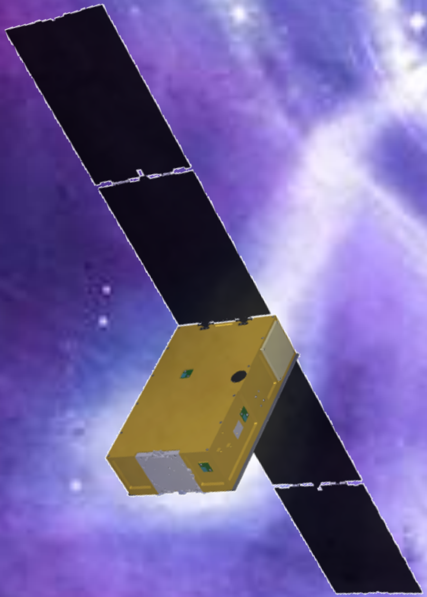
A Prediction Code for the BurstCube Satellite

What is BurstCube?

- Many space telescopes observe in the x-ray and gamma ray, because the ozone layer effectively shields us from them
- BurstCube is a relatively inexpensive satellite that will expand our ability to observe more high-energy events, and corroborate the ones we already detect
- It's a type of CubeStat, a 6U (20×30×10cm), that can share the rocket as a "rideshare" payload
  - They can even just be "tossed" off the ISS into orbit with a special arm or ejector
- The planned mission is only one year after launch in 2022, but could be up to three years, so we want to have all the programming ready to go before launch



Pi Nuessle, mentored by Dr. Judith Racusin, NASA Goddard

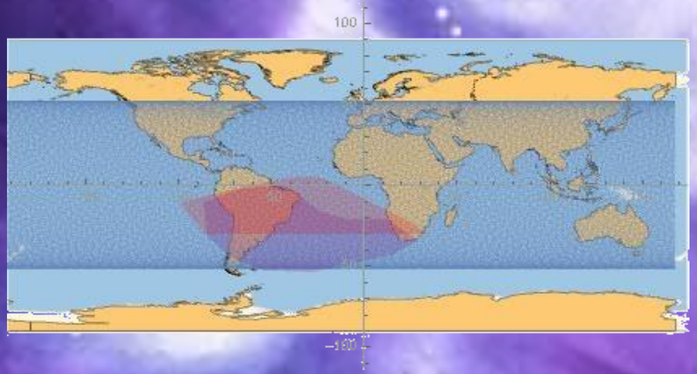
Presented to Royal Astronomical Society

The Code

a starting point for the science operations software

Purpose

- We always need to know the satellite's orientation and position to know what it was, is, or will be looking at
- There are areas, namely the Southern Atlantic Anomaly caused by the Van Allen Belts, where the satellite will need to be turned off
- The angle to the Sun (i.e. beta angle) will also be needed to estimate the power level



This map of BurstCube's orbit matches expectations. (the red SAA is the Fermi-LAT polygon, the purple one NICER)

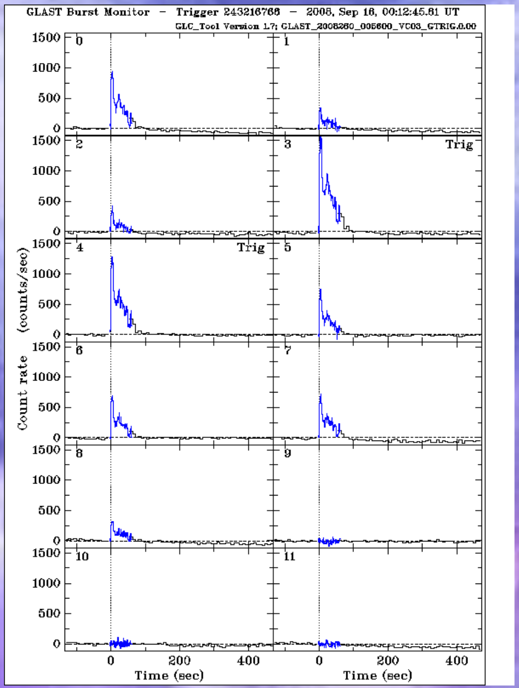
Workings

- The code is written in Python to maximize accessibility by other teams
- It uses SGP4, a first-order Euler method, to predict the position
- It will implement quaternion propagation to fully account for the reaction wheels, orbital precession, etc. when predicting the pointing
- A simple function is used to find the beta angle
- All code is heavily commented and includes a list of command line Python packages

Code Functionality

When BurstCube triggers, we get a light curve from each of the four detectors with coarse energy resolution

- These outputs, along with the localization code, is used to pinpoint the source



