Using the NOAA GFS Model to Predict High-Altitude Balloon Flight Trajectories for Arbitrary Flight Profiles

Austin Langford

Submitted under the supervision of Professor James Flaten to the University Honors Program at the University of Minnesota-Twin Cities in partial fulfillment of the requirements for the degree of Bachelor Science, *magna cum laude* in Aerospace Engineering and Mechanics.

May 12, 2018

Acknowledgements

The author gratefully acknowledges the support of Dr. James Flaten, the advisor for this paper and the associate director of the Minnesota Space Grant Consortium, as well as Shane Mayer-Gawlik of the Montana Space Grant Consortium for his guidance and experience in this topic.

Abstract

For the purpose of predicting the flight path of a high-altitude balloon with payload, an automated process to retrieve future wind velocity datasets from the NOAA's GFS model for all altitudes is explored and described. This wind data is then used along with arbitrary initial conditions in latitude, longitude, altitude, and time, plus an altitude profile for the balloon to predict the latitude and longitude of the balloon at each time in the altitude profile, generating a predicted trajectory in latitude, longitude, altitude, and time for the balloon. An example program implementing this process is illustrated, and its results are compared to currently available online tools for balloon trajectory prediction. The example program is found to match the results of the available tools closely, with some deviation that is expected due to the method of integration selected, demonstrating that integration of the wind datasets acquired from the NOAA GFS model is a valid method of balloon flight trajectory prediction. Additionally, the methods used are customizable, which allows for trajectory prediction for any arbitrary altitude profile.

Table of Contents

Introduction

Studying the stratosphere is a difficult task due to the challenges in reaching and operating at such high altitude. Helium or Hydrogen filled high altitude balloons, such as Latex weather balloons, give relatively inexpensive access to the lower stratosphere, with the only limitations being the weight of balloon-borne payloads. By using latex balloons that rise until they burst at altitude, or zero-pressure balloons that rise to an altitude and then continue to float, scientific instruments can be lofted to study the stratosphere. At the end of a balloon flight, whether the balloon bursts or the payload is cut free of the balloon, the payloads will parachute to the ground for recovery.

One major challenge of recovery is determining where the payload is likely to land. It is difficult to accurately predict the flight path of a balloon because atmospheric conditions vary with altitude. Over the course of a 2 to 20 hour balloon flight, it is likely to drift over potentially hundreds of miles, possibly into terrain that would make recovery difficult or impossible. It is for this reason that balloon flight path prediction is an important aspect of choosing a balloon launch site, launch time, and altitude profile.

This thesis will demonstrate how National Oceanic and Atmospheric Administration (NOAA) weather models can be acquired and used to predict the flight path of a balloon. There will first be an explanation of the relevant data that is needed for trajectory prediction, and where it can be found. Then, there will be an example of how this data can be used to make a flight trajectory prediction for a variety of ascent and descent profiles. Finally, the results of these trajectory predictions will be compared to existing, freely available online tools to demonstrate why the limitations of existing software products motivate this effort.

Balloon Equations of Motion

The equations of motion for a balloon in free flight are the following [1]:

$$m\ddot{x} = \overrightarrow{F_d}x,$$

$$m\ddot{y} = \overrightarrow{F_d}y,$$

$$m\ddot{z} = \left(\overrightarrow{F_d} + \overrightarrow{F_g} + \overrightarrow{F_b}\right)z,$$

where m is the mass of the balloon and payload, $\overrightarrow{F_d}$ is the aerodynamic drag force resulting from the difference between the balloon velocity and the velocity of the surrounding air, $\overrightarrow{F_g}$ is the force of gravity, and $\overrightarrow{F_b}$ is the force from buoyancy. The z-direction is the vertical direction, while the x and y directions are horizontal. For stratospheric ballooning, the gravitational force is considered to be constant regardless of position or altitude. The drag force is dependent the physical characteristic of the balloon, as well as the relative wind speed between the balloon and surrounding air, and therefore depends on altitude and position. These equations of motion show that motion in the vertical direction is a combination of buoyancy, drag, and gravity. This relationship is somewhat complicated, but in practice there is very little effect from wind on the vertical velocity of the balloon. This is primarily because the velocity of the wind in the vertical direction is usually small. Renegar from the University of Maryland, College Park [1], as well as Voss, Ramm, and Dailey from Taylor University [2], have published formulae for calculating the vertical velocity of a balloon with payload. These depend on air density and various balloon parameters such as coefficient of drag and surface area. These factors depend on the geometry and physical parameters of the balloon, and since a balloon expands as it rises, they depend on the balloon's altitude.

These equations also demonstrate that the horizontal motion of the balloon will only depend on the aerodynamic drag force. This drag force depends in part on the velocity of the air relative to the balloon velocity. In practice, the balloon's velocity will rapidly approach the velocity of the wind and, over the course of a long balloon flight, the balloon velocity can be reasonably estimated to be equal to the velocity of the wind.

Regardless of which model altitude vs. time profile model is used, the horizontal motion of the balloon will only depend on the horizontal wind speed over the chosen altitude profile. This report will therefore go into detail only about predicting the horizontal motion, assuming that an altitude vs. time profile is generated via some model or assumption that is consistent with experience from actual balloon flights.

