GNSS Data Processing Software

Alessandro Gatti
Felix Enyimah Toffah

Supervisor: Prof. Ludovico Biagi

Prof. Gianluca Palermo

September 15, 2021

Introduction

- **GNSS Point Positioning involves measurement of the signals emitted by a satellite for the determination of the position of a receiver on the surface of the Earth. The emitted signal contains information about the clock, ephemerides and integrity of the satellite.**

- **From the ephemeris, the receiver can compute the position of the satellite. From the positions of different satellites, the receiver can derive its own.**

- **The signal propagation through the atmosphere is delayed by the presence of free electrons in the ionosphere (about 100 and 1000 km altitude) and the water vapour content of the troposphere. These delays affect the precision in the estimated position of a receiver.**

Objectives

GNSS Data Processing Software has been developed to work with the GPS constellation standards.

It fulfills the following objectives:

- **Compute the time varying position and velocity of a satellite**
- **Produce a map of the orbit**
- **Produce plots of azimuth & elevation with respect to a receiver**
- **Produce maps showing time varying effects of the ionosphere on the propagated signal**
- **Incorporate a Graphical User Interface for accessing the software**

Development(1)

The program was developed in Python using the following libraries:

- Matplotlib (3.2.0)
- Cartopy (0.18.0)
- Astroplan (0.8)
- Wx (4.1.1)
- Geopandas (0.9.0)
- Numpy (1.18.1)

During the development, some changes were made from the initial draft: the core functionalities stayed the same while the tropospheric and relativistic effect computations were replaced in favor of dedicating more importance to the graphical interface and presentation of the results.

Development(2)

The main implemented libraries are:

- **read_rinex.py:** *reads parameters from the navigation message*

- **sat_orbit.py:** *compute position and velocity of a satellite in time*

- **ionosphericCorrectionSF.py:** *compute ionospheric correction parameters (single frequency receivers)*
- **ionosphericCorrectionDF.py:** *compute ionospheric correction parameters (dual frequency receivers)*

- **geod2cart.py:** *coordinate conversion*
- **cart2geod.py**

- **Main.py:** *defines GUI and the visualized maps*

Data types & formats

- GDPS works with GPS RINEX navigation messages up to version 3.05.

- The parameters and acceptable range of values for user input are defined as follows:

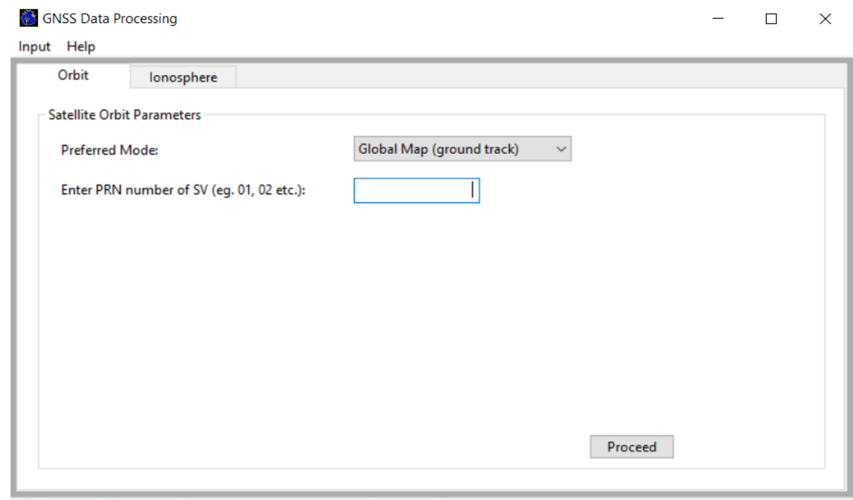
Parameter	Minimum	Maximum
Azimuth (Deg Decimal)	-180	180
Elevation (Deg Decimal)	0	90
Latitude (Deg Decimal)	-90	90
Longitude (Deg Decimal)	-180	180
Time (HH:MM:SS)	00:00:00	23:59:59

