

Team Andromeda

Version 1

Requirements Specification

November 25, 2019

Accepted as baseline requirements for the project:

For the client: _____ _____ For the team: _____ _____
Signature Date Signature Date

_____ _____
Signature Date

Clients - Dr. Audrey Thirouin and Dr. Will Grundy

Mentor - Isaac Shaffer

Members - Matthew Amato-Yarbrough, Batai Finley, Bradley Kukuk, John Jacobelli, and Jessica Smith

Table of Contents

1. Introduction	1
2. Problem Statement	2
3. Solution Vision	3
4. Project Requirements	5
5. Potential Risks	14
6. Project Plan	18
7. Conclusion	19

1. Introduction

Since the time of Galileo, humans have been relentlessly studying the universe. Currently, billions of dollars are spent every year on sending probes to other planets and small bodies in a desperate attempt to understand what lies in the cosmos. This continuous research has driven technological development in areas not directly related to space. For example, home insulation, baby formula, and portable computers are a few of the side results discovered through space exploration. Other valuable data has been derived from space exploration as well, including lesser known phenomenon such as early planet formation and creation hypotheses. Many observatories like Lowell are gathering information daily to further our comprehension of the space that surrounds humanity.

Our clients Dr. Audrey Thirouin and Dr. Will Grundy work for Lowell Observatory. Dr. Thirouin is a research scientist interested in the characteristics of small bodies within the Solar System, while Dr. Grundy is an astronomer that researches Kuiper Belt objects. Kuiper Belt objects are a region of leftover bodies from the solar system's early history, making them valuable for observing conditions similar to that of early planet formation. Together, Dr. Thirouin and Dr. Grundy focus on collecting and analyzing data about small bodies that lay far from Earth and are hard to directly observe. Currently, they are working on modeling binary systems in the Kuiper Belt and need to use techniques which make the most use of the data available to achieve this. For distant objects in the solar system, only a pinpoint source is observable which can be used to determine the brightness of the object at a given point in time. These luminosity recordings can be combined to form a light curve, which is a graph of brightness values over a given time.

Light curves can be used to interpret properties about the objects that they are derived from. For instance, an asteroid that is non-spherical will reflect more light when a larger portion of its surface is facing an observer. This is because more light from the Sun is being reflected to said observer. Since the object is reflecting more light at certain points in its rotation, the brightness will differ depending on when it is observed in its rotation. The various brightness values can then be graphed and, in most cases, this will generate a wave graph that represents the light curve. Information like rotational speed and rough proportions can then be found based on graph data such as the period or amplitude of the curve. A broad number of other characteristics can be found using light curves by an adept astronomer.

Our clients wish to use light curves to better understand binary systems. A binary system is composed of two objects that orbit a common barycenter, which is a point in space that represents the center of mass of two or more orbiting bodies. The gravity from both bodies affect the other and can cause unique situations, such as tidal locking or precession. Binary systems introduce new challenges while also offering opportunities for light curve modeling.

Dr. Thirouin and Dr. Grundy can model binary systems using spheres and faceted objects using the previous iteration of the project, which was developed by Paired Planet Technologies. Our clients need new software to improve the application by including triaxial ellipsoids, a Markov chain Monte Carlo (MCMC) algorithm, a graphical user interface (GUI), and a video generator. The solution for this iteration of the project will allow for more accurate modeling of binary systems and the generation of estimated parameters of observed light curves.

2. Problem Statement

2.1 Current Workflow

Our clients use the implemented solution to model binary systems by inputting the parameters into an IDL command line. From here they choose to either use spheres or faceted objects to represent the binary system they are simulating. After this is done, the Forward Model is executed and returns a simulated light curve of the observed system. Depending on the predicted light curve produced, they will either adjust parameters originally input or keep the produced predicted light curve.

