e3doubt

A python- and R-based tool for EISCAT_3D experiment design and uncertainty analysis

S. M. Hatch[1], I. Virtanen[2], ISSI Team 506[3]

[1]Department of Physics and Technology, University of Bergen, Bergen, Norway
[2]Space Physics and Astronomy Research Unit, University of Oulu, Oulu, Finland
[3]International Space Science Institute, Bern, Switzerland (link to team website)

Outline

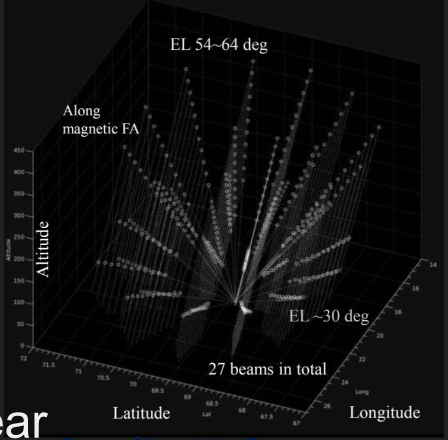
- Motivation for e3doubt: a python frontend to ISgeometry
- Practical demo
- In(put)s and Out(put)s of e3doubt
- Demonstration I: Uncertainties of maps of ionospheric convection reconstructed from E3D measurements using SECS
- Demonstration II: Uncertainty of calculations of electromagnetic work

Motivation

- EISCAT_3D is an extremely advanced ISR system.
    - How can we (plan to) use it?
- Ilkka Virtanen has written a powerful set of tools in R for ISR uncertainty analysis: the ISgeometry package
    - But no one uses R … (do they?)
- Enter: e3doubt

Practical demo

In(put)s and Out(put)s of e3doubt



Credit: Y. Ogawa

Required inputs

- Azimuth and elevation for each beam
- Heights at which to sample each beam
- Default is ~Ogawa's suggested CP1 program from last year

Optional inputs

- Information about Tx, Rx (location, min el, FWHM, power, duty cycles, T_{noise}, ...)
  - Can specify Tx site and completely arbitrary combination of Rx sites
- Relative beam dwell times
- Range resolution for each altitude
- Ionosphere and atmosphere parameters
  - Default is IRI and MSIS via iri2016 and pymsis
- … And more!

In(put)s and Out(put)s of e3doubt - 2

Possible workflow

1. Select elevations and azimuths
2. Initialize ‘Experiment’ object
   e.g., `exp = Experiment(az=az,el=el)`
3. Run IRI and MSIS models, calculate collision frequency
   `exp.run_models()`
4. Calculate plasma parameter uncertainties using ISgeometry
   `unc = exp.get_uncertainties(integrationsec=600)`
5. Perform analysis using uncertainty estimates
   …

In(put)s and Out(put)s of e3doubt - 3

Custom ionosphere and atmosphere parameters: set_ionos/set_atmos functions

```
def get_datacov_e3doubt(ddict):

    exp = e3doubt.Experiment(el=ddict['el'], az = ddict['az'],h=ddict['alts'])

    exp.run_models()

    exp.set_ionos('ne',ddict['ne'])
    exp.set_ionos('Te',ddict['Te'])
    exp.set_ionos('Ti',ddict['Ti'])

    uncert =exp.calc_uncertainties(integrationsec=5*60)

    cov = exp.get_velocity_cov_matrix()

    ddict['cov_vi'] = cov
    ddict['var_ne'] = uncert.dnemulti.values

    return ddict
```

Credit: J. Reistad

In(put)s and Out(put)s of e3doubt - 4

Output of get_uncertainties()

```
In [7]: df = exp.get_uncertainties()

In [8]: df.columns
Out[8]:
Index(['dne1', 'dne2', 'dne3', 'dnemulti', 'dTe1', 'dTe2', 'dTe3', 'dTemulti',
       'dTi1', 'dTi2', 'dTi3', 'dTimulti', 'dVi1', 'dVi2', 'dVi3', 'dVimulti'],
      dtype='object')

In [9]: df
Out[9]:
             dne1          dne2          dne3  ...        dVi2        dVi3     dVimulti
0    2.775050e+09  1.209054e+10  1.328753e+10  ...   97.543539  107.200598   364.616562
1    1.004876e+10  8.203632e+09  8.750000e+09  ...  136.358919  145.440520   727.078870
2    1.116370e+10  9.308450e+09  9.745883e+09  ...  222.350262  232.799176  1308.073572
3    1.601766e+10  1.184437e+10  1.212833e+10  ...  288.004052  294.908699  2066.109336
4    4.124264e+10  2.217895e+10  2.203723e+10  ...  230.818517  229.343612  2442.379623
..            ...           ...           ...  ...         ...         ...          ...
238  4.057360e+10  1.860394e+10  1.920834e+10  ...  196.892688  203.289253  2066.275313
239  1.104095e+11  4.237233e+10  4.268149e+10  ...  232.971586  234.671381  3363.021721
240  1.520652e+11  6.531773e+10  6.502533e+10  ...  270.932762  269.719925  4226.699972
241  1.390351e+11  7.586721e+10  7.544972e+10  ...  406.871173  404.632213  6215.150742
242  1.157246e+11  7.850811e+10  7.826261e+10  ...  624.767342  622.813650  9547.465558

[243 rows x 16 columns]
```

