**GeoData Software Package**

**Motivation**

The GeoData software is a set of classes in functions in Python and MATLAB to plot and interpolated data from various sources in the space physics community. The main goal is to simplify the plotting and processing of geophysical data from various remote sensors such as radar systems and cameras.

The software set will be tied together by using a class structure. These classes will be written in both python and matlab so users can use either language. These classes will generalize the different data types so code can be reused more effectively. Currently tasks such as interpolating data points to different coordinate spaces require code to be rewritten specifically for new data types. This class structure will hopefully remedy that by making an overarching API so data from different sources can be easily brought in and used effectively.

**Class Structure**

The GeoData software is built around a class where each instance is a data set from a sensor. The user will simply have to make a function to read the data into the class and which at that point will give them access to other tools such as interpolations and plotting methods that can be used to augment or display the data.

The class will be made up of the following variables

data - This will hold the data for the data set. In python this will be a dictionary where the keys are the names of the data and the values will be numpy arrays that hold the data. In MATLAB the field names will be the data names and the arrays will be the values. Each data set will be held in a flattened array structure or can be an NxT array where N is the number of locations of measurements and T will be the number of times. If this is not possible both dataloc and times

coordnames – This string will hold the types of coordinates for the data. There will be a set number of coordinate types seen in the table below. More can be added as needed,

|  |  |
| --- | --- |
| string name | Definition |
| wgs84 | Latitude Longitude altitude (deg,deg,m) |
| Spherical | Range azimuth and elevation (km, deg, deg) elevation angle is referenced to z=0 plane. |
| Spherical2 | Range azimuth and elevation (km, deg, deg) elevation angle is referenced to x=y=0 line. |
| ENU | East north up (m,m,m,lat,long,alt). |
| ECEF | Earth centered earth fixed (m,m,m). |
| Cartesian | Local Cartesian grid (km,km,km). |

dataloc – This will be a NxP array of locations in the coordinate system of choice. P is the number of elements

sensorloc – This will be an array that holds the location of the sensor in wgs84. If there are multiple sensors such as a set of satellite measurements the array will be filled with nans.

times – A Tx2 array of times in posix format showing the ending and beginning of a measurement.

**Workflow**

The intended workflow will entail the user first reading the data into the class structure. This will be done by creating a read function that will output the variables listed above. The other method will be making their own function to fill in the data structure and output an instance of the class.

From there the user can take advantage of pre-programed functions for interpolation and plotting. This will allow for maximum code reuse.

**Input/Output**

The data in this format can be saved into structured hdf5 files. Either MATLAB or Python can read these files. This will allow for easy transfer between the two languages so the features between the two can be utilized.

**Installation**

The software package can be installed in MATLAB by adding the necessary file to the MATLAB path. The python version can be installed using the setup tools package.