2.2 Current Workflow Issues

Currently, the clients are facing a host of issues that are disruptive to their workflow and limit the capabilities with which they can model binary systems. They need the addition of new modules to strengthen the software solution they currently have implemented. These modules will address the client's following issues:

- Inability to represent a triaxial ellipsoid-shaped body
- Parameter input into the command line is inefficient
- Predicted parameter estimates are done by hand
- No option to produce videos from rendered images

The current solution can only use spheres or faceted objects to represent binary systems, which in turn limits our clients to two choices. They can either choose to render binary systems quickly with low accuracy using spheres, or more accurately with longer render times using faceted objects. Additionally, command line input makes for a needlessly arduous and time consuming data entry process.

Furthermore, the workflow currently lacks an efficient means to determine the best fitting parameters for an observed light curve. To determine these parameters, our clients need to manually perform the operations that a MCMC algorithm is capable of. This again further impedes the workflow of our clients, adding a significant amount of overhead to what could otherwise be an automated process. Lastly, without a means to take rendered images and compile them into a video format, our clients lack the ability to visualize the movement of the binary systems being simulated without manually doing so.

3. Solution Vision

3.1 Solution Overview

Our solution will address our clients' problems by using the current API, frameworks, and math libraries to improve performance. A framework will be used to provide a GUI to facilitate the entry of parameters for the Forward Model. Additionally, the GUI will allow for the option to compile rendered images into a video format. The solution will have a Shape subclass for the implementation of triaxial ellipsoids. Lastly, the solution will use an HMC API to handle the implementation of the HMC algorithm.

The API will be well documented to allow users to maintain and update the code base. The design of the API will be modular to allow for new functionality to be added in the future.

3.2 Solution Statement

Due to the requirements given and the existing code base, the software that will be developed must be compatible with the existing code base. The existing code base was developed using C/C++. With that in mind the software will be built using C/C++ and Python.

For the GUI implementation for the Forward Model, the programming language we will be using is Python due to its compatibility with C/C++. The framework that will be used to develop the GUI will be Kivy, a Python framework that is used to create cross platform compatible software. This will allow the user to use this software anywhere and at any time.

For the acceleration of the Forward Model we will be creating a Triaxial Ellipsoid class to the existing API using C/C++. The Triaxial Ellipsoid class will be dependant on the existing API's library and will use all of the same libraries.

The API will be well documented to allow users to maintain and update the code base. The design of the API will be modular to allow for new functionality to be added in the future.

3.3 Solution Steps

The solution for GUI implementation is to wrap the existing API in a GUI. The clients want the interface to take in parameters for the Forward Model so that it can call the Forward model, and generate a light curve to be displayed to the screen. The GUI that will be developed will need to be able to do the following:

1. Run the Forward Model
2. Take in input parameters and accuracy settings
3. Display the generated light curve
4. Allow different Forward Model settings

The solution for Forward Model Acceleration is to create a Triaxial Ellipsoids class. The clients want a Triaxial Ellipsoid class that can be used to calculate and render objects more accurately so that the Forward Model can more accurately model what the observed object looks like. The Triaxial Ellipsoids implementation that will be developed will need to be able to do the following:

1. Calculate rotation
2. Evaluate Spin State
3. Render Objects

The solution for the HMC algorithm implementation is to wrap the Forward Model with a HMC algorithm. The clients want the best range of parameters that can be used for the Forward Model, so they can generate more accurate light curves. The Hamiltonian Monte Carlo algorithm that will be developed will need to be able to do the following:

