S.T.A.R.S. - Stellar Teaching and Astronomical Realism Simulator

Octavian-Mihai Matei
Technical University of Cluj-Napoca
Str. G. Barițiu 28, 400027,
Cluj-Napoca, România
mateitavis@gmail.com

Adrian Sabou
Technical University of Cluj-Napoca
Str. G. Barițiu 28, 400027,
Cluj-Napoca, România
adrian.sabou@cs.utcluj.ro

ABSTRACT
This paper describes the development of a physics-based simulation application for educa- tional purposes. We propose a solution that has the main aim to enhance the visual learning experience of students during Physics classes through the simulation of the behaviors of stars using Newtonian-based formulae in different scenarios. The target users will be high school students enrolled in the 9-12th grade in the physics classes. The users will be able to get real-time information about different statis- tics of real-world stars using the ESA GAIA database. This research paper presents the de- velopment steps, beginning with the conceptu- alization of the application's scope and con- text, it progresses through the mathematical background computations used in the applica- tion, the actual implementation, testing meth- ods used and the results found, validating its educational efficacy in engaging students with complex physics concepts through interactive games.

Author Keywords
Educational application; Physics; Unity; OpenCL

ACM Classification Keywords
H.5.2. User Interfaces

DOI: 10.37789/ icusi.2024.19

INTRODUCTION
The world's usage of electronic devices has steadily increased since the creation of the first personal computer and now, more rapidly, since the introduction of the smartphone. As a consequence, the adoption of the Internet has skyrocketed in the last few years, and as of April 2024, a total of 5.44 billion people, which accounts for around 67.1% of the Earth's population, have Internet access, as reported by the article [1] by Kepios, which states that a new "supermajority" was established for the first time in human history on Internet usage. As a consequence, this enables the majority of the population to have access to scientific discoveries and, at the same time, to educational applications and websites, which would further advance the development of humankind.

In the educational and scientific side of the Internet, the field of astronomical visual- izations has seen substantial advancements in recent years, driven by the increasing availability of real astronomical data and powerful computational tools provided by reputable in- stitutions in the field like National Aeronau- tics and Space Administration (NASA), and European Space Agency (ESA), the European Space Agency. In addition, these improvements are accelerated by the open-source data that those organizations release to a wider pub- lic. One such example is the three releases of data from the Global Astrometric Interferom- eter for Astrophysics (GAIA) experiment by ESA that enabled non-affiliated scientists all around the world to research a spectacularly sized database, containing detailed astrometric, photometric, and spectroscopic data for ap- proximately 1.8 billion stars in the Milky Way galaxy and beyond. In fact, the total size of Gaia DR3 data of around 1.5 petabytes (PB) and validated by third parties in various arti- cles, such as [2] provides the right environment for scientists and hobbyists to test numerous theories and property values of all those objects released. Furthermore, the scientific potential is huge, as seen in the proofs and releases in articles such as [3] or [4].

As a response to the rapid development of these educational astrophysics applications, this article will focus on the project enti- tled Stellar Teaching and Astronomical Real- ism Simulator, shortly S.T.A.R.S., which aims to contribute to this field by developing an application that integrates real astronomical data with interactive visual methods to ease the effort of physics teachers in high-schools in displaying and experimenting with different physics related scenarios in a controlled way.
The students would also highly benefit from this project, as they would be able to quickly vi- sualize all the aspects that the teacher presents during classes.

The S.T.A.R.S. project represents a signifi- cant step forward in astronomical simulations, offering a tool that combines accuracy, com- putational efficiency, and user-friendliness. By addressing the existing gaps in the field, this project has the potential to greatly enhance the educational experience and facilitate a deeper understanding of celestial mechanics. It will be tested by a variety of methods to ensure high- quality data visualization and full engagement from the users, including performance-oriented tests, user experience tests performed by high school teachers and students and an open beta session on public platforms to test the demand for such an application in the real world.

The main features of the application include the possibility of visualizing star information, such as the size, speed, spectral output, and many more properties, which are fundamen- tal attributes of an educational-focused project. Moreover, the ability of the teachers to modify the information guide ”on the fly” to accom- modate the student’s needs and the curriculum represents a step up to the traditional physics- oriented applications on the market. Another crucial feature is the capability to retrieve data from any source, provided it adheres to a spe- cific format. Currently, this is implemented us- ing GAIA R3 data, but it can be extended to other data sources that follow the same formatting, granting teachers and students un- precedented control over the simulation’s input data.

