraction, bolometric X-ray luminosity, stellar mass fraction of satellite galaxies to central galaxy, and weighted centre shift between X-ray emission and mass distribution within R_{500c} .

quantity	unit	name
ID	[integer]	id
x	[kpc/h]	x
y	[kpc/h]	y
z	[kpc/h]	z
M_{star}	[M_{\odot}/h]	M_solh
M_{gas}	[M_{\odot}/h]	M_gas
sfr	[$M_{\odot}/year$]	sfr_msoly
host ID	[integer]	cluster_id
radial distance	[kpc/h]	dist
peculiar v_x	[km/s]	vx
peculiar v_y	[km/s]	vy
peculiar v_z	[km/s]	vz
velocity	[km/s]	dv
radial velocity	[km/s]	dr
tangential velocity	[km/s]	dt
mass ratio to cD	[km/s]	log10_mCD_m

Table 4.4: Metadata for galaxies, from top to bottom: galaxy ID, position (in comoving coordinates), mass, star-formation rate, ID of cluster or group where the galaxy belongs to, distance to the center of the cluster or group it belongs to, different velocity components relative to the cluster center (as peculiar velocities) and the ratio of the stellar mass in satellite galaxies in respect to the central one.


```
xissim \
  clobber=yes \
  instrume="XIS1" \
  ea1=0 ea2=0 ea3=90 \
  infile1="phlist_xissim.fits" \
  infile2=none \
  date_obs="2009-09-01T00:00:00" \
  xis_rmffile=suzaku/xis/cpf/ae_BI_ao4_20090901.rmf" \
  xis_contamifile="suzaku/xis/bcf/ae_xi1_contami_20061016.fits" \
  outfile="suzaku_xis_events.fits"
```

XMM(EPIC):

```
runsixt \
  EventList="sixtxmm_events.fits" \
  PatternList="sixtxmm_pattern.fits" \
  Mission="XMM" Instrument="EPICPN" \
  Mode="FFTHIN" \
  XMLFile="xmm/epicpn/fullframe_thinfilter.xml" \
  Simput="phlist.fits" \
  Exposure=1.0e4E \
  RA=10.0 Dec=10.0 \
  MJDREF=50814.0
epicpn_events \
  PatternList="sixtxmm_pattern.fits" \
  EPICpnEventList="xmm_epic_events.fits"
```

eROSITA:

```
erosim \
  prefix="erosita_" \
  PhotonList=events_pv.fits \
  RawData=events_allpv.fits \
  background=yes \
  XMLFile="srg/erosita_1.xml" \
  XMLFILE1="srg/erosita_1.xml" \
  XMLFILE2="srg/erosita_2.xml" \
  XMLFILE3="srg/erosita_3.xml" \
  XMLFILE4="srg/erosita_4.xml" \
  XMLFILE5="srg/erosita_5.xml" \
  XMLFILE6="srg/erosita_6.xml" \
  XMLFILE7="srg/erosita_7.xml" \
  Simput="phlist.fits" \
  Exposure=1.0e4 \
  SkipInvalids=yes \
```

```

seed=-1 \
clobber=yes \
RA=10.0 Dec=10.0 \
MJDREF=50814.0

```

Athena(XIFU):

```

xifupipeline \
  prefix="athena_xifu_" \
  PixImpactList=impact.fits \
  XMLFile=athena/1469mm_xifu/xifu_baseline.xml \
  AdvXml=athena/1469mm_xifu/xifu_detector_hex_baseline.xml \
  Background=yes \
  RA=10.0 Dec=10.0 \
  Simput="phlist.fits" \
  Exposure=1.0e4 \
  UseRMF=T \
  clobber=yes

```

Athena(WFI):

```

athenawfisim \
  prefix=athena_wfi_ \
  XMLFile0="athena/1469mm_wfi_w_filter/depfet_b_1l_ff_chip0.xml" \
  XMLFile1="athena/1469mm_wfi_w_filter/depfet_b_1l_ff_chip1.xml" \
  XMLFile2="athena/1469mm_wfi_w_filter/depfet_b_1l_ff_chip2.xml" \
  XMLFile3="athena/1469mm_wfi_w_filter/depfet_b_1l_ff_chip3.xml" \
  Simput="phlist.fits" \
  Exposure=1.0e4 \
  Background=yes \
  Attitude="athena/attitude_wfi_ra10_dec10.fits" \
  RA=10.00 Dec=10.00 \
  chatter=0 \
  MJDREF=52000.0 \
  clobber=yes

```

XraySurveyor(HDXI)

```

pset simx mode=hl
pset simx Exposure=1.0e4
pset simx UseSimput=yes
pset simx MissionName=XraySurveyor
pset simx InstrumentName=HDXI
pset simx ScaleBkgnd=0.0
pset simx RandomSeed=24
pset simx SimputFile=phlist.fits
pset simx PointingRA=10.0

```

```
pset simx PointingDec=10.0
pset simx OutputFileName=surveyor_hdxi_events
simx
```

XraySurveyor(XCAL)

```
pset simx mode=hl
pset simx Exposure=1.0e4
pset simx UseSimput=yes
pset simx MissionName=XraySurveyor
pset simx InstrumentName=XCAL
pset simx ScaleBkgnd=0.0
pset simx RandomSeed=24
pset simx SimputFile=phlist.fits
pset simx PointingRA=10.0
pset simx PointingDec=10.0
pset simx OutputFileName=surveyor_xcal_events
simx
```

Hitomi(SXS)

```
pset simx mode=hl
pset simx Exposure=1.0e4
pset simx UseSimput=yes
pset simx MissionName=Hitomi
pset simx InstrumentName=SXS
pset simx ScaleBkgnd=0.0
pset simx RandomSeed=24
pset simx SimputFile=phlist.fits
pset simx PointingRA=10.0
pset simx PointingDec=10.0
pset simx OutputFileName=hitomi_sxs_events
simx
```

Hitomi(SXI)

```
pset simx mode=hl
pset simx Exposure=1.0e4
pset simx UseSimput=yes
pset simx MissionName=Hitomi
pset simx InstrumentName= SXI
pset simx ScaleBkgnd=0.0
pset simx RandomSeed=24
pset simx SimputFile=phlist.fits
pset simx PointingRA=10.0
pset simx PointingDec=10.0
pset simx OutputFileName=hitomi_sxi_events
```

```
simx
```

```
Hitomi(HXI)
```

```
pset simx mode=hl
pset simx Exposure=1.0e4
pset simx UseSimput=yes
pset simx MissionName=Hitomi
pset simx InstrumentName=HXI
pset simx ScaleBkgnd=0.0
pset simx RandomSeed=24
pset simx SimputFile=phlist.fits
pset simx PointingRA=10.0
pset simx PointingDec=10.0
pset simx OutputFileName=hitomi_hxi_events
simx
```

```
Chandra(ACIS-S)
```

```
marx S-SIMPOT-Source="phlist.fits" \
ExposureTime=1.0e4 TStart=2012.5 \
GratingType="NONE" DetectorType="ACIS-S" \
DitherModel="INTERNAL" RANom=10 Dec_Nom=10 Roll_Nom=50 \
SourceRA=10 SourceDEC=10 \
Verbose=yes mode=h OutputDir=point
marx2fits point chandra_acis-s_evt.fits
```

```
Chandra(ACIS-I)
```

```
marx S-SIMPOT-Source="phlist.fits" \
ExposureTime=1.0e4 TStart=2012.5 \
GratingType="NONE" DetectorType="ACIS-I" \
DitherModel="INTERNAL" RANom=10 Dec_Nom=10 Roll_Nom=50 \
SourceRA=10 SourceDEC=10 \
Verbose=yes mode=h OutputDir=point
marx2fits point chandra_acis-i_evt.fits
```

Chapter 5

Gadget3 on GPUs

Antonio Ragagnin^{1,2,3}, Mathias Wagner⁴ Klaus Dolag^{3,5}, Margarita Petkova^{2,3}, Alexander Arth^{3,5}, David Hubber³, Conradin Roffler⁶, Claudio Gheller⁶

¹ Leibniz-Rechenzentrum (LRZ), Boltzmannstrasse 1, D-85748 Garching, Germany

² Excellence Cluster Universe, Boltzmannstrasse 2r, D-85748 Garching, Germany

³ Universitäts-Sternwarte, Fakultät für Physik, Ludwig-Maximilians Universität München, Scheinerstrasse 1, D-81679 München, Germany

⁴ NVIDIA Research

⁵ Max-Planck-Institut für Astrophysik, Karl-Schwarzschild Strasse 1, D-85748 Garching bei München, Germany

⁶ CSCS-ETH, Via Trevano 131, Lugano 6900, Switzerland

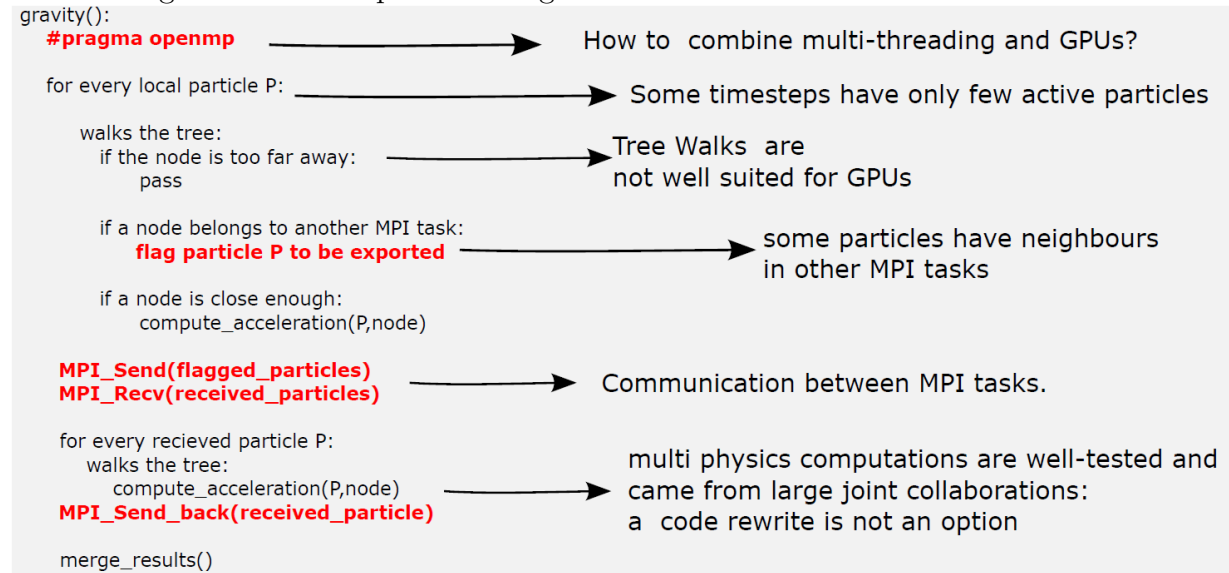
5.1 Gadget3 Parallelisation Infrastructure

Gadget3 is one of the most used codes in cosmological simulations, and it is well known to weak-scale up to hundred of thousands of core. In order to prepare the code to the new generation of simulations, we improved its Single Instruction Multiple Data (SIMD) and Single Instruction Multiple Thread (SIMT) parallelism by porting it to the GPUs. In its porting to the GPUs, we exploited the fact that all main Gadget modules (gravitational interaction, SPH, conduction) share the same pattern of processing a list of active particles and find the list of nearby particles (“neighbours”) in order to update particles properties. These neighbours are typically selected within a given searching sphere, defined by a given searching radius, defined by local conditions of the active particles (see, e.g. Hernquist and Katz (1989)).

Here below we discuss some important points that must taken into account when porting Gadget to GPUs.