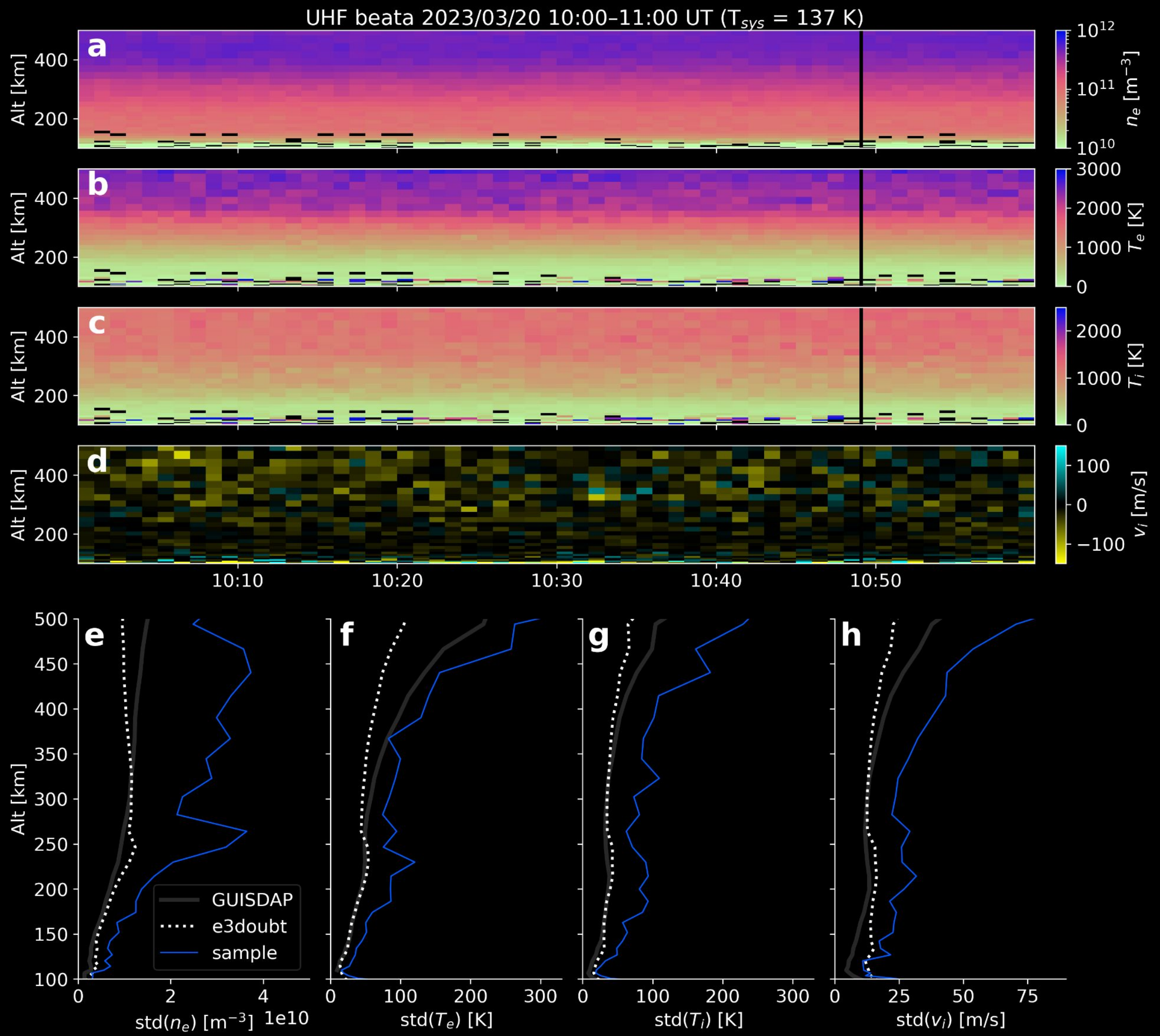
In(put)s and Out(put)s of e3doubt - 5

Lots of helper functions

- get_velocity_cov_matrix
  - velocity covariance matrix in ENU or ECEF coordinates for each point
- get_beam_info
  - Azimuth, elevation, dwell time, and beam number for each beam
- get_atmos
  - Get a pandas DataFrame containing all atmospheric parameters
- get_ionos
  - Get pandas DataFrame containing all ionospheric parameters
- get_points
  - az, el, h, beam, gdlat gclat, glon, xecef, yecef, zecef, resR
- radar_utils.py, geodesy.py