Acquiring the Necessary Atmospheric Data

Determining the drift of a balloon in terms of latitude and longitude will only depend on the wind velocity over its altitude profile. Therefore, the necessary datasets needed for prediction are wind velocity and direction at a given time, latitude, longitude, and altitude. Because determining a balloon's future trajectory in terms of wind velocity is only useful for prediction, it is not a useful to acquire measured wind data from the past. What is needed is a model of future wind velocity.

The Global Forecast System (GFS) is a weather forecast model that is produced by the National Center for Environmental Prediction (NCEP) [3]. It integrates many sources of information to create an accurate model of the atmosphere at high spatial resolution, with predictions spanning 16 days. It provides datasets of many atmospheric and weather-relevant quantities over a grid spanning the entire globe, and is published every 6 hours. Not unexpectedly, the models are less accurate further from the publishing time. Therefore, it is important to use the newest model whenever possible. Because of the high spatial resolution and wind velocity predictions that reach out to over two weeks, the GFS is the model that will be used to acquire wind datasets for the use in balloon flight path prediction.

Acquiring weather forecast datasets can be done through the NOAA Operational Model Archive and Distribution System (NOMADS). The GFS contains many atmospheric parameters that are not relevant to balloon trajectory prediction and it also covers much more of the Earth than is necessary for a typical weather balloon flight. This results in very large amounts of data, very little of which actually needs to be downloaded. Downloading the entire set over the NOMADS system is impractical, as this needs to be done frequently for more accurate predictions and takes significant time for large amounts of data. Fortunately, NOMADS facilitates access to the GFS data through OPeNDAP, which allows for selective access to pieces of the total dataset. The main URL for NOMADS is:

http://nomads.ncep.noaa.gov

At this site there is a table containing information and links about each of the available datasets.

To access datasets over OPeNDAP, a URL that leads to the desired dataset needs to be generated. The base URL, which will begin any URL that leads to datasets provided by NOMADS, is:

http://nomads.ncep.noaa.gov:9090/dods/

From here, an identifier for the specific dataset can be appended to the end. For the high resolution GFS model (0.25 degree), the identifier is gfs_0p25, leading to the URL for the 0.25 degree resolution GFS model:

http://nomads.ncep.noaa.gov:9090/dods/gfs_0p25

There are many datasets that can be accessed by using different identifiers, such as a lower resolution (0.5 degree and 1 degree) GFS models. These identifiers can be found by selecting the OPeNDAP links for the desired sets at the NOMADS main page listed above.

Next, a date for the dataset must be selected. The format for this date identifier is "gfs[year][month][day]", where year is four digits and month and day are two digits. This is appended to the end of the previous URL to generate something similar to the following, which is for 0.25 degree resolution GFS model datasets for April 29, 2018:

http://nomads.ncep.noaa.gov:9090/dods/gfs_0p25_1hr/gfs20180429

Model outputs are uploaded to NOMADS four times daily at even 6-hour increments. Therefore, the desired data set on the date provided must be specified. This is done by selecting either 00z, 06z, 12z, or 18z as the UTC hour into the date for the dataset desired, and inserting it into the identifier. The "z" at the end of these numbers stands for "zulu", which represents the time in UTC (Coordinated Universal Time). UTC is offset from the US central time zone by 5 hours during daylight savings time, and by 6 hours otherwise. For example, the 0.25 degree resolution dataset for 06 UTC hours into the date would be selected by appending “gfs_0p25_06z”, which can be seen in the following URL for the 0.25 degree resolution GFS model dataset published at 06 hours UTC on April 29, 2018:

http://nomads.ncep.noaa.gov:9090/dods/gfs_0p25/gfs20180429/gfs_0p25_06z

This is now the complete URL that will allow access to the GFS model published at the time specified. Appending “.info” to this URL will lead to a page describing this dataset. It will show the longitude, latitude, and time range covered, as well as a description of each atmospheric parameter that can be attained from the set. Here, it becomes clear which pieces of data are needed to make balloon flight predictions. The parameters “ugrdprs” and “vgrdprs” are the u and v (East and North respectively) components of the wind velocity in meters per second, which are the sets that will be needed. To get the dataset over OPeNDAP, append “.ascii?parameter” to the dataset URL. It is generally not a good idea to acquire the entire dataset for any parameter, as it will be a very large amount of data. Therefore, a subset of the parameter set must be acquired by using an index range.

The u and v wind velocity sets are each given as functions of four parameters: time, lev, lat, and lon. Lat and Lon correspond to latitude and longitude respectively, and lev corresponds to the GFS altitude level. Each of these parameters is an index into the grid of the total span of each parameter, and each input can be a range of indices. Therefore, to get the wind velocity at a given time, level, latitude, and longitude, append “.ascii?parameter[timeIndex][levIndex][latIndex][lonIndex]” to the dataset URL.

The time index is the simplest to determine. The GFS model outputs forecasts at a three hour interval. Therefore, the time index will be given by the difference between the prediction time and the GFS time, divided by three hours.

