amisrsynthdata: A Case Study in Expandable Packages

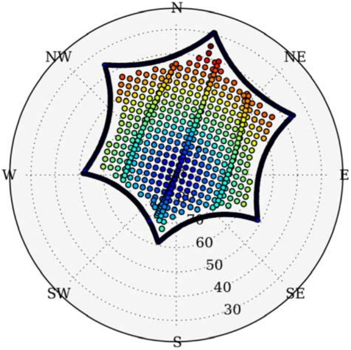
Leslie Lamarche[1], Asti Bhatt[1], Pablo Reyes[1]

(1) SRI International, Menlo Park, CA, USA

2023 CEDAR Workshop
San Diego, CA, USA
June 25 – June 30, 2023

AMISR Background

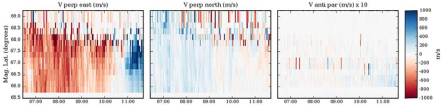
- Advanced Modular Incoherent Scatter Radar (AMISR)
- Electronically steerable phased array
  - A single scan cycles through multiple beam positions extremely quickly, effectively creating a simultaneous image of ionospheric plasma parameters in 3D
  - Experiments can be easily customized to generate beams in a wide variety of configurations
- Experiment design can be extremely flexible
  - Number/Position of beams
  - Beam coding/Altitude regimes of interest
  - Time resolution

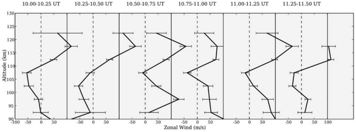


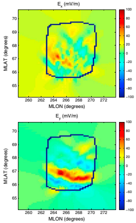
AMISR Advanced Data Products

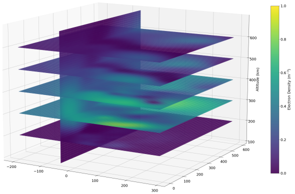
- Resolved Vector Velocities
- E-region Wind Profile
- 2D Resolved Vector Velocities
- Volumetric Interpolation
- Energetic Particle Precipitation Spectra

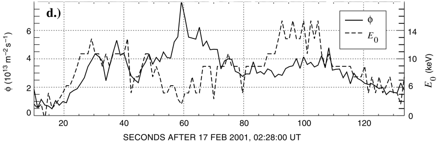
These high-level data products require inversion of the original fitted data, which is often complex and requires assumptions and/or regularization.







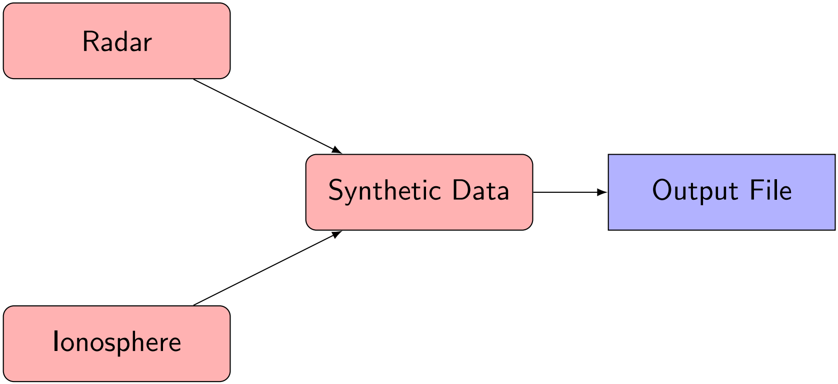




Synthetic Data

- "Data" generated with software that represents what a particular instrument expects to see when observing a defined system
- Extremely useful for developing inversion techniques because it provides "truth" values to compare against
- Test how well an instrument (or a particular instrument mode) can observe a particular phenomena, especially rare events
- Optimize observing modes
- Basically equivalent to an Observing System Simulation Experiment (OSSE)

General Concept



The amisrsynthdata Package

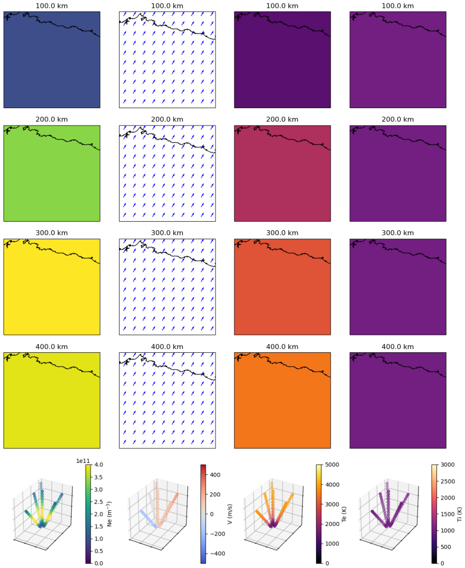
Repository: `https://github.com/amisr/amisrsynthdata`
Documentation: `https://amisrsynthdata.readthedocs.io`

1. Install: $ pip install git+https://github.com/amisr/amisrsynthdata.git
2. Create config file - information in documentation
3. Run program: $ amisrsynthdata config.ini

- Output mimics the format of an SRI AMISR processed data file
- Currently only have a handful of options coded for common ionosphere states (i.e., polar cap patches, uniform plasma flows), but it is easy to add custom options by adding new functions to the Ionosphere class
- Can be called from command line or imported into python script to access the Ionosphere or Radar classes directly