  Many tools for radar geometry and geodesy calculations

Demonstration I: Compare with measurements

- How does e3doubt stack up against GUISDAP?



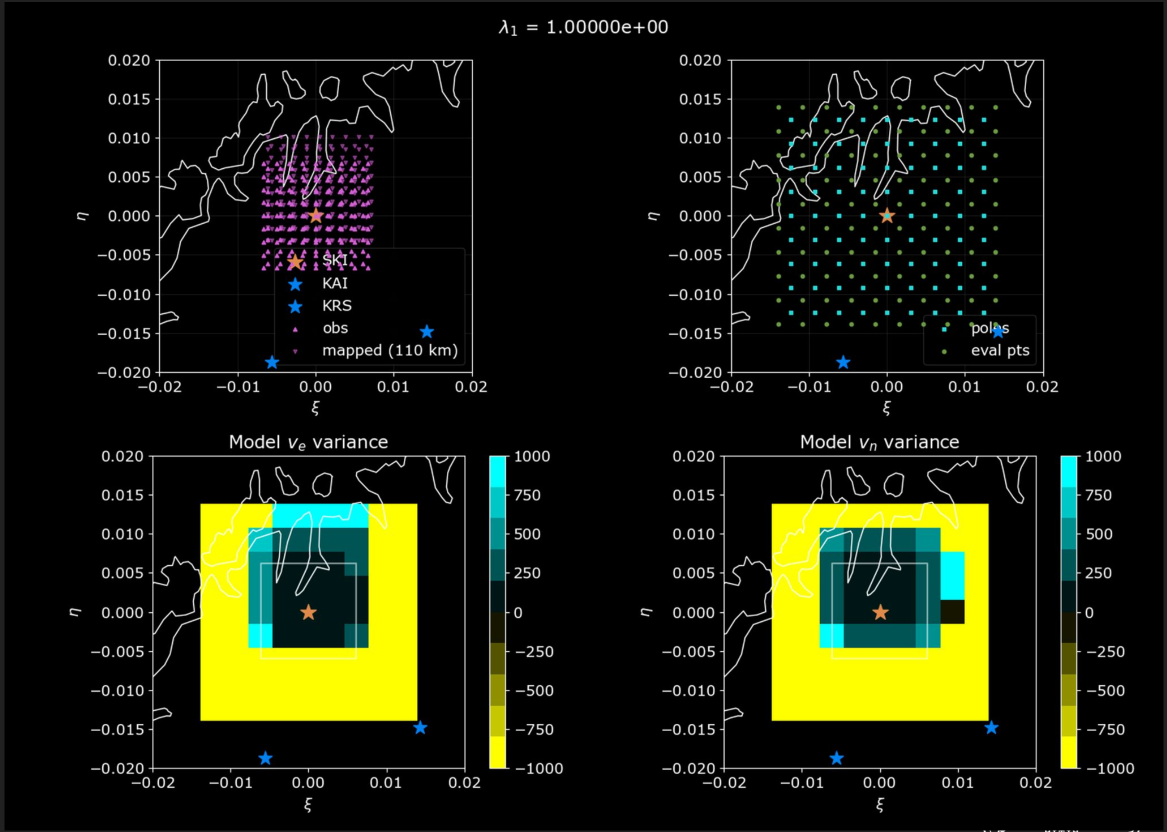
UHF beata 2023/03/20 10:00–11:00 UT (T_{sys} = 137 K)
a
b
c
d
Alt [km]
400
200
n_e [m^{-3}]
10^{12}
10^{11}
10^{10}
T_e [K]
3000
2000
1000
0
T_i [K]
2000
1000
0
v_i [m/s]
100
0
−100
10:10
10:20
10:30
10:40
10:50
e
f
g
h
Alt [km]
500
450
400
350
300
250
200
150
100
GUISDAP
e3doubt
sample
std(n_e) [m^{-3}] 1e10
0
2
4
std(T_e) [K]
std(T_i) [K]
0
100
200
300
std(v_i) [m/s]
0
25
50
75