RELATED WORK IN EDUCATIONAL GAMES FOR PHYSICS

From the beginning of our species, people played games, either to distract them or just for fun, one thing is sure: games are as old as time and continue to play an essential part in our lives. One of the first pieces of writing that mention playing is in 345 B.C. in ”Plato Laws 1 & 2” [5] translated and commented by Susan Sauvé Meyer, where Plato describes playing as an essential development part of a child. There, he describes the usage of replica tools and equipment that would help a boy grow into his future occupation and become a good grown man. Furthermore, in his works, he also relates to the obligation of caretakers to tend to such playful activities, as riding with the children, playing war games with them etc.

Additionally, Elliot Avedon and Brian Sutton-Smith’s pioneering examination of ed- ucational games over time, ”The Study of Games” [6], focuses on games invented before the 1970s. The authors analyze how playing games affects people’s cognitive, psychological, and cultural development as well as how society standards are shaped by gaming. By studying games from ancient civilizations until the mid- twentieth century, the authors highlight the origins and history of educational games. This historical context allows us to better compre- hend how games continue to be useful teaching tools since it shows how flexible they may be to promote learning in a variety of cultural and chronological contexts. One fascinating exam- ple they discuss is how the ancient Greeks and Romans used games like Ludus latrunculorum and Astragaloi to teach strategic thinking and mathematical skills, as these games were not just for entertainment, they were actually de- signed to teach important knowledge and skills to the pupils. Moreover, in medieval Europe, games such as Chess were employed by edu- cational institutions to teach military tactics, moral values, and social etiquette.

In conclusion, a lot of studies argue that education through games is one of the oldest forms of learning, from Plato’s example of war games to the medieval times when chess was used as a simulation of military strategies, to today when the abstract physical properties of stars can be simulated in a user-friendly way. The hands-on method was approached by every important institution and with the clear rise in videogames, this trend doesn’t seem to slow down any time soon.

Physicus

A video-game called ”Physicus” was created to educate high school students fundamental physics principles, related to electricity, mag- netism, light and color. Its usefulness was ex- amined in a recent study titled ”Game-Based Learning of Physics Content: The Effectiveness of a Physics Game for Learning Basic Physics Concepts” [7]. This game creates an immer- sive virtual laboratory, which is a departure from standard classroom approaches. Students participate actively in this activity by moving objects around the game world to explore with basic physics concepts like force and gravity. In contrast to simulations, the game runs in real time, letting students see the results of their decisions right away. Compared to textbook graphics, this promotes a deeper comprehension of cause-and-effect linkages.

Cosmonium

Cosmonium[1] is a free, open-source program that

[1] Available at https://cosmonium.org

allows users to explore the universe in a virtual 3D environment. Unlike some astron- omy software that relies on static images, Cos- monium offers an interactive experience. Users can navigate the solar system, visiting planets, moons, and even venturing out to explore dis- tant galaxies. As they travel, they can zoom in on celestial bodies to get a closer look or zoom out to see the vastness of space. Moreover, with just a click, one can access detailed descriptions, physical properties, and interesting facts, such as the names and positions of constellations in the Milky Way galaxy as seen in Figure 1, which promotes a deeper understanding of the universe. This accessibility of knowledge en- hances the user experience, making exploration more engaging and meaningful. Cosmonium is a useful resource for our research because of its user-friendly design, which captivates peo- ple who are learning astronomy. Through an analysis of Cosmonium's effective integration of exploration and information acquisition, we gained important insights into the design prin- ciples of educational games centered around galactic simulations, which have the potential to excite and encourage scientific students.

Kerbal Space Program

Since its introduction in 2015, the highly regarded space flight simulation game "Kerbal Space Program" (KSP)[2] has fascinated play- ers and space enthusiasts community. With Squad's space program builder and manager, rocket designer and launcher, and realistic physics simulation, KSP lets users explore plan- ets and manage their own space program in the fictional star system of Kerbol. Space fans and even some real-world engineers have been drawn to KSP because of its realistic yet sim-

Figure 1. Cosmonium - Names and Constellations

ple orbital physics, which has garnered the game's devoted fan base. The impact of the game goes beyond simple enjoyment. In 2016, a NASA-funded initiative took advantage of KSP's widespread appeal to involve players in a virtual asteroid-sampling endeavor, as seen in Figure 2, as reported in Karl B. Hille's arti- cle "Gamers Tackle Virtual Asteroid-Sampling Mission" [8], this project demonstrates the po- tential of games like KSP not only to entertain but also to serve as valuable training and sim- ulation tools.