1. Produce range of likely parameters for Forward Model
2. Will take in initial conditions and observed data

4. Project Requirements

4.1 Functional Requirements

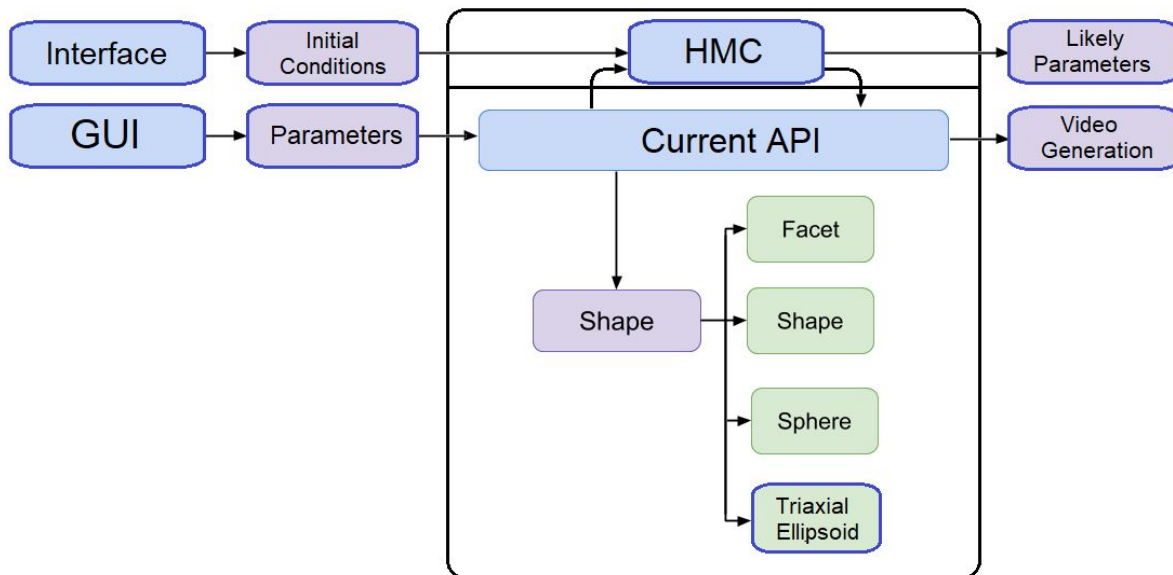


Figure 1: Envisioned Forward Model

There is one major use case that must be satisfied for this project which is the Forward Model. A simplified, modified version only showing existing modules relevant to our work this year is displayed in Figure 1 above. The portions of the Forward Model that have a blue outline are what our team plans to implement, with everything else being implemented by last year's team.

Our changes to the Forward Model involves 3 major steps: implementing the Hamiltonian Monte Carlo algorithm, rendering and calculating the light curves of triaxial ellipsoids, and improving user input and model output through the addition of a GUI and video generation. These extensions will be used to improve the current Forward Model by allowing for a greater ease of use for our clients.

FR1. Simulate Triaxial Ellipsoids

The software must be capable of performing calculations for the Forward Model using triaxial ellipsoids. It must also be able to render a triaxial ellipsoid object when the user requests so when calling the Forward Model. This output will be generated through the use of the already implemented ray tracing feature.

Additionally, this output must be compatible with the modifications that the user may request of other implemented objects by way of input parameters. These adjustments to the output include the ability to adjust the resolution or pixel count of the image. The requirements of simulation and calculations of triaxial ellipsoids can be broken down into four sections:

1. Calculate light curve using triaxial ellipsoids

The current iteration of the API takes in a given amount of parameters to calculate the light curve for a system, including:

- Ephemeris table
- Keplerian orbital elements
- Time value(s)
- Object shape parameters
- Spin states
- Hapke parameters
- Optional accuracy setting

The software must be capable of performing calculations for the Forward Model using triaxial ellipsoids. All of these parameters work in conjunction to calculate the forward model, and provide a light curve of a system. The triaxial ellipsoid object will need to be compatible with all the other parameters that the API utilizes in order to calculate a light curve.

Rendering images of the system during these calculations is also possible. This means that a triaxial ellipsoid object must be renderable by the software so that the system can be visualized if necessary.

2. Render triaxial ellipsoids as objects

The software must be able to render a triaxial ellipsoid object when the user requests so when calling the Forward Model. As the ellipsoid object will be used by the Forward Model, its model output will be generated through the use of the already implemented ray tracing feature.

The current program simulates objects, meaning that it will also be required of triaxial ellipsoids. Rendering the triaxial ellipsoids will allow users to visualize how the objects are interacting with each other. Ray tracing is the method currently used to render objects in the program. Spheres and faceted objects are both composed of triangular facets in order to utilize ray tracing. Similar to these objects, triaxial ellipsoids will need to be composed of triangular facets to be renderable.