- **parallel scheme:** Concerning the parallelisation scheme, Gadget3 is parallelised using a hybrid OpenMP/MPI approach. Each MPI task own a number of chunks of the Hilbert-ordered particles, and this ensures that every MPI task has a set of spatially nearby particles. When computing the acceleration on a given particle, such particles may interact with a region of space that belongs to a different CPU.
- **Gadget kernels:** The list of active particles is processed in multi-threaded parallel fashion within the framework of OpenMP. For this reason MPI ranks receive guest-particles from other MPI ranks, and compute the interaction (and the relative tree walk) for these guest particles. Once a MPI task completes the computation of guest particles, it sends the results back to the original MPI task, and, since forces sum linearly, the MPI rank can simply sum the local contributions with the imported ones.
- **Adaptive time steps:** Gadget has an adaptive timestep, thus only the position and velocity of currently active particles are updated at the end of every time bin. For this reason, during the tree walk, if Gadget encounters a particle (or a whole tree branch with this kind of particles) whose position wasn’t updated previously, it will drift it in a thread-locking fashion. Besides the thread-locking parts, there are additional multi-node blocking communications during the pseudo-particles exchanges.
- **Memory pattern:** Additionally, it should be mentioned that the memory storage of Gadget3 particles is organises in a so called Array of Structures (AoS) fashion. This means that all properties of a single particle are grouped together in memory. This poses a performance problem because Gadget stores a high number of data for each particle (up to $200B$ used by the gravity module and $600B$ used by all the hydrodynamics modules), while the number of properties used in each single Gadget module is low. For this reason, every time the code reads the property of a particle, it also loads several hundred of bytes of unused information into the cache. While

Figure 5.1: A sketch of the gravity loop in the standard OpenMP version of Gadget and the challenges we faced to port such algorithm to the GPUs.



AoS approach is preferred for making cleaner code, e.g., in C it is possible to copy all properties of particle i to a new structure by just writing `struct particle_data myP = P[i]`; as opposed to a more cache-friendly Structure of Arrays (SoA, where, in the previous example, one would have to manually copy all i -th properties).

A limitation for a full parallelisation is given by some non-thread safe regions in its tree walk: (1) the first lock in the standard implementation of Gadget is that all threads concur to fill a shared export buffer for the purpose of later sending neighbour lists to other MPI ranks; (2) the other lock is given by the non thread safe operation of drifting.

5.2 GPU porting

Figure 5.1 shows the current status of the Barnes Hut implementation of Gadget and the main challenges in porting the code to GPUs. Although the figure refers specifically to a gravity loop, it can be generalised to all other Gadget kernels in the sense that:

- Gadget can't benefit from vectorisation because it stores data in array of structures (AoS).
- The use of blocking MPI communications in order to exchange neighbouring particles between MPI ranks poses a limit in fully exploiting the computational power of both CPUs and GPUs.

- The code uses an adaptive timestepping scheme, and running the code only on the GPUs will not give any speedup when running time bins with only few of active particles.
- The code needs to perform some thread-locking operation at each tree walk because of the necessity of drifting particles and to fill the shared export buffer.
- Gadget has been built over a decennial effort of developers who implemented various versions of gravity and SPH solvers, and sub-grid models (star formation and feedback, black hole growth and AGN feedback, etc..) that have been extensively tested. Rewriting Gadget3 in a unified architecture language (e.g. CUDA) or heterogeneous computing language (e.g. OpenCL) would imply rewriting portions of such modules with the associated risk of adding mistakes.

We therefore decided to develop our porting using the OpenACC architecture due to the long developed history of Gadget and to limit the modification of the code and to make it possible to still run it on CPU only systems.

OpenACC is a programming standard supported by various C/C++/Fortran compilers (it is currently supported by GCC and PGI compilers). The user decorates region of codes with so-called "pragmas", directives that the compiler will further interpret. Such directives can tell, for instance, that a certain variable or array must be transferred to the accelerator, or that a certain block of code should be executed in the accelerator instead of the host.

5.2.1 Memory Transfer

To minimize communication between CPUs and GPUs, one would ideally load the initial conditions of the simulation in the memory of the GPU and run the whole simulation in the GPU. Unluckily, (i) each Gadget module needs its own specific subset of properties, and given that the average GPU would only be able to load a relatively low amount of particles (compared to the memory of a nowadays computing node) this means that large runs, capable of filling the whole node memory, would need a larger number of GPUs and (ii) small time bins won't perform on the GPUs.

To solve this issue and to be able to exploit the GPU memory at its best, we decided to only upload, for each Gadget module, the properties that are necessary for such module or for other successive modules in the current timestep. The drawback of this approach is that at each timestep we need to download the data back to the GPU, with its associated overhead.

For instance, the most recent Magneticum/Box0/mr run, with $2 \cdot 6 \cdot 10^6$ (the code was run with 2 MPI tasks per node) particles per node, was allocating *4GB* for the Barnes Hut tree, *22GB* for the basic quantities used in gravity (e.g. position, mass, acceleration ecc..), and, while part of these quantities (e.g. position, mass, particle-type..) are used in the tree walk of the successive modules, Gadget needs additional *14GB* for the additional SPH-only part (that is split in density computation and hydro-force computation), *0.6GB*

for the metal evolution and additional $4GB$ for the active particle list and the Peano keys, (the keys are used in the domain decomposition only).

Summing all the contributions we get $40GB$ usage per node, while one of the most modern super computer with accelerators as Piz Daint (6th in the top 500 list of super computers) has a GPU with only $16GB$. On the other hand, it is clear that a $16GB$ GPU system would be able to store the particle properties of a single Gadget module.

To further minimise the data transfer of the particle properties, we send only properties readed in each module (masses, positions, ecc..) and download only properties that are actually updated (e.g. acceleration). In the GPU porting we exploit the already present Gadget MPI communication infrastructure. We used the same Gadget routines to produce guest particles and to send the data to the GPUs. And in the same way, the GPU uses the already existing routines to exchange data, but in this case, particles are returned to the host (instead of the CPU-CPU communication of guest-particles between different MPI tasks).

5.2.2 Adaptive Timesteps

Large, high resolution, cosmological simulations have both void regions and clustered regions. Particles in void regions evolve with large timesteps because of the small force acting on them, compared to clustered regions where the stronger force requires very small timesteps.

After nearly half of the simulation time it is very common to have timebins with only one or few active particles, as shown in the Gadget output below (where the columns contains the timebin, the number of particles inside the timebin and the fraction of total time spent in this time bin).

Occupied timebins:	cumulative A D	cpu-frac
bin=22	186659106642	27.3%
bin=21	133675153886	15.9%
bin=20	93435782615	30.0%
bin=19	54791154	1.7%
bin=18	1209026	1.1%
bin=17	47667	1.9%
bin=16	9706	3.5%
bin=15	835	6.4%
bin=14	1	12.4%

We can see that, even in simulations with 10^{11} particles, there are timebins with $\simeq 1000$ or less active particles. Since timesteps with such a low amount of active particles won't benefit from the single instruction multiple thread (SIMT) architectures of GPUs, we decided to keep small timebins (with less than $2 \cdot 10^4$ active particles) to run on the CPU only. OpenACC turned out to be the best tool to implement this decision because it makes it possible to use the same code on both GPU and CPU with zero effort.

When a timestep run on the GPU, to remove all thread-locking operations from the

GPU tree walk, we drift all particles at the beginning of the timestep.

5.2.3 MPI Communication

One of the main advantage of our porting is that it overlaps CPU work with the GPU computation. We decided to overlap the CPU and the GPU computation in the following way: while the GPU loops over the active particles and computes local interactions, the CPU loops over the active particles in order to fill the export buffer.

While the GPU is busy with computing the interaction from the contribution of the same-node particles (i.e. local interactions), the CPU will take care of walking the tree for each active particle in order to perform all MPI send/receive of guest particles. If the amount of received guest particles is high enough (we chose a threshold of $2 \cdot 10^4$), then CPU will queue the interaction computation to the GPU, otherwise, if the number of guest particles is lower, such interactions will be computed in the CPU.

5.2.4 Barnes-Hut, SPH and Conduction porting

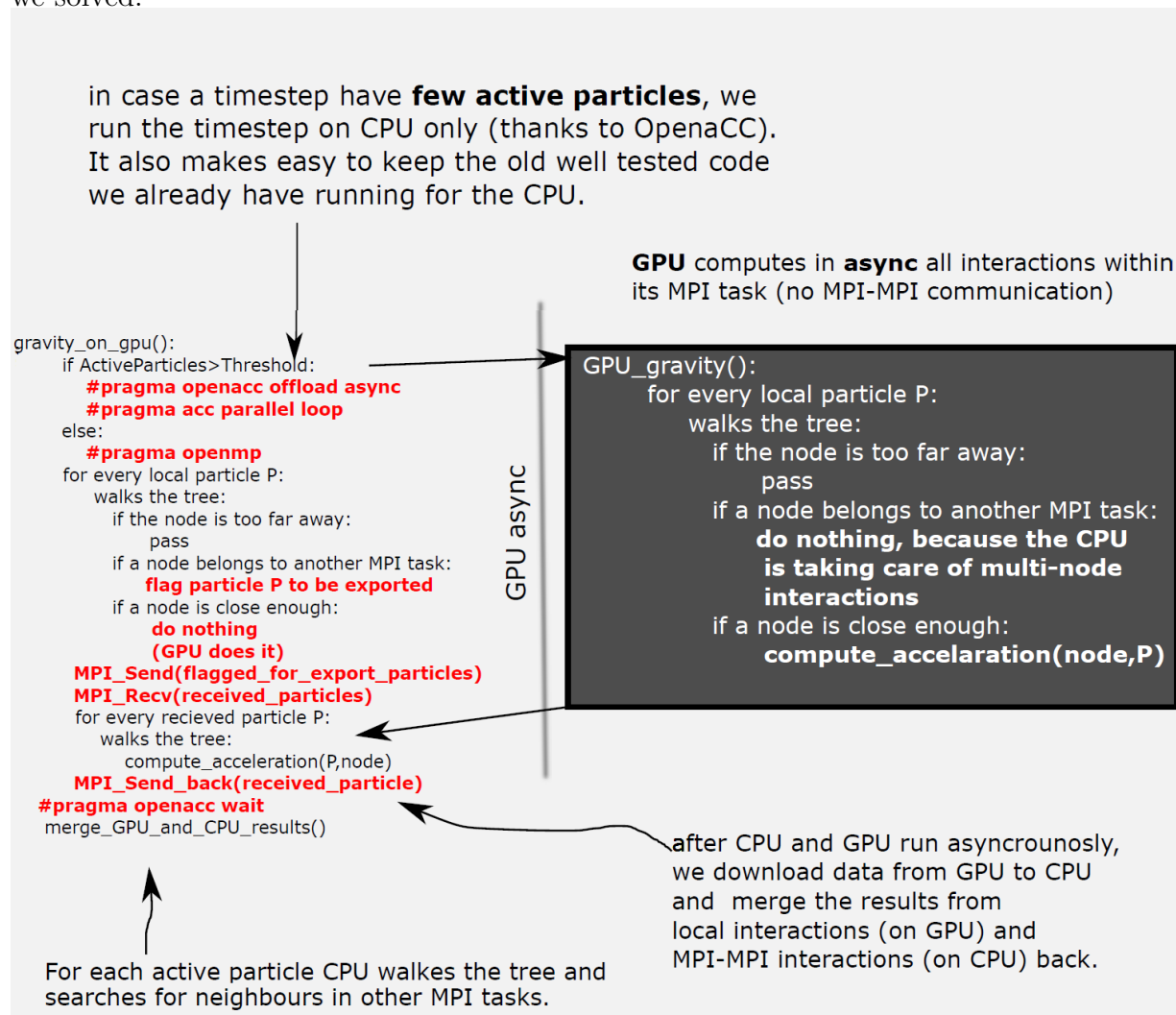
Although SPH, conduction and Barnes Hut algorithms compute the interaction acting over a list of active particles, there are some main differences to take into account when porting Gadget to the GPUs. First of all, SPH and conduction have only particle-particle interactions while Barnes Hut techniques collapse distant tree nodes in point-like pseudo particles. As a consequence Gadget's Barnes Hut algorithm implementation has the force computation embedded in the tree walk, as opposed to SPH and conduction that first build a neighbour list and only after they loop over it. Additionally, SPH and conduction require a set of neighbours within a distance that is well known before the tree walk. On the other hand, Barnes Hut opening criteria depends on a angle between the target particle and the cells (and thus is not known a priori). In a OpenACC porting this implies that gravity acceleration computation will be inside a tree walk branch, which will limit the maximum performance.

On the other hand, in the SPH and conduction modules it is possible to disentangle the tree walk from the force computation. The drawback is that it is not well known a priori the amount of neighbours of a given SPH particle (especially in zoom-in simulations). The CPU implementation overcomes this problem by allocating a neighbour buffer for each thread of a size that is equal to the number of local particles. Since it is practically impossible to allocate such a long buffer on each GPU thread, our porting performs a tree walk and neighbour interactions in chunks of 32 neighbours.