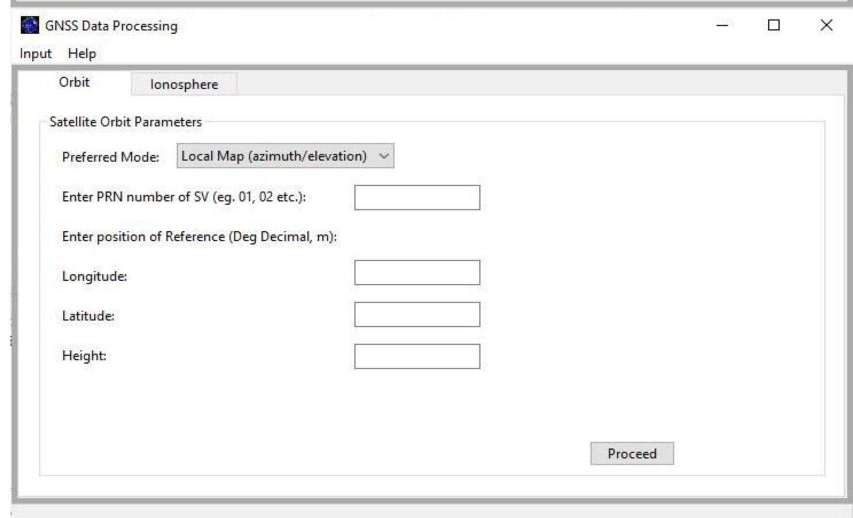
GUI(1)

GDPS has 2 main modules:

1) Satellite Orbit:

- Calculates the positions and velocities of a satellite vehicle over time.
- Allows to visualize the ground track of any chosen satellite over the surface of the Earth, and to show the evolution of azimuth & elevation with respect to an arbitrary user inserted position.





Orbit(1)

Global map

- The software computes position and velocity of satellites every 5 minutes, in a range of 2 hours, after every epoch (entry) in the data. With the available dataset a map is produced visualizing the shape of the orbit of the chosen vehicle, i. e. the ground track with respect to the Earth's surface.

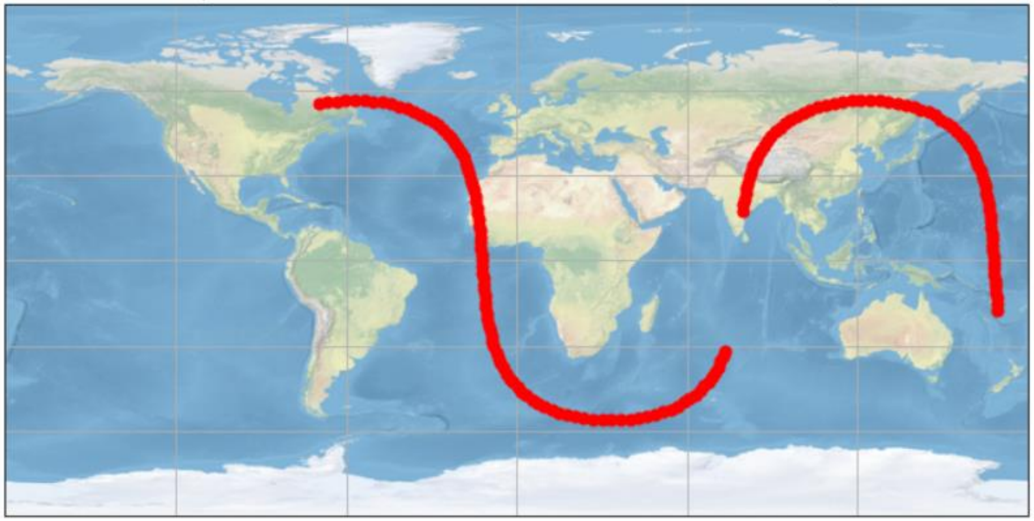
- Checks on the input are put in place to verify that the inserted SV number is valid, if one of the inserted parameters is not valid a warning message will appear.