THEORETICAL CONCEPTS

This chapter will lay the foundation for the N-body simulation by exploring relevant com- putational techniques and the underlying physi- cal principles, as well as the theoretical problem related to simulating the galactic center of the Milky Way, which are critical for interpreting the results and understanding the limitations of our simulation environment.

N-Body Simulation

It is clear that, considering the number of objects that need to be simulated, there should be a streamlined method to compute the at- traction of planets and their movements. For this purpose, the two algorithms were analyzed side by side to decide what to use in the final product.

Naive Method

The main idea of this method is very straightforward, by directly computing the forces of attraction using Newton's law of gravi- tation for every pair of stellar objects we get the approximate values that we need. The main dis- advantage of this method is that it is computa- tionally expensive, leading to a Time Complex- ity of $O(N^2)$ and Space Complexity of $O(N)$.

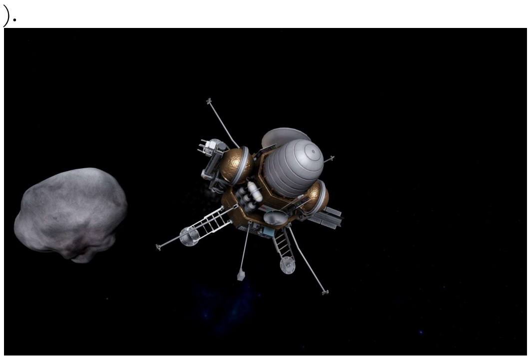

Figure 2. KSP - Asteroid Mission

[2] Available at https://www.kerbalspaceprogram.com

Fast Multipole Method

In galactic simulations, Fast Multipole Method (FMM) approximates the gravitational influence of distant groups of stellar objects us- ing multipole expansions, allowing for efficient and accurate force calculations.

Computational complexity was computed for the 2 most important features, Time Complexity of $O(N)$ or $O(N \log N)$, depending on the implementation and the specifics of the problem and Space Complexity is $O(N)$.

Fast Multipole Method versus Naive Method

Both algorithms were implemented in a ba- sic and simple way in Python to run simula- tions to determine, in a controlled way, the best method to be further taken into account to be implemented. There were 213 snapshots of performance for 10000 game objects with randomized initial positions, velocities and ac- celerations, which would be approximate to the scope of the final project. So the results of these tests are shown in Table 1 and contrary to the- oretical computations for time complexity, the Naive Method (List processing time) was more efficient, by having a median value of 27 versus the 94 from the Fast Multipole Method. This difference can also be seen in the mean values for the algorithms, where the Naive Method has lower bounds for the delay, but the stability of the FMM method cannot be contested by looking at Figure 3.

TABLE 1. FMM VS NAIVE STATISTICS

Statistical Method	Fast Multipole Method Delay[ms]	Naive Delay[ms]
Mean	95.38	69.77
Median	94	27
Minimum	90	26

The main reason why the Naive Method is more efficient in this particular case, even though it yielded a $O(N^2)$ theoretical complexity, is the movement of the main camera, alias the object from which the octree is spanned, so the whole tree needs to be rebalanced at almost every update tick in order to update the position and precision of each child tree. By looking at the density distribution of these two algorithms in Figure 4, we can see the two behaviors in action, where FMM has a constant processing time, in order to update the octree at every step, whereas the Naive Method has two areas of tendencies, depending on the size of data that needs to be computed, either com- putes the whole N sized set or the smaller and better to compute K sized set from the filtering function.

In conclusion, contrary to theoretical computations, the best method to use is the Naive one, as the number of game objects is still under the point of exhaustive processing delays for such simulations.

Galactic Simulation Physics Problem

Using the Universal Law of Attraction [9] between the Sun and Sagittarius A*:For the objects in the simulation to prop- erly orbit the galactic center, we would need to compute their mass by doing computations regarding the centripetal force, as well as the Universal Law of Attraction for the Newtonian-based physics that are integral to our application.