5.3 Results

Figure 5.2 shows a schematic review of our OpenACC porting of the gravity module in Gadget. The porting is very similar to the SPH porting, with the difference that the SPH loop is performed over chunks of 32 particles. We run our implementation with

Figure 5.2: A sketch of the gravity loop in the GPU porting of Gadget and the challenges we solved.



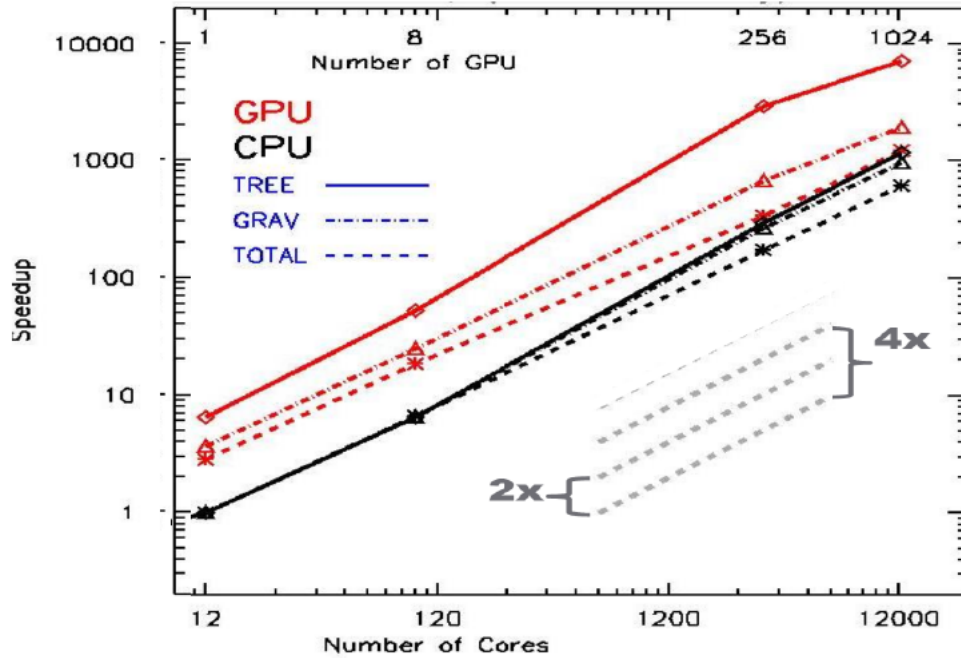


Figure 5.3: Scaling of the OpenACC porting of the gravity solver of Gadget in the PizDaint machine. The plot shows the speed up vs. the number of CPU cores for both GPU and CPU runs. Courtesy of Klaus Dolag.

the Magneticum/Box4/uhr initial conditions on the Piz Daint (CSCS) machine (12 cores, 64GB RAM, and NVIDIA Tesla P100 16GB) and tested its speed up against the same architecture.

The simulation is a full cosmological box with $2 \cdot 216^3$ particles, covers a volume of 48Mpc/h, has a mass resolution of $6.9 \cdot 10^8 M_\odot$ for dark matter particles and $1.4 \cdot 10^8 M_\odot$ for gas particles, with a softening length of 3.7kpc/h.

Both the OpenACC run and the CPU run are executed with the fastest MPI/OpenMP configuration on the CPU run. In particular it runs over 8 MPI tasks, each with 4 OpenMP threads. The speedup we found at the final time is

- Total speedup: 2.15
- Gravity solver speedup: 1.8
- SPH speedup: 2.6
- Conduction speedup: 3.0

5.4 Conclusions

We ported the most time-consuming modules of Gadget on the GPUs using OpenACC. We managed to obtain a speedup higher than 2 in a full cosmological box because large timesteps are executed in the GPU and small timesteps are executed in the CPU, and because we overlap MPI communication (performed on the CPU) and force computations (performed on the GPU if there are at least 10^4 active particles). From further analyses we found that there is a non-negligible fraction of time spent in CPU-GPU communication. The future plan is to test the code on more modern GPU systems (e.g. the Volta architecture from NVIDIA).

During the implementation of the algorithm, (at the 2016 EuroHack at CSCS) we were able to scale the Gadget code with the GPU gravity implementation up to 1024 GPUs. In this occasion we were able to show how this implementation is able to scale up to thousand of GPUs with a constant speedup (see Figure 5.3).

As the porting is giving a constant speedup over a very large number of GPUs, we are confident that we prepared the code for the GPU runs and for the next generations of cosmological simulations (which will probably need to fully exploit SIMT and SIMD architectures). Nevertheless future work is still needed to be done in order to fully exploit GPU architectures (e.g. by further minimising GPU–CPU communication and by running using multiple GPUs per MPI rank).

Chapter 6

Conclusions

The current generation of codes for hydrodynamic cosmological simulations have to exploit single instruction multiple thread (SIMT) and single instruction multiple data (SIMD) in order to perform on most of the modern super computers.

In order to run Magneticum/Box0/mr simulation and to fully exploit the super computer SuperMUC, we studied Gadget3 bottlenecks and partially refactored the code in order to minimise overheads so that most of the execution time is spent in computations related to physics only. In Chapter 2 we show how the neighbour search implemented in the N-body code Gadget is the most time consuming part of the code. The neighbour search is performed by Gadget at every timestep for all active particles and consists of the process of walking the tree in order to find neighbouring particles. For this reason, we developed and implemented a way to recycle neighbours to accelerate the neighbour search. We exploited the fact that particles are ordered with a space filling curve and thus particles that are close-by in memory are also close-by in the simulated volume. Our technique makes a single neighbour search for a whole group of particles that are close enough. As a result, we perform less neighbour searches over larger searching lengths. Our technique should work, in principle, for any N-Body code with a space-filling-curve ordering of particles. After refactoring the code, the neighbour search is no more the most time consuming part of Gadget.

We then combine three Magneticum boxes (Box0/mr, Box2b/hr and Box4/uhr) to cover a mass range of $3 \cdot 10^{11} - 6 \cdot 10^{15} M_{\odot}$ of well resolved clusters from redshift zero to redshift 2, and we computed the concentration for all well resolved halos. Then we fit the concentration as a power law of mass and redshift. This is the first study of the mass-concentration relation in hydrodynamic simulations covering several orders of magnitude in mass. In the high mass regime, we found a good agreement between the concentration predicted by the Magneticum clusters and past theoretical and observational works in the literature. While in the low mass regime, the Magneticum simulation concentration is systematically lower than the concentration found in studies based on dark matter only simulations. Such different behaviour is in agreement with the studies of simulations with AGN feedback. They show how the effect of AGN feedback in low mass haloes, is capable of lowering the concentration up to $\approx 15\%$ (see Figure 8 in Duffy et al., 2010) by

removing the baryons from the inner region of the halo. Due to the high mass range of the Magneticum simulations, we are able to capture this effect and its disappearance as the halo mass increases. In the second part of Chapter 3 we discussed the origin of the large scatter of concentration in the mass-concentration plane by studying its dependency on fossilness. Since we are working with hydrodynamic simulations, we were able to compute the halo fossilness by exploiting the stellar masses of galaxies (see Eq. 3.4) and comparing our values to observations.

Since the output of the most modern cosmological simulations can occupy several hundred of TB of data, we presented a data center based on a multi-layer infrastructure for large cosmological hydrodynamic simulations (see Chapter 4). In fact hydrodynamic simulations may include different kind of matter content (gas, dark matter, stars, black holes) and various physical processes that shape the formation of objects and structures in our universe, and all those factors dramatically increase the output size of the current simulations. The web portal provides the possibility to the wide scientific community to perform analysis tools on data from several large simulations. This data center allows users to configure workflows to run individual analysis tools for specific objects of a given snapshot on the raw simulation data. The web interface helps users to find interesting galaxy clusters by (1) building compounded queries on metadata of the different simulations, which in our case are galaxies and galaxy clusters as obtained by SUBFIND, (2) select objects from plots of arbitrary cluster and galaxy properties and (3) by browsing 2D maps of the simulation. Those queries can be restrictions to global properties like mass or other observables, as well as complex queries on various internal aspects. This gives the possibility to the users to select different general classes of objects (like merging clusters, compact groups or fossil groups) and to run some post-processing jobs over them. The jobs are executed in a dedicated computing cluster and the web portal users do not need to request an access to the HPC facility.

Finally, since some of the new super computers are including more and more accelerators (e.g. GPUs), in Chapter 5 we present a GPU porting of the code Gadget3 using the OpenACC framework. The porting is performed by decorating the code with OpenACC "pragmas" directives. For this reason, it is not invasive and it still provides the possibility of running the code on CPU only systems. OpenACC turned out to be a useful tool in order to use the same exact code for both CPUs and GPUs runs, allowing the scientific community to work on a single source code that run on multiple architectures. We were able to fully exploit both the CPU and GPU of the supercomputer by overlapping computations (performed by the GPU) and communication of MPI ranks that exchanges neighbouring lists of active particles (performed by the CPU). Additionally, since GPU systems won't perform well in small time bins, when a time bin has less than 10^4 active particles, the timestep runs in the CPU only. We tested our algorithm and found a total speedup > 2 in the run of the Magneticum/Box4/hr box. This porting also opens and tests the viability of making Gadget vectorisation, which is a necessity in order to run the future large cosmological hydrodynamic simulations.

For simulations that will run in the future generation of super computers, it will be necessary to move toward vector-friendly codes. Our GPU porting can be a base for a

SIMD friendly code. Additionally, codes should move to a Structure of Array paradigm, in contrast with the current Array of Structure paradigm of Gadget3 in order to exploit the cache memory of the underlying systems. Finally, Gadget still has a serial tree build and a serial domain decomposition, which could be a bottleneck for future simulations, and the PM potential computation perform all to all communications between all MPI tasks which may slow down the next large runs.

Acknowledgements

I am grateful to my family and the cats we had during my childhood, Omero, Cartesio, Penelope, Ciccillo, Kuga and Milo who inspired me with their life full of curiosity. Thanks to Vittoria Musante for the design guide lines of my posters and for helping me with the German translations. Thanks to Rupam Bhattacharya for carefully proof reading some parts of this manuscript, and, with the Champignons, for the musical jam sessions and drinks we had during my time in Munich. I'm also grateful to Vasileios Karakasis, Nicolay J. Hammer, Luigi Iapichino and Michele Martone for the various discussions on super computers. Thanks to the people at the observatory in Trieste for the interesting discussions on the future of cosmological simulations, that shaped the final part of this thesis. Additionally, thanks to the never ending candies and coffee inside Klaus' office, thanks to my IMPRS committee (Klaus Dolag, Michael Bader, Nicolay J. Hammer) for always pointing me to a possible direction when I found myself stuck in solving a problem, and thanks to everyone who attended my talks at the CAST group at the observatory of Munich, for the many ideas and discussions.

