TECHNICAL ASPECTS OF ILLUSTRIS- A COSMOLOGICAL SIMULATION

Dr Sheshappa SN[1], Megha S Kencha Reddy[2] , N Chandana[3], Ranjitha N[4], Sushma Fouzdar[5]

[1]*Faculty, Dept. of ISE, Sir MVIT, Karnataka, India, sheshappasn@gmail.com,*

[2]*VIII semester, Dept. of ISE, Sir MVIT*

Abstract - THE BIG QUESTION - Where did we come from and where are we going?Humans have been trying to figure out the origin of the known universe since the beginning of time and Since the early part of the 1900s, the Big Bang theory is the most accepted explanation and has dominated the discussion of the origin and fate of the universe. And the future of the universe can only be hypothesized based on our knowledge but the question does not seem to get irrelevant anytime in the near future. With the advancement technology and the available research, data and techniques like data mining and machine learning the problems can be visualised through accurate digital representations and simulations which are being used extensively to facilitate further developments in all fields of science.

Key Words: Illustris, Cosmological simulation, AREPO, Tree particle mesh, Subfind algorithm.

1. INTRODUCTION:

Modern astronomical surveys provide enormous amounts of observational data that confront our theories of structure and galaxy formation. Interpreting these observations requires accurate theoretical predictions. However, galaxy formation is a challenging problem due to its intrinsic multiscale and multiphysics character. Cosmological computer simulations are, hence, the method of choice for tackling these complexities when studying the properties, growth and evolution of galaxies. These simulations are extremely essential to understand the detailed workings of the structure and galaxy formation. Dark matter builds the backbone for structure formation and is therefore a key ingredient of all these simulations. In addition, dark energy is responsible for the accelerated expansion of the Universe and must also be considered. Despite the fact that the nature of dark matter and dark energy are not known, simulations can make detailed and reliable predictions for these dark components based on their general characteristics. Ordinary matter, such as stars and gas, contribute only about five percent to the energy budget of the Universe. Nevertheless, simulating this matter component is essential to study galaxies, but, unfortunately, it is also the most challenging aspect of galaxy formation. Recent simulations follow the formation of individual galaxies and galaxy populations from well-defined initial conditions and yield realistic galaxy properties. At the core of these simulations are detailed galaxy formation models. Of the many aspects these models are capable of, they describe the cooling of gas, the formation of stars, and the energy and momentum injection caused by supermassive black holes and massive stars. Nowadays, simulations also model the impact of radiation fields, relativistic particles and magnetic fields, leading to an increasingly complex description of the galactic ecosystem and the detailed evolution of galaxies in the cosmological context. Galaxy formation simulations have also become irreplaceable in cosmological studies. Cosmological simulations of galaxy formation hence provide important insights into a wide range of problems in astrophysics and cosmology.

2. LITERATURE SURVEY

Dylan Nelsona et al. [1] addressed scientific issues relevant for the interpretation of the simulations, which served as a pointer to publish and on-line documentation of the project and describes planned future additional data releases, and discussed technical aspects of the release.

Rainer Weinberger et al. [2] studied Time-integration which is performed by adopting local time-step constraints for each cell individually, solving the fluxes only across active interfaces, and calculating gravitational forces only between active particles, using an operator-splitting approach. This allows simulations with high dynamic range to be performed efficiently.

Volker Springel et al. [3] studied the simulation of the assembly of a massive rich cluster and the formation of its constituent galaxies in a flat, low-density universe.

Paul Bode et al. [4] tested the new TPM algorithm by comparing with results from Ferrell & Bertschinger'sP^{3}M code and found that, except in small clusters, the TPM results are atleast as accurate as those obtained with the well-established P^{3}M algorithm, while taking significantly less computing time.

Rainer Weinberger et al. [5] studied the population of supermassive black holes (SMBHs) and their effects on massive central galaxies in the IllustrisTNG cosmological hydrodynamical simulations of galaxy formation.

Mark Vogelsberger et al. [6] introduced the Illustris Project, a series of large-scale hydrodynamical simulations of galaxy formation.

Debora Sijacki[7] studied the properties of black holes and their host galaxies across cosmic time in the Illustris simulation.