3. Allow for manipulation of dimensions

Triaxial ellipsoids describe a large group of shapes. Spheres that have been shortened or lengthened along their x, y, and/or z axis in a 3D space are considered to be a triaxial ellipsoid. Being able to modify the value of triaxial ellipsoid's axes is key to implementing these shapes and allowing the user to gain the most from them. These parameters must be as accessible and simple to change as they are for currently implemented objects.

As the ellipsoid object will be used by the Forward Model, its model output will be generated through the use of the already implemented ray tracing feature. This output must be modifiable by the user in the same way that the user may manipulate the output of currently implemented objects via input parameters. These adjustments to the output include the ability to adjust the resolution or pixel count of the image.

4. Rotate object

The sphere class that is implemented in the current program does not have to consider rotation, as a sphere is uniform throughout. Triaxial ellipsoids are not uniform throughout, and therefore must consider the possible fluctuations in reflectance of the shape itself. An orientation function must be implemented for the shape to account for these fluctuations, and to allow for the model to accurately represent the shape.

The rotation will be the most difficult part of the shape to implement due to the math that is needed to calculate it. This will be accomplished through matrix multiplication that will update the ellipsoid object depending on what orientation is required.

FR2. Produce Best Fit Parameters Using HMC

A key requirement of our project is the implementation of an HMC algorithm. This algorithm will serve the purpose of finding the best set of parameters that our clients are interested in testing. This is done in order to determine the best fit parameters of an observed light curve. Additionally, the algorithm will show how well our clients have pinned down the values that they are interested in, and visualize this cloud of solutions using a model. Lastly, the algorithm will save the solutions produced to an external file. The requirements of the algorithm are explored in further detail in the following sections.

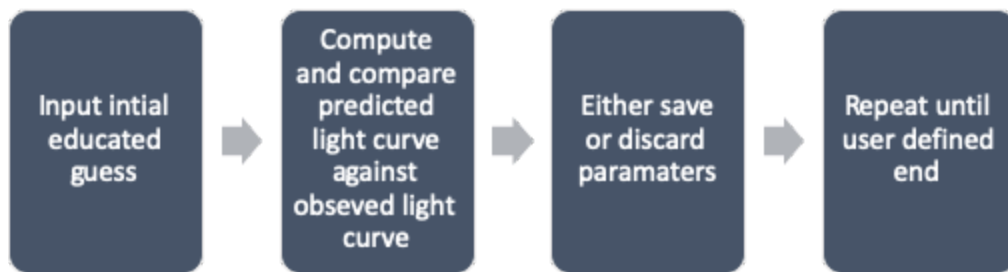


Figure 2: Steps of the HMC Algorithm

1. Produce likely parameters that fit the observed data

The main functionality of the HMC algorithm will be to produce parameters that best fit the observed data collected by our clients. The algorithm accomplishes this by inputting an educated guess into the Forward Model, which in turn produces a predicted light curve that is then compared to the observed light curve. If the comparison falls within a certain range of acceptability, then the parameters used to create the predicted light curve are saved as a possible set of solutions. Otherwise, the parameters are discarded and a new educated guess is made. This process is repeated, with each set of tested parameters being determined by the previous solution. This process continues until a user defined end. The steps of the algorithm are laid out in Figure 2 above.

Moreover, our clients asked us to include the functionality of annealing within the HMC algorithm in order to disregard produced solutions that are too extreme to fit the solution space that our clients are concerned with. They asked us to include this functionality because the produced extreme solutions would affect the usability of the data created by the algorithm.

2. Display the range of solutions using a model

Using the model produced by the HMC algorithm, our clients will be able to observe the relationship between the parameters that were input into the algorithm and the possible solutions determined by

the algorithm. Our clients suggested a Corner Plot model to represent our data. It is a type of model that is used to visualize multidimensional samples, such as those produced by an HMC algorithm, using a scatterplot matrix. The points that make up the Corner Plot would be possible solutions that match the data of the observed light curve, to a degree of error.

3. Saving the solutions to an external file

Our clients would also like for the HMC algorithm to produce a file, such as a .csv file, that would contain the raw data produced by the HMC algorithm. If we imagine the layout of the file as a table, the number of columns would be equal to the number of variables that were sampled from, and the lines would be equal to the number of solutions produced. This information can be used in testing outside of the algorithm.

