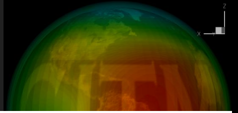
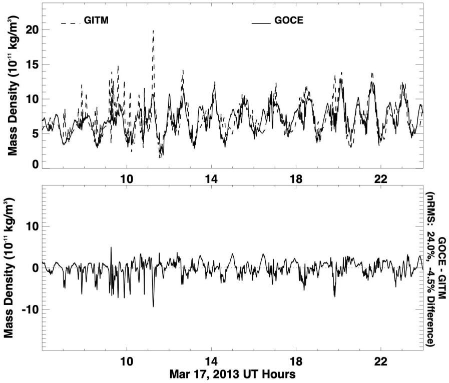
GITM and Aether

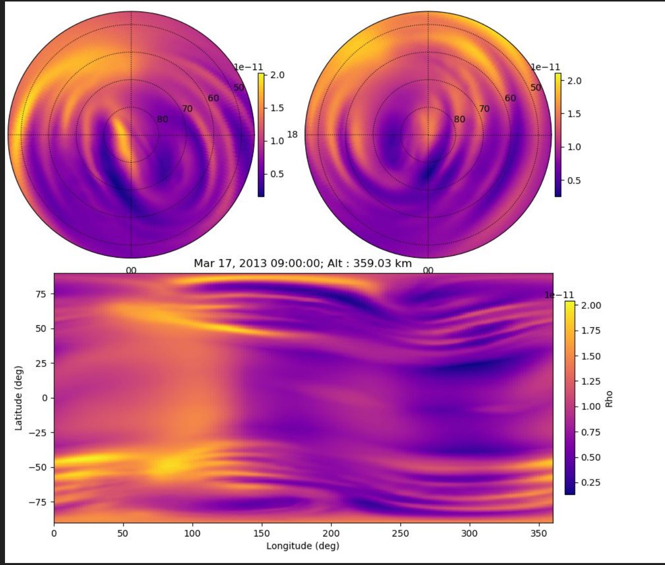
Aaron Ridley

The Global Ionosphere Thermosphere Model

- 3D geographic grid structure
  - Longitude, Latitude, Altitude (not pressure)
  - Fully parallel with 2D block-based domain decomposition
  - Typical resolution of 4° x 1° (lon x lat) on 200 processors; about 4x faster than real time
    - Highest resolution runs are 2° x 0.5° (lon x lat) on 800 processors
      - Each file is about a GB, so this is complicated
    - Stretched grid in altitude at about ~1/3rd "scale height" (100 - ~500 km)
  - Can run in 1D (fixed longitude/latitude, rotating with the Earth - debugging mostly)
  - Can run in limited longitude / latitude domain, with different horizontal boundary conditions
- State Variables:
  - 11 Neutral Densities (O, O_2, N_2, $N(^4S)$, NO, He, $N(^2D)$, $N(^2P)$, CO_2, $O(^1D)$)
  - 8 Ion densities ($O^+(^4S)