- For every map produced, a button on the window allows to export it as image.

Orbit(2)

Global map

Satellite G06
(from 2021-01-01 06:00:00 to 2021-01-01 22:00:00)



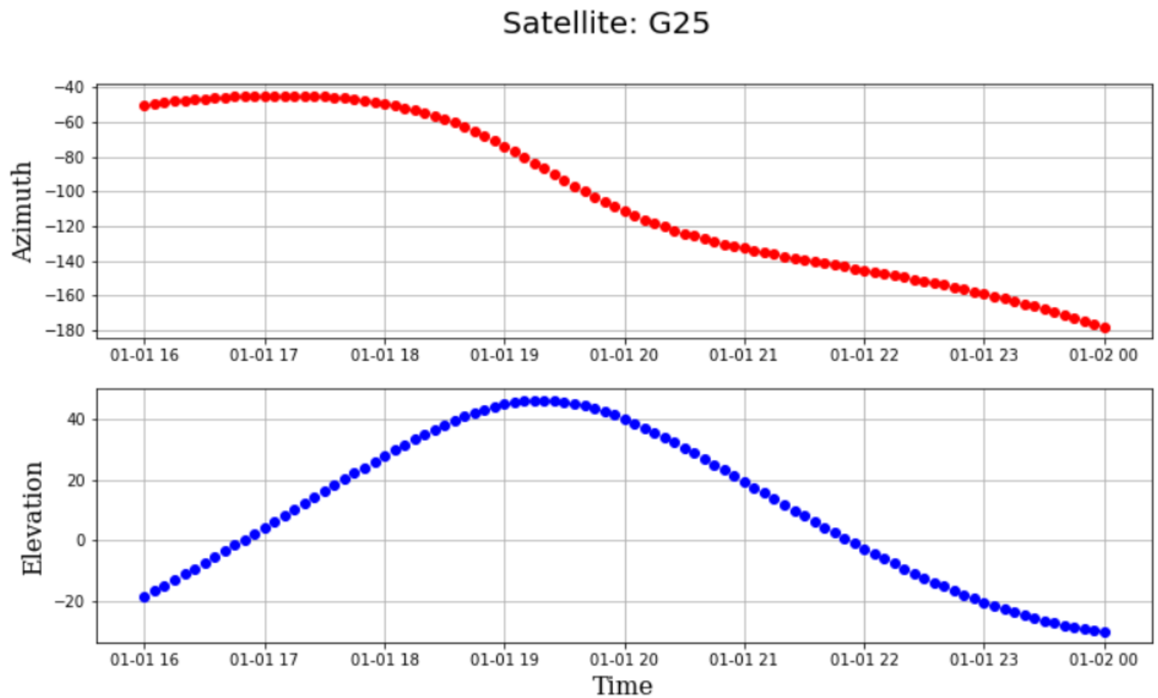
Example: the ground track of sv G06

Note: since the orbit is visualized only for the 2 valid hours after each entry in the navigation data, if there are not enough entries some ‘gaps’ in the orbit may appear, as seen from the above figure