FR3. Improved GUI

A key requirement that we have is the implementation of a new GUI for the forward model. This GUI will make the Forward Model more user friendly and allow the user to enter parameters from the interface rather than entering them at the command line. Additionally it will allow the user to see the predicted light curve directly on the screen where the user will be able to compare the predicted light curve to the observed light curve.

1. Input to Forward Model

The GUI must allow the user to enter in variables from the interface. There will be text boxes that will take in a specific type of input, and if the input type is invalid, the GUI will prompt the user to put in an appropriate input. Once all variables are entered, there will be a button that will allow the user to run the Forward Model with the input parameters.

Due to the large amount of parameters to be input for the Forward Model, there will be a scroll wheel that will allow the user to navigate through the parameter list faster.

2. Output to Forward Model

Once the software has acceptable parameters, it will enter those parameters into the Forward Model and generate a light curve that will be displayed to the screen. Once the light curve is generated to the GUI the user will be able to save the output as a series of images locally on their device.

There will be a tab at the top of the output panel that will allow the user to switch between the generated light curve and the rendered images from the Forward Model.

3. Settings

The GUI will start with a user defined default for setting configurations. Before the software runs the Forward Model, the user will be able to change specific settings to more accurately run the Forward Model. The settings will be found in a tab at the top of the GUI that will prompt the user with a new page containing all the settings.

4. Command Line Interface

The software will allow the user to opt out of using the Graphical User Interface, and will then redirect the user to a Command Line Interface. This will allow users to run brute force scripts and bypass all restrictions within the settings tab. With this, the user will be able to run the Forward Model API by itself with no HMC or GUI integration.

FR4. Generate Video From Forward Model

1. Output File

When the video generation software is run, it will take in a collection of images and condense them down to a .MP4 file. The video generation software will be accessible from the Forward Model GUI, and the file location will be put into a text box. From there, the images will be converted to a .MP4 file at the press of the “Compile Button”.

2. Image Collection

As the API renders objects, it will save images of the binary system model into a folder. The images will be a visual representation of the binary system as the individual objects complete their orbit. These images will be rendered from the Forward Model while it is running, and then later used by the video generator to create a video.

4.2 Performance Requirements

Currently our clients' solution has room for growth in order to help streamline the overall process. Through our additions, our clients will have an easier work flow for the model. We still must be conscious of the user, especially with requirements such as usability, and understand how the program is expected to perform. Though many of our implementations' performance will be up to the users, we have broken down and detailed our core requirements below.

PR1. Runtime of HMC Algorithm

The runtime of the HMC Algorithm varies depending on how long the user wishes to run the algorithm. More steps result in a longer runtime which can lessen our clients' productivity. Without a user being able to choose when the algorithm ends, it could run up to an indefinite amount of time depending on how many parameters were specified to be estimated. A user defined end will be used for this algorithm to ensure that the process can be halted when the user believes it has collected enough data in regard to the variables they are interested in.

While a minimum runtime cannot be determined because of the user defined end, the user must be aware of what they are looking for and consider this when utilizing HMC. Still considering the defined end, there is also no defined accuracy of the algorithm. Though it may give output that is not what the user is looking for on occasion, it overall should move its output towards what is desired.

PR2. Graphical User Interface (GUI)

The GUI that is planned to be implemented to help simplify the process for our clients will be simple to use. Though we are in continuous communication with our clients to ensure that the design of the GUI is to their specification, other users must be considered. The design will be simplistic and not require much learning from the user. It is fairly straightforward and will have each parameter labeled. The user will be expected to be familiar with the current program to understand what each parameter does.

We can consider the video generator in this section as well, as the user must use the GUI to determine whether a video should be output. This will be a simple button that is fairly intuitive.

PR3. General Usability and Readability

The modules implemented within our solution will be supported with extensive documentation. Explanations for all of the functions we develop and their parameters will be provided. The documentation we supply will enable other users beyond our clients with the information necessary to utilize this application.