The centripetal force required to keep an object, in this case, the Sun in circular motion is given by:

$$F_c = \frac{m_{\text{sun}} v^2}{r} \quad (1)$$

Using the Universal Law of Attraction [9] between the Sun and Sagittarius A*:

$$F_{\text{gravity}} = \frac{G m_{\text{sun}} m_{\text{Sag A*}}}{r^2} \quad (2)$$

We know that the force of gravity in both equations 1 and 2 is equal, so we get:

$$m_{\text{sun}} \frac{v^2}{r} = \frac{G m_{\text{sun}} m_{\text{Sag A*}}}{r^2} \quad (3)$$

Solving for *m*Sag A*:

$$m_{\text{Sag A*}} = \frac{v^2 r}{G} \quad (4)$$

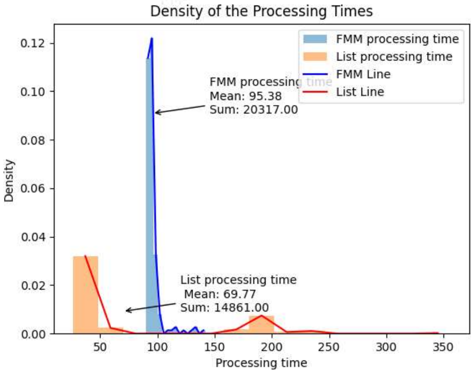

Figure 3. Data distribution of FMM vs Naive Method

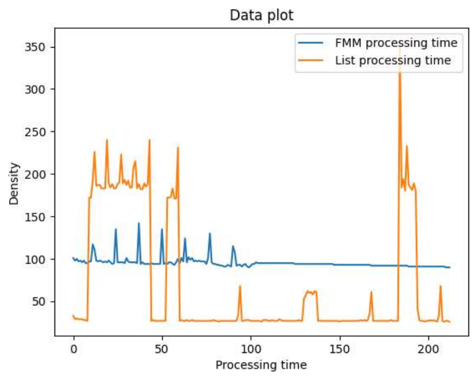

Figure 4.Density of FMM vs Naïve Method

We know the orbital velocity of the Sun around Sagittarius A* is approximately $2,3 \times 10^5$ m/s, as discussed in the article The Circular Velocity Curve of the Milky Way from 5 to 25 kpc [10], and the distance between the Sun and Sagitarrius A* is about 2.55×10^{20} meters, the mass of the Sun is 1.988×10^{30} kg. Plugging in these values in the Equation 4, we find the mass of Sagittarius A* required to keep the Sun in its orbit is 2.021×10^{41}.

In conclusion, Sagittarius A* would need to be approximately 2.021×10^{41} kg in the simulation, which means it is about $2.3655{\times}10^4$ times more massive than it was measured by third parties. This difference in mass is explained firstly by the usage of Newtonian-based equations, which provide a simpler and rougher view of celestial dynamics. Another reason is the assumption that the total mass of the galac- tic center is just the mass of Sagittarius A*, which, as reviewed in the article by V. V. Bobylev and A. T. Bajkova [11], is within the general consensus for the mass of the galactic center. For simplification reasons, we will use this computed value to act as the center of the simulation.

IMPLEMENTATION DETAILS

OpenCL (Open Computing Language) is an open standard supported by multiple hardware vendors, including AMD, Intel, and NVIDIA, as well as by various CPUs, GPUs, FPGAs, and other accelerators. This broad compatibility allows for greater flexibility and portability of ap- plications across different hardware platforms. The main operating flow is similar to other hardware acceleration APIs, such as CUDA, with some variations of the actual implemen- tation of the drivers and the main architecture. The main arguments for using this framework are:

1) **Open Source** - transparency and accessi- bility, as well as the possibility of personal development or tweaking of the framework for one's needs.
2) **Vendor-Neutral** - being able to run on diverse devices from different manufactur- ers without being tied to a specific vendor's ecosystem.
3) **Heterogeneous Programming Model** - the framework allows us to program for various processing units within a system, thus taking advantage of the strengths of CPUs, GPUs, FPGAs etc. for different tasks within a single application.

All things considered, OpenCL provides portability and flexibility, which makes it an important tool for developers who have to build code that works across several hardware platforms. This is highly beneficial for het- erogeneous computing environments and crossplatform apps. However, OpenCL tries to serve a wide range of devices, since it cannot be as closely tuned for the unique capabilities of any one platform as a solution created for a single hardware type.

The User Interface of the final application can be seen in Figure 5. It can be split into several regions that will be explained below in a counterclockwise manner:

Right top corner represents the scenario map that includes the camera frustrum,

Left top corner represents the player speed and FOV, field that changes shape and color according to the relativistic Doppler effect,

Left bottom corner represents the TIME speed of the simulation in millions of years per second,

The input field Stellar Search enables the user to search for a specific star and go to its location,

The button Wiki opens the wiki pages of the application which takes data dynami- cally from a JSON file,

One of the main problems with the initial design using the naive implementation of the simulation was the number of active objects in the memory. Two types of tests need to be performed, one before the object pooling and another after the implementation of the pool- ing. For both of these experiments, the system was left to stabilize and then the number of frames, *N*, was set to 15 and the speeds were set gradually in the set [0, 1, 5]. Following this procedure, the results show a net improvement of the delay in milliseconds, from an average of 220 ms delay before the object pooling implementation versus an average delay of 150.5 ms delay afterward, so a net decrease of 70.081 ms, or 31.76%, in the delay.

