Evaluation of Global Ionospheric Electron Density using Simultaneous Observations from Amateur Radio Networks, International Space Station, and NeQuickG Model for Space Weather Prediction

Gamal Zayed[1]

Marcin Leśniowski

P.B.S. Harsha[2] VU3HPH

Matt Downs 2E1GYP

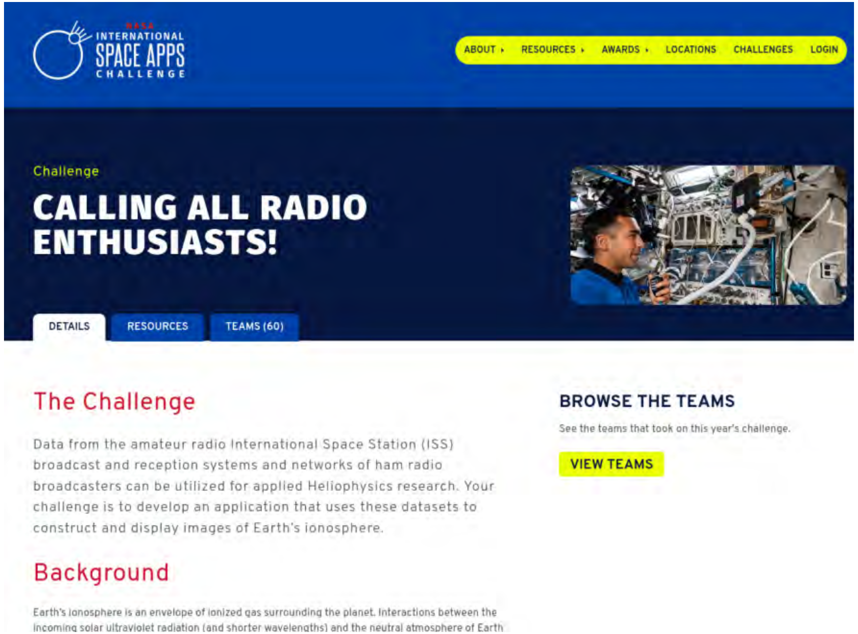
Daniel Metcalfe

Sıla Kardelen Karabulut

[1]Cairo University, Egypt
[2]CentraleSupelec, France

Image credit: NASA Space Apps

Space Apps Challenge



- Develop an application that uses information from ISS and HAM radio broadcasts
- Display images of Earth's ionosphere
- High-temporal and high-spatial resolution

https://2022.spaceappschallenge.org/challenges/2022-challenges/radio-enthusiasts/details

Hackathon time: 48 hours (Weekend)

The Fellowship of the Ionosphere

Daniel / Frontend Developer

Gamal / Researcher / Ionospheric model

Harsha / Researcher / ISS data

Matt / RF tech / HAM Data

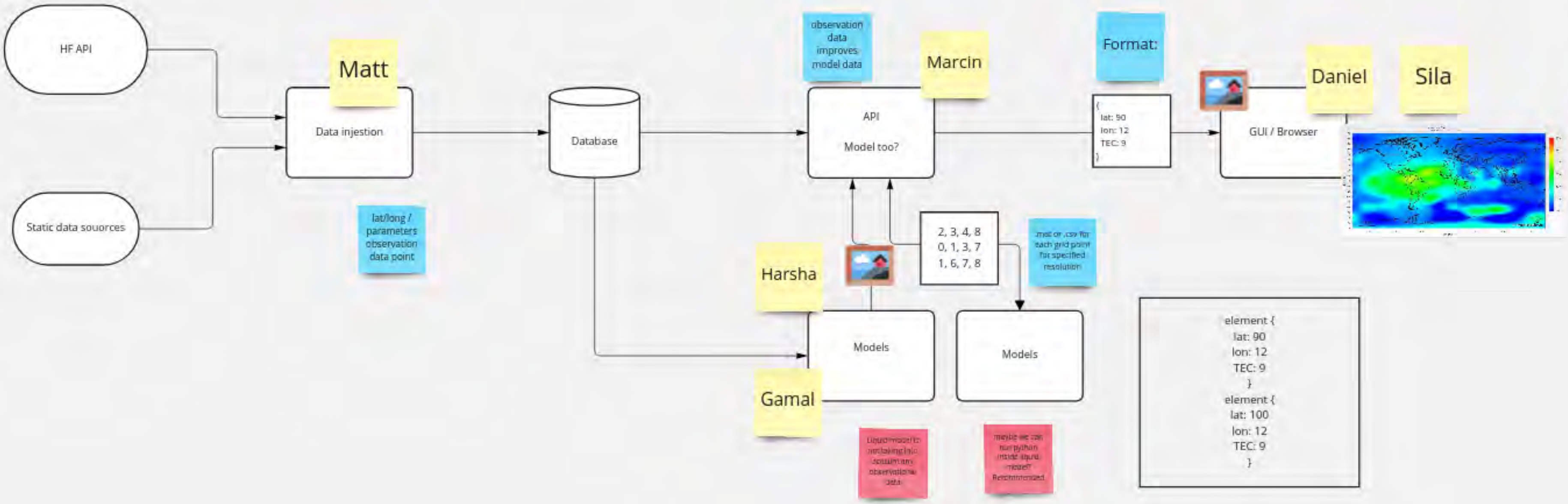
Sıla / Project Manager / Management

Marcin / Developer / Glue

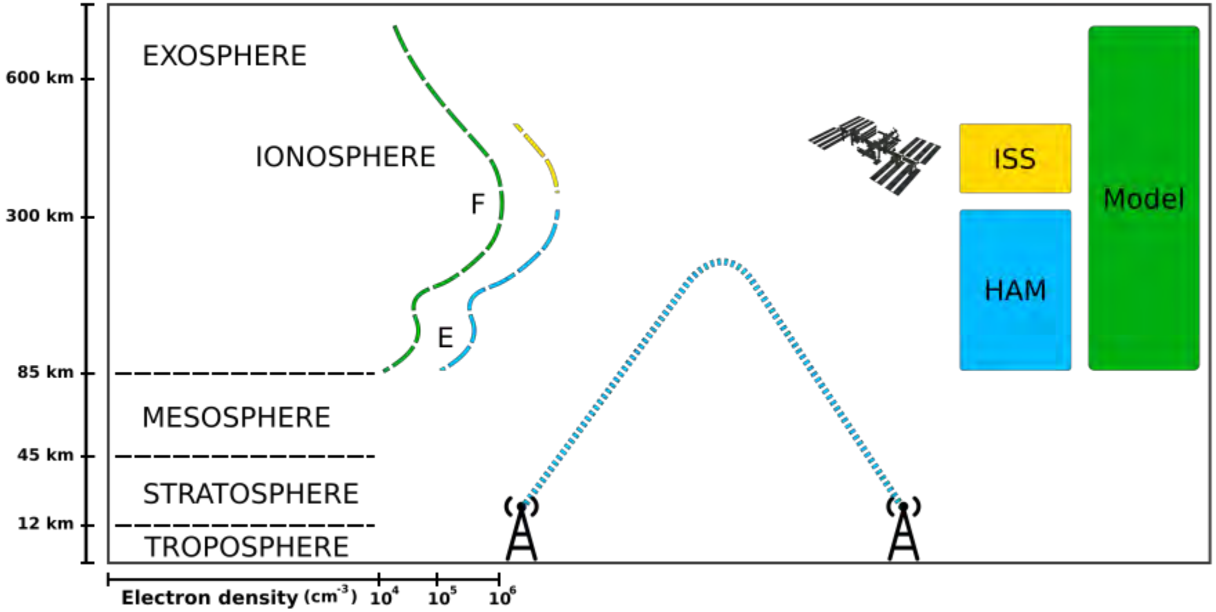
Our Goal

- Web application to allow more exposure (broader audience)
  - no technical knowledge needed to use it
  - public website can promote ionospheric research
  - splitting frontend and backend allows for independent development
- Be flexible / future proof
  - We might want to eventually combine multiple sources of data
- Be able to see / compare multiple sources of data
  - HAM
  - ISS
  - Model
  - (possibly others)

Individual work schedule



Why multiple sources (vertically)