4.3 Environmental Requirements

ER1. Cross Cross-Platform Compatibility

The solution we implement needs to be able to compile and run on Linux, Windows and Mac. This is an environmental requirement met by the team that worked on the previous iteration of this project. Our clients asked that we keep this functionality with the solution we implement in order to maintain code usability and shareability.

ER2. Compatible with CPU and CPU-GPU Configurations

The solution will need to be compatible with computers that only have a CPU, as well as computers that utilize both a CPU and a GPU. We are considering this environmental requirement as it was met by the previous

team's solution. As such, our solution would have to follow this requirement in order to be compatible with the existing solution.

4.4 Conclusion

The solution that we plan to implement must satisfy the aforementioned functional, performance and environmental requirements. These functional requirements include simulating objects in binary systems as triaxial ellipsoids, and computing the best fit parameters of an observed light curve using an HMC algorithm. Additionally, the solution will need to contain a constructed GUI for the Forward Model, as well as the ability to create a video from rendered images produced by the Forward Model.

Within meeting these functional requirements, our solution must also consider several performance and environmental requirements. This includes requirements such as the runtime of the HMC algorithm, the knowledge needed to utilize the GUI, and the general usability and readability of the code we implement. Lastly, the solution will consider the requirements of cross-platform compatibility and the limitations of computers that will be used by our clients.

5. Potential Risks

Since our project requirements have been deliberated, it is important that we clarify any risks that could arise during the implementation process. While the risks we identified are subject to change, it is important that they are addressed with a plan for mitigation. While combing through our project requirements, we found three specific risks. The first risk is scope expansion, which regards the idea that our project will have future iterations. The second risk references the high learning curve derived from the HMC algorithm. Lastly, the third risk addresses the rotation function of the triaxial ellipsoid class.

5.1 Scope Expansion

5.1.1 Shape Class

As the project stands, the sphere shape and faceted shape classes can be used to visualize binary asteroids. For our project requirement, we will be

finishing the triaxial ellipsoid implementation. This will give our clients a diverse option to represent these binary systems without the cost associated with the faceted shape class.

Mitigation

From a logical standpoint, there are not any other classes that our clients will need to simulate the asteroids. Our clients have not mentioned any future plans for more shapes either. Therefore, we do not have a mitigation strategy for the case where we need to add more shape classes currently. If this becomes a risk as we continue discussion with our client, then we will update our Requirement Specification document accordingly.

5.1.2 Cluster Parallelization

Running the program with a faceted object representation of binary asteroids causes the program runtime to skyrocket. Adding the HMC functionality to the program will also cause the runtime to grow depending on how many parameters need to be estimated. In addition to those factors, our clients have mentioned that they envision the program being run on a computer cluster like Northern Arizona University's Monsoon in the near future. To accommodate our clients' vision for the future of this project, we need a mitigation plan.

Mitigation

Paired Planet Technologies, the team that created this program last year, implemented modular design concepts to uphold future integrations. To mitigate the future risk of scope expansion for cluster parallelization, we will hold ourselves to those same modular design standards.

5.2 Hamiltonian Monte Carlo

5.2.1 Learning Curve

The Hamiltonian Monte Carlo algorithm is the most important part of our project. After much deliberation among our team, our mentor, and our clients, it is apparent that it is also the most challenging factor. Our team does not have a strong background in Bayesian statistics, and our clients do not have formal education on the inner workings of the Hamiltonian

Monte Carlo algorithm. Since the HMC appears to have a steep learning curve, we need a mitigation strategy to counteract the likelihood for problems to arise due to misunderstanding.

Mitigation

To counteract the steep learning curve stemming from HMC, we need a vast support system. This support system includes our mentor Isaac Shaffer, who has a Master of Science degree in Statistics. In addition, we will be reaching out to the Statistics department of Northern Arizona University. It is important that we keep our clients updated on any knowledge gained from this support system. To sustain a solid understanding for our clients, we will update them during our client meetings.

5.3 Triaxial Ellipsoid

5.3.1 Rotation

As previously discussed in the Scope Expansion section pertaining to Shape Class, the triaxial ellipsoid class is mostly finished. Paired Planet Technologies had attempted to implement the shape last year on top of the other challenges of their project. However, since the radii of triaxial ellipsoids vary greatly, it is challenging but important to have a rotation factor to get characteristics from all angles.