Demonstration II: Map of convection uncertainty

- 81(!) beams on cubedsphere grid
- *Covariance* of $\mathbf{v}_\perp$ from E3D mapped to 110 km using Apex basis vectors:

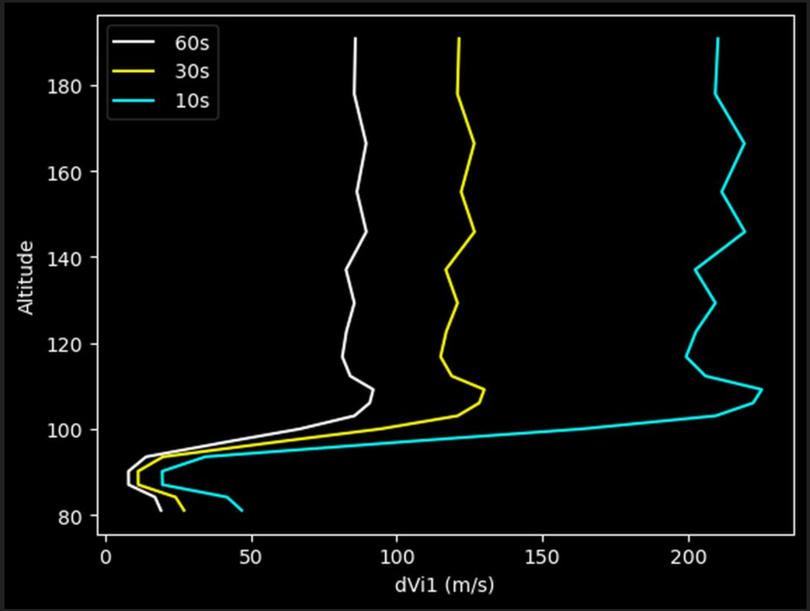
$$\Sigma_{v'} = \mathbf{B}\Sigma_v\mathbf{B}^T$$

- Reconstruct ionosph. potential using curl-free SECS functions

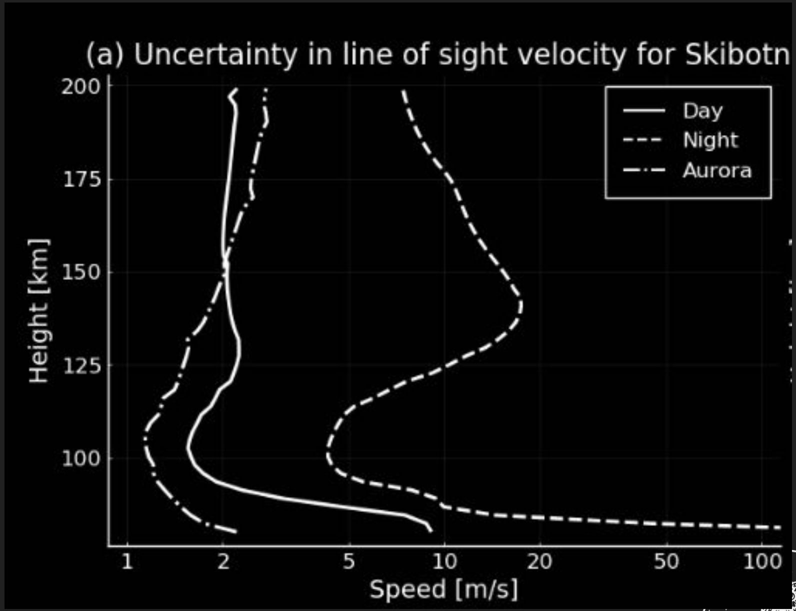


Comparison with Stamm et al (2021) (credit: Habtamu Tesfaw, UOulu)

e3doubt



Stamm et al (2021)



Why is e3doubt so much more uncertain? Self noise …

Summary

- We need dedicated, open-source tools to make the most of E3D measurements

- We've produced a tool for experiment design and uncertainty estimation (try it!)
  - e3doubt (GitHub: https://github.com/Dartspacephysiker/e3doubt)

Thanks!