3. DATA ACCESS

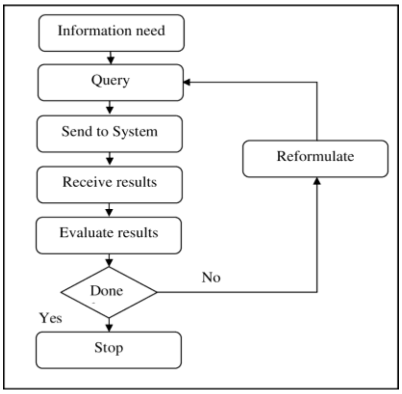

Fig -1: A simplified diagram of the standard model of the information access processes

The following are two complementary ways to access the Illustris data products [1].

1. Directly download the raw files and example scripts which can be provided as a starting point for local analysis.
2. A web-based API can be used, either through a web browser or programmatically in an analysis script, to perform common search and extraction tasks.

The above two approaches can be combined. For example, a user may be forced to download the full redshift zero group catalog in order to perform a complex search which is not supported by the API.

3.1 Direct File Download and Example Scripts

All primary data products are released in HDF5 format for Illustris. This is a portable, self-describing, binary specification suitable for large numerical datasets, for which file access routines are available in all common computing languages. We use only the basic features of the format: groups, attributes, and datasets, with one and two dimensional numeric arrays.

In order to maintain the reasonable file sizes, most outputs are split across multiple file "pieces"(or "chunks"). Individual links to each file chunk are available through the web-based API, and a snapshot can be downloaded in its entirety using a single wget command. Pre-computed sha256 checksums are provided for all files so that their integrity can be verified.

The provided example scripts [1] (in IDL, Python, and Matlab) give basic I/O functionality such as: (i) reading a given particle type and/or data field from the snapshot files, (ii) From the snapshot corresponding to a halo or subhalo, read the particle subset (iii) extracting the full subtree from either SubLink or LHaloTree for given subhalo, (iv) walking a tree to count the number of mergers, (v) read entire group catalog at one snapshot (vi) reading specific fields from the group catalog, or the entries for a single halo or subhalo,expecting they will serve as a useful starting point for writing any analysis task, and intend them as a 'minimal working examples' which are short and simple enough that they can be quickly understood and extended.

3.2 Web-based API

Web-based interface (API) have been implemented which can respond to a variety of user requests and queries. It is a well-defined interface between the user and the Illustris data products, which is expressed in terms of the required input(s) and expected output(s) for each type of request. The provided functionality is independent, as much as possible, from the underlying data structure, heterogeneity, format, and access methods. The API can be used in addition to, or in place of, the download and local analysis of large data files. At a high level, the API allows a user to search, extract, visualize, and analyze [1]. In every case, the goal is to reduce the data response size, either by extracting an unmodified subset, or by calculating a derivative quantity.

By specific example, the following types of requests can be handled through the current API, for any simulation at any snapshot:

- To List all the available simulations, their snapshots, and associated metadata.
- To list all objects and their properties in the Sub find group catalog .
- To search in the Subfind group catalogs with numeric range over any field .
- For a specific halo or subhalo, return all fields from the group catalog.
- Return a subset of this 'group cutout' containing only specified particle/cell type(s), and/or specific field(s) for each type.

• Traverse descendant and primary progenitor links across adjacent snapshots, as available in the SubLink merger trees, we recommend a user consult the up to date API reference available on the website [1].

3.3 API Access Details

Each API endpoint can return a response in one or more data types. When multiple options exist, a specific return format can be requested through one of the following methods [1].

• "(?format=)" indicates that the return type is chosen by supplying such a querystring, appended to the URL.

• "(.ext)" indicates that the return type is chosen by supplying the desired file extension in the URL.

3.4 Authentication

All API requests requires authentication, and therefore also user registration. Each request must provide, along with the details of the request itself, the unique "API Key" of the user making the request. A user can send their API key in the querystring, by appending it to the URL as:

• ?api key=d22d1f16b894a0b894ec31

A user can also alternatively send their API key in HTTP header. This is particularly useful for wget commands or within scripts [1]. Note that if a user is logged in to the website, then requests from the browser are automatically authenticated. Navigating the Browsable API works in this way.

