# Instructions:

**Please submit all work products, including documentation, through your open public github account and discuss these during the technical interview**. Please complete as many assessment challenges as you would like. No hard timeline and open-bookstyle, so submit when you feel ready and have a good high-level understanding of what we do. These exercises are extracted from our working code and are intended to provide realistic examples of what we do to support AFRL R&D ionospheric impact application development. Evaluation criteria are based on thoroughness, attention to detail, problem-solving, and communication in writing and verbal.

## Assessment IRI EDP: Create a C-based modeling and simulation program that drive IRI model Fortran code. The code should capture and generate vertical EDP (Electron Density Profile) for a given time and location of interest.

time of interest: Mar 3 2021 UT 11:00:00 and Mar 4, 2021 UT 23:00:00

location o interest: Lat 37.8N and Lon 75.4W

1. Create a simple Makefile that can compile iri2016 (<http://irimodel.org>) and generate a shared object/library
2. Write a C-program that links with the shared object created and create all data needed for step
3. Use gnuplot ([www.gnuplot.info](http://www.gnuplot.info)) or other similar C-based plotting tools to generate plots of EDP parameters using the shared objective created in step 1.
4. alternatively, use F2PY (<https://www.numfys.net/howto/F2PY/>) and Python to create EDP plots using the shared object created in step 1. (Although C-based plotting is the preferred solution)
5. Furnish instructions/documentation, etc. on how to run the code and lesson/insights learned by doing this exercise.

I don’t believe I accomplished this as intended, but I was able to create a shared object which was compiled with a c program that generates IRI plots through gnuplot. I believe this program was intended to pass data from the C script itself into the IRI\_WEB subroutine and then return the desired data into arrays in C. While I was able to accomplish passing data from c into fortran, I struggled to both pass data back from fortran to C. I ended up wrapping IRI\_test.for in a subroutine which I called from my C script which requires user input on the command line.

Additionally, I wanted to focus on getting the program up and running before generating a makefile, so this was unfortunately not accomplished either. Instead, the compilation is done through a simple shell script. See iri\_plot.c and iri\_compile.bash.

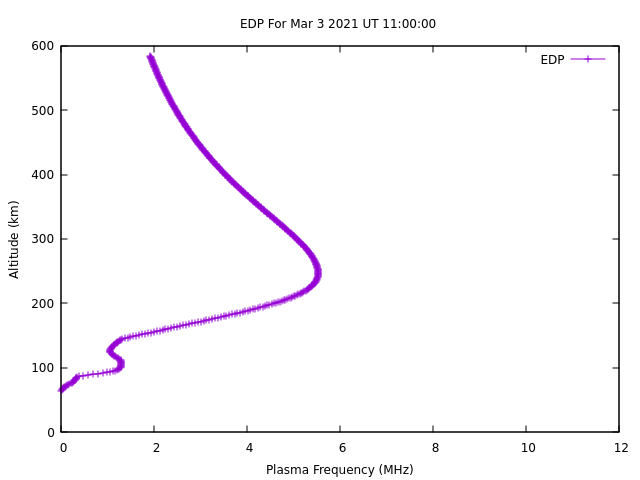
To compile the program, simply run iri\_compile.bash in the same directory as iri\_plot.c, the modified version of iri\_test.for which I created, and the rest of the IRI 2016 fortran and data files. Then the program can be run by simply executing iri\_plot from the command line and entering the desired ionosphere parameters when requested. These parameters include date, time, location, ionosphere boundaries, among others.