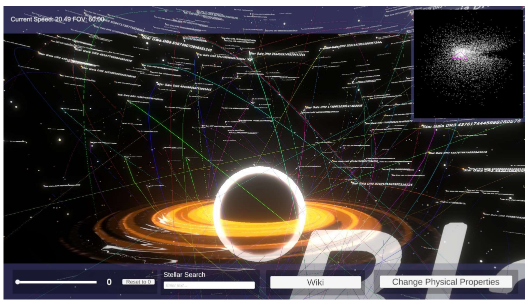

Figure 5. In-game Image

The output conditions were selected following the GAIA DATA RELEASE 3 documentation published by the Gaia Collaboration members [12]. This algorithm was derived from the methods and considerations discussed in the Articles [13], [14] for GAIA R1 and The Article "Gaia Data Release 2 - Using Gaia par- allaxes" [15] for Gaia R2.

TESTING

Functional Testing

Functional testing was used in the project to test how the application responds when the user performs an action that changes the state of the system. The testing was carried out to detect the errors in the project that appeared during the development of the application. This type of testing is a form of white-box testing, as it involves verifying the software's function- ality through the examination of the internal code structure during the development of the project.

Throughout the project, rigorous testing of the GAIA API was required. The availability of the data was extremely important. The main problem that we try to test and solve is the temperature of the stars which is not always available. For validating the velocities, we can use Newton's law of universal gravitation [9], Equation 5, and the formula for centripetal acceleration, Equation 6.

$$F = G\frac{m * m_{\text{center}}}{r^2} \tag{5}$$

$$a = (v^2)/r \tag{6}$$

$$v = \sqrt{\frac{G(m + m_{\text{center}})}{r}} \tag{7}$$

From Equation 5 and Equation 6, we get Equation 7, which is the formula for the speed of the object relative to the galactic center, that can be transformed easily in a 3D vectorial problem. Solving for each velocity, we get the expected velocities in a margin of 1% of the accepted error percentage.

User testing

The user testing was done to test the educa- tional value of this project. This type of testing is a form of black-box testing, as it involves verifying the software's functionality without examining the internal code structure.

The application was tested by a group of 18 students (ages 14-19) recruited from a high school in Romania. The participants had vary- ing levels of experience with astrophysics. They played the game and completed a Google form about their experience. The main objective of this endeavor was to test the efficiency and ap- plicability of such applications in real classroom environments. The questions were split into two categories, the first part is related to the current way of teaching astrophysics and the second part is a review of the given project.

The application can be deemed successful based on the responses to the questions posed, where the total score for each respondent is 60 points and the minimum satisfactory result is 30 points. As computed from the form results, the average points, per respondent is 46.78, as we can see in Figure 6, and the average per question is 7.79 out of 10. Most students thought the application would be very help- ful in their astrophysics classes since it would make it easier

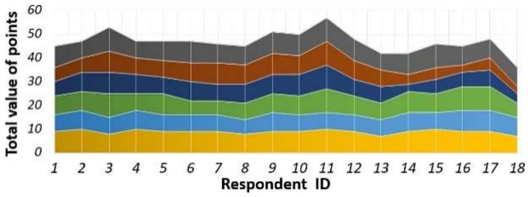

Figure 6. Form Results

for them to understand what the teacher was saying. This remark highlights how well the program works to improve learning by giving complicated astrophysical ideas interesting and understandable visual aids.

CONCLUSIONS

S.T.A.R.S. project made contributions to the field of astronomical simulations by devel- oping a robust application that integrates real astronomical data from the GAIA API with the ease of use of visual methods deployed in Unity3d.

The application successfully implemented essential features such as basic planetary move- ment and detailed information visualization, enhancing the educational and research poten- tial of the tool. The user interface was thought- fully designed, incorporating elements like a search bar for game objects, an orientation com- pass, and in-game menus, which collectively im- proved user interaction and overall experience. These foundational functionalities provided a solid basis for simulating and exploring astro- nomical phenomena interactively.

Following the development of the applica- tion, the results are satisfactory. The main objective of the application was achieved, by creating an application that can be used in a classroom by a teacher with students to explain Newtonian-based physics.