Bibliography

- S. J. Aarseth. Direct n-body calculations. In J. Goodman and P. Hut, editors, *Dynamics of Star Clusters*, volume 113 of *IAU Symposium*, pages 251–258, 1985.
- F. Abdalla, J. Annis, D. Bacon, S. Bridle, F. Castander, M. Colless, D. DePoy, H. T. Diehl, M. Eriksen, B. Flaugher, J. Frieman, E. Gaztanaga, C. Hogan, S. Jouvel, S. Kent, D. Kirk, R. Kron, S. Kuhlmann, O. Lahav, J. Lawrence, H. Lin, J. Marriner, J. Marshall, J. Mohr, R. C. Nichol, M. Sako, W. Saunders, M. Soares-Santos, D. Thomas, R. Wechsler, A. West, and H. Wu. The Dark Energy Spectrometer (DESPEC): A Multi-Fiber Spectroscopic Upgrade of the Dark Energy Camera and Survey for the Blanco Telescope. *ArXiv e-prints*, September 2012.
- S. W. Allen, S. Ettori, and A. C. Fabian. Chandra measurements of the distribution of mass in the luminous lensing cluster Abell 2390. *MNRAS*, 324:877–890, July 2001. doi: 10.1046/j.1365-8711.2001.04318.x.
- L. Amendola, S. Appleby, D. Bacon, T. Baker, M. Baldi, N. Bartolo, A. Blanchard, C. Bonvin, S. Borgani, E. Branchini, C. Burrage, S. Camera, C. Carbone, L. Casarini, M. Cropper, C. de Rham, C. Di Porto, A. Ealet, P. G. Ferreira, F. Finelli, J. García-Bellido, T. Giannantonio, L. Guzzo, A. Heavens, L. Heisenberg, C. Heymans, H. Hoekstra, L. Holtenstein, R. Holmes, O. Horst, K. Jahnke, T. D. Kitching, T. Koivisto, M. Kunz, G. La Vacca, M. March, E. Majerotto, K. Markovic, D. Marsh, F. Marulli, R. Massey, Y. Mellier, D. F. Mota, N. J. Nunes, W. Percival, V. Pettorino, C. Porciani, C. Quercellini, J. Read, M. Rinaldi, D. Sapone, R. Scaramella, C. Skordis, F. Simpson, A. Taylor, S. Thomas, R. Trotta, L. Verde, F. Vernizzi, A. Vollmer, Y. Wang, J. Weller, and T. Zlosnik. Cosmology and Fundamental Physics with the Euclid Satellite. *Living Reviews in Relativity*, 16:6, September 2013. doi: 10.12942/lrr-2013-6.
- P. Anninos and M. L. Norman. *Nested grid methods for cosmological hydrodynamic and N-body systems*. 1994.
- K. A. Arnaud. XSPEC: The First Ten Years. In G. H. Jacoby and J. Barnes, editors, *Astronomical Data Analysis Software and Systems V*, volume 101 of *Astronomical Society of the Pacific Conference Series*, page 17, 1996.

- A. Arth, K. Dolag, A. M. Beck, M. Petkova, and H. Lesch. Anisotropic thermal conduction in galaxy clusters with MHD in Gadget. *ArXiv e-prints*, December 2014.
- Uri M. Ascher and Linda R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM: Society for Industrial and Applied Mathematics, 1998. ISBN 0898714125. URL <https://www.amazon.com/Computer-Ordinary-Differential-Equations-Differential-Algebraic/dp/0898714125?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbiori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0898714125>.
- Jasjeet S Bagla. TreePM: A code for cosmological n-body simulations. *Journal of Astrophysics and Astronomy*, 23(3-4):185–196, 2002.
- R. Ballouz. *Numerical Simulations of Granular Physics in the Solar System*. PhD thesis, University of Maryland, College Park, 2017.
- J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324: 446–449, December 1986a. doi: 10.1038/324446a0.
- J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324: 446–449, December 1986b. doi: 10.1038/324446a0.
- J. E. Barnes. An Efficient N-Body Algorithm for a Fine-Grain Parallel Computer. In P. Hut and S. L. W. McMillan, editors, *The Use of Supercomputers in Stellar Dynamics*, volume 267 of *Lecture Notes in Physics*, Berlin Springer Verlag, page 175, 1986. doi: 10.1007/BFb0116409.
- I. Bartalucci, M. Arnaud, G. W. Pratt, and A. M. C. Le Brun. Resolving the hydrostatic mass profiles of galaxy clusters at $z \sim 1$ with XMM-Newton and Chandra. *A&A*, 617: A64, September 2018. doi: 10.1051/0004-6361/201732458.
- A. M. Beck, G. Murante, A. Arth, R.-S. Remus, A. F. Teklu, J. M. F. Donnert, S. Planelles, M. C. Beck, P. Förster, M. Imgrund, K. Dolag, and S. Borgani. An improved SPH scheme for cosmological simulations. *MNRAS*, 455:2110–2130, January 2016. doi: 10.1093/mnras/stv2443.
- M. Bernyk, D. J. Croton, C. Tonini, L. Hodkinson, A. H. Hassan, T. Garel, A. R. Duffy, S. J. Mutch, G. B. Poole, and S. Hegarty. The Theoretical Astrophysical Observatory: Cloud-based Mock Galaxy Catalogs. *ApJS*, 223:9, March 2016. doi: 10.3847/0067-0049/223/1/9.
- E. Bertschinger. Gravitational Simulations of Cosmic Structure Formation. In *American Astronomical Society Meeting Abstracts #186*, volume 27 of *Bulletin of the American Astronomical Society*, page 849, May 1995.

- E. Bertschinger. Simulations of Structure Formation in the Universe. *ARA&A*, 36:599–654, 1998. doi: 10.1146/annurev.astro.36.1.599.
- S. Bhattacharya, S. Habib, K. Heitmann, and A. Vikhlinin. Dark Matter Halo Profiles of Massive Clusters: Theory versus Observations. *ApJ*, 766:32, March 2013. doi: 10.1088/0004-637X/766/1/32.
- V. Biffi, K. Dolag, H. Böhringer, and G. Lemson. Observing simulated galaxy clusters with PHOX: a novel X-ray photon simulator. *MNRAS*, 420:3545–3556, March 2012. doi: 10.1111/j.1365-2966.2011.20278.x.
- V. Biffi, K. Dolag, and H. Böhringer. Investigating the velocity structure and X-ray observable properties of simulated galaxy clusters with PHOX. *MNRAS*, 428:1395–1409, January 2013. doi: 10.1093/mnras/sts120.
- J. Binney and S. Tremaine. *Galactic Dynamics: Second Edition*. Princeton University Press, 2008.
- A. Biviano, A. Moretti, A. Paccagnella, B. M. Poggianti, D. Bettoni, M. Gullieuszik, B. Vulcani, G. Fasano, M. D’Onofrio, J. Fritz, and A. Cava. The concentration-mass relation of clusters of galaxies from the OmegaWINGS survey. *A&A*, 607:A81, November 2017. doi: 10.1051/0004-6361/201731289.
- M. R. Blanton, M. A. Bershadsky, B. Abolfathi, F. D. Albareti, C. Allende Prieto, A. Almeida, J. Alonso-García, F. Anders, S. F. Anderson, B. Andrews, and et al. Sloan Digital Sky Survey IV: Mapping the Milky Way, Nearby Galaxies, and the Distant Universe. *AJ*, 154:28, July 2017. doi: 10.3847/1538-3881/aa7567.
- S. Bocquet, A. Saro, K. Dolag, and J. J. Mohr. Halo mass function: baryon impact, fitting formulae, and implications for cluster cosmology. *MNRAS*, 456:2361–2373, March 2016. doi: 10.1093/mnras/stv2657.
- H. Böhringer, P. Schuecker, L. Guzzo, C. A. Collins, W. Voges, S. Schindler, D. M. Neumann, R. G. Cruddace, S. De Grandi, G. Chincarini, A. C. Edge, H. T. MacGillivray, and P. Shaver. The ROSAT-ESO flux limited X-ray (REFLEX) galaxy cluster survey. I. The construction of the cluster sample. *A&A*, 369:826–850, April 2001. doi: 10.1051/0004-6361:20010240.
- H. Bondi. On spherically symmetrical accretion. *MNRAS*, 112:195, 1952. doi: 10.1093/mnras/112.2.195.
- H. Bondi and F. Hoyle. On the mechanism of accretion by stars. *MNRAS*, 104:273, 1944. doi: 10.1093/mnras/104.5.273.

- C. M. Booth and J. Schaye. Simulations of the Growth of Black Holes and Feedback from Active Galactic Nuclei. In S. Heinz and E. Wilcots, editors, *American Institute of Physics Conference Series*, volume 1201 of *American Institute of Physics Conference Series*, pages 21–24, December 2009. doi: 10.1063/1.3293041.
- S. Borgani. Cosmology with Clusters of Galaxies. In M. Plionis, O. López-Cruz, and D. Hughes, editors, *A Pan-Chromatic View of Clusters of Galaxies and the Large-Scale Structure*, volume 740 of *Lecture Notes in Physics*, Berlin Springer Verlag, page 24, 2008. doi: 10.1007/978-1-4020-6941-3_9.
- S. Borgani and A. Kravtsov. Cosmological Simulations of Galaxy Clusters. *Advanced Science Letters*, 4:204–227, February 2011. doi: 10.1166/asl.2011.1209.
- S. Borgani, G. Murante, V. Springel, A. Diaferio, K. Dolag, L. Moscardini, G. Tormen, L. Tornatore, and P. Tozzi. X-ray properties of galaxy clusters and groups from a cosmological hydrodynamical simulation. *MNRAS*, 348:1078–1096, March 2004. doi: 10.1111/j.1365-2966.2004.07431.x.
- F. R. Bouchet and L. Hernquist. Cosmological simulations using the hierarchical tree method. *ApJS*, 68:521–538, December 1988. doi: 10.1086/191299.
- G. L. Bryan and M. L. Norman. Simulating X-ray Clusters with Adaptive Mesh Refinement. In *American Astronomical Society Meeting Abstracts*, volume 27 of *Bulletin of the American Astronomical Society*, page 1421, December 1995.
- G. L. Bryan and M. L. Norman. A Hybrid AMR Application for Cosmology and Astrophysics. *ArXiv Astrophysics e-prints*, October 1997.
- G. L. Bryan, M. L. Norman, B. W. O’Shea, T. Abel, J. H. Wise, M. J. Turk, D. R. Reynolds, D. C. Collins, P. Wang, S. W. Skillman, B. Smith, R. P. Harkness, J. Bordner, J.-h. Kim, M. Kuhlen, H. Xu, N. Goldbaum, C. Hummels, A. G. Kritsuk, E. Tasker, S. Skory, C. M. Simpson, O. Hahn, J. S. Oishi, G. C. So, F. Zhao, R. Cen, Y. Li, and Enzo Collaboration. ENZO: An Adaptive Mesh Refinement Code for Astrophysics. *ApJS*, 211:19, April 2014. doi: 10.1088/0067-0049/211/2/19.
- J. Buchner, A. Georgakakis, K. Nandra, L. Hsu, C. Rangel, M. Brightman, A. Merloni, M. Salvato, J. Donley, and D. Kocevski. X-ray spectral modelling of the AGN obscuring region in the CDFS: Bayesian model selection and catalogue. *A&A*, 564:A125, April 2014. doi: 10.1051/0004-6361/201322971.
- J. S. Bullock, T. S. Kolatt, Y. Sigad, R. S. Somerville, A. V. Kravtsov, A. A. Klypin, J. R. Primack, and A. Dekel. Profiles of dark haloes: evolution, scatter and environment. *MNRAS*, 321:559–575, March 2001. doi: 10.1046/j.1365-8711.2001.04068.x.

- Hans-Joachim Bungartz, Miriam Mehl, Tobias Neckel, and Tobias Weinzierl. The PDE framework Peano applied to fluid dynamics: an efficient implementation of a parallel multiscale fluid dynamics solver on octree-like adaptive Cartesian grids. *Computational Mechanics*, 46(1):103–114, 2010.
- D. A. Buote. The Unusually High Halo Concentration of the Fossil Group NGC 6482: Evidence for Weak Adiabatic Contraction. *ApJ*, 834:164, January 2017. doi: 10.3847/1538-4357/834/2/164.
- R. Cen. Hydrodynamic Simulations of Cosmological Systems. In *Bulletin of the American Astronomical Society*, volume 22 of *BAAS*, page 1305, September 1990.
- R. Cen. A hydrodynamic approach to cosmology - Methodology. *ApJS*, 78:341–364, February 1992. doi: 10.1086/191630.
- S. Chandrasekhar. *Hydrodynamic and hydromagnetic stability*. 1961.
- C. A. Correa, J. S. B. Wyithe, J. Schaye, and A. R. Duffy. The accretion history of dark matter haloes - III. A physical model for the concentration-mass relation. *MNRAS*, 452:1217–1232, September 2015. doi: 10.1093/mnras/stv1363.
- H. M. P. Couchman. Mesh-refined P3M - A fast adaptive N-body algorithm. *ApJ*, 368:L23–L26, February 1991. doi: 10.1086/185939.
- G. Covone, M. Sereno, M. Kilbinger, and V. F. Cardone. Measurement of the Halo Bias from Stacked Shear Profiles of Galaxy Clusters. *ApJ*, 784:L25, April 2014. doi: 10.1088/2041-8205/784/2/L25.
- W. Cui, V. Springel, X. Yang, G. De Lucia, and S. Borgani. Properties of fossil groups in cosmological simulations and galaxy formation models. *MNRAS*, 416:2997–3008, October 2011. doi: 10.1111/j.1365-2966.2011.19248.x.
- W. Cui, S. Borgani, K. Dolag, G. Murante, and L. Tornatore. The effects of baryons on the halo mass function. *MNRAS*, 423:2279–2287, July 2012. doi: 10.1111/j.1365-2966.2012.21037.x.
- W. Cui, C. Power, S. Borgani, A. Knebe, G. F. Lewis, G. Murante, and G. B. Poole. On the dynamical state of galaxy clusters: insights from cosmological simulations - II. *MNRAS*, 464:2502–2510, January 2017. doi: 10.1093/mnras/stw2567.
- R. Davé, K. Finlator, and B. D. Oppenheimer. Galaxy evolution in cosmological simulations with outflows - II. Metallicities and gas fractions. *MNRAS*, 416:1354–1376, September 2011. doi: 10.1111/j.1365-2966.2011.19132.x.
- A. J. Davis, A. D’Aloisio, and P. Natarajan. Virialization of high-redshift dark matter haloes. *MNRAS*, 416:242–247, September 2011. doi: 10.1111/j.1365-2966.2011.19026.x.