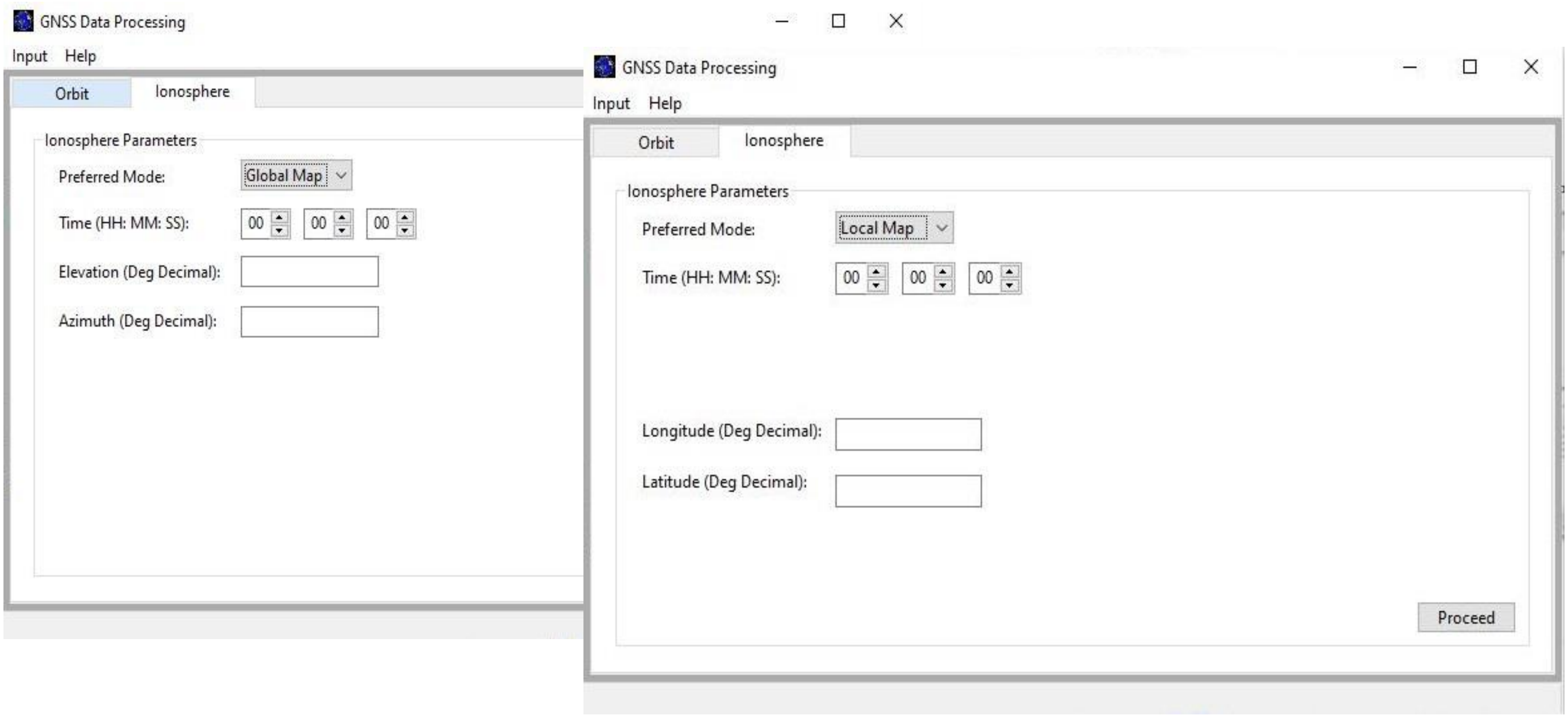
Orbit(3)

Local map

- This functionality shows the difference in time of the azimuth and elevation of a satellite, calculated with respect to an arbitrary position inserted by the user.