- **Intro**

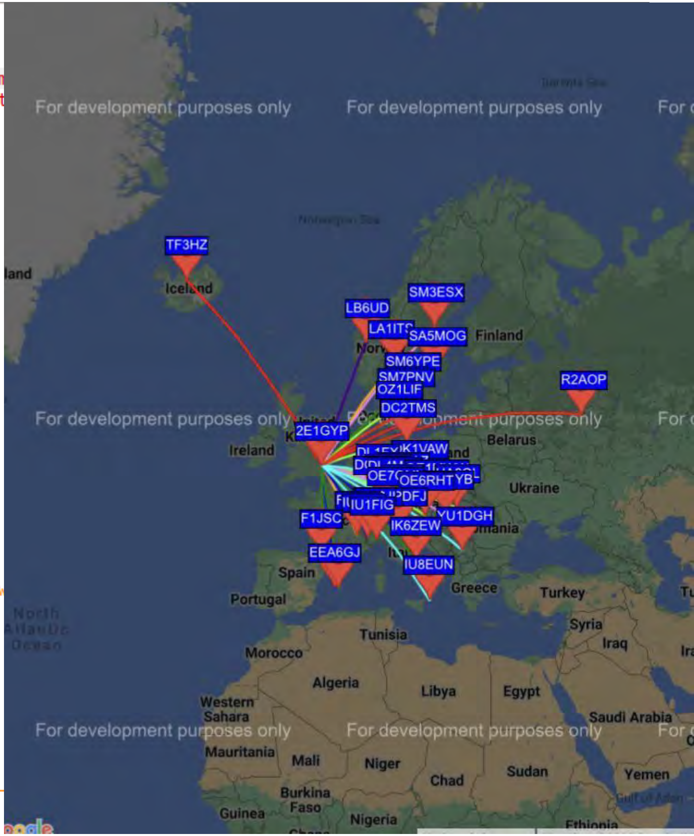
1. HAM radio reporting network (WSPR)
2. Floating Potential Measurement Unit (FPMU) onboard ISS
3. NeQuickG global ionospheric model

- **Summary**

Amateur Radio Networks

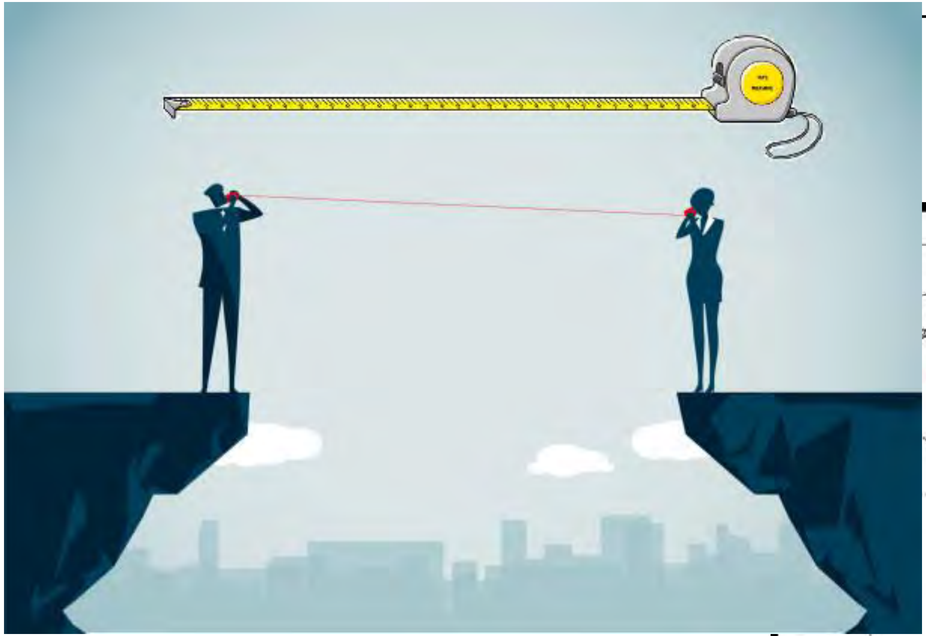
Weak Signal Propagation Reporter Net.

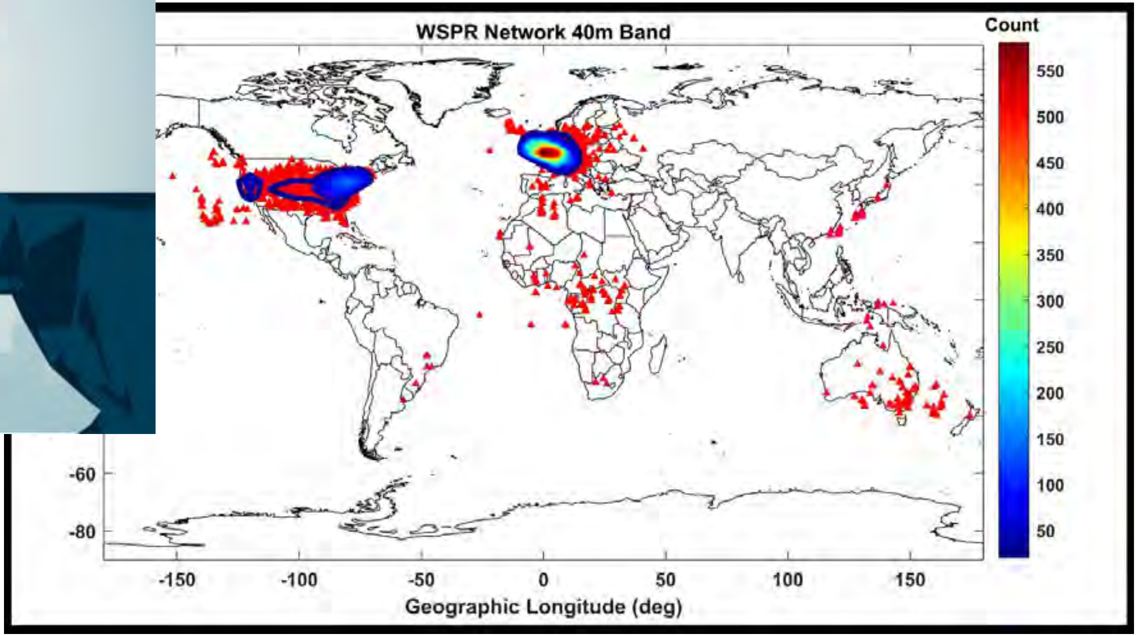
```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# Matthew Downs, 2E1GYP, FELLOWSHIP OF THE IONOSPHERE
# This script queries the WSPRnet database for 2 hours (up to a maxim
# Each spot link is derived to a geographic mid-point.  This midpoint
# and ingested into the database serving our mapping application.
# In the finished project, this process would auto-execute hourly.
import requests
import json
import pandas as pd
import csv
from cmath import pi
from math import radians, cos, sin, sqrt, degrees, asin, atan2, acos
import math
import os
from csv import reader
import re

print("Retrieving data, Please wait...")

# Remove old data if it is present
try:
    os.remove("HF_Midpoint.csv")
except:
    pass

# Query WSPRnet database for last 1 hour of spots
url = 'http://dbsrv1.wspr.live/?query=select%20*%20from%20wspr.rx%20w
x = requests.post(url)
output = x.text

print("Data succesfully retrieved")

# Create csv to hold spots
f = open("HF_Midpoint.csv", "a")
# Write column headers to csv
f.write("id,time,band,rx_sign,rx_lat,rx_lon,rx_loc,tx_sign,tx_lat,tx_
f.write("\n")
line=""
```