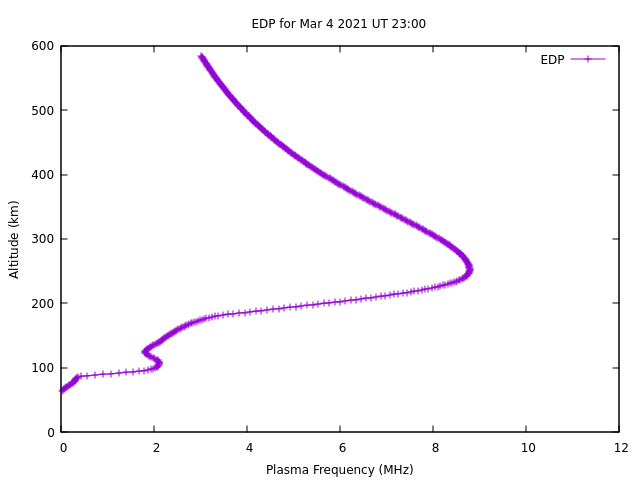
The included EDP profiles were generated using the two times of interest with the UT switch (1) activated and the location of interest using the geog switch (0). A variable ionosphere height was requested from 0 to 600 km with a step of 1km and the standard table of IRI parameters was then requested, with total electron count extending to 600 km.

Ultimately, this was a very useful exercise in that it required the consolidation of many disparate aspects of software creation I knew but had never combined as well giving me a lot of insight into aspects of software that I had very little experience with. I was especially shocked by the difficulty of navigating and working with a large fortran 77 codebase, which I had never fully appreciated before this. While I’m a little unhappy I wasn’t able to fully complete this exercise as intended, I’m very satisfied with how much I learned while doing it. Chart

Description automatically generatedChart

Description automatically generated





Ionosphere Modeling and Ray Tracing

A - Quasi Parabolic Model

Let us look at a simplified and naive model of the Ionosphere in one dimension (r - the distance from the center of earth) where:

Text

Description automatically generated

This defines a single layer model of the Ionosphere with a quasi-parabolic description.

The layer starts at distance rb from the center of earth and ends at rmrb/(rb-ym).

* Ne(r) is the electron density number (EDP) i.e. number of electrons per cubic meter at distance r from the center of earth.
* Nm is the maximum EDP
* rb is the bottom of the ionosphere.
* ym is called the semi-thickness of the layer, the distance from the bottom of the layer to the maximum, i.e. ym=rm-rb.

1. At what height does the maximum EDP occur?

Maximum EDP occurs at and height, , where re is the radius of the earth

1. What is the critical frequency ( highest frequency where a HF signal will be completely reflected) for this one layer Ionosphere?

The critical frequency is

1. What parameters define uniquely and completely this Ionosphere?

Nm, rb, and rm define the ionospheric properties with the condition that

Alternatively, one can use Nm, ym, and rm with the condition that

1. Plot using any software a Quasi Parabolic curve for five different values of ym 20,50,100,200,250, rm=300 and Nm=1012 electrons per m3

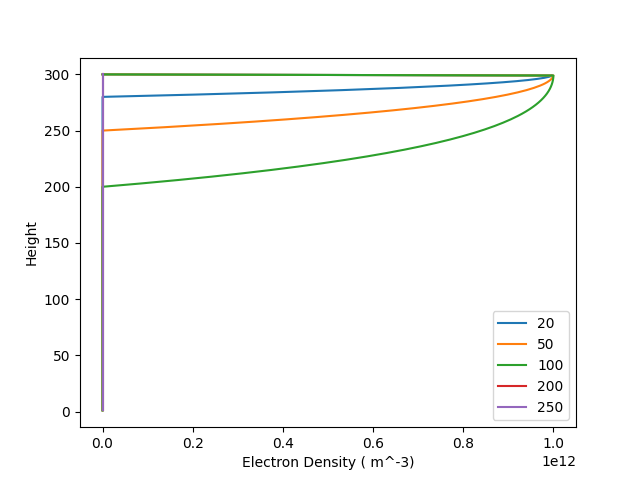


Figure EDP Profiles for varying Ionosphere semi-thickness values

B - Ionosphere Reflection

The refractive index **μ** in the ionosphere is directly related to the EDP and the ionospheric plasma frequency, (when magnetic and collisions are neglected) by the following equation:

=1-

Where fN is the plasma frequency, frequency of a wave that will be totally reflected by a plasma with EDP Ne(r), f is the frequency of the incident wave and k is the conversion factor

Where e is the electron charge, ε0 is the electric permittivity in vacuum and m is the electron mass.

Consider a very simple case of ray in the ionosphere where the EDP only depends on the height, the ray trajectory will follow Snell’s law i.e. .