4. AREPO – CODE

Arepo is a massively parallel gravity and magnetohydrodynamics code for astrophysics, designed for the large dynamic range of problems. It employs a finite-volume approach to discretize the equations of hydrodynamics on a moving Voronoi mesh, and a tree-particle-mesh method for gravitational interactions. It is originally optimized for cosmological simulations of structure formation, and also been used in many other applications in the astrophysics. It contains a finite-volume of magnetohydrodynamics algorithm on the unstructured, dynamic Voronoi tessellation coupled to the tree-particle-mesh algorithm for the Poisson equations on a Newtonian or cosmologically expanding spacetime. Time-integration is performed adopting local timestep constraints for each of the cells individually, solving the fixes only across active interfaces, and calculating gravitational forces only between active particles, using an operator-splitting approach. This allows simulations with high dynamic range to be performed efficiently. It is a massively distributed-memory parallel code, using the Message Passing Interface (MPI) communication which is standard and employing a dynamical work-load and memory balancing scheme to allow optimal use of multi-node parallel computers. The employed parallelization algorithms of Arepo are deterministic and produce the binary-identical results when re-run on the same machine and with the same number of MPI ranks. A simple primordial cooling and the star formation model is included as an example of the sub-resolution models commonly used in simulations of galaxy formation. It also contains a suite of computationally inexpensive test problems, ranging from idealized tests for automated code for the verification of the scaled-down versions of cosmological galaxy formation simulations, and is extensively documented in order to assist adoption of the code by new scientific users.

4.1 Computational methods

Over the past few decades, computer simulations of the evolution of the universe, including the effects of gravity, have reached a point of maturity and accuracy. Those which attempt to also include a treatment for gas, such as Illustris, have proven to be significantly harder. A number of fundamentally different methods exist for simulating gas on a computer. In astrophysics, most researchers have used one of the two approaches: (i) "smoothed particle hydrodynamics", or SPH, where the mass of the gaseous fluid is parceled out to a discrete number of particles. These particles move in response to the combined forces of gravity and hydrodynamics, and their position at any time indicates where the gas is, and how it is moving. (ii) The second approach of "Eulerian" or "mesh-based" methods, typically utilizing a scheme called "adaptive mesh refinement" or AMR [2]. In this method, space itself is divided up into a grid, and the flow of gas between neighboring cells of this grid is computed over time. The Illustris simulation uses a different approach, as implemented in the AREPO code, which we typically refer to as employing a "moving, unstructured mesh". Like in AMR, the volume of space is discretized into many individual cells, but as in SPH, these cells move with time, adapting to the flow of gas in their vicinity. As a result, the mesh itself, called a Voronoi tessellation of space, has no preferred directions or grid-like structure. Over the past few years we have shown that the new type of approach for simulating gas has significant advantages over the other two methods, particularly for large, cosmological simulations like Illustris. In addition to being accurate, the AREPO code is also efficient – it can run on tens of thousands of computer cores simultaneously, leveraging some of the largest computers that currently exist for scientific research within the "high performance computing" (HPC) community. The Illustris simulations were run on supercomputers in France, Germany, and the US. The largest was run on 8,192 compute cores, and took 19 million CPU hours .

4.2 Algorithms used

4.2.1 Particle Mesh Algorithm

Particle Mesh (PM) is a computational method for determining the forces in a system of particles. These particles could be atoms, stars, or fluid components and so the method is applicable to many fields, including molecular dynamics and also astrophysics. The basic principle is that a system of particles is converted into a grid (or "mesh") of density values. The potential is then solved for this density grid, and forces are applied to each particle based on what cell it is in, and where in the cell it lies. Various methods for converting a system of particles into a grid of densities exist. One method is that each particle simply gives its mass to the closest point in the mesh. Another method is the Cloud-in-Cell (CIC) method, where the particles are modelled as constant density cubes, and one particle can contribute mass to several cells.

4.2.1.1 TPM algorithm

1. Find the total density on a grid.
2. Based on the grid density, decompose the volume into a background PM volume and a large number of isolated high density regions. Every particle is then assigned to either the PM background or a specific tree.
3. Integrate the motion of the PM particles (those not in any tree) using the PM gravitational potential computed on the grid.
4. For each tree in turn, integrate the motion of the particles using a smaller time step if needed; forces internal to the tree are found with a tree algorithm ,added to the tidal forces from the external mass distribution taken from the PM grid [4].
5. Step global time forward, go back to step 1.

4.2.2 Friends-of-Friends algorithm

Algorithmic Description: The algorithm has a single free parameter which is called as the linking length. Any two particles which are separated by a distance less than or equal to the linking length are called " friends". The FOF group is then defined by the set of particles for which each particle within the set is connected to every other particle in the set through a network of friends.