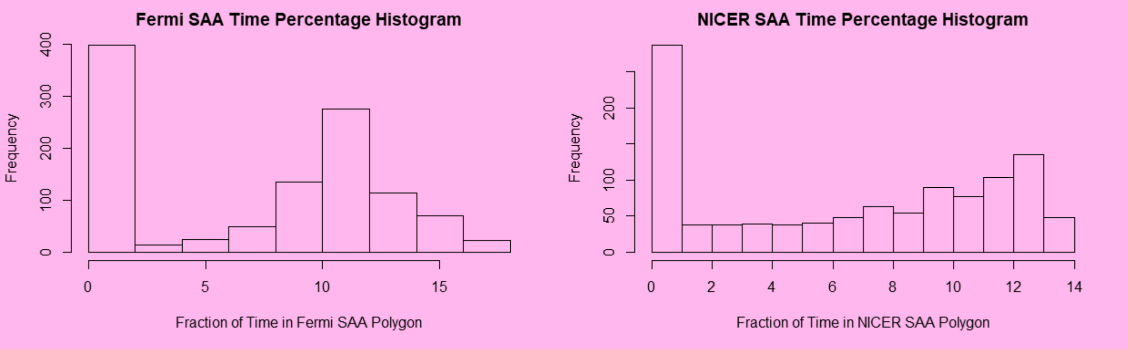
An example image from FERMI GBM displaying the slight difference in the twelve detectors that we can use to locate the burst.

How the Code is Used

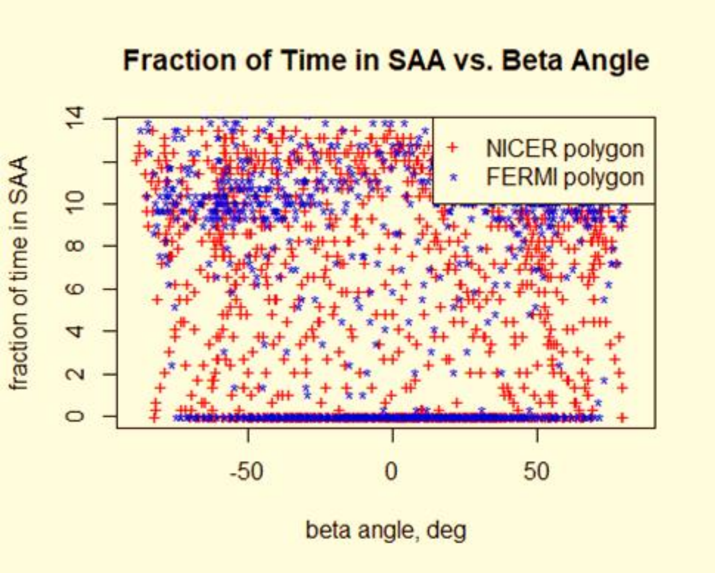
1. It asks where BurstCube is, what it's looking at, and when it is
2. It asks when the user would like to get information about
3. It figures out the latitude and longitude BurstCube will be at on the requested date
4. It figures out what angle the sun will be at, relative to BurstCube, at that time
5. It calculates the pointing at that time
6. It figures out if the satellite will be in a background anomaly
7. Finally, the program outputs that information to a .csv file, for use in programs like Excel, R, or Python

If wanted, there is also a version of the code that steps through a year of data each time; it also outputs an annual adjustment to the beta angle to correct for nodal precession

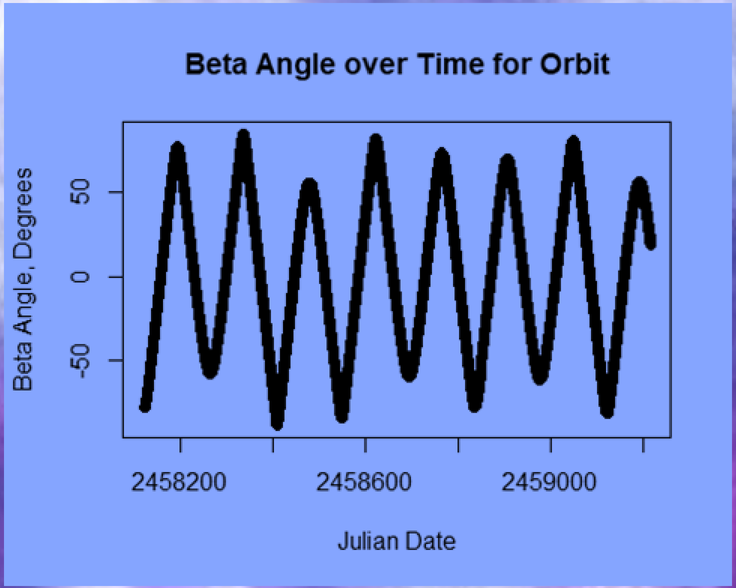
Code Checks



Here we observe the percentage of time spent in each SAA. Note that the two graphs have different y-axes but agree that the satellite spends about 11% of the day in them.



Here, we confirm that the time in the Southern Atlantic Anomaly and the beta angle are unrelated, with a correlation of 0.0050.007 and a p-value of 0.9/0.8



We observe appropriate behavior in the beta angle over time due to the orbits and nodal precession.

Thanks

- Judy Racusin
- Sean Griffin
- Sean Semper
- Hasnaa Khalifi
- Jeremy Perkins
- StackOverflow
- STLTracker
- NASA Thermal and Fluids Analysis Workshop