The index into the latitude and longitude sets can be determined by using the sets themselves. Each of these sets reports the grid of latitudes and longitudes covered in decimal degrees. To download the latitude set, append ".ascii?lat" to the end of the dataset URL. This will return a comma separated set of the latitudes in the grid of the GFS model. Similarly, the longitude set can be acquired by appending ".ascii?lon" to the end of the dataset URL. From here, simply compare the desired latitude or longitude to the values in the sets to determine the index for that latitude or longitude.

Determining the level index requires converting from altitude to GFS level. The "hgtprs" parameter is the geopotential height for a given time, level, latitude, and longitude. Therefore, once the time, latitude, and longitude indices are known, the entire set of levels for those indices can be acquired. The resulting set will be a set of altitudes for that time, latitude, and longitude. The level index is the index into this set whose value most closely matches the altitude desired. Therefore, the process involves first determining the time, latitude, and longitude indices as demonstrated above. Then, appending ".ascii?hgtprs[timeIndex][0:30][latIndex][lonIndex]" to the dataset URL. For example, the appended section could appear as ".ascii?hgtprs[0][0:30][540][1040]". This will return the set of altitudes in the grid at the given time, latitude, and longitude indices. Then, the index of the value of this set which most closely matches the altitude is the level index.

Now that the process of locating and downloading wind speeds for a given time, altitude, latitude, and longitude is understood, the subset of the total GFS dataset that is needed can be downloaded. Assuming the dataset desired spans from *minTime* to *maxTime,* from *minLat* to *maxLat,* and from *minLon* to *maxLon,* the process is as follows.

First, determine the minimum and maximum time indices that correspond to *minTime* and *maxTime* by using the following equation:

$$timeIndex = \frac{(time - GFSTime)}{10800},$$

where both time and GFSTime are in seconds, as 10800 is the number of seconds in three hours.

Next, determine the minimum and maximum latitude and longitude indices that correspond to *minLat* and *maxLat.* Download the latitude and longitude sets by appending to following to the dataset URL:

.ascii?lat

.ascii?lon

Search the sets for *minLat*, *maxLat, minLon* and *maxLon* to determine the minimum and maximum indices.

Next, download the level set by appending the following to the dataset URL:

.ascii?hgtprs[*minTimeIdx:maxTimeIdx*][0:30][*minLatIdx:maxLatIdx*][*minLonIdx:maxLonIdx*]

Then, download the u and v wind speed sets by appending the following to the dataset URL:

.ascii?ugrdprs[*minTimeIdx:maxTimeIdx*][0:30][*minLatIdx:maxLatIdx*][*minLonIdx:maxLonIdx*]

.ascii?vgrdprs[*minTimeIdx:maxTimeIdx*][0:30][*minLatIdx:maxLatIdx*][*minLonIdx:maxLonIdx*]

For example, for the region surrounding Minneapolis by 5 degrees in the each direction, *minLat* and *maxLat* are 40 degrees and 40 degrees respectively. These correspond to indices in the 0.25 degree resolution latitude set of 520 and 560. For the longitude range -98 degrees to -88 degrees, the corresponding indices in the 0.25 degree resolution longitude set are 1048 and 1088. Therefore, the string appending to the URL to download the altitude set for this region would be similar to the following:

".ascii?hgtprs[3:11][0:30][520:560][1048:1088]"

where the time indices depend on the date and time length desired. Weather balloon flights usually last 2 to 3 hours, though the time of day when a flight begins can vary significantly.

Each of these will return a text version of the requested subset of the GFS model. Therefore, each will need to be processed before they can be used to make predictions. The example program in the Appendix shows an example of handling this data as multidimensional NumPy arrays, along with a file keeping track of the offsets. NumPy is a free scientific computing library for Python.

Using Atmospheric Data to Make Balloon Flight Path Predictions

Once the required datasets are downloaded, they can be used to make a balloon flight path prediction. Prediction of latitude and longitude is accomplished via numerical integration from the wind speeds at given locations and times. Therefore, what is needed to determine the flight path of the balloon is the following:

1. Initial Conditions for Latitude, Longitude, and Time
2. A flight profile of Altitude vs. Time

Then, beginning with the initial conditions and assuming the altitude of the balloon follows the profile, the latitude and longitude for each subsequent altitude/time in the profile can be determined by integrating from the wind dataset.

The initial conditions correspond to the launch site and time of the flight. These can be varied based on the output of the prediction to find a trajectory and landing spot that is acceptable. The altitude vs. time profile can be generated based on any model for balloon vertical velocity and for any time step size. This gives freedom in choosing a flight profile that takes any shape, whether it be to ascend to burst or to float for some amount of time.

The example program in the Appendix demonstrates an Euler's method approach to determining the flight path of a balloon. Any integration technique can be used, but for this example Euler's method was selected due to its simplicity. Also included in the Appendix is a program to generate an altitude profile for either a simple balloon flight with constant ascent and descent rates and a burst at the maximum altitude, or a profile that ascends at a constant rate to a float altitude, floats for a certain amount of time, and then descends at a constant rate. In reality, the ascent rate of a balloon is close to constant, however the descent rate is typically not, with the descent rate decreasing with altitude. Using actual flight data, a more accurate model of the descent rate of a balloon could be generated and used to create flight profiles.