Mitigation

The orient function of the triaxial ellipsoid class was made previously to perform the rotation. However, Paired Planet Technologies did not have sufficient time to get it functioning. Since the shape class for triaxial ellipsoid was already created and we only need to fix the orient function, there is no mitigation plan needed. There will be a slight learning curve to the math behind the rotation however, and therefore we will update for a mitigation plan if needed.

5.4 Risk Analysis

In order to present these values in a simple layout, we created a risk analysis table. The risk analysis table has a tab that holds the challenges

that we face. The severity tab ranks (1-10) how important that challenge is to our current project. The likelihood tab shows a percentage (1% - 99%) for the challenge, representing how likely it is to cause problems. The risk value is calculated by multiplying the severity by the likelihood. For risk values above a value of 3, a mitigation strategy will be enacted.

Risk Analysis Table				
Challenge	Severity (1 - 10)	Likelihood (1%-99%)	Risk Value	Mitigation
Scope Expansion: Shape Class	3	40%	1.2	Not a problem
Scope Expansion: Cluster Parallelization	5	80%	4.0	Modular design
HMC: Learning Curve	9	99%	8.91	Heavy communication
Triaxial Ellipsoid: Rotation	8	10%	.8	Not a problem

Table 3: Project risk factors briefly outlined

After considering the severity of these risks and how likely they are to cause issues, cluster parallelization and the learning curve associated with HMC needs to be handled. Since this project will be revisited in future capstone projects, we will deploy modular design tactics to uphold scope expansion for cluster parallelization. We will also heavily research HMC and build a knowledgeable support system to counteract the learning curve associated with it. We conclude that adding another shape class will not be a problem in the future, nor will fixing the triaxial ellipsoid rotation function. Team Andromeda recognizes that the risk values related to the rotation function and the shape class may change and will be updating our requirements specification as adjustments are needed.

6. Project Plan

To ensure the success of our project, we reviewed our short term and long term tasks. Our current work encompasses requirements specification, finalizing our webpage, and, most importantly, the technology demonstration. The long term tasks, or future work, peer into second semester tasks. These tasks can be viewed in Figure 4, which is located at the end of the document.

6.1 Current Work

We are working on finishing our final website and starting the technology demonstration. It will require us to take our first steps towards our complete solution. The technology demonstration will be finished by the end of the semester and requires us to show the following:

1. Demonstrating the HMC algorithm and its constructed interface by replicating a walkthrough of an agreed upon tutorial in the proposed HMC API
2. Demonstrating the Triaxial Ellipsoid Shape by creating a unit test that generates a picture in the build folder
3. Demonstrating the GUI by calling the Forward Model using the Sila-Nunam settings, allowing the user to change at least one parameter, and then re-run the Forward Model

6.2 Future Work

Over the winter break, our team has made a personal goal to familiarize ourselves with the C++ language. Looking into the second semester, there are four portions of our project we will need to implement. Since the triaxial ellipsoid class is mostly implemented, we will only need to correct the orient function so that spin state can be calculated. We will also be considering merging the Sphere class into the Ellipsoid class. Our client would like us to keep a copy of the Sphere class separate until we are able to confirm that their runtimes are roughly equivalent. We will also need to fully integrate the HMC algorithm, the GUI, and the video generator.

Since Paired Planet Technologies previously designed this project modularly, we will maintain that design throughout our integration. Once we have the alpha prototype finished in March, we will spend the rest of the semester optimizing our solution.

7. Conclusion

Space is an exciting, mysterious territory. Despite thousands of years of research, humankind has only begun to scratch the surface of understanding the universe. The New Horizons mission, launched in 2006, started its maiden voyage to the Kuiper Belt in pursuit of gathering characteristics of Pluto and other Kuiper Belt objects. To this day, there are numerous objects floating in the Kuiper Belt that we have yet to explore. Up to 15% of those asteroids are considered to have at least one secondary in their orbit¹. Studying these binary gives the power to unlock answers for how they are formed, how their orbits work, and how they fit in to the solar system.