Set FOF group counter j=1.
 For each of the particle, n, not yet associated with any group:
 Assign n to the group j, initialize a new member list, mlist, for the group j with particle n as first entry,
 Recursively, for each of the new particle p in mlist:
 Find the neighbors of p lying within a distance less than or equal to the linking length, add to mlist those not already assigned to group j,
 Record mlist for group j, set j=j+1.

4.2.3 SUBFIND algorithm

The algorithm identifies subhalos in each FOF group. SUBFIND is used to extract substructure, which we define as locally overdense, self-bound particle groups within a larger parent group [3]. The parent group will be a particle group pre-selected with a standard FOF linking length, although SUBFIND could operate on arbitrary particle groups, or with slight modifications on all of the particles in a simulation at once. Our algorithm tries to identify all locally overdense regions by imitating such a lowering of a global density threshold. To this end, we sort the finite number of particles according to their density, and we ‘rebuild’ the particle distribution by adding them in the order of decreasing density. Whenever a new particle i with density pi is considered, we find the Nngb nearest neighbours within the full particle set. Within this set Ai of Nngb particles, we also determine the subset of particles with density larger than pi, and among them we select a set Bi containing the two closest particles [3]. Rememb, that this set may contain only one particle, or it may be empty. We now consider three cases.
(i) The set Bi is empty, i.e., among the Nngb neighbours is no particle that has a higher density than particle i. In this case, particle i is considered to mark a local density maximum, and it starts growing a new subgroup around it.
(ii) If Bi contains a single particle, or two particles that are attached to the same subgroup, the particle i is also attached to this subgroup.
(iii) Bi contains two particles that are currently attached to different subgroups. In this case, the particle i is considered to be a saddle point, and the two subgroups labelled by the particles in Bi are registered as subhalo candidates. Afterwards, the particle i is added by joining the two subgroups to form a single subgroup.
Arepo has a number of additional features that influence practical aspects of how the code can be used in specific situations [2].

These are described in the following:-

On the fly structure and substructure finding:- Arepo includes a friends-of-friends (FOF) and a substructure identification algorithm that can be used both on-the- fly or in postprocessing.
First, a friends-of-friends algorithm is applied to the particle distribution to define groups as equivalence classes, where any pair of particles is in the same group if they are closer to each other than lFOF times the mean inter-particle separation. Next, these groups are subjected to substructure identification with the Subfind algorithm. To this end, the local density is estimated in an SPH-like approach at the position of all member particles, by using a smoothing length that encloses Nngb, sub (weighted) nearest neighbors. Then, an excursion set algorithm is used to identify locally overdense regions as substructure candidates. Each candidate is then treated with a gravitational unbinding

procedure which iteratively excludes all particles with positive total energy from the member list of the substructure candidate. The potential binding energy is calculated using a tree-algorithm analogous to the one used for the gravitational force calculation, with opening criterion Vsub. If a gravitationally bound set of particle survives that contains at least a minimum number of particles, the substructure is retained in the list of final subhalo [2].
The algorithm can detect nested sets of subhalos, but each particle/cell is counted only towards the mass of one substructure. Groups and subhalos are allowed to be much larger than the ones that fit into the memory of a single MPI rank. The particles and cells of a snapshot dump are automatically stored on disk such that particles belonging to individual (FOF) groups are grouped in the output as one block, in order of descending group size.

Within a given group, the particles are further sorted according to subhalo membership, again in descending order of subhalo size [2]. Finally, within a subhalo, the particles are sorted by their binding energy, with the particle with the lowest total energy coming first.This allows efficient random access to the particle data belonging to groups or subhalos even of very large simulation outputs.

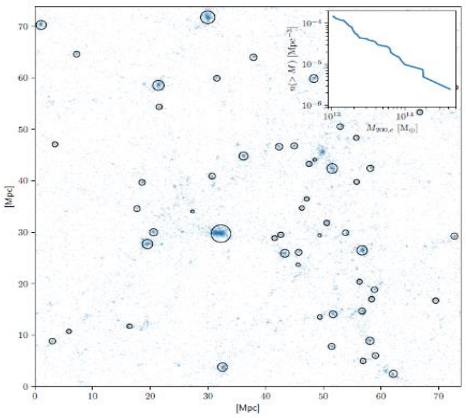

Fig -2: The cosmic large scale structure in a gravity-only cosmological simulation.