- J. E. Davis, M. W. Bautz, D. Dewey, R. K. Heilmann, J. C. Houck, D. P. Huenemörder, H. L. Marshall, M. A. Nowak, M. L. Schattenburg, N. S. Schulz, and R. K. Smith. Raytracing with MARX: x-ray observatory design, calibration, and support. In *Space Telescopes and Instrumentation 2012: Ultraviolet to Gamma Ray*, volume 8443 of *Proc. SPIE*, page 84431A, September 2012. doi: 10.1117/12.926937.
- M. Davis, G. Efstathiou, C. S. Frenk, and S. D. M. White. The evolution of large-scale structure in a universe dominated by cold dark matter. *ApJ*, 292:371–394, May 1985. doi: 10.1086/163168.
- C. De Boni. The galaxy cluster concentration-mass relation in dark energy cosmologies. *ArXiv e-prints*, February 2013.
- T. Di Matteo, V. Springel, and L. Hernquist. Energy input from quasars regulates the growth and activity of black holes and their host galaxies. *Nature*, 433:604–607, February 2005. doi: 10.1038/nature03335.
- T. Di Matteo, J. Colberg, V. Springel, L. Hernquist, and D. Sijacki. Direct Cosmological Simulations of the Growth of Black Holes and Galaxies. *ApJ*, 676:33–53, March 2008. doi: 10.1086/524921.
- K. Dolag, M. Jubelgas, V. Springel, S. Borgani, and E. Rasia. Thermal Conduction in Simulated Galaxy Clusters. *ApJ*, 606:L97–L100, May 2004. doi: 10.1086/420966.
- K. Dolag, F. K. Hansen, M. Roncarelli, and L. Moscardini. The imprints of local superclusters on the Sunyaev-Zel’dovich signals and their detectability with Planck. *MNRAS*, 363:29–39, October 2005. doi: 10.1111/j.1365-2966.2005.09452.x.
- K. Dolag, S. Borgani, S. Schindler, A. Diaferio, and A. M. Bykov. Simulation Techniques for Cosmological Simulations. *Space Sci. Rev.*, 134:229–268, February 2008a. doi: 10.1007/s11214-008-9316-5.
- K. Dolag, M. Reinecke, C. Gheller, and S. Imboden. Splotch: visualizing cosmological simulations. *New Journal of Physics*, 10(12):125006, December 2008b. doi: 10.1088/1367-2630/10/12/125006.
- K. Dolag, S. Borgani, G. Murante, and V. Springel. Substructures in hydrodynamical cluster simulations. *MNRAS*, 399:497–514, October 2009. doi: 10.1111/j.1365-2966.2009.15034.x.
- K. Dolag, B. M. Gaensler, A. M. Beck, and M. C. Beck. Constraints on the distribution and energetics of fast radio bursts using cosmological hydrodynamic simulations. *MNRAS*, 451:4277–4289, August 2015. doi: 10.1093/mnras/stv1190.

- K. Dolag, E. Komatsu, and R. Sunyaev. SZ effects in the Magneticum Pathfinder simulation: comparison with the Planck, SPT, and ACT results. *MNRAS*, 463:1797–1811, December 2016. doi: 10.1093/mnras/stw2035.
- J. Dubinski. A parallel tree code. *New A*, 1:133–147, October 1996. doi: 10.1016/S1384-1076(96)00009-7.
- A. R. Duffy. Probing the nature of dark energy through galaxy redshift surveys with radio telescopes. *Annalen der Physik*, 526:283–293, August 2014. doi: 10.1002/andp.201400059.
- A. R. Duffy, J. Schaye, S. T. Kay, and C. Dalla Vecchia. Dark matter halo concentrations in the Wilkinson Microwave Anisotropy Probe year 5 cosmology. *MNRAS*, 390:L64–L68, October 2008. doi: 10.1111/j.1745-3933.2008.00537.x.
- A. R. Duffy, J. Schaye, S. T. Kay, C. Dalla Vecchia, R. A. Battye, and C. M. Booth. Impact of baryon physics on dark matter structures: a detailed simulation study of halo density profiles. *MNRAS*, 405:2161–2178, July 2010. doi: 10.1111/j.1365-2966.2010.16613.x.
- A. A. Dutton and A. V. Macciò. Cold dark matter haloes in the Planck era: evolution of structural parameters for Einasto and NFW profiles. *MNRAS*, 441:3359–3374, July 2014. doi: 10.1093/mnras/stu742.
- J. W. Eastwood. Low dissipation fluid and MHD simulation. *Computer Physics Communications*, 48:17–23, January 1988. doi: 10.1016/0010-4655(88)90019-7.
- Carl Eckart. Variation principles of hydrodynamics. *The Physics of Fluids*, 3(3):421–427, 1960.
- G. Efstathiou and J. W. Eastwood. On the clustering of particles in an expanding universe. *MNRAS*, 194:503–525, February 1981. doi: 10.1093/mnras/194.3.503.
- G. Efstathiou, M. Davis, S. D. M. White, and C. S. Frenk. Numerical techniques for large cosmological N-body simulations. *ApJS*, 57:241–260, February 1985. doi: 10.1086/191003.
- A. Einstein. Die Grundlage der allgemeinen Relativitätstheorie. *Annalen der Physik*, 354:769–822, 1916. doi: 10.1002/andp.19163540702.
- Rüdiger Esser and Renate Knecht. Intel paragon xp/s-architecture and software environment. pages 121–141, 1993.
- A. C. Fabian. Cooling Flows in Clusters of Galaxies. *ARA&A*, 32:277–318, 1994. doi: 10.1146/annurev.aa.32.090194.001425.
- A. C. Fabian, K. A. Arnaud, M. W. Bautz, and Y. Tawara. ASCA observations of cooling flows in clusters of galaxies. *ApJ*, 436:L63–L66, November 1994. doi: 10.1086/187633.

- A. C. Fabian, R. F. Mushotzky, P. E. J. Nulsen, and J. R. Peterson. On the soft X-ray spectrum of cooling flows. *MNRAS*, 321:L20–L24, February 2001. doi: 10.1046/j.1365-8711.2001.04285.x.
- D. Fabjan, S. Borgani, L. Tornatore, A. Saro, G. Murante, and K. Dolag. Simulating the effect of active galactic nuclei feedback on the metal enrichment of galaxy clusters. *MNRAS*, 401:1670–1690, January 2010. doi: 10.1111/j.1365-2966.2009.15794.x.
- G. J. Ferland, K. T. Korista, D. A. Verner, J. W. Ferguson, J. B. Kingdon, and E. M. Verner. CLOUDY 90: Numerical Simulation of Plasmas and Their Spectra. *PASP*, 110: 761–778, July 1998. doi: 10.1086/316190.
- R. Ferrell and E. Bertschinger. Particle-Mesh Methods on the Connection Machine. *International Journal of Modern Physics C*, 5:933–956, 1994. doi: 10.1142/S0129183194001069.
- A. Friedmann. Über die Krümmung des Raumes. *Zeitschrift für Physik*, 10:377–386, 1922. doi: 10.1007/BF01332580.
- J. A. Frieman, B. Bassett, A. Becker, C. Choi, D. Cinabro, F. DeJongh, D. L. Depoy, B. Dilday, M. Doi, P. M. Garnavich, C. J. Hogan, J. Holtzman, M. Im, S. Jha, R. Kessler, K. Konishi, H. Lampeitl, J. Marriner, J. L. Marshall, D. McGinnis, G. Miknaitis, R. C. Nichol, J. L. Prieto, A. G. Riess, M. W. Richmond, R. Romani, M. Sako, D. P. Schneider, M. Smith, N. Takanashi, K. Tokita, K. van der Heyden, N. Yasuda, C. Zheng, J. Adelman-McCarthy, J. Annis, R. J. Assef, J. Barentine, R. Bender, R. D. Blandford, W. N. Boroski, M. Bremer, H. Brewington, C. A. Collins, A. Crofts, J. Dembicky, J. Eastman, A. Edge, E. Edmondson, E. Elson, M. E. Eyler, A. V. Filippenko, R. J. Foley, S. Frank, A. Goobar, T. Gueth, J. E. Gunn, M. Harvanek, U. Hopp, Y. Ihara, Ž. Ivezić, S. Kahn, J. Kaplan, S. Kent, W. Ketzbeck, S. J. Kleinman, W. Kollatschny, R. G. Kron, J. Krziesiński, D. Lamenti, G. Leloudas, H. Lin, D. C. Long, J. Lucey, R. H. Lupton, E. Malanushenko, V. Malanushenko, R. J. McMillan, J. Mendez, C. W. Morgan, T. Morokuma, A. Nitta, L. Ostman, K. Pan, C. M. Rockosi, A. K. Romer, P. Ruiz-Lapuente, G. Saurage, K. Schlesinger, S. A. Snedden, J. Sollerman, C. Stoughton, M. Stritzinger, M. Subba Rao, D. Tucker, P. Vaisanen, L. C. Watson, S. Watters, J. C. Wheeler, B. Yanny, and D. York. The Sloan Digital Sky Survey-II Supernova Survey: Technical Summary. *AJ*, 135:338–347, January 2008. doi: 10.1088/0004-6256/135/1/338.
- Markus Geimer, Felix Wolf, Brian JN Wylie, Erika Ábrahám, Daniel Becker, and Bernd Mohr. The Scalasca performance toolset architecture. *Concurrency and Computation: Practice and Experience*, 22(6):702–719, 2010.
- Paul Gibbon, Wolfgang Frings, and Bernd Mohr. Performance analysis and visualization of the n-body tree code PEPC on massively parallel computers. In *PARCO*, pages 367–374, 2005.

- RA Gingold and JJ Monaghan. Kernel estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics*, 46(3):429–453, 1982.
- Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.
- Leslie Greengard and Vladimir Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta numerica*, 6:229–269, 1997.
- A. M. Groener, D. M. Goldberg, and M. Sereno. The galaxy cluster concentration-mass scaling relation. *MNRAS*, 455:892–919, January 2016. doi: 10.1093/mnras/stv2341.
- L. Hernquist. Performance characteristics of tree codes. *Astrophysical Journal Supplement Series*, 64:715–734, August 1987. doi: 10.1086/191215.
- L. Hernquist and N. Katz. TREESPH - a unification of SPH with the hierarchical tree method. *Astrophysical Journal Supplement Series*, 70:419–446, June 1989. doi: 10.1086/191344.
- M. Hirschmann, T. Naab, R. Davé, B. D. Oppenheimer, J. P. Ostriker, R. S. Somerville, L. Oser, R. Genzel, L. J. Tacconi, N. M. Förster-Schreiber, A. Burkert, and S. Genel. The effect of metal enrichment and galactic winds on galaxy formation in cosmological zoom simulations. *MNRAS*, 436:2929–2949, December 2013. doi: 10.1093/mnras/stt1770.
- M. Hirschmann, K. Dolag, A. Saro, L. Bachmann, S. Borgani, and A. Burkert. Cosmological simulations of black hole growth: AGN luminosities and downsizing. *MNRAS*, 442:2304–2324, August 2014. doi: 10.1093/mnras/stu1023.
- R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. 1988.
- R. W. Hockney, S. P. Goel, and J. W. Eastwood. Quiet High-Resolution Computer Models of a Plasma. *Journal of Computational Physics*, 14:148–158, February 1974. doi: 10.1016/0021-9991(74)90010-2.
- F. Hofmann, J. S. Sanders, N. Clerc, K. Nandra, J. Ridl, K. Dennerl, M. Ramos-Ceja, A. Finoguenov, and T. H. Reiprich. eROSITA cluster cosmology forecasts: Cluster temperature substructure bias. *A&A*, 606:A118, October 2017. doi: 10.1051/0004-6361/201730742.
- E. Holmberg. On the Clustering Tendencies among the Nebulae. II. a Study of Encounters Between Laboratory Models of Stellar Systems by a New Integration Procedure. *ApJ*, 94:385, November 1941. doi: 10.1086/144344.
- F. Hoyle and R. A. Lyttleton. The effect of interstellar matter on climatic variation. *Proceedings of the Cambridge Philosophical Society*, 35:405, 1939. doi: 10.1017/S0305004100021150.