space.foti@gmail.com

Method 1



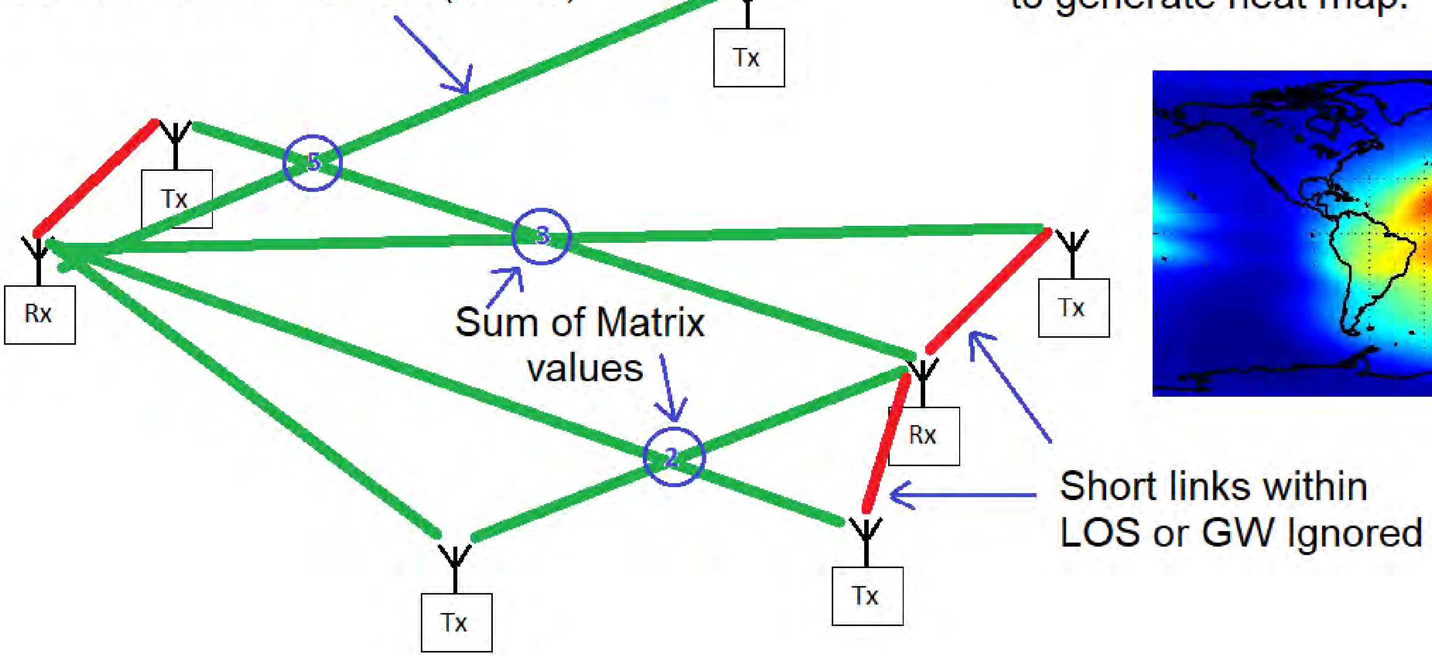


WSPR Network 40m Band
Count
550
500
450
400
350
300
250
200
150
100
50
-60
-80
-150
-100
-50
0
50
100
150
Geographic Longitude (deg)

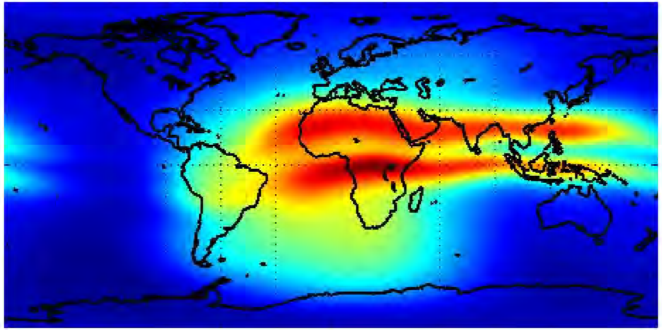
Method 2

Assigned a Matrix value with...

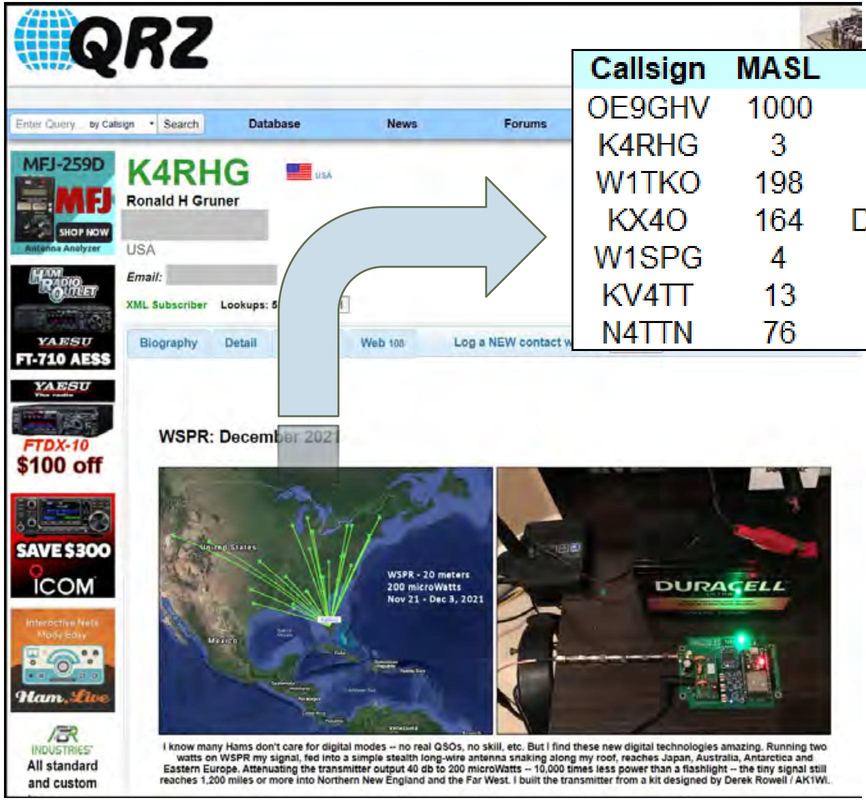
TxdB – FSPLdB + RxdB x (km/10)



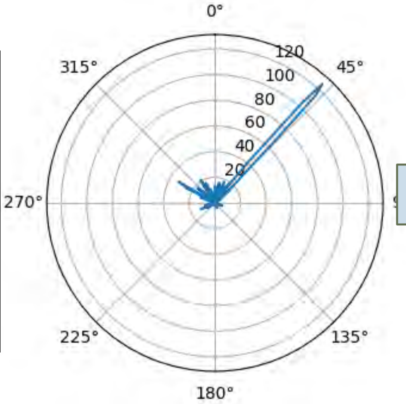
Transpose lat/lon & Intersect Values to generate heat map.

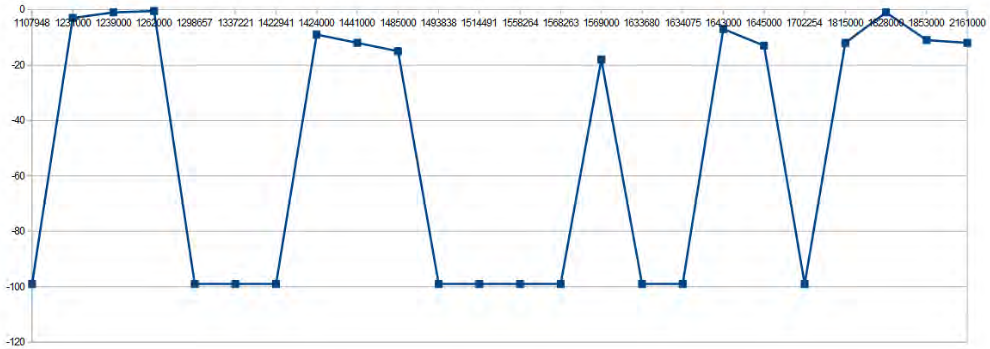


Future research

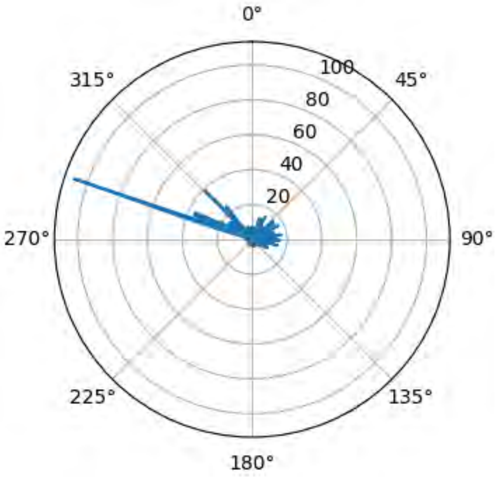


Callsign	MASL	Antenna	V/H
OE9GHV	1000	Windom	H
K4RHG	3	Long-wire	H
W1TKO	198	Fan dipole	H
KX4O	164	DXE MBVE-5A-4UPR	V
W1SPG	4	Dipole	H
KV4TT	13	Trap dipole	H
N4TTN	76	Cushcraft R8	V