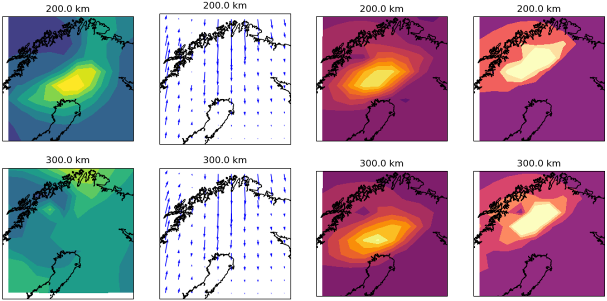
Basic Setup and Example

```
|| GENERAL:
||   starttime: 2016—09—13 00:00:00
||   endtime: 2016—09—13 00:30:00
||   output_filename: synthetic_data.h5
||   ion_mass: [16., 32., 30., 28., 14.]
||   err_coef: [1., 1.e—9, 5.e—9, 1.e—9]
||   noise: False
||   summary_plot: synthdata_summary.png
||   summary_plot_time: 2016—09—13 00:10:00
|| RADAR:
||   full_name: Poker Flat
||   abbreviation: PFISR
||   site_coords: [65.13, —147.47, 213.]
||   beamcodes: [64016, 64157, 64964, 65066]
||   beam_azimuth: [120., 180.]
||   beam_elevation: [60., 50.]
||   acf_slant_range: [80000., 800000., 3000.]
||   altitude_bins:
||     — [80000., 150000., 10000.]
||     — [150000., 800000., 50000.]
||   integration_period: 60.
|| DENSITY:
||   uniform:
||     value: 5.0e+10
||   chapman:
||     N0: 4.0e+11
||     H: 100000.
||     z0: 300000.
||     sza: 0.
|| VELOCITY:
||   uniform:
||     value: [300., 500., 0.]
|| ETEMP:
||   hypertan:
||     maxtemp: 4000.
||     scale_height: 300000.
|| ITEMP:
||   uniform:
||     value: 1000.
```



Ionospheric State Functions

- When the Ionosphere class is initialized, it sets up the density, velocity, itemp, and etemp methods based on the config file
- Include the names of the functions to use and auxiliary parameters (i.e., NmF2, hmF2, position of structure) needed to evaluate them
- The auxiliary parameters that need to be defined are described in the documentation
- Can include multiple functions for a single physical ISR parameter - these are summed to create the complete function



Challenges with Specifying Ionosphere

Potential ionospheric structures one may want to consider:

- Polar Cap Patches
- Auroral Arcs
- Flow Shears
- Sun-Aligned Arcs
- TIDs/LSTIDs/MSTIDs
- Sporadic E Layer
- Pulsating Aurora
- Energetic Particle Precipitation
- Plasma Heating
- Infinite other possibilities

A single interface to describe all of these would be challenging to develop and difficult to use.

Approach:

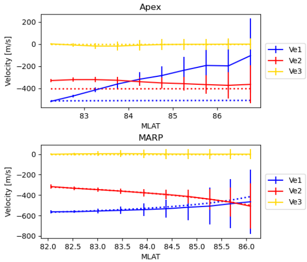
- Don't try to program in every possible functionality, but make it easy to add more
- Allow for combinations of phenomena (i.e., Gaussian patch on top of a Chapman layer)
- Specify which functions to use in config file

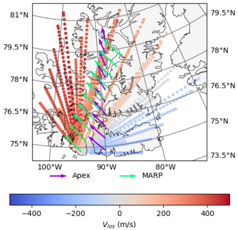
Design Expandable Software

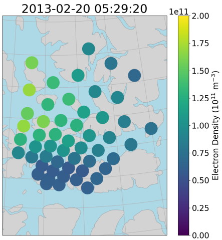
Goal: Make it easy for users to contribute new ionospheric state functions, even when they do not have extensive experience contributing to open source projects

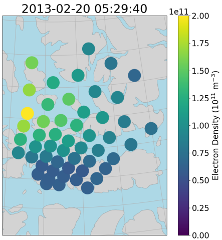
1. Only have one location in the source code where new functions should be added
2. Strictly define the format of functions, particularly in terms of input and output parameters
   - input = time, glat, glon, galt
   - output = ionospheric parameter
   - ND arrays must be accepted
   - any coordinate system or array manipulation can happen internally as long as input/output remains the same
3. Document these requirements VERY clearly
4. Accept that this format and code design may need modifications, especially in early iterations

Example Applications









- Verify that a new algorithm for resolving 3D plasma drift velocities (Marp) performs better than the original algorithm (Apex)
- Sample GEMINI output in beams to understand how RISR-N would observe a polar cap with developing gradient drift instability

Conclusions and Caveats

- Synthetic data are a useful tool for testing observing geometries, instrument modes, and inversion techniques
- When designing software, sometimes it is easier to make it straight forwards to add more features than try initially code in all possible uses

Features not currently included in amisrsynthdata:

1. Any kind of proper treatment or simulation of ISR theory - The module effectively assumes the radar measures plasma parameters perfectly at a particular location, although empirical errors can be added.
2. Integration over a time period, smearing along the length of pulses, pulse coding, or beam spreading.
3. Madrigal data format - Currently files are only generated in the SRI data format.