Since space voyages are time consuming and costly, telescopic observations from earth are the most consistent way to form hypotheses of binary systems. Scientists can prepare observed light curves but cannot easily confirm the correctness of the characteristics formed from it. Our project was purposed to help our clients at Lowell Observatory solidify these attributes.

The Hamiltonian Monte Carlo algorithm we are implementing is the key to producing a carefully estimated lightcurve. This estimated light curve can be compared to the scientist's observed lightcurve and form genuine binary asteroid characteristics. The addition of the triaxial ellipsoid shape will accelerate render speeds for visualization without a high cost of accuracy lost. The HMC algorithm will allow for computed estimations of parameters for observed light curves, allowing for easier gathering of characteristic information for a binary system. The GUI will streamline the data input process for our clients, easing their use of the system. Finally, the video generator will compile a movie for our clients to use at will. We understand that the HMC algorithm is daunting but important task, and therefore we are deeply researching the problem.

¹

https://www.esa.int/Safety_Security/Hera/Binary_asteroids_key_to_Earth_s_planetary_defence

Team Andromeda is extremely excited to continue working with Dr. Audrey Thirouin and Dr. Will Grundy. Their observations at Lowell Observatory are yielding continuous advancements in the understanding of our solar system. We are delighted to be contributing to the progression of space knowledge and are ready for the challenge.

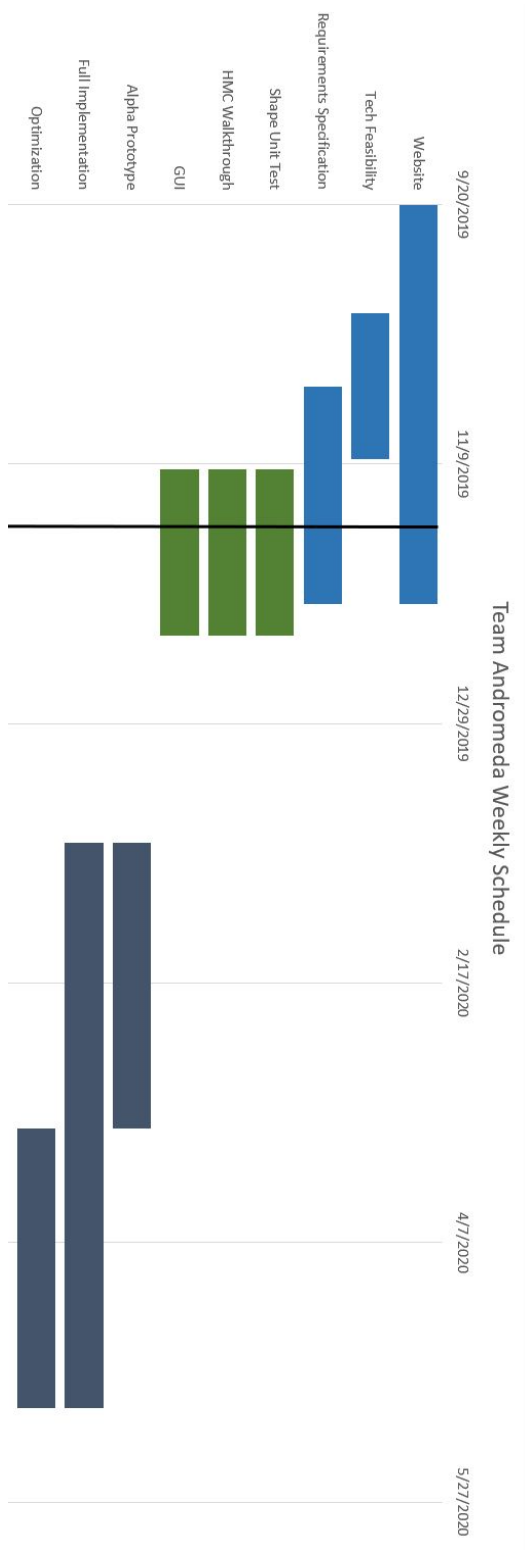


Figure 4: Gantt chart outlining weekly schedule