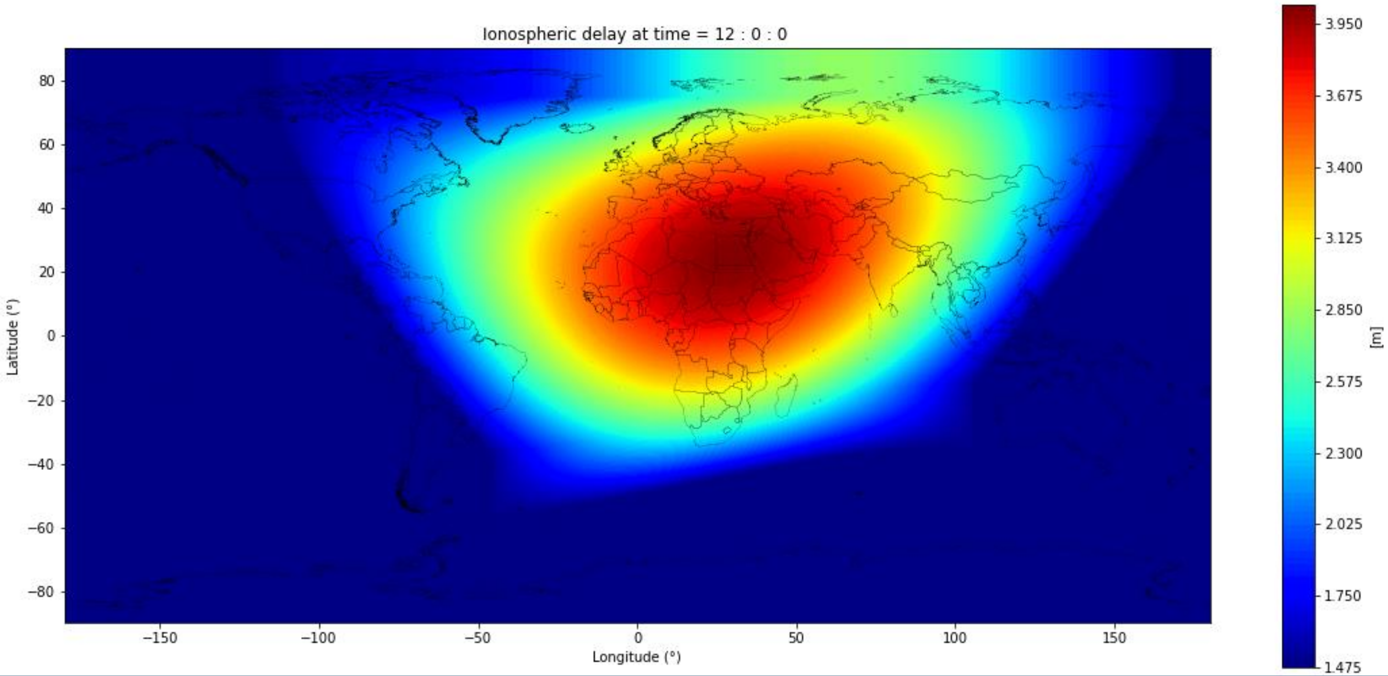
2) Ionospheric error correction



Ionospheric Effect(1)