The process for making trajectory predictions is as follows:

First, insert the initial conditions into the altitude profile to make the profile specific to the launch site and time. Next, determine the wind velocity for the first latitude, longitude, altitude, and time as given by the initial conditions by converting them to indices as described in the previous section, and using the indices to search the u and v wind speed sets. Use these wind velocities, along with the time step selected in the altitude profile, to determine the latitude and longitude of the next point in the altitude profile. Continue this process with the new latitude and longitude until the altitude profile is completed.

Once the integration is complete, it will have produced a set of time, latitude, longitude and altitude. This is the prediction of the balloon's trajectory. The method of visualizing or utilizing this data depends on the scenario. Included in the program in the Appendix is a function to generate HTML and Javascript for a Google Maps plot of the predicted flight path to visualize the trajectory. In the Appendix, there is a worked example of using the example programs to create altitude profiles, download datasets, and run a flight prediction.

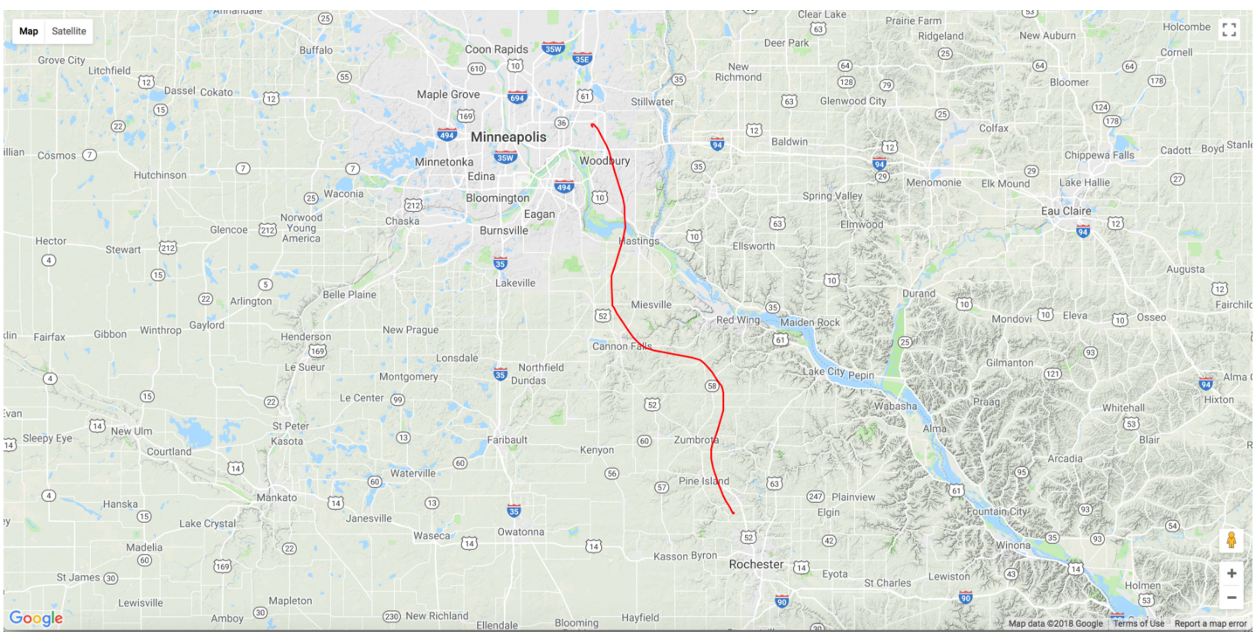
Comparison to the Available Flight Prediction Tools

Renegar's paper *A Survey of Current Balloon Trajectory Prediction Technology* [1] provides a comparison of three online balloon trajectory predictors: the Cambridge University Space Flight predictor (predict.habhub.com), ASTRA predictor (astra-planner.soton.ac.uk), and the University of Wyoming predictor (weather.uwyo.edu/upperair/balloon_traj.html).