- P. J. Humphrey, D. A. Buote, C. R. Canizares, A. C. Fabian, and J. M. Miller. A Census of Baryons and Dark Matter in an Isolated, Milky Way Sized Elliptical Galaxy. *ApJ*, 729:53, March 2011. doi: 10.1088/0004-637X/729/1/53.
- P. J. Humphrey, D. A. Buote, E. O’Sullivan, and T. J. Ponman. The ELIXr Galaxy Survey. II. Baryons and Dark Matter in an Isolated Elliptical Galaxy. *ApJ*, 755:166, August 2012. doi: 10.1088/0004-637X/755/2/166.
- Y. Ishisaki, Y. Maeda, R. Fujimoto, M. Ozaki, K. Ebisawa, T. Takahashi, Y. Ueda, Y. Ogasaka, A. Ptak, K. Mukai, K. Hamaguchi, M. Hirayama, T. Kotani, H. Kubo, R. Shibata, M. Ebara, A. Furuzawa, R. Iizuka, H. Inoue, H. Mori, S. Okada, Y. Yokoyama, H. Matsumoto, H. Nakajima, H. Yamaguchi, N. Anabuki, N. Tawa, M. Nagai, S. Katsuda, K. Hayashida, A. Bamba, E. D. Miller, K. Sato, and N. Y. Yamasaki. Monte Carlo Simulator and Ancillary Response Generator of Suzaku XRT/XIS System for Spatially Extended Source Analysis. *PASJ*, 59:113–132, January 2007. doi: 10.1093/pasj/59.sp1.S113.
- A. Jenkins, C. S. Frenk, S. D. M. White, J. M. Colberg, S. Cole, A. E. Evrard, H. M. P. Couchman, and N. Yoshida. The mass function of dark matter haloes. *MNRAS*, 321: 372–384, February 2001. doi: 10.1046/j.1365-8711.2001.04029.x.
- J. G. Jernigan. Direct n-body simulations with a recursive center of mass reduction and regularization. In J. Goodman and P. Hut, editors, *Dynamics of Star Clusters*, volume 113 of *IAU Symposium*, pages 275–283, 1985.
- S. Johnston, R. Taylor, M. Bailes, N. Bartel, C. Baugh, M. Bietenholz, C. Blake, R. Braun, J. Brown, S. Chatterjee, J. Darling, A. Deller, R. Dodson, P. Edwards, R. Ekers, S. Ellingsen, I. Feain, B. Gaensler, M. Haverkorn, G. Hobbs, A. Hopkins, C. Jackson, C. James, G. Joncas, V. Kaspi, V. Kilborn, B. Koribalski, R. Kothes, T. Landecker, E. Lenc, J. Lovell, J.-P. Macquart, R. Manchester, D. Matthews, N. McClure-Griffiths, R. Norris, U.-L. Pen, C. Phillips, C. Power, R. Protheroe, E. Sadler, B. Schmidt, I. Stairs, L. Staveley-Smith, J. Stil, S. Tingay, A. Tzioumis, M. Walker, J. Wall, and M. Wolleben. Science with ASKAP. The Australian square-kilometre-array pathfinder. *Experimental Astronomy*, 22:151–273, December 2008. doi: 10.1007/s10686-008-9124-7.
- L. R. Jones, T. J. Ponman, A. Horton, A. Babul, H. Ebeling, and D. J. Burke. The nature and space density of fossil groups of galaxies. *MNRAS*, 343:627–638, August 2003. doi: 10.1046/j.1365-8711.2003.06702.x.
- N. Katz, D. H. Weinberg, and L. Hernquist. Cosmological Simulations with TreeSPH. *ApJS*, 105:19, July 1996. doi: 10.1086/192305.
- H. G. Khosroshahi, B. J. Maughan, T. J. Ponman, and L. R. Jones. A fossil galaxy cluster: an X-ray and optical study of RX J1416.4+2315. *MNRAS*, 369:1211–1220, July 2006. doi: 10.1111/j.1365-2966.2006.10357.x.

- A. Klypin. Numerical Simulations in Cosmology I: Methods. *ArXiv Astrophysics e-prints*, May 2000.
- A. Klypin, G. Yepes, S. Gottlöber, F. Prada, and S. Heß. MultiDark simulations: the story of dark matter halo concentrations and density profiles. *MNRAS*, 457:4340–4359, April 2016. doi: 10.1093/mnras/stw248.
- A. A. Klypin and S. F. Shandarin. Three-dimensional numerical model of the formation of large-scale structure in the Universe. *MNRAS*, 204:891–907, September 1983. doi: 10.1093/mnras/204.3.891.
- A. A. Klypin, S. Trujillo-Gomez, and J. Primack. Dark Matter Halos in the Standard Cosmological Model: Results from the Bolshoi Simulation. *ApJ*, 740:102, October 2011. doi: 10.1088/0004-637X/740/2/102.
- A. Kolodzig, M. Gilfanov, R. Sunyaev, S. Sazonov, and M. Brusa. AGN and QSOs in the eROSITA All-Sky Survey. I. Statistical properties. *A&A*, 558:A89, October 2013. doi: 10.1051/0004-6361/201220880.
- E. Komatsu, K. M. Smith, J. Dunkley, C. L. Bennett, B. Gold, G. Hinshaw, N. Jarosik, D. Larson, M. R. Nolta, L. Page, D. N. Spergel, M. Halpern, R. S. Hill, A. Kogut, M. Limon, S. S. Meyer, N. Odegard, G. S. Tucker, J. L. Weiland, E. Wollack, and E. L. Wright. Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation. *ApJS*, 192:18, February 2011. doi: 10.1088/0067-0049/192/2/18.
- A. V. Kravtsov. On the Origin of the Global Schmidt Law of Star Formation. *ApJ*, 590: L1–L4, June 2003. doi: 10.1086/376674.
- A. Kundert, F. Gastaldello, E. D’Onghia, M. Girardi, J. A. L. Aguerri, R. Barrena, E. M. Corsini, S. De Grandi, E. Jiménez-Bailón, M. Lozada-Muñoz, J. Méndez-Abreu, R. Sánchez-Janssen, E. Wilcots, and S. Zarattini. Fossil group origins - VI. Global X-ray scaling relations of fossil galaxy clusters. *MNRAS*, 454:161–176, November 2015. doi: 10.1093/mnras/stv1879.
- L. D. Landau and E. M. Lifshitz. *Fluid mechanics*. 1959.
- G. Lemson and t. Virgo Consortium. Halo and Galaxy Formation Histories from the Millennium Simulation: Public release of a VO-oriented and SQL-queryable database for studying the evolution of galaxies in the LambdaCDM cosmogony. *ArXiv Astrophysics e-prints*, August 2006.
- W. P. Lin, Y. P. Jing, S. Mao, L. Gao, and I. G. McCarthy. The Influence of Baryons on the Mass Distribution of Dark Matter Halos. *ApJ*, 651:636–642, November 2006. doi: 10.1086/508052.

- Pangfeng Liu and Sandeep N Bhatt. Experiences with parallel n-body simulation. *Parallel and Distributed Systems, IEEE Transactions on*, 11(12):1306–1323, 2000.
- Leon B Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82:1013–1024, 1977.
- A. D. Ludlow, J. F. Navarro, M. Li, R. E. Angulo, M. Boylan-Kolchin, and P. E. Bett. The dynamical state and mass-concentration relation of galaxy clusters. *MNRAS*, 427:1322–1328, December 2012. doi: 10.1111/j.1365-2966.2012.21892.x.
- A. D. Ludlow, J. F. Navarro, R. E. Angulo, M. Boylan-Kolchin, V. Springel, C. Frenk, and S. D. M. White. The mass-concentration-redshift relation of cold dark matter haloes. *MNRAS*, 441:378–388, June 2014. doi: 10.1093/mnras/stu483.
- A. V. Macciò, A. A. Dutton, F. C. van den Bosch, B. Moore, D. Potter, and J. Stadel. Concentration, spin and shape of dark matter haloes: scatter and the dependence on mass and environment. *MNRAS*, 378:55–71, June 2007. doi: 10.1111/j.1365-2966.2007.11720.x.
- A. V. Macciò, A. A. Dutton, and F. C. van den Bosch. Concentration, spin and shape of dark matter haloes as a function of the cosmological model: WMAP1, WMAP3 and WMAP5 results. *MNRAS*, 391:1940–1954, December 2008. doi: 10.1111/j.1365-2966.2008.14029.x.
- U. Maio, K. Dolag, B. Ciardi, and L. Tornatore. Metal and molecule cooling in simulations of structure formation. *MNRAS*, 379:963–973, August 2007. doi: 10.1111/j.1365-2966.2007.12016.x.
- R. Mandelbaum, U. Seljak, and C. M. Hirata. A halo mass-concentration relation from weak lensing. *J. Cosmology Astropart. Phys.*, 8:006, August 2008. doi: 10.1088/1475-7516/2008/08/006.
- A. B. Mantz, S. W. Allen, and R. G. Morris. Cosmology and astrophysics from relaxed galaxy clusters - V. Consistency with cold dark matter structure formation. *MNRAS*, 462:681–688, October 2016. doi: 10.1093/mnras/stw1707.
- A. Marconi, G. Risaliti, R. Gilli, L. K. Hunt, R. Maiolino, and M. Salvati. Local super-massive black holes, relics of active galactic nuclei and the X-ray background. *MNRAS*, 351:169–185, June 2004. doi: 10.1111/j.1365-2966.2004.07765.x.
- A. M. Martin, E. Papastergis, R. Giovanelli, M. P. Haynes, C. M. Springob, and S. Stierwalt. The Arecibo Legacy Fast ALFA Survey. X. The H I Mass Function and $\Omega_{\text{H I}}$ from the 40% ALFALFA Survey. *ApJ*, 723:1359–1374, November 2010. doi: 10.1088/0004-637X/723/2/1359.