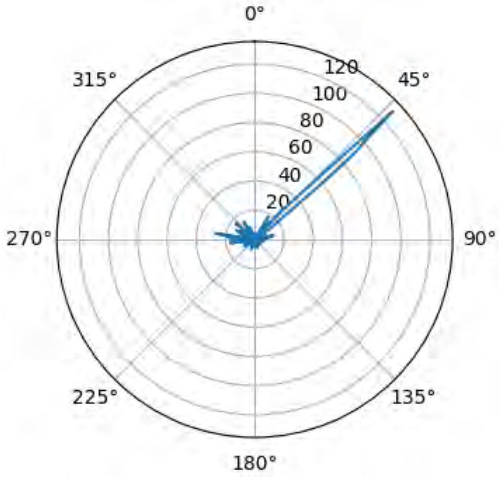




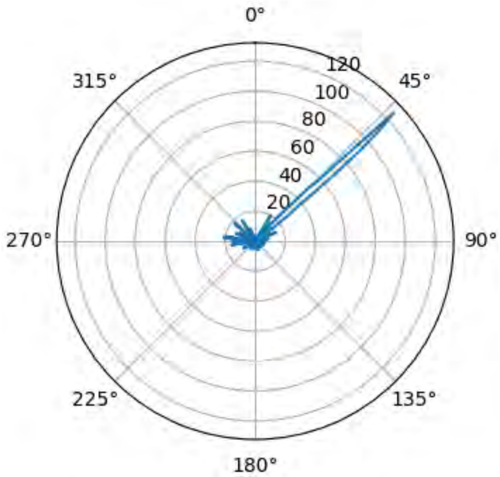
space.foti@gmail.com



Gwyn G3ZIL
South Coast England

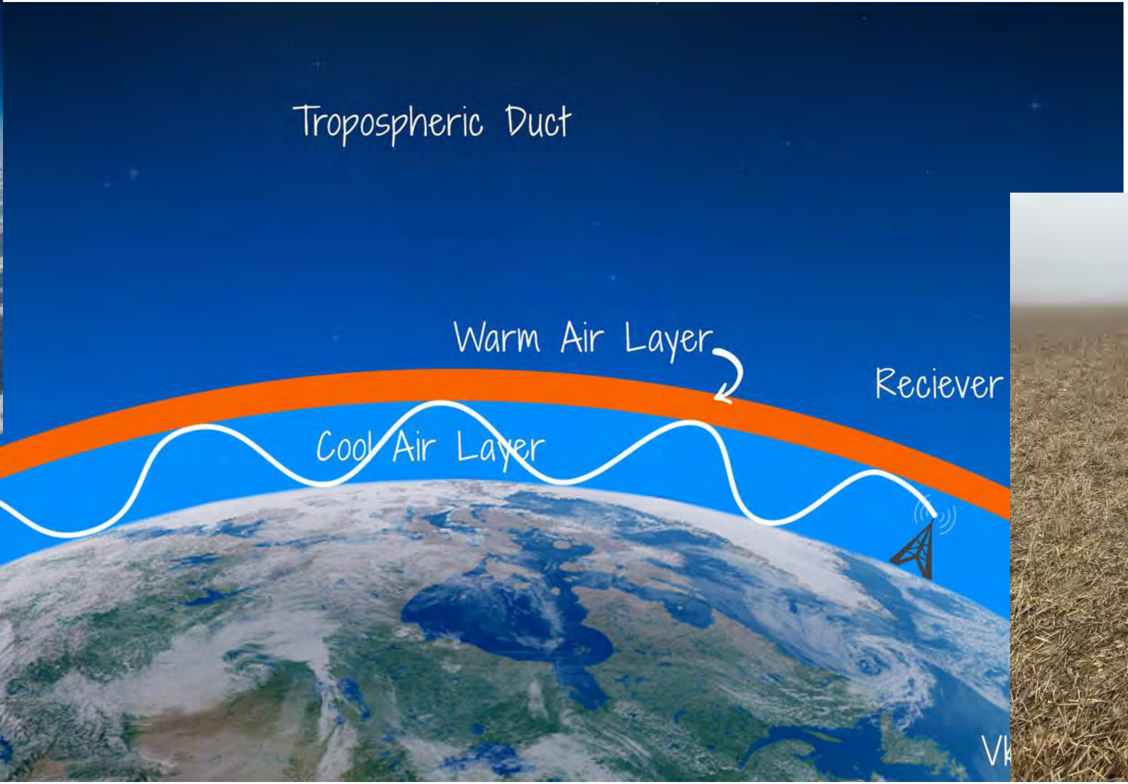


John N8OBJ
Case Western Reserve University



W8EDU
Case Amateur Radio Club

Future research



space.foti@gmail.com

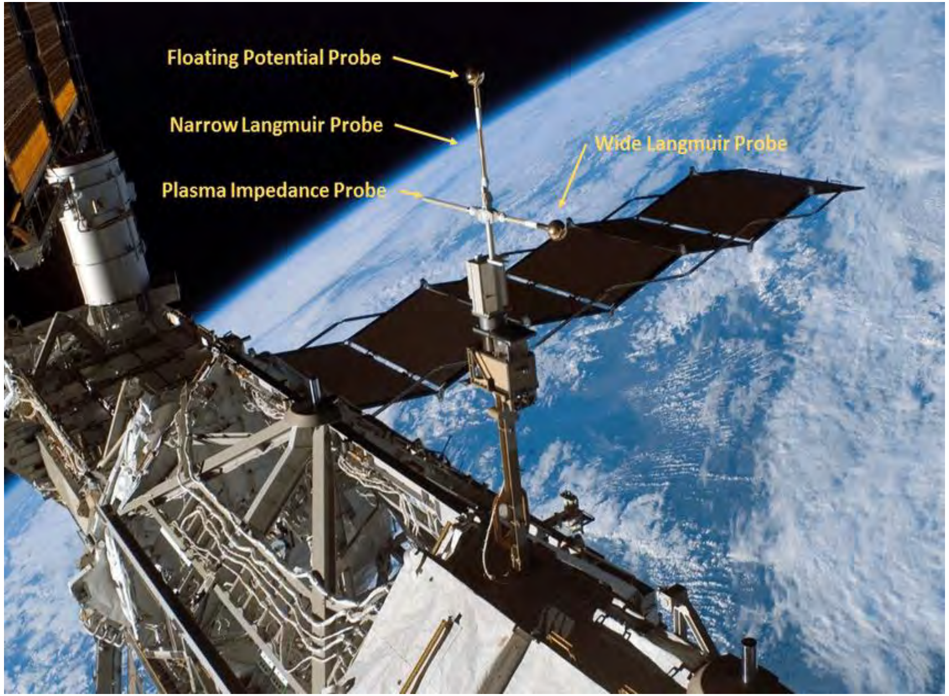
International Space Station (ISS)

1. ISS is equipped with a Floating Potential Measurement Unit (FPMU).

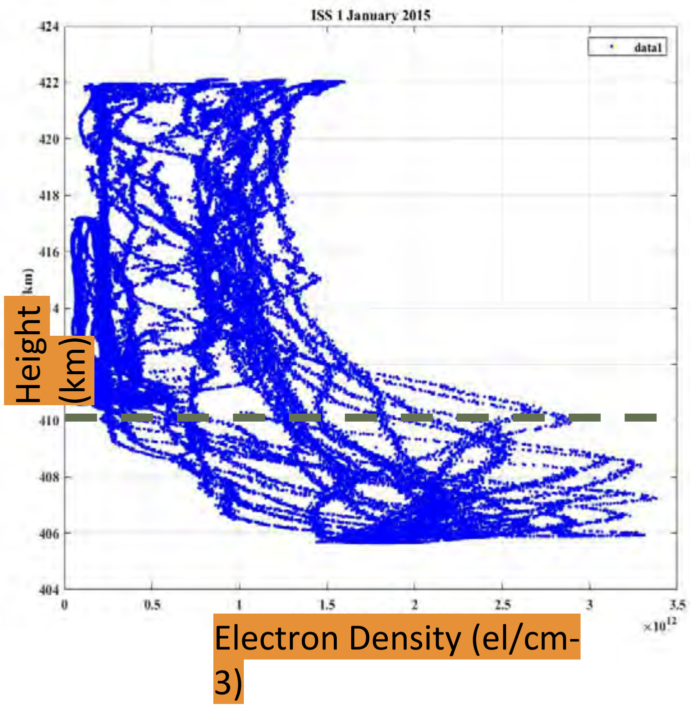
2. FPMU is a collection of 4 probes that are used to measure the ISS floating potential as well as the electron density and temperature of the local plasma environment.