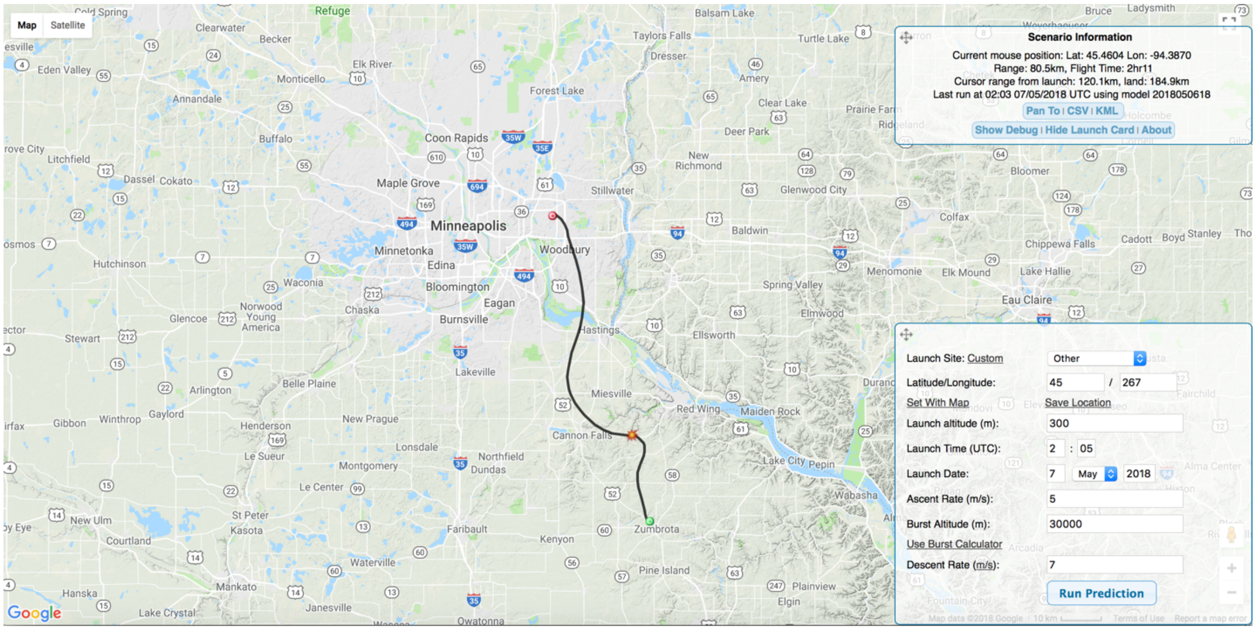
The Cambridge predictor is simple to use, freely available online, and is the flight prediction tool most often used by the Minnesota Space Grant ballooning team. It allows users to select initial conditions of the flight, as well as ascent rate, descent rate and burst altitude. There is an optional burst and ascent rate calculator that will attempt to calculate the ascent rate and burst altitude based on the model of balloon and gas type being used. The University of Wyoming predictor is the least fully featured of the three, with no built-in visualization tool, and very limited options for the altitude profile of the balloon. It does however have a calculator for descent velocity based on drag coefficient, mass, and chute size. The ASTRA predictor offers the most features of the three, with the advanced modelling of latex balloon ascent and descent. Additionally, it utilizes a Monte Carlo approach to simulate variation in balloon parameters that could lead to slightly different flight trajectories, and gives a heat map of landing locations. ASTRA predictor is also the only tool that allows for the simulation of flights with a float segment. While there are advantages to these features, teams such as the Minnesota Space Grant ballooning team at the University of Minnesota have flight data for over 100 flights, and as a result can predict the altitude profile accurately without such models of the balloon. Unfortunately the ASTRA predictor does not allow for any other method of altitude profile simulation, limiting its potential. One common feature of the three predictors is that they each do not allow customization of the integration technique or changing of the timestep used in the prediction.

As a simple demonstration of the example program in the Appendix, a sample of the prediction for a flight that begins at 45° latitude, -93° longitude at 300 meters above sea level, and ascends at a constant 5 m/s to a burst at 30000 meters, and then descends at a constant 7 m/s is run. The results are compared to the results of the Cambridge predictor and the University of Wyoming predictor, as the ASTRA predictor does not give the options to generate such a flight profile, as altitude profiles can only be generated using their model.

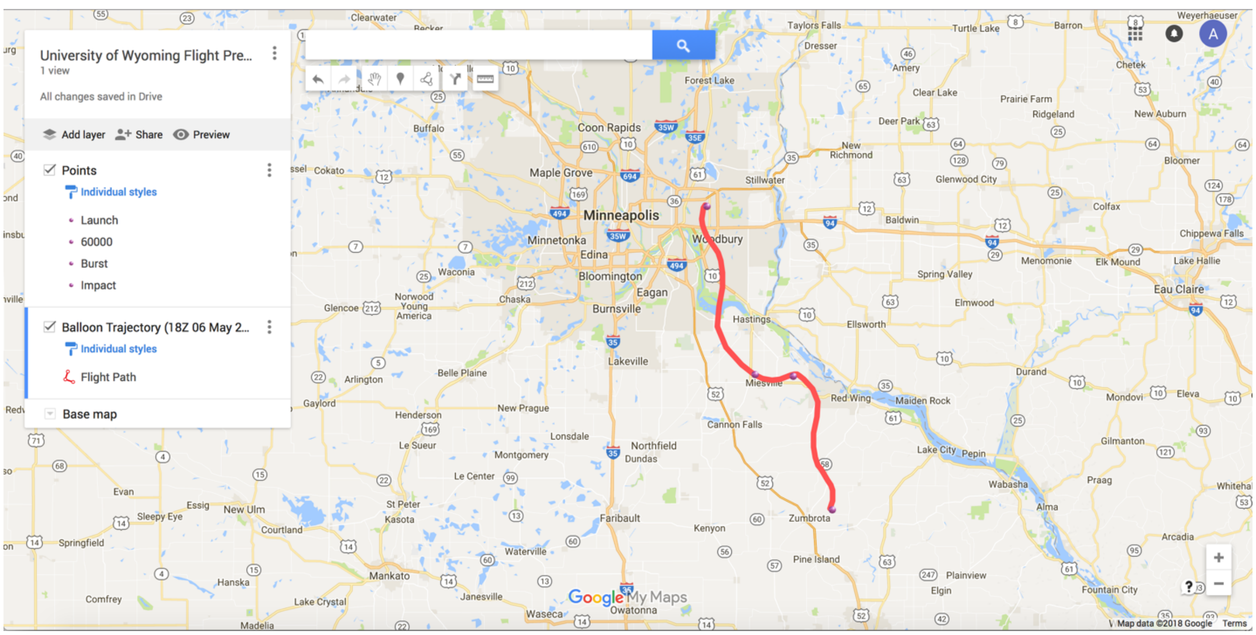
The results of each are shown here:



The Example Program



The Cambridge University Predictor – The asterisk is the location of burst



The KML output of the University of Wyoming Predictor

From this simple comparison, it is clear that each program predicts the same general flight path. The University of Wyoming and the Cambridge University predictors show excellent agreement with each other, while the example program overshoots their prediction slightly. The differences could be related to the Euler's method approach taken by the example program, while the Cambridge University predictor uses a 4th order Runge-Kutta technique [1]. The overshoot could also be related to the constant decent speed used, while the Cambridge and Wyoming predictors likely use a descent profile that is more realistic and therefore faster. A modification to the example program could be made to determine which factors are significant. Regardless, it appears that the techniques used to generate balloon trajectory predictions in this paper are valid.

Conclusions

Predicting the flight path of a balloon is a critical part of high altitude ballooning. It is possible to predict the trajectory of a balloon assuming a known altitude profile by using information acquired from the NOAA's GFS model that is freely accessible through NOMADS. Downloading a subset of this dataset is important to limit the amount of data that is requested from the servers, and is handled by using OPeNDAP. Once the datasets are local to the machine running the prediction, any model can be used to generate an altitude profile, and any integration technique can be used to translate from wind speed to latitude and longitude based on initial conditions.

While an Euler's method approach to numerically integrating the trajectory of a balloon was demonstrated in the example programs, more robust numerical integration techniques could be explored in the future to perhaps achieve more accurate results. Additionally, more user-intuitive methods of generating altitude profiles could be explored to simplify the process of creating flight predictions.

Appendix

The following repository holds the example program that will download GFS model data for a given latitude and longitude region for a given amount of time, as well as the example program that uses this data to predict the trajectory of a balloon.

https://github.com/MNSGC-Ballooning/GFS-Predictions

The following is a worked example of how to utilize these programs to predict a balloon flight trajectory:

First, use the “createAltPlan.py” program to generate altitude profiles in the correct format:

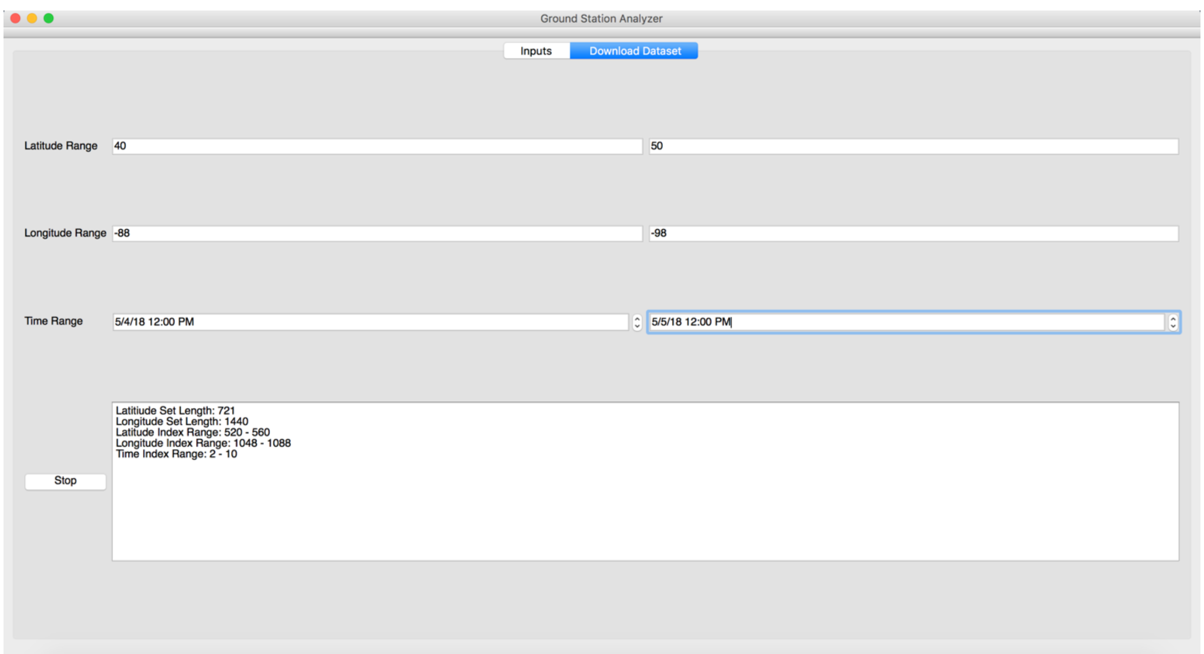
Burst Example:

```
python createAltPlan.py
Float or Burst Profile? (F/B) B
Alt Profile Name: burstProfile1
Timestep (s): 30
Start Alt (m): 300
Max Alt (m): 33000
Ascent Rate (m/s): 5
Descent Rate (m/s): 7
```