Diagram

Description automatically generated

1. What happens to rays for the different relations between fN and f?

Frequencies above fN begin being transmitted through the ionosphere and frequency-based changes in the permittivity exhibit an inversion in their slope.

When the ratio approaches the limit of 1 the wave is strongly reflected away from the normal and begins experiencing attenuation.

At the ratio of exactly 1 there is strong absorption as the imaginary portion of the index of permittivity peaks and the wave is strongly reflected.

As the ratio trends towards infinity and the frequency decreases towards 0 the permittivity becomes strongly negative, a 180 degree phase shift occurs, waves remain fully reflected but have less ability to couple to the waveguide. These waves potentially become evanescent and propagate very long distances before decaying.

1. Find a relationship between the angle of incidence θi the frequency f and the EDP N(h) that enables total reflection?

Total reflection occurs when .

Hence,

1. Calculate the maximum frequency fob allowing total reflection for the ionospheres of part A question 4

For ym = 200 and 250, Ne = 0 and there is no maximum frequency for total reflection

For ym = 20, 50, and 100 the ionospheres achieve Nmax = 1e12 and the maximum frequency is the frequency where

Therefore

1. Use the concept of virtual height to calculate the formula for the maximum range D (distance along earth’s path) for a ray that is totally reflected?
2. What is this distance for the above ionospheres?

All of the above ionospheres have the same maximum height and the same maximum electron density. However the last two do not have valid electron density profiles and can be assumed to not reflect.

Setting ,

C - Ray Tracing

Let us look at a ray, a wave front of eletro-magnetic radiation, in the HF radio spectrum. The ray, of frequency f, begins its path at earth’s surface with an initial elevation angle of β0, and is totally reflected by an ionosphere. The ionosphere is described by a single layer Quasi-Parabolic model as in part A . Look at the diagram below:

Diagram

Description automatically generated

The path of the ray follows Snells law, in a spherically symmetric system one can use a special form called Bouger’s Law which states that the product rμ(r)cosβ is a constant of motion. So .

Everywhere along the ray path.

1. Find an equation in integral form that will represent the range of the ray in terms of r and μ(r)? (the range is the ray path length along earth’s surface , note that it is related to the change in dθ)
2. Solve the integral analytically. (hint: use a parametrization of the denominator of the integral to the form: )
3. Use the above parametrization to write an equation for the apogee height, solve it.
4. At what angle is the apogee height maximal?
5. Plot the ray path using the formula from 1, for the 5 ionospheres from section A, the frequencies of 5,10,15,20 MHz and elevations of 5,10,15,20 degrees, along with the ionosphere as a heatmap, how are the results related to 3 and 4?

Unfortunately I didn’t leave myself enough time to work through this problem.

## Assessment: "convolve\_raw\_\*.dat" (attached) have data arranged in 3 columns.

## 1) Column 1 and 2 designate x and y coordinates respectively.

## 2) Column 3 is the main data defined at the corresponding (x, y) points.

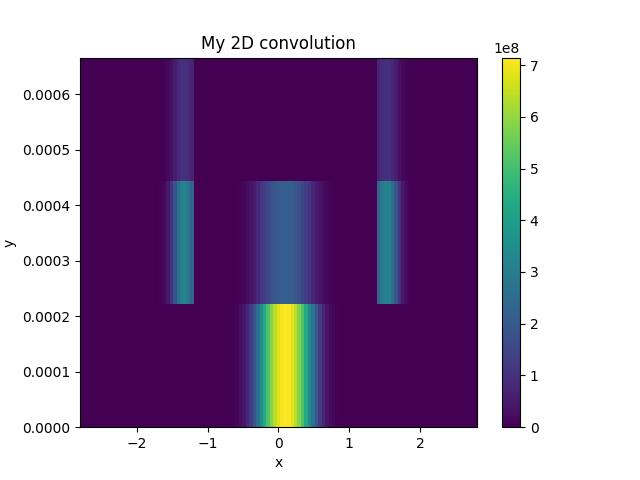
## 3) Compute the 2-d convolution of these data and provide visualization.

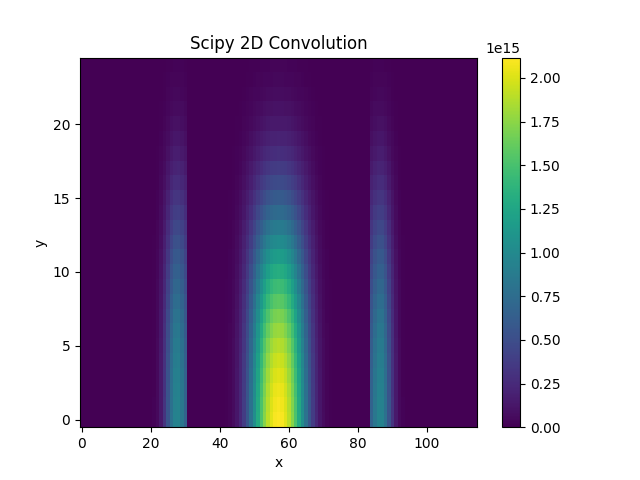
## 4) Use any programming language or library as necessary.

For this assessment I tried to implement a 2D convolution algorithm in python, however the result does not seem to be correct when compared with the output from scipy’s inbuilt 2D convolution function. There are clear similarities, but I believe the magnitude of my result should match that of the scipy result and something is wrong with the overlap of the data 0 and data 1 in the y direction resulting in a dramatic y-resolution difference.

A large part of my work for this assessment was involved in simply preparing the input data. This was accomplished in several steps. First the regions where data\_0 and data\_1 overlapped were determined to save on run time, since the convolution result for regions that didn’t overlap would be zero. The range of x and y values spanned by the overlapping regions were used to create two uniform structured arrays and the values from data\_0 and data\_1 were assigned to these arrays to match their original locations. Data\_1 was then transposed and these grids were padded by zeros to create equal sized arrays to be used in the convolution algorithm. The scipy convolution function did not require the second array to be transposed.

See 2D\_convolution.py. This script can simply be run through python on the command line when placed into the same directory as the two data files.





## Assessment Temporal relationship: In C, without using any third-party library except standard C library, write a function that can sort the following entries by date and time, from the oldest to the newest then write a median filter function that calculates foF2 and hmF2.

See sort\_and\_filter.c and the output file AU930\_filtered.txt

To run the script simply place it in the same directory as the data file in need of sorting/filtering and compile it with no added parameters. For example, on my system this is simply ‘gcc sort\_and\_filter.c’. Then it can be run by simply executing the a.out file.

## Assessment Interpolation: Using any language and plotting packages you like, interpolate the given point location values to a structured grid of longitude and latitude locations. This task will ask the candidate to implement a procedure that interpolates known grid points and values to a structured grid

|  |  |  |
| --- | --- | --- |
| lon | lat | value |
| 121.39 | 13.51 | 1.494 |
| 126.19 | 12.02 | 1.934 |
| 130.27 | 13.11 | 2.148 |
| 127.42 | 10.09 | 9.155 |
| 126.14 | 15.33 | 2.221 |
| 125.96 | 14 | 8.1 |
| 123.15 | 10.88 | 2.039 |
| 130.5 | 11.18 | 1.916 |
| 129.08 | 15.78 | 3.729 |
| 122.74 | 15.82 | 7.137 |

You are free to choose any interpolation method you like. The spatial grid should have 50 rows and 70 columns with a longitude range [121.0, 131.0] and latitude range [10.0, 16.0]. Provide a plot of your results - including a short summary on the interpolation procedure you implemented.

My interpolation method takes each of the known locations and compares it to the nearest 4 points in the structured grid. These points are . This results in 4 rectangles formed by the and these 4 points, , with one corner of each rectangle being formed by the and the opposite corner of each rectangle being formed by one of the 4 nearest points,. Weights are assigned to each point, , by dividing the area of by the area of the rectangle formed by comprising each corner. Values are then proportionally applied to by multiplying the known value at by the corresponding weight for each point.

See point\_interpolation.py

Chart, scatter chart

Description automatically generated