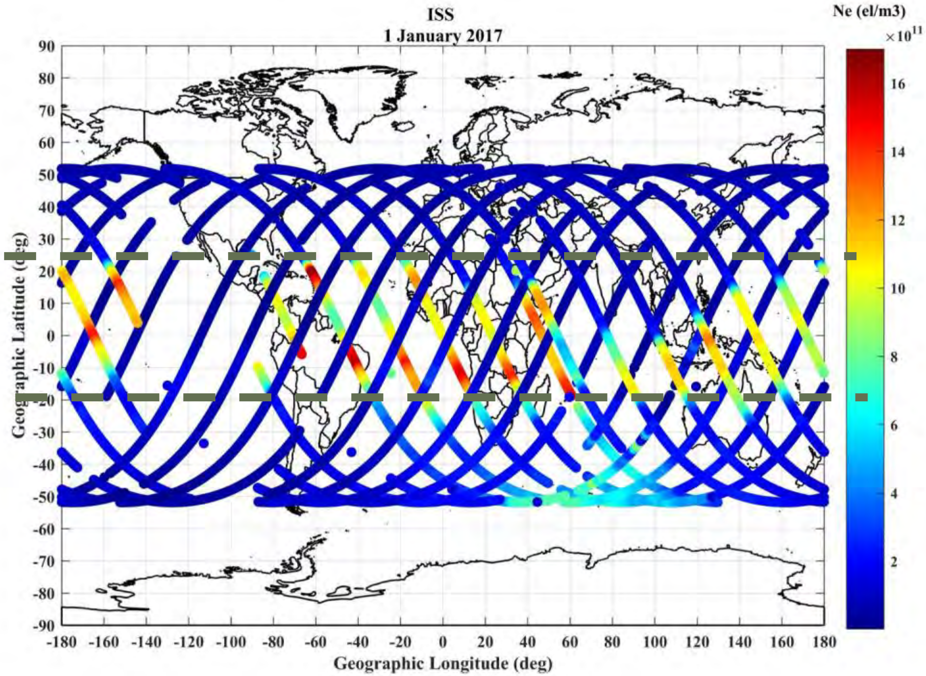
3. Download the data from below link: https://spdf.gsfc.nasa.gov/pub/data/international_space_station_iss/sp_fpmu/

Top-side Electron Density Profile (from ~400 km to ~450 km)



Diurnal variation of electron density



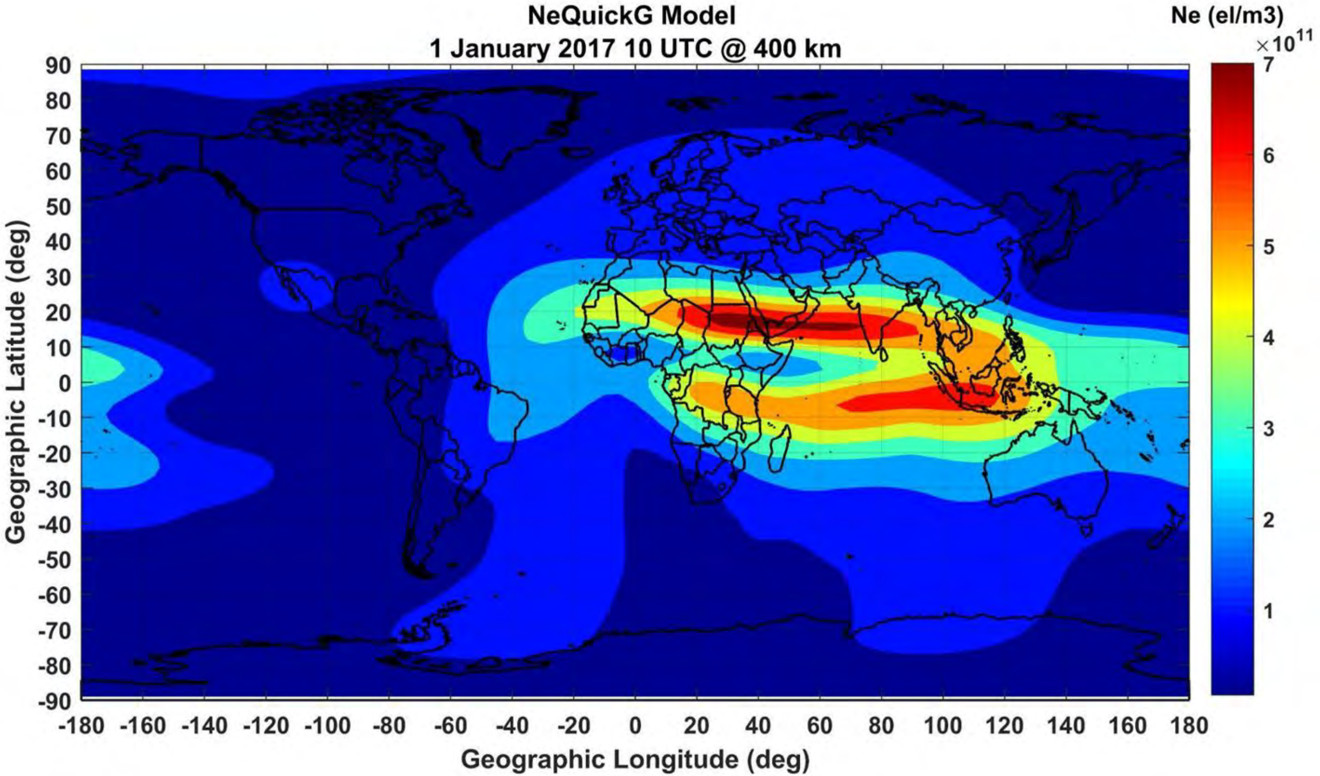


ISS data could be useful for better prediction of space weather impacts.

ISS daily orbital trajectory for the 1 January 2017

space.foti@gmail.com

NeQuick G Model

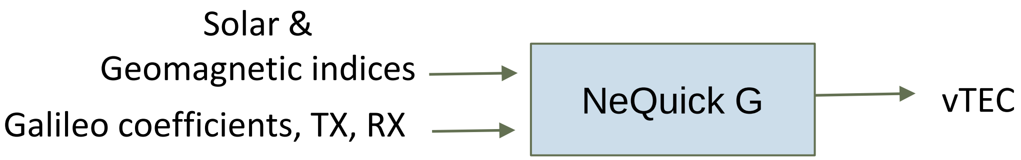


space.foti@gmail.com

NeQuick G Model: About

1. NeQuick G model is a global ionospheric model
2. provides better spatial and temporal resolution.
3. developed by
   a. **I**nternational **C**enter for **T**heoretical **P**hysics (ICTP)
   b. University of Gruz
4. As a single frequency model to provide ionospheric corrections for the GNSS user community.