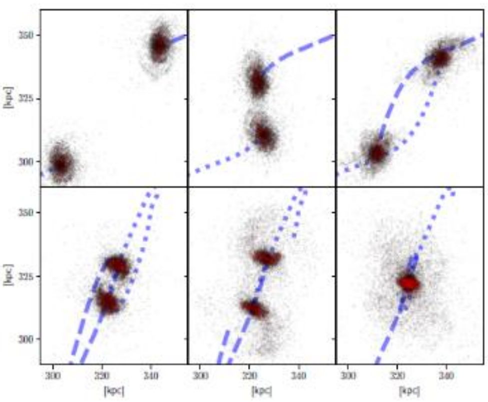

Fig -3:Time series of the position of existing (black) and newly formed (red) stars in a galaxy merger of two equal mass galaxies.

5. SPACE SIMULATORS

5.1 Space Engine

The Space Engine simulation is free to download and run on Windows PCs. It uses real astronomical data to recreate the universe, from planet Earth to distant galaxies. In patches where data is lacking, the program generates star systems and planets procedurally. Where Space Engine really shines is its transition at scale. Seamlessly fly from the craggy surface of an alien moon to the reaches of deep space. The option to move with inertia makes it feel like you're in a ship. Exploration is both satisfying and awe-inspiring.

5.2 Universe Sandbox

We all know that the most important force in the observable universe is gravity. This simulation lets you mess around with it using accurate Newtonian physics. Add a black hole next to Jupiter and watch it swallow the solar system. Blow up the moon and ruin everyone's day. Tweak the gravitational constant and send two galaxies hurtling toward each other.

5.3 Orbiter 2010

Orbiter is focused on the mechanics of space flight, rather than re-creating the entire universe. You can dock with space stations, deploy satellites and land on any planetary surface. The physics and timescale are purportedly accurate, and you can compress time if you don't have six years to spend on a trip to Saturn. Like Space Engine, this is a non-commercial project, and thus free to download and run on Windows PCs. Universe Sandbox lets you speed up time to watch your cosmic experiments unfold at a macro level.

5.4 The Solar System: Explore Your Backyard

The model can recreate the movement of the solar system at any point in time, past or present. Slice open a planet to see a cross-section of its core (or a theorized representation for those on which astronomers have no data). "Country mode" gives you info and borders of every country on Earth, and 88 constellations are represented by the artwork of 17th century astronomer Johannes Hevelius. An intuitive interface and beautiful design make this a great educational tool for six bucks.

5.5 Celestia

A veteran of the space simulation frontier, Celestia was originally released in 2001 and set the bar for scientifically accurate, open-universe exploration. You can fly between any of the celestial bodies in the Hipparcos Catalog of 118,322 stars. Scaling is fluid, and you can travel at speeds up to millions of light years per second, in case you need to hightail it out of the galaxy. View the known universe from any vantage, and grab screenshots and HD movies as you swing through nebulae.

5.6 WorldWide Telescope

WorldWide Telescope is really an astronomical map that overlays real images of the cosmos on a 3D environment. It's developed by Microsoft and utilizes imagery from both ground and space telescopes, including the Hubble. The program is even projected in some dome planetariums. It's free and also runs as a web client, provided you have the appropriate Silverlight browser plugins.

5.7 Kerbal Space Program

This one simulates a fictional universe, but it's widely hailed as a great educational game about astrophysics. Construct viable spacecraft that can overcome the effects of gravity and fuel consumption to get your little Kerbals into orbit. Sound easy? It's not. The position, size and quantity of components you add to your ship will determine whether it reaches the stratosphere or explodes on the scaffold. Timed rocket booster sequences and jettisoning depleted fuel cells are critical to success.

5.8 Illustris

The Illustris project is a large cosmological simulation of galaxy formation, completed in late 2013, using a state of the art numerical code and a comprehensive physical model. Building on several years of effort by members of the collaboration, the Illustris simulation represents an unprecedented combination of high resolution, total volume, and physical fidelity.