Furthermore, an explicit user interface was designed and implemented that guides the user to the most important parts of the application. The wiki page is the perfect example of a user- defined part of the application, where teach- ers and hobbyists alike can define their own concepts and present them in an easy-to-follow manner.

Future improvements include the addition of support for planets which would be a signifi- cant advancement in enriching the educational experience. Moreover, the implementation of multiple scenarios would enhance the user ex- perience, such as our star system, to visualize the planets, moons and mechanics that play a crucial role in our existence, a simulation that includes more stars or a different sets of stars for the end user to explore and learn

References

1. S. Kemp, "Digital around the world," https://datareportal.com/global-digital-overview, Kepios, n.d., accessed: May 2024.
2. C. Babusiaux, C. Fabricius, S. Khanna, T.

Mu- raveva, C. Reylé, F. Spoto, A. Vallenari, X. Luri,
F. Arenou, M. Alvarez et al., "Gaia data release 3- catalogue validation," Astronomy & Astrophysics, vol. 674, p. A32, 2023.

3. R. J. Jennings, D. L. Kaplan, S. Chatterjee, J. M. Cordes, and A. T. Deller, "Binary pulsar distances
and velocities from gaia data release 2," The Astro- physical Journal, vol. 864, no. 1, p. 26, 2018.
4. A. Tanikawa, K. Hattori, N. Kawanaka, T. Kinu- gawa, M. Shikauchi, and D. Tsuna, "Search for a black hole binary in gaia dr3 astrometric binary stars with spectroscopic data," The Astrophysical Journal, vol. 946, no. 2, p. 79, 2023.
5. S. S. Meyer, Plato: Laws 1 and 2. OUP Oxford, 2015.
6. E. M. Avedon and B. Sutton-Smith, "The study of games," (No Title), 1971.
7. A. Foster, M. Koehler, and P. Mishra, "Game- based learning of physics content: the effectiveness of a physics game for learning basic physics con- cepts," in World conference on educational mul- timedia, hypermedia and telecommunications Vol- ume: 1, 01 2006, pp. 2119–2125.
8. K. B. Hille, "Gamers tackle vir- tual asteroid-sampling mission," June 2016, accessed: 2024. [Online]. Avail- able: https://www.nasa.gov/solar-system/gamers-tackle-virtual-asteroid-sampling-mission/
9. I. Newton, Philosophiae Naturalis Principia Math- ematica ("Mathematical Principles of Natural Phi- losophy"). London: Royal Society, 1687, also pub- lished in Cambridge, 1713; London, 1726.
10. A.-C. Eilers, D. W. Hogg, H.-W. Rix, and M. K. Ness, "The circular velocity curve of the milky way from 5 to 25 kpc," The Astrophysical Journal, vol. 871, no. 1, p. 120, jan 2019. [Online]. Available: https://dx.doi.org/10.3847/1538-4357/aaf648
11. V. Bobylev and A. Bajkova, "Review of current estimates of the galaxy mass," Publications of the Pulkovo Observatory, vol. 228, p. 1–20, May 2023. [Online]. Available: http://dx.doi.org/10. 31725/0367-7966-2023-228-3
12. Gaia Collaboration, Vallenari, A., Brown, A. G. A., Prusti, T., and d. et al., "Gaia data release 3
- summary of the content and survey properties," A&A, vol. 674, p. A1, 2023. [Online]. Available: https://doi.org/10.1051/0004-6361/202243940
13. O. Malkov, D. Kovaleva, A. Zhukov, and O. Dluzh- nevskaya, "Theoretical mass-luminosity relations in gaia g-band," Astrophysics and Space Science, vol. 367, no. 4, April 2022. [Online]. Available: http://dx.doi.org/10.1007/s10509-022-04066-1
14. M. Delbo, J. Gayon-Markt, G. Busso, A. Brown, L. Galluccio, C. Ordenovic, P. Bendjoya, and P. Tanga, "Asteroid spectroscopy with gaia," planss, vol. 73, pp. 86–94, 12 2012.
15. Luri, X., Brown, A. G. A., Sarro, L. M., Arenou, F., Bailer-Jones, C. A. L., Castro-Ginard, A., de Bruijne, J., Prusti, T., Babusiaux, C., and Delgado, H. E., "Gaia data release 2 - using gaia parallaxes," A&A, vol. 616, p. A9, 2018. [Online]. Available: https://doi.org/10.1051/0004-6361/201832964