1. Now, the **E**uropean **S**pace **A**gency officially approves the NeQuick G model as a signal of service for the Galileo Users.



space.foti@gmail.com

Methodology: Galileo Coefficients

European Union Agency for the Space Program

BRDC00IGS_R_20190010000_01D_MN.rnx - Notepad

File Edit Format View Help

```
     3.03           N: GNSS NAV DATA    M: MIXED            RINEX VERSION / TYPE
MergeMNfile.tcl     IGS                 20190711 014358 GMT PGM / RUN BY / DATE
GAL    2.7750D+01  3.7500D-01  1.0681D-03                   IONOSPHERIC CORR
GPSA   7.4506D-09 -1.4901D-08 -5.9605D-08  1.1921D-07       IONOSPHERIC CORR
GPSB   9.2160D+04 -1.1469D+05 -1.3107D+05  7.2090D+05       IONOSPHERIC CORR
BDSA   6.5193D-09  8.1956D-08 -8.3447D-07  1.7285D-06 X 02  IONOSPHERIC CORR
BDSA   6.5193D-09  8.1956D-08 -8.3447D-07  1.7285D-06 W 05  IONOSPHERIC CORR
BDSA   6.5193D-09  8.1956D-08 -8.3447D-07  1.7285D-06 X 06  IONOSPHERIC CORR
```


```
     3.03           N: GNSS NAV DATA    M:
MergeMNfile.tcl     IGS                 201
GAL    2.7750D+01  3.7500D-01  1.0681D-03
```

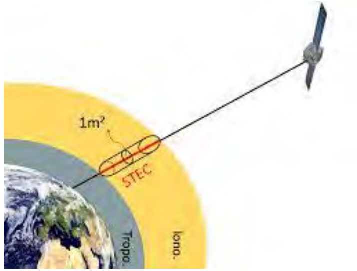

```
BDSA   6.5193D-09  8.1956D-08 -1.2517D-06  3.3975D-06 U 19  IONOSPHERIC CORR
BDSA   6.5193D-09  8.1956D-08 -9.5367D-07  1.9670D-06 U 25  IONOSPHERIC CORR
BDSA   5.5879D-09  8.9407D-08 -1.0133D-06  2.2650D-06 O 26  IONOSPHERIC CORR
BDSA   6.5193D-09  8.1956D-08 -9.5367D-07  1.9670D-06 U 27  IONOSPHERIC CORR
BDSA   5.5879D-09  8.1956D-08 -9.5367D-07  1.9670D-06 W 28  IONOSPHERIC CORR
BDSA   5.5879D-09  8.9407D-08 -1.0133D-06  2.2650D-06 F 29  IONOSPHERIC CORR
BDSA   5.5879D-09  8.9407D-08 -1.0133D-06  2.2650D-06 V 30  IONOSPHERIC CORR
```

Code

```python
""" Input Parameters here """
mth = 11 #January
UT = 0 #Universal Time
electron_density = []  # two-dimensional array  lat x
lat = []  # [y/4 for y in range(-89, 91)]  #90#40
element_lat = -89
while element_lat <= 90:
    lat.append(element_lat)
    element_lat += 2.5
lon = []
element_lon = -180
while element_lon <= 180:
    lon.append(element_lon)
    element_lon += 5
required_height_of_study = 400  #in Kms
## Gallileo Coefficients backing to January 1st, 2017
a0 = 4.4000e+01
a1 = 3.8281e-01
a2 = -1.8616e-03
""" Processing Below """
# Create input objects
TX = NEQTime(mth, UT)
BX = GalileoBroadcast(a0, a1, a2)
hs = required_height_of_study
```

```python
""" Input Parameters here """
mth = 11 #4
UT = 10 #2
electron_density = []  # two dimensional array  lat x lon
lat = []  # [y/4 for y in range(-89, 91)]  #90#40
element_lat = -89
while element_lat <= 90:
    lat.append(element_lat)
    element_lat += 2.5
```

NeQuick G Parameters




```python
class NequickG_parameters:
    def __init__(self, pos, broadcast, time):
        self.Position = pos  # Nequick position object
        self.Broadcast = broadcast  # Nequick broadcast object
        self.Time = time  # Nequick time object
        self.compute_parameters()
```

```python
def compute_parameters(self):
    # Stage 1
    self.__compute_MODIP__()
    self.__effective_ionization__()
    self.__effective_sunspot_number__()
    self.__solar_declination__()
    self.__solar_zenith__()
    self.__effective_solar_zenith__()
```

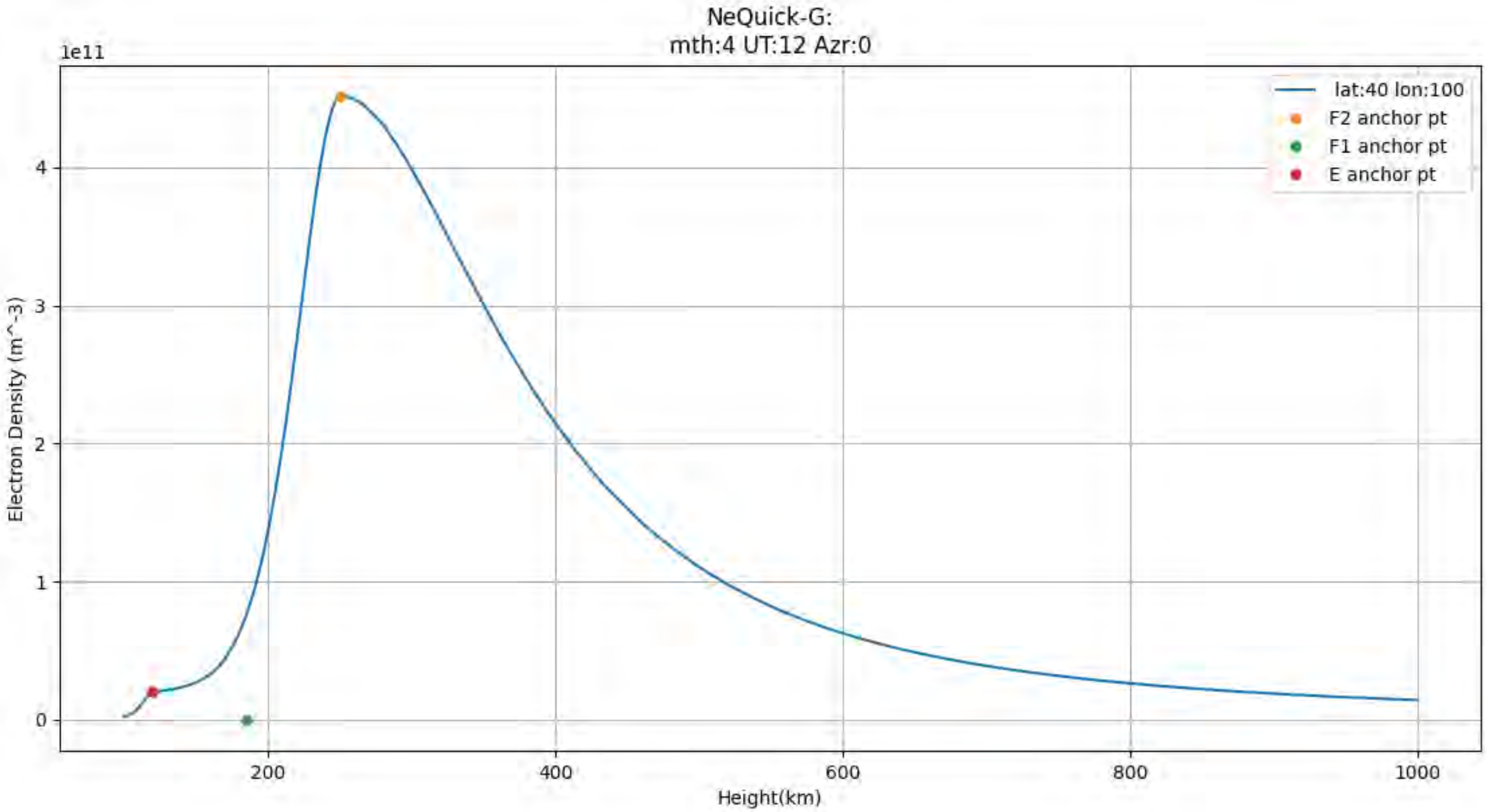
```python
# Stage 2
self.__readccirXXfiles__()
self.__interpolate_AZR__()
self.__F2fouriertimeseries__()
self.F2Layer()
self.ELayer()
self.F1Layer()
```

```python
# Stage 3
self.get_hmE()
self.get_hmF2()
self.get_hmF1()
```

```python
# Stage 4
self.get_B2bot()
self.get_B1top()
self.get_B1bot()
self.get_BEtop()
self.get_BEbot()
```