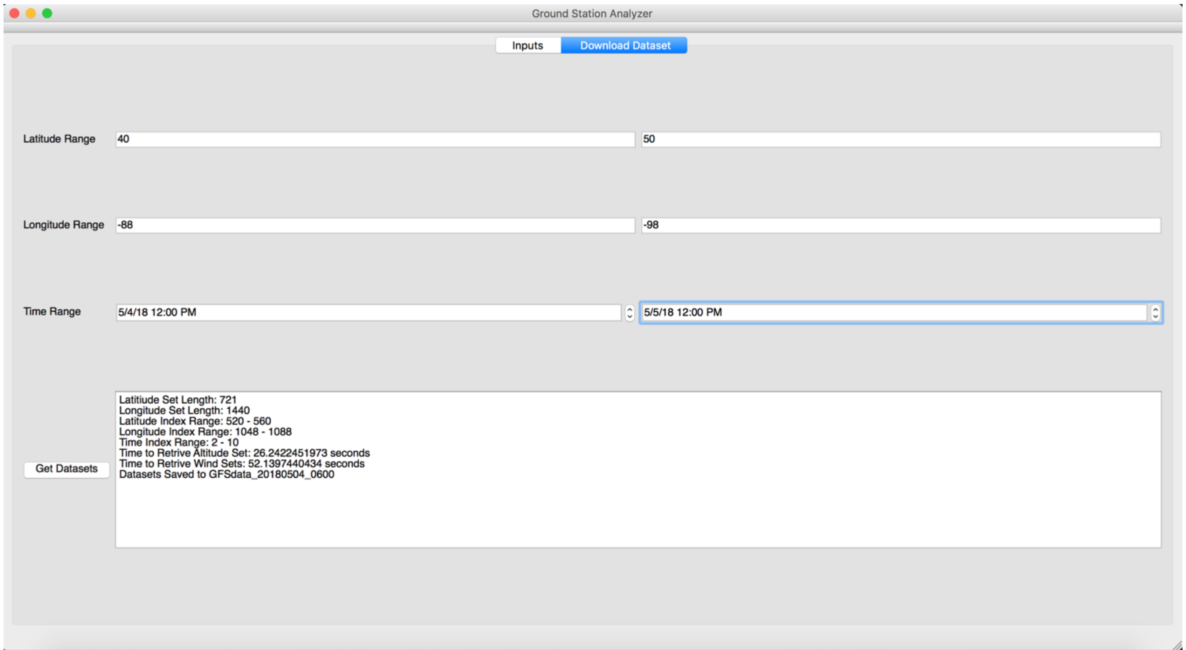
Float Example:

python createAltPlan.py
Float or Burst Profile? (F/B) F
Alt Profile Name: floatProfile1
Timestep (s): 30
Start Alt (m): 300
Float Alt (m): 25000
Float Time (s): 7200
Ascent Rate (m/s): 5
Descent Rate (m/s): 7

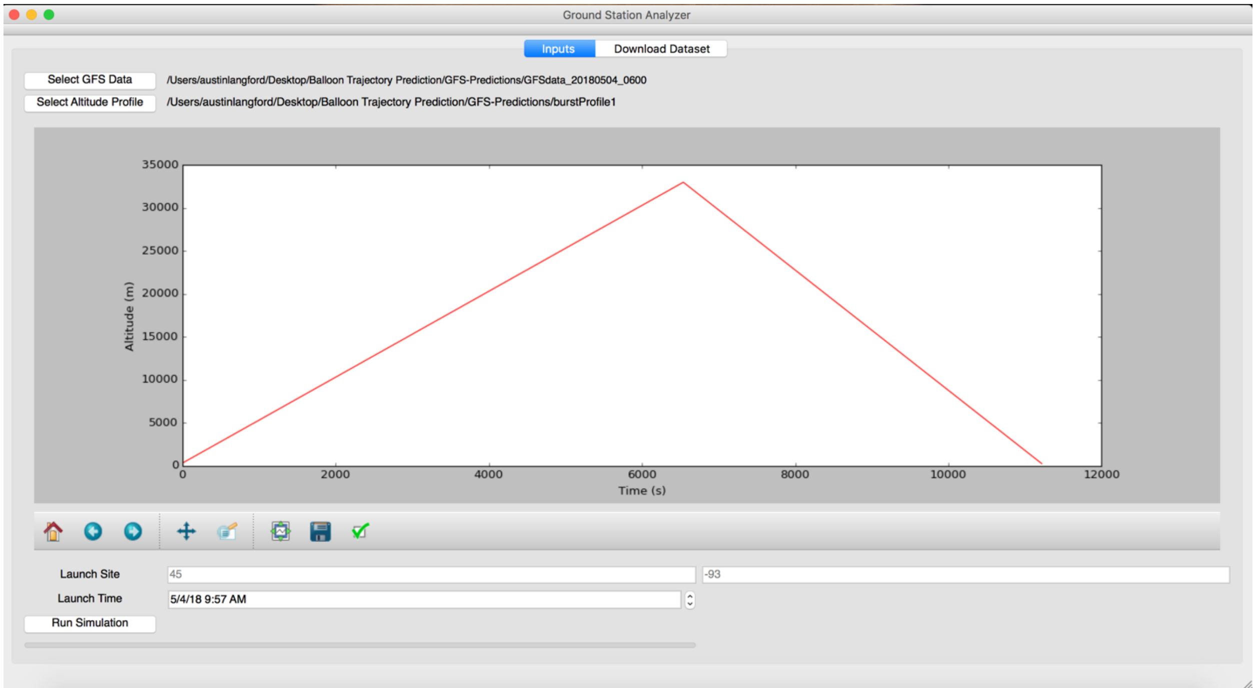
Next, run the GUI-based simulation program “simulator.py”, and go to the download datasets tab. Select the region that the flight will remain inside, and click the download datasets button:



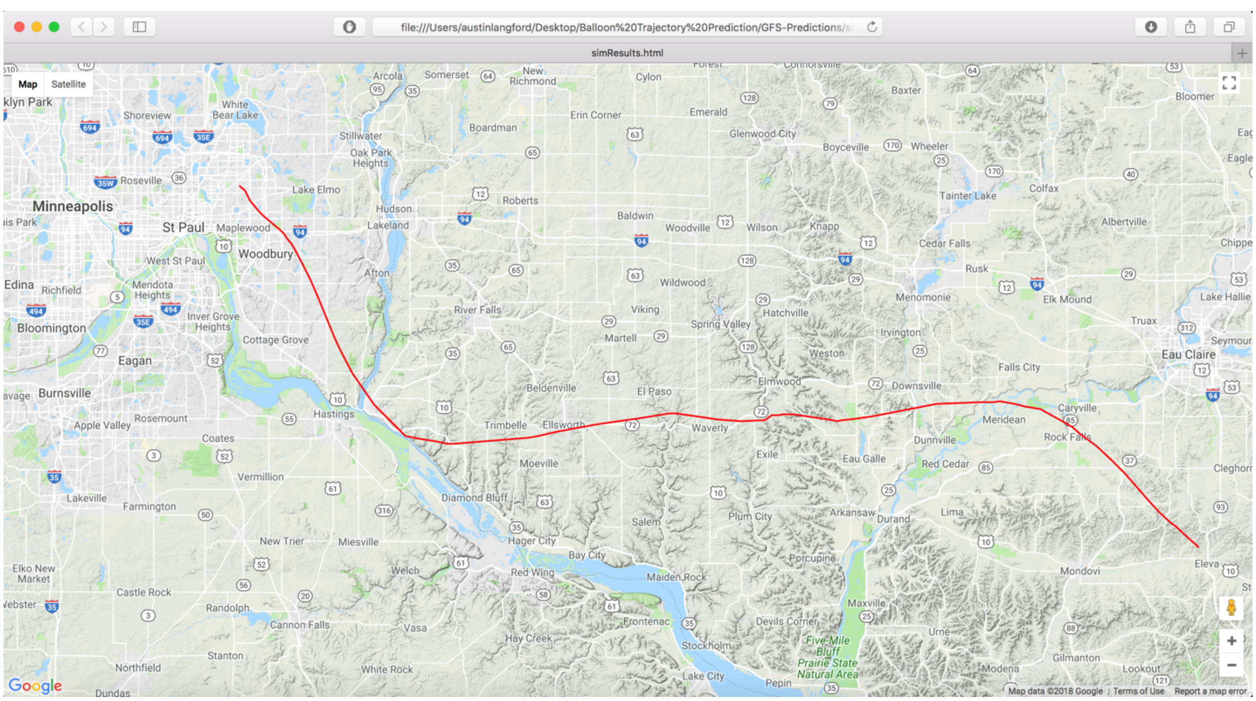
The program will immediately report the latitude, longitude, and time index ranges that correspond to the range specified above. Once the program has retrieved and saved the datasets, it will display the amount of time taken to download the sets, and the directory in which the data was saved:



Select the Inputs tab, press the Select GFS Data button, and select the GFS dataset directory that was downloaded. Then, press the Select Altitude Profile button, and select the altitude profile for the flight. A plot of the altitude profile will appear in the center of the window, and the path to the altitude profile file and GFS directory will be displayed above:



Finally, enter the initial conditions of the flight, and press the Run Simulation button. When the simulation is complete, a Google Maps plot of the flight path will open in the default web browser:



If the flight path displayed is not satisfactory, possibly because it lands in unfavorable terrain, modify the initial conditions and/or try a new altitude vs. time before running another simulation.

Bibliography

1. *A Survey of Current Balloon Trajectory Prediction Technology* – Luke W. Renegar, University of Maryland, College Park, AHAC 2017

2. *Understanding High-Altitude Balloon Flight Fundamentals* – Hank D. Voss, Natalie A. Ramm, and Jeff Dailey, Taylor University, AHAC 2012

3. *The Global Forecast System (GFS) – Global Spectral Model* – NCEP/NOAA, http://www.emc.ncep.noaa.gov/GFS/