- T. P. K. Martinsson, M. A. W. Verheijen, K. B. Westfall, M. A. Bershad, D. R. Andersen, and R. A. Swaters. The DiskMass Survey. VII. The distribution of luminous and dark matter in spiral galaxies. *A&A*, 557:A131, September 2013. doi: 10.1051/0004-6361/201321390.
- N. J. McConnell and C.-P. Ma. Revisiting the Scaling Relations of Black Hole Masses and Host Galaxy Properties. *ApJ*, 764:184, February 2013. doi: 10.1088/0004-637X/764/2/184.
- M. McDonald, B. A. Benson, A. Vikhlinin, K. A. Aird, S. W. Allen, M. Bautz, M. Bayliss, L. E. Bleem, S. Bocquet, M. Brodwin, J. E. Carlstrom, C. L. Chang, H. M. Cho, A. Clocchiatti, T. M. Crawford, A. T. Crites, T. de Haan, M. A. Dobbs, R. J. Foley, W. R. Forman, E. M. George, M. D. Gladders, A. H. Gonzalez, N. W. Halverson, J. Hlavacek-Larrondo, G. P. Holder, W. L. Holzapfel, J. D. Hrubes, C. Jones, R. Keisler, L. Knox, A. T. Lee, E. M. Leitch, J. Liu, M. Lueker, D. Luong-Van, A. Mantz, D. P. Marrone, J. J. McMahon, S. S. Meyer, E. D. Miller, L. Mocuano, J. J. Mohr, S. S. Murray, S. Padin, C. Pryke, C. L. Reichardt, A. Rest, J. E. Ruhl, B. R. Saliwanchik, A. Saro, J. T. Sayre, K. K. Schaffer, E. Shirokoff, H. G. Spieler, B. Stalder, S. A. Stanford, Z. Staniszewski, A. A. Stark, K. T. Story, C. W. Stubbs, K. Vanderlinde, J. D. Vieira, R. Williamson, O. Zahn, and A. Zenteno. The Redshift Evolution of the Mean Temperature, Pressure, and Entropy Profiles in 80 SPT-Selected Galaxy Clusters. *ApJ*, 794:67, October 2014. doi: 10.1088/0004-637X/794/1/67.
- M. Meneghetti and E. Rasia. Reconciling extremely different concentration-mass relations. *ArXiv e-prints*, March 2013.
- M. Meneghetti, E. Rasia, J. Vega, J. Merten, M. Postman, G. Yepes, F. Sembolini, M. Donahue, S. Ettori, K. Umetsu, I. Balestra, M. Bartelmann, N. Benítez, A. Biviano, R. Bouwens, L. Bradley, T. Broadhurst, D. Coe, N. Czakon, M. De Petris, H. Ford, C. Giocoli, S. Gottlöber, C. Grillo, L. Infante, S. Jouvel, D. Kelson, A. Koekemoer, O. Lahav, D. Lemze, E. Medezinski, P. Melchior, A. Mercurio, A. Molino, L. Moscardini, A. Monna, J. Moustakas, L. A. Moustakas, M. Nonino, J. Rhodes, P. Rosati, J. Sayers, S. Seitz, W. Zheng, and A. Zitrin. The MUSIC of CLASH: Predictions on the Concentration-Mass Relation. *ApJ*, 797:34, December 2014. doi: 10.1088/0004-637X/797/1/34.
- J. Merten, M. Meneghetti, M. Postman, K. Umetsu, A. Zitrin, E. Medezinski, M. Nonino, A. Koekemoer, P. Melchior, D. Gruen, L. A. Moustakas, M. Bartelmann, O. Host, M. Donahue, D. Coe, A. Molino, S. Jouvel, A. Monna, S. Seitz, N. Czakon, D. Lemze, J. Sayers, I. Balestra, P. Rosati, N. Benítez, A. Biviano, R. Bouwens, L. Bradley, T. Broadhurst, M. Carrasco, H. Ford, C. Grillo, L. Infante, D. Kelson, O. Lahav, R. Massey, J. Moustakas, E. Rasia, J. Rhodes, J. Vega, and W. Zheng. CLASH: The Concentration-Mass Relation of Galaxy Clusters. *ApJ*, 806:4, June 2015. doi: 10.1088/0004-637X/806/1/4.

- C. A. Metzler and A. E. Evrard. A simulation of the intracluster medium with feedback from cluster galaxies. *ApJ*, 437:564–583, December 1994. doi: 10.1086/175022.
- M. J. Meyer, M. A. Zwaan, R. L. Webster, L. Staveley-Smith, E. Ryan-Weber, M. J. Drinkwater, D. G. Barnes, M. Howlett, V. A. Kilborn, J. Stevens, M. Waugh, M. J. Pierce, R. Bhathal, W. J. G. de Blok, M. J. Disney, R. D. Ekers, K. C. Freeman, D. A. Garcia, B. K. Gibson, J. Harnett, P. A. Henning, H. Jerjen, M. J. Kesteven, P. M. Knezek, B. S. Koribalski, S. Mader, M. Marquarding, R. F. Minchin, J. O’Brien, T. Oosterloo, R. M. Price, M. E. Putman, S. D. Ryder, E. M. Sadler, I. M. Stewart, F. Stootman, and A. E. Wright. The HIPASS catalogue - I. Data presentation. *MNRAS*, 350:1195–1209, June 2004. doi: 10.1111/j.1365-2966.2004.07710.x.
- J. J. Monaghan. Smoothed particle hydrodynamics. *ARA&A*, 30:543–574, 1992. doi: 10.1146/annurev.aa.30.090192.002551.
- J. J. Monaghan and R. A. Gingold. Shock Simulation by the Particle Method SPH. *Journal of Computational Physics*, 52:374–389, November 1983. doi: 10.1016/0021-9991(83)90036-0.
- C. Murali, N. Katz, L. Hernquist, D. H. Weinberg, and R. Davé. The Growth of Galaxies in Cosmological Simulations of Structure Formation. *ApJ*, 571:1–14, May 2002. doi: 10.1086/339876.
- T. Naderi, M. Malekjani, and F. Pace. Evolution of spherical overdensities in holographic dark energy models. *MNRAS*, 447:1873–1884, February 2015. doi: 10.1093/mnras/stu2481.
- R. Narayan and M. V. Medvedev. Thermal Conduction in Clusters of Galaxies. *ApJ*, 562:L129–L132, December 2001. doi: 10.1086/338325.
- J. F. Navarro, C. S. Frenk, and S. D. M. White. The Structure of Cold Dark Matter Halos. *ApJ*, 462:563, May 1996. doi: 10.1086/177173.
- J. F. Navarro, C. S. Frenk, and S. D. M. White. A Universal Density Profile from Hierarchical Clustering. *ApJ*, 490:493–508, December 1997. doi: 10.1086/304888.
- D. Nelson, A. Pillepich, V. Springel, R. Weinberger, L. Hernquist, R. Pakmor, S. Genel, P. Torrey, M. Vogelsberger, G. Kauffmann, F. Marinacci, and J. Naiman. First results from the IllustrisTNG simulations: the galaxy colour bimodality. *MNRAS*, 475:624–647, March 2018. doi: 10.1093/mnras/stx3040.
- A. F. Neto, L. Gao, P. Bett, S. Cole, J. F. Navarro, C. S. Frenk, S. D. M. White, V. Springel, and A. Jenkins. The statistics of Λ CDM halo concentrations. *MNRAS*, 381:1450–1462, November 2007. doi: 10.1111/j.1365-2966.2007.12381.x.

- F. R. Pearce, C. S. Frenk, A. Jenkins, J. M. Colberg, P. A. Thomas, H. M. P. Couchman, S. D. M. White, G. Efstathiou, J. A. Peacock, and A. H. Nelson. Cosmological galaxy formation. *ArXiv Astrophysics e-prints*, June 1999.
- P. J. E. Peebles. Structure of the Coma Cluster of Galaxies. *AJ*, 75:13, February 1970. doi: 10.1086/110933.
- P. J. E. Peebles. *The large-scale structure of the universe*. 1980.
- P. J. E. Peebles and R. H. Dicke. Origin of the Globular Star Clusters. *ApJ*, 154:891, December 1968. doi: 10.1086/149811.
- A. Pillepich, D. Nelson, L. Hernquist, V. Springel, R. Pakmor, P. Torrey, R. Weinberger, S. Genel, J. P. Naiman, F. Marinacci, and M. Vogelsberger. First results from the IllustrisTNG simulations: the stellar mass content of groups and clusters of galaxies. *MNRAS*, 475:648–675, March 2018. doi: 10.1093/mnras/stx3112.
- Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, F. Atrio-Barandela, J. Aumont, C. Baccigalupi, A. Balbi, A. J. Banday, and et al. Planck intermediate results. V. Pressure profiles of galaxy clusters from the Sunyaev-Zeldovich effect. *A&A*, 550:A131, February 2013. doi: 10.1051/0004-6361/201220040.
- D. Potter, J. Stadel, and R. Teyssier. PKDGRAV3: beyond trillion particle cosmological simulations for the next era of galaxy surveys. *Computational Astrophysics and Cosmology*, 4:2, May 2017. doi: 10.1186/s40668-017-0021-1.
- C. Power, J. F. Navarro, A. Jenkins, C. S. Frenk, S. D. M. White, V. Springel, J. Stadel, and T. Quinn. The inner structure of Λ CDM haloes - I. A numerical convergence study. *MNRAS*, 338:14–34, January 2003. doi: 10.1046/j.1365-8711.2003.05925.x.
- F. Prada, A. A. Klypin, A. J. Cuesta, J. E. Betancort-Rijo, and J. Primack. Halo concentrations in the standard Λ cold dark matter cosmology. *MNRAS*, 423:3018–3030, July 2012. doi: 10.1111/j.1365-2966.2012.21007.x.
- G. W. Pratt and M. Arnaud. XMM-Newton observations of three poor clusters: Similarity in dark matter and entropy profiles down to low mass. *A&A*, 429:791–806, January 2005. doi: 10.1051/0004-6361:20041479.
- G. W. Pratt, E. Pointecouteau, M. Arnaud, and R. F. J. van der Burg. The hot gas content of fossil galaxy clusters. *A&A*, 590:L1, May 2016. doi: 10.1051/0004-6361/201628462.
- W. H. Press and P. Schechter. Formation of Galaxies and Clusters of Galaxies by Self-Similar Gravitational Condensation. *ApJ*, 187:425–438, February 1974. doi: 10.1086/152650.

- D. J. Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231:759–794, February 2012. doi: 10.1016/j.jcp.2010.12.011.
- V. Quilis, J. M. A. Ibanez, and D. Saez. A Multidimensional Hydrodynamic Code for Structure Evolution in Cosmology. *ApJ*, 469:11, September 1996. doi: 10.1086/177753.
- T. Quinn, N. Katz, J. Stadel, and G. Lake. Time stepping N-body simulations. *ArXiv Astrophysics e-prints*, October 1997.
- A. Ragagnin, K. Dolag, V. Biffi, M. Cadolle Bel, N. J. Hammer, A. Krukau, M. Petkova, and D. Steinborn. A web portal for hydrodynamical, cosmological simulations. *Astronomy and Computing*, 20:52–67, July 2017. doi: 10.1016/j.ascom.2017.05.001.
- A. Ragagnin, K. Dolag, L. Moscardini, A. Biviano, and M. D’Onofrio. Dependency of halo concentration on mass, redshift and fossilness in Magneticum hydrodynamic simulations. *ArXiv e-prints*, October 2018.
- Antonio Ragagnin, Nikola Tchipev, Michael Bader, Klaus Dolag, and Nicolay Hammer. Exploiting the space filling curve ordering of particles in the neighbour search of gadget3. In Gerhard R. Joubert, Hugh Leather, Mark Parsons, Frans Peters, and Mark Sawyer, editors, *Parallel Computing: On the Road to Exascale*, volume 27 of *Advances in Parallel Computing*, pages 411–420. IOS Press, May 2016. doi: 10.3233/978-1-61499-621-7-411.
- E. Rasia, M. Meneghetti, and S. Ettori. X-ray Morphological Estimators for Galaxy Clusters. *The Astronomical Review*, 8(1):40–70, January 2013. doi: 10.1080/21672857.2013.11519713.
- R.-S. Remus, K. Dolag, T. Naab, A. Burkert, M. Hirschmann, T. L. Hoffmann, and P. H. Johansson. The co-evolution of total density profiles and central dark matter fractions in simulated early-type galaxies. *MNRAS*, 464:3742–3756, January 2017. doi: 10.1093/mnras/stw2594.
- J. Salmon, P. J. Quinn, and M. Warren. *Using parallel computers for very large N-body simulations: shell formation using 180 K particles.*, pages 216–218. 1990.