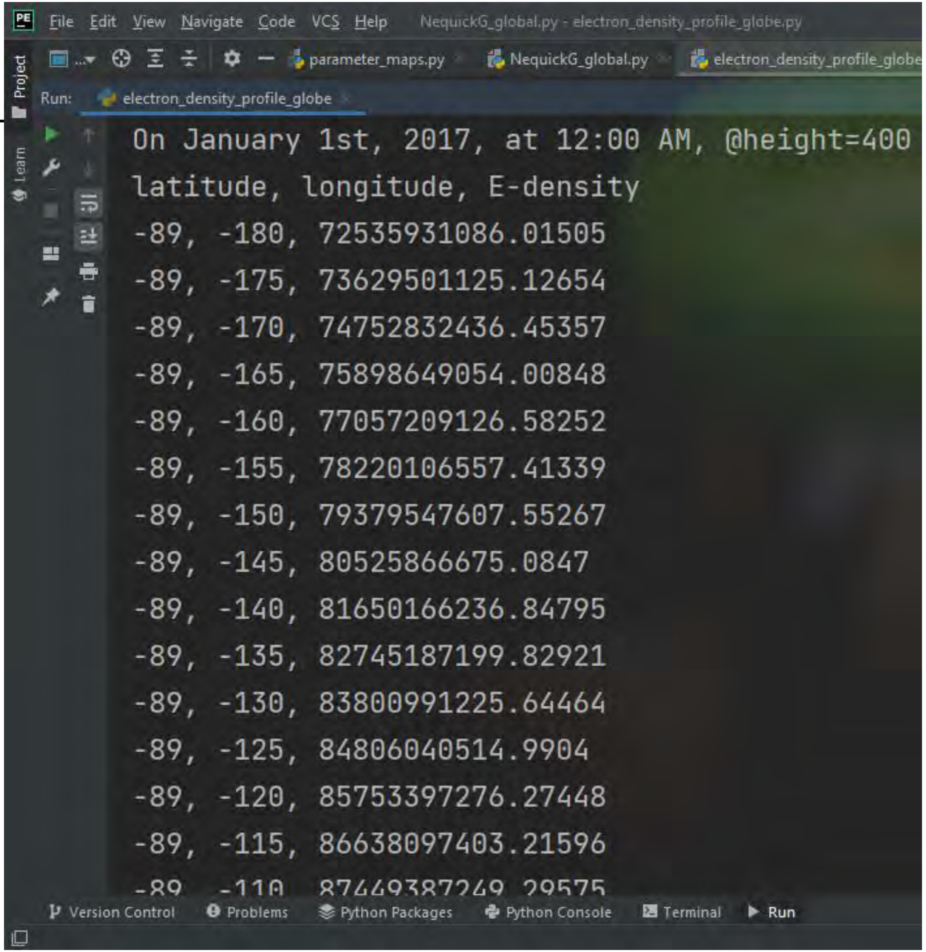
```python
# Stage 5
self.get_A1()
self.get_A2A3()
self.shape_parameter()
self.get_H0()
```


space.foti@gmail.com

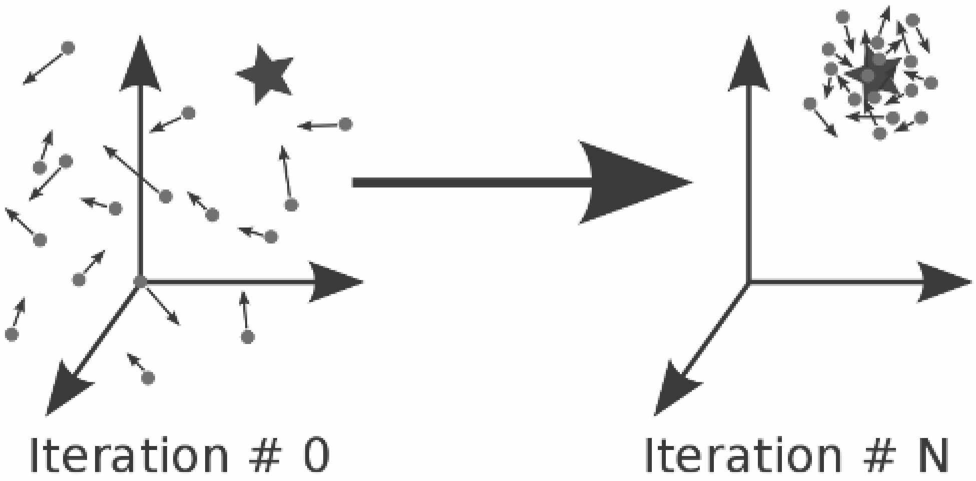


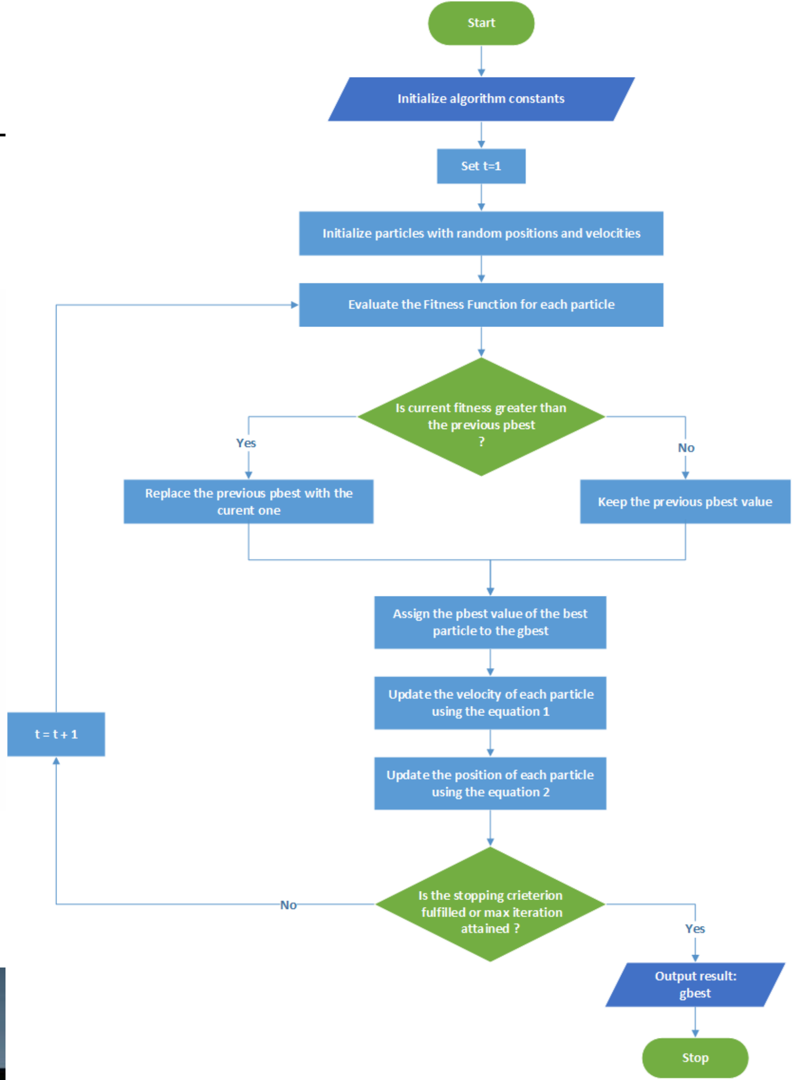
NeQuick-G:
mth:4 UT:12 Azr:0
1e11
Electron Density (m^-3)
Height(km)
0
1
2
3
4
200
400
600
800
1000
lat:40 lon:100
F2 anchor pt
F1 anchor pt
E anchor pt