6. RESULT ANALYSIS

In 2014, a team of scientists amazed the world with a simulation of the universe from its birth to the present. The most detailed simulation of the universe to date, starting 12 million years after the big bang and tracing 13 billion years of cosmic evolution: Illustris. Having first confirmed that the cosmological model actually leads to the galaxy distribution that we see in space, the project went on to yield numerous discoveries-for instance about the properties of galaxies and the impact of supermassive black holes on cosmic structures. As it has been named it illustrated the formation of the universe in a gargantuan cube with a side length of 350 million lightyears-from the generation of the first hydrogen and helium atomic nuclei to the aggregation of these gases and the creation of the first supermassive black holes, galaxies and stars that form the cosmic web that we see today, with interlaced gas filaments, clusters of galaxies and vast voids in between. Illustris is a suite of large volume, cosmological hydrodynamical simulations run with the moving- mesh code AREPO and including a comprehensive set of physical models critical for following the formation and evolution of galaxies across cosmic time [2].

7. CONCLUSION

Previous simulations of the universe had either been limited in resolution or forced to focus on only a small portion of the universe. In contrast, Illustris was the first simulation able to reproduce the universe on both large and small scales simultaneously. Most importantly, it generated a mixed population of spiral, elliptical and irregular galaxies as we see them in our universe-something previous simulations had failed to do because of numerical inaccuracies and incomplete physical models. Illustris also accurately simulated certain characteristics of galaxies, such as their rotation speed or their metal and hydrogen content.

8. FUTURE WORK

Recently, Illustris was taken one step further under the new project name IllustrisTNG. This sequel project is even more ambitious, consisting of several different universe simulations that vary in size, resolution, and complexity of the physics included. Among them is a smaller version to Illustris named TNG50 — in a cube with a side length of 165 million light years. Because it is smaller, TNG50 can simulate the universe at even higher resolution, which allows researchers to track the morphology of galaxies in finer detail — down to star populations of several hundred solar masses, in fact. Also, hydrogen and helium gas turbulence and shock waves are much better resolved in TNG50. In contrast, the new TNG300 is a much larger simulation in a cube with a side length of almost 1 billion light years. This version allows observation of rare events and objects with great distance between them, such as very large galaxy clusters. The new set of simulations also includes more accurate and complete physics, most importantly the effects of baryonic particles on dark matter and the modelling of magnetic fields.

9. REFERENCES

[1] Dylan Nelson, Annalisa Pillepich, Shy Genel, Mark Vogelsberger, Volker Springel, Paul Torrey, Vicente Rodriguez-Gomez, Debora Sijacki, Gregory F. Snyder, Brendan Griffen, Federico Marinacci, Laura Blecha, Laura Sales, Dandan Xu, Lars Hernquist , "The Illustris Simulation: Public Data Release", arXiv:1504.00362v2[astro-ph.CO] 27 Oct 2015.

[2] Rainer Weinberger, Volker Springel, and Rudiger Pakmor, "The Arepo public code release", arXiv:1909.04667v1[astro-ph.IM] 10 sep 2019.

[3] Volker Springel, Simon D. M. White, Giuseppe Tormen and Guinevere Kauffmann, "Populating a cluster of galaxies – I. Results at z = 0", Mon. Not. R. Astron. Soc. 328, 726–750 (2001).

[4] Paul Bode, Jeremiah P. Ostriker, Guohong Xu, "The Tree–Particle–Mesh N-body Gravity Solver", Astrophys.J.Suppl. 128 (2000) 561-570.

[5] Rainer Weinberger, Volker Springel, Rüdiger Pakmor, Dylan Nelson, Shy Genel, Annalisa Pillepich, Mark Vogelsberger, Federico Marinacci, Jill Naiman, Paul Torrey, Lars Hernquist, "Supermassive black holes and their feedback effects in the IllustrisTNG simulation",
arXiv:1710.04659 [astro-ph.GA].

[6] Mark Vogelsberger, Shy Genel, Volker Springel, Paul Torrey, Debora Sijacki, Dandan Xu, Greg Snyder, Dylan Nelson, Lars Hernquist, "Introducing the Illustris Project: simulating the coevolution of dark and visible matter in the Universe", Monthly Notices of the Royal Astronomical Society (Volume: 444 , Issue: 2 , Aug. 2014).

[7] Debora Sijacki , Mark Vogelsberger, Shy Genel , Volker Springel , Paul Torrey, Greg Snyder, Dylan Nelson, Lars Hernquist, "The Illustris simulation: the evolving population of black holes across cosmic time", arXiv:1408.6842 [astro-ph.GA].