- A. Saro, J. Liu, J. J. Mohr, K. A. Aird, M. L. N. Ashby, M. Bayliss, B. A. Benson, L. E. Bleem, S. Bocquet, M. Brodwin, J. E. Carlstrom, C. L. Chang, I. Chiu, H. M. Cho, A. Clocchiatti, T. M. Crawford, A. T. Crites, T. de Haan, S. Desai, J. P. Dietrich, M. A. Dobbs, K. Dolag, J. P. Dudley, R. J. Foley, D. Gangkofner, E. M. George, M. D. Gladders, A. H. Gonzalez, N. W. Halverson, C. Hennig, J. Hlavacek-Larrondo, W. L. Holzappel, J. D. Hrubes, C. Jones, R. Keisler, A. T. Lee, E. M. Leitch, M. Lueker, D. Luong-Van, A. Mantz, D. P. Marrone, M. McDonald, J. J. McMahon, J. Mehl, S. S. Meyer, L. Mocanu, T. E. Montroy, S. S. Murray, D. Nurgaliev, S. Padin, A. Patej, C. Pryke, C. L. Reichardt, A. Rest, J. Ruel, J. E. Ruhl, B. R. Saliwanchik, J. T. Sayre, K. K. Schaffer, E. Shirokoff, H. G. Spieler, B. Stalder, Z. Staniszewski, A. A. Stark, K. Story, A. van Engelen, K. Vanderlinde, J. D. Vieira, A. Vikhlinin, R. Williamson, O. Zahn, and A. Zenteno. Constraints on the CMB temperature evolution using multiband measurements of the Sunyaev-Zel'dovich effect with the South Pole Telescope. *MNRAS*, 440: 2610–2615, May 2014. doi: 10.1093/mnras/stu575.
- J. Schaye, C. Dalla Vecchia, C. M. Booth, R. P. C. Wiersma, T. Theuns, M. R. Haas, S. Bertone, A. R. Duffy, I. G. McCarthy, and F. van de Voort. The physics driving the cosmic star formation history. *MNRAS*, 402:1536–1560, March 2010. doi: 10.1111/j.1365-2966.2009.16029.x.
- H. Shan, J.-P. Kneib, R. Li, J. Comparat, T. Erben, M. Makler, B. Moraes, L. Van Waerbeke, J. E. Taylor, A. Charbonnier, and M. E. S. Pereira. The Mass-Concentration Relation and the Stellar-to-halo Mass Ratio in the CFHT Stripe 82 Survey. *ApJ*, 840: 104, May 2017. doi: 10.3847/1538-4357/aa6c68.
- R. K. Smith, N. S. Brickhouse, D. A. Liedahl, and J. C. Raymond. Collisional Plasma Models with APEC/APED: Emission-Line Diagnostics of Hydrogen-like and Helium-like Ions. *ApJ*, 556:L91–L95, August 2001. doi: 10.1086/322992.
- V. Springel. The cosmological simulation code GADGET-2. *MNRAS*, 364:1105–1134, December 2005a. doi: 10.1111/j.1365-2966.2005.09655.x.
- V. Springel. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364:1105–1134, December 2005b. doi: 10.1111/j.1365-2966.2005.09655.x.
- V. Springel. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *MNRAS*, 401:791–851, January 2010. doi: 10.1111/j.1365-2966.2009.15715.x.
- V. Springel. Hydrodynamic simulations on a moving Voronoi mesh. *ArXiv e-prints*, September 2011.

- V. Springel and L. Hernquist. Cosmological smoothed particle hydrodynamics simulations: a hybrid multiphase model for star formation. *MNRAS*, 339:289–311, February 2003a. doi: 10.1046/j.1365-8711.2003.06206.x.
- V. Springel and L. Hernquist. The history of star formation in a Λ cold dark matter universe. *MNRAS*, 339:312–334, February 2003b. doi: 10.1046/j.1365-8711.2003.06207.x.
- V. Springel, S. D. M. White, G. Tormen, and G. Kauffmann. Populating a cluster of galaxies - I. Results at $z=0$. *MNRAS*, 328:726–750, December 2001a. doi: 10.1046/j.1365-8711.2001.04912.x.
- V. Springel, N. Yoshida, and S. D. M. White. GADGET: a code for collisionless and gasdynamical cosmological simulations. *New A*, 6:79–117, April 2001b. doi: 10.1016/S1384-1076(01)00042-2.
- V. Springel, T. Di Matteo, and L. Hernquist. Modelling feedback from stars and black holes in galaxy mergers. *MNRAS*, 361:776–794, August 2005a. doi: 10.1111/j.1365-2966.2005.09238.x.
- V. Springel, S. D. M. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. Navarro, R. Thacker, D. Croton, J. Helly, J. A. Peacock, S. Cole, P. Thomas, H. Couchman, A. Evrard, J. Colberg, and F. Pearce. Simulations of the formation, evolution and clustering of galaxies and quasars. *Nature*, 435:629–636, June 2005b. doi: 10.1038/nature03597.
- J. G. Stadel. *Cosmological N-body simulations and their analysis*. PhD thesis, UNIVERSITY OF WASHINGTON, 2001.
- L. K. Steinborn, K. Dolag, M. Hirschmann, M. A. Prieto, and R.-S. Remus. A refined sub-grid model for black hole accretion and AGN feedback in large cosmological simulations. *MNRAS*, 448:1504–1525, April 2015. doi: 10.1093/mnras/stv072.
- L. K. Steinborn, K. Dolag, J. M. Comerford, M. Hirschmann, R.-S. Remus, and A. F. Teklu. Origin and properties of dual and offset active galactic nuclei in a cosmological simulation at $z=2$. *MNRAS*, 458:1013–1028, May 2016. doi: 10.1093/mnras/stw316.
- M. Steinmetz and E. Müller. Galaxy formation in a cluster environment: first steps to a consistent description. In G. Klare, editor, *Astronomische Gesellschaft Abstract Series*, volume 8 of *Astronomische Gesellschaft Abstract Series*, page 16, 1993.
- A. F. Teklu, R.-S. Remus, K. Dolag, A. M. Beck, A. Burkert, A. S. Schmidt, F. Schulze, and L. K. Steinborn. Connecting Angular Momentum and Galactic Dynamics: The Complex Interplay between Spin, Mass, and Morphology. *ApJ*, 812:29, October 2015. doi: 10.1088/0004-637X/812/1/29.

- T. Theuns. Parallel P3M with exact calculation of short range forces. *Computer Physics Communications*, 78:238–246, January 1994. doi: 10.1016/0010-4655(94)90002-7.
- L. Tornatore, S. Borgani, V. Springel, F. Matteucci, N. Menci, and G. Murante. Cooling and heating the intracluster medium in hydrodynamical simulations. *MNRAS*, 342:1025–1040, July 2003. doi: 10.1046/j.1365-8711.2003.06631.x.
- L. Tornatore, S. Borgani, K. Dolag, and F. Matteucci. Chemical enrichment of galaxy clusters from hydrodynamical simulations. *MNRAS*, 382:1050–1072, December 2007. doi: 10.1111/j.1365-2966.2007.12070.x.
- K. Umetsu, A. Zitrin, D. Gruen, J. Merten, M. Donahue, and M. Postman. CLASH: Joint Analysis of Strong-lensing, Weak-lensing Shear, and Magnification Data for 20 Galaxy Clusters. *ApJ*, 821:116, April 2016. doi: 10.3847/0004-637X/821/2/116.
- V. Karakasis et al. EMEA IPCC user form meeting, Dublin, 2015.
- G. B. van Albada. Evolution of clusters of galaxies under gravitational forces. *AJ*, 66:590, December 1961. doi: 10.1086/108469.
- Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967. doi: 10.1103/PhysRev.159.98. URL <http://link.aps.org/doi/10.1103/PhysRev.159.98>.
- A. Voevodkin, K. Borozdin, K. Heitmann, S. Habib, A. Vikhlinin, A. Mescheryakov, A. Hornstrup, and R. Burenin. Fossil Systems in the 400d Cluster Catalog. *ApJ*, 708:1376–1387, January 2010. doi: 10.1088/0004-637X/708/2/1376.
- M. Vogelsberger, J. Zavala, and A. Loeb. Subhaloes in self-interacting galactic dark matter haloes. *MNRAS*, 423:3740–3752, July 2012. doi: 10.1111/j.1365-2966.2012.21182.x.
- M. S. Warren and J. K. Salmon. A portable parallel particle program. *Computer Physics Communications*, 87:266–290, May 1995a. doi: 10.1016/0010-4655(94)00177-4.
- M. S. Warren and J. K. Salmon. A parallel, portable and versatile treecode. In D. H. Bailey, P. E. Bjorstad, J. R. Gilbert, M. V. Mascagni, R. S. Schreiber, H. D. Simon, V. J. Torczon, and L. T. Watson, editors, *Parallel Processing for Scientific Computing*, 1995b.
- Michael S Warren and John K Salmon. A portable parallel particle program. *Computer Physics Communications*, 87(1):266–290, 1995.
- S. D. M. White. The dynamics of rich clusters of galaxies. *MNRAS*, 177:717–733, December 1976. doi: 10.1093/mnras/177.3.717.

- S. D. M. White and M. J. Rees. Core condensation in heavy halos - A two-stage theory for galaxy formation and clustering. *MNRAS*, 183:341–358, May 1978. doi: 10.1093/mnras/183.3.341.
- L. M. Widrow, D. Ryu, D. R. G. Schleicher, K. Subramanian, C. G. Tsagas, and R. A. Treumann. The First Magnetic Fields. *Space Sci. Rev.*, 166:37–70, May 2012. doi: 10.1007/s11214-011-9833-5.
- G. Xu. A new parallel N-Body gravity solver: TPM. *Astrophysical Journal Supplement Series*, 98:355, May 1995. doi: 10.1086/192166.
- D. G. York, J. Adelman, J. E. Anderson, Jr., S. F. Anderson, J. Annis, N. A. Bahcall, J. A. Bakken, R. Barkhouser, S. Bastian, E. Berman, W. N. Boroski, S. Bracker, C. Briegel, J. W. Briggs, J. Brinkmann, R. Brunner, S. Burles, L. Carey, M. A. Carr, F. J. Castander, B. Chen, P. L. Colestock, A. J. Connolly, J. H. Crocker, I. Csabai, P. C. Czarapata, J. E. Davis, M. Doi, T. Dombeck, D. Eisenstein, N. Ellman, B. R. Elms, M. L. Evans, X. Fan, G. R. Federwitz, L. Fiscelli, S. Friedman, J. A. Frieman, M. Fukugita, B. Gillespie, J. E. Gunn, V. K. Gurbani, E. de Haas, M. Haldeman, F. H. Harris, J. Hayes, T. M. Heckman, G. S. Hennessy, R. B. Hindsley, S. Holm, D. J. Holmgren, C.-h. Huang, C. Hull, D. Husby, S.-I. Ichikawa, T. Ichikawa, Ž. Ivezić, S. Kent, R. S. J. Kim, E. Kinney, M. Klaene, A. N. Kleinman, S. Kleinman, G. R. Knapp, J. Korienek, R. G. Kron, P. Z. Kunszt, D. Q. Lamb, B. Lee, R. F. Leger, S. Limmongkol, C. Lindenmeyer, D. C. Long, C. Loomis, J. Loveday, R. Lucinio, R. H. Lupton, B. MacKinnon, E. J. Mannery, P. M. Mantsch, B. Margon, P. McGehee, T. A. McKay, A. Meiksin, A. Merelli, D. G. Monet, J. A. Munn, V. K. Narayanan, T. Nash, E. Neilsen, R. Neswold, H. J. Newberg, R. C. Nichol, T. Nicinski, M. Nonino, N. Okada, S. Okamura, J. P. Ostriker, R. Owen, A. G. Pauls, J. Peoples, R. L. Peterson, D. Petravick, J. R. Pier, A. Pope, R. Pordes, A. Prosapio, R. Rechenmacher, T. R. Quinn, G. T. Richards, M. W. Richmond, C. H. Rivetta, C. M. Rockosi, K. Ruthmansdorfer, D. Sandford, D. J. Schlegel, D. P. Schneider, M. Sekiguchi, G. Sergey, K. Shimasaku, W. A. Siegmund, S. Smee, J. A. Smith, S. Snedden, R. Stone, C. Stoughton, M. A. Strauss, C. Stubbs, M. SubbaRao, A. S. Szalay, I. Szapudi, G. P. Szokoly, A. R. Thakar, C. Tremonti, D. L. Tucker, A. Uomoto, D. Vanden Berk, M. S. Vogeley, P. Waddell, S.-i. Wang, M. Watanabe, D. H. Weinberg, B. Yanny, N. Yasuda, and SDSS Collaboration. The Sloan Digital Sky Survey: Technical Summary. *AJ*, 120:1579–1587, September 2000. doi: 10.1086/301513.
- N. L. Zakamska and R. Narayan. Models of Galaxy Clusters with Thermal Conduction. *ApJ*, 582:162–169, January 2003. doi: 10.1086/344641.
- A. A. Zdziarski, J. Poutanen, and W. N. Johnson. Observations of Seyfert Galaxies by OSSE and Parameters of Their X-Ray/Gamma-Ray Sources. *ApJ*, 542:703–709, October 2000. doi: 10.1086/317046.

Y. B. Zel'dovich. Gravitational instability: An approximate theory for large density perturbations. *A&A*, 5:84–89, March 1970.

D. H. Zhao, Y. P. Jing, H. J. Mo, and G. Börner. Accurate Universal Models for the Mass Accretion Histories and Concentrations of Dark Matter Halos. *ApJ*, 707:354–369, December 2009. doi: 10.1088/0004-637X/707/1/354.