Output Sample

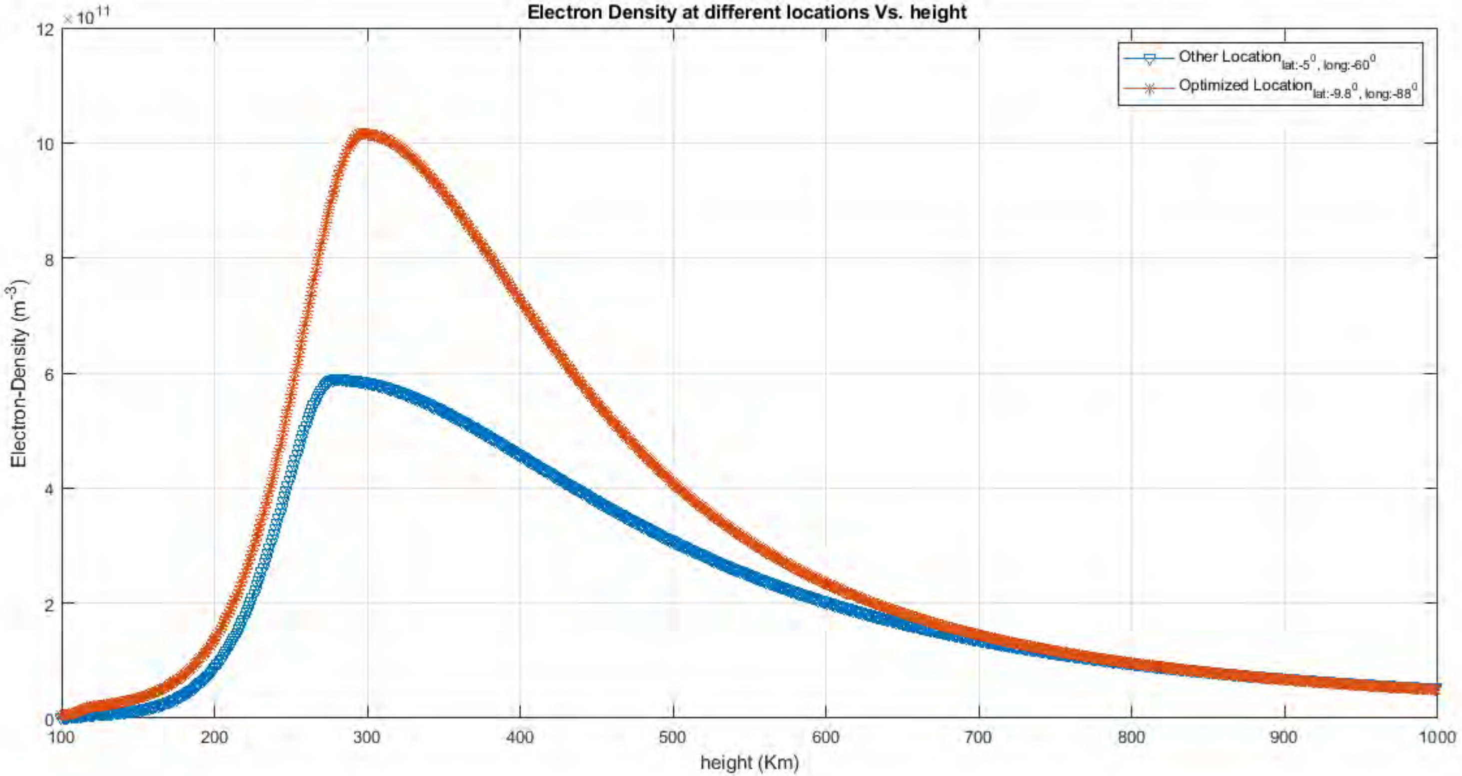


Particle Swarm Optimization



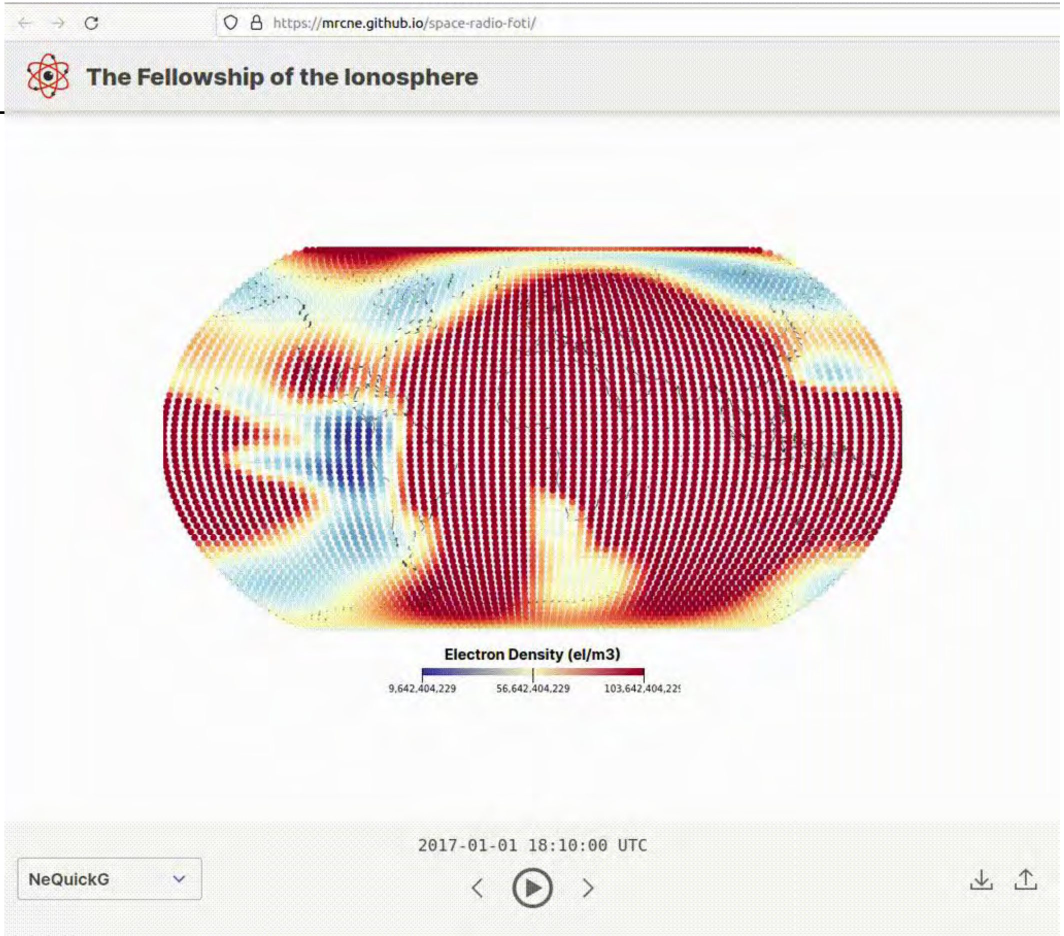


PSO optimized location



Results / App

https://mrcne.github.io/space-radio-foti/



Summary

1. The project is still in early state of development, but we’re still collaborating and having regular meetings.
2. We invite people to try the app and give us feedback despite it being far completion.
3. In the future it might help users to understand ionospheric electron density and its hourly variations.
4. HAM radio broadcast data from WSPR Network is utilized to approximate the bottom side of the ionosphere.
5. ISS provides measurements of the top-side electron density.
6. NeQuick G model is useful to study electron density variations with great spatial and temporal resolution.
7. Utilization of the three data sources could help in better ionospheric state prediction in the future.

space.foti@gmail.com

Links and Resources

- Contact us at: space.foti@gmail.com
- Feel free to explore the web app: https://mrcne.github.io/space-radio-foti/
- Source code: https://github.com/mrcne/space-radio-foti
- Space Apps Challenge: https://2022.spaceappschallenge.org/challenges/2022-challenges/radio-enthusiasts/details
- Challenge project: https://2022.spaceappschallenge.org/challenges/2022-challenges/radio-enthusiasts/teams/fellowship-of-the-ionosphere/project
- Weather balloon data: https://www.ncei.noaa.gov/products/weather-balloon/integrated-global-radiosonde-archive
- A survey of the techniques for measuring the radio refractive index (nist.gov)

Thank you / QA

space.foti@gmail.com