Global map

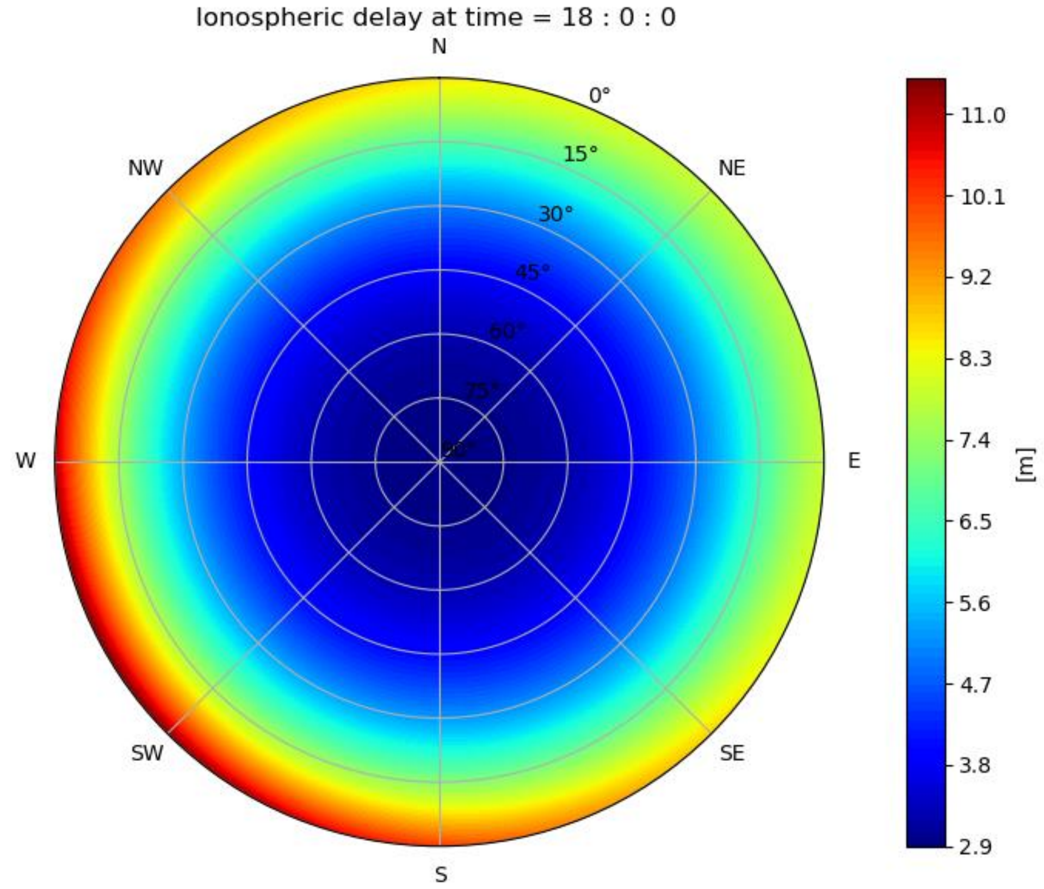
- Shows global ionospheric effect at the message specified time, with regions of higher ionospheric effect shaded red and regions of lower ionospheric effects shaded blue.



Ionospheric Effect(2)

Local map

- Local map of ionospheric effect for varying elevation and azimuth.
- From the graph, it's clear that the ionospheric error is higher for lower elevation of the station with respect to the satellite vehicle.



Testing

A separate python script was developed for comparing the output of the software with the precise ephemerides (i.e. an historical archive where precise positions of satellites are stored). The script takes in input the produced file of positions and the precise ephemerides file (.sp3) and elaborates a text file of the differences.

Epoch	svPRN	x_diff	y_diff	z_diff	x_sv	y_sv
['2021', '1', '1', '0', '0']	G10	-0.744607	-0.440898	-0.102201	2.0298e+07	1.17726e+07
['2021', '1', '1', '0', '0']	G12	-0.469383	-0.181252	0.0983903	-1.28158e+07	1.13929e+07
['2021', '1', '1', '0', '0']	G15	-0.153541	-0.394982	0.267578	-7.53408e+06	1.62594e+07

After testing with thousands of results, the difference has always been in the range of some meters.

['2021', '1', '1', '23', '45']	G29	-0.522345	0.0534896	-0.48207	3.91971e+06	2.58871e+07
['2021', '1', '1', '23', '45']	G31	0.0695322	0.335648	-0.0839948	1.77013e+07	1.50995e+06
['2021', '1', '1', '23', '45']	G32	-0.632529	-0.412062	0.39824	1.56971e+07	1.71673e+07

```
Maximum abs difference detected: 3.719988513737917
```

Conclusions

Python libraries for the estimation of the position of a satellite by reading the observation data of RINEX format as well as the ephemerides information has been implemented.

In addition a GUI has also been developed to enable a user easy access to the implemented libraries and for visualization of the results in maps.

Future development:

- The software can be upgraded adding support for newer versions of navigation message files.
- Other modules can be integrated, adding functionalities for computation of tropospheric effect delay and for correction of the relativistic effect.

References

Data specifications and algorithms implemented from:

- Interface Specification IS-GPS-200L, August 2020
- Interface Specification IS-GPS-705G, August 2020
- Interface Specification IS-GPS-800G, August 2020
- Rinex300, November 2007
- Rinex305, December 2020