CS Capstone Design

Technical Demo Grading Sheet (100 pts)

TEAM: Team Andromeda

Overview: Team Andromeda wants to ensure the utmost success for delivering a full-bodied solution to our clients Dr. Will Grundy and Dr. Audrey Thirouin. To provide this solution, we need to create technical demonstrations in our early phase to kick-start our project. Although these demonstrations are minimal in comparison to the scope of our project, it is vital that we keep the ideologies of modularity, parallelization, and consistency throughout the scope of the project.

Risky technical challenges

Based on our requirements acquisition work and current understanding of the problem and envisioned solution, the following are the key technical challenges that we will need to overcome in implementing our solution:

C1: HMC API: The difficulty with this challenge will be working with the HMC API that our team chooses to implement into our project. This is predominantly due to our team's lack of familiarity with using an MCMC algorithm. The demonstration will need to prove to our mentor that we have a basic knowledge on how to write functional code for the HMC API. The API we chose to implement is Stan.

C2: HMC API Interface: The difficulty with this challenge will be implementing an interface that can interact with the selected HMC API. While we have experience with making interfaces, we do not have experience making an interface for this specific API. The demonstration will need to prove to our mentor that we can create a basic interface that is responsible for inputting data into the HMC API and displaying or saving the result. The interface we chose to implement for the HMC API is Pystan.

C3: Triaxial Ellipsoid Shape: When Paired Planet Technologies generated the lightcurve modeling last year, they were able to finish a large chunk of the triaxial ellipsoid class. However, they were unable to figure out the correct math to do the rotations necessary for calculating a light curve from the model. To ensure that the triaxial shape class has been implemented correctly, we will need to create a unit test that produces this shape before we move forward with rotation. After winter break, we will need to move forward with fixing the orient function that handles the spin state of the triaxial ellipsoid.

C4: GUI for the Forward Model: The difficulty of this challenge will be wrapping the GUI around the current API and calling it from within the GUI. This will be a challenge due to the crossover from C/C++ to Python. The Demonstration will need to prove to our mentor that we will be able to run and compile a C/C++ functions within the Python GUI.

Challenges covered by demos:

In this section, we outline the demonstrations we have prepared, and exactly which of the challenge(s) each one of them proves a solution to.

Demonstration 1: HMC API Tutorial Walkthrough

Challenges addressed: **HMC API and Interface**

Flight Plan:

1. Walk-through of agreed upon tutorial in the proposed HMC API interface. Tutorial: Eight Schools example - https://pystan.readthedocs.io/en/latest/getting_started.html

1a. Review each segment of code within the tutorial

2. Demonstrate the success of the tutorial with a visual model produced by the API

Evaluation:

✓ Convincingly demo'd each of listed challenges?

✓ Other evaluative comments:

Demonstration 2: Triaxial Ellipsoid Shape Test

Challenges addressed: **Triaxial Ellipsoid Shape**

Flight Plan:

1. Comb over the previously implemented triaxial ellipsoid class
2. Create a unit test that confirms the object is an ellipsoid
3. Reflect the successful unit test via picture generated in build folder

Evaluation:

✓ Convincingly demo'd each of listed challenges?

✓ Other evaluative comments:

Demonstration 3: GUI for the Forward Model Interaction

Challenges addressed: **GUI for the Forward Model**

Flight Plan:

1. Demonstrate GUI call of Forward Model
2. Demonstrate Forward model call using Sila-Nunam settings by showing a predicted light curve.
3. Allow user to change at least one parameter via textbox and re-run Forward model call.

Evaluation:

✓ Convincingly demo'd each of listed challenges?

✓ Other evaluative comments:

Other challenges recognized by not addressed by demo:

If there were challenges you listed earlier that were *not* covered by a demo, list here. This will hopefully be a short list…but better to be clear about where you are. If you have items here, you could list (if applicable) any pending plans to reduce these risks.

- Integrating the HMC API in a way that it calls the Forward Model: The demo will demonstrate our ability to use the HMC API, but it will not demonstrate the difficulty of integrating the HMC API as a wrapper to the Forward Model. We will consult with Brian Donnelly, a member of the team that worked on the previous iteration of our project, for advice on integrating the HMC API into the codebase.
- For the triaxial ellipsoid representation, the biggest challenge moving forward is implementing the rotation or spin state of the shape. The previous team attempted to implement this as it is necessary in order to calculate a light curve from the triaxial ellipsoid shape. However, they were not successful mostly because of the intense math behind this solution. Since we have little understanding of how this math works, we will be researching triaxial rotation problems as well as talking to professors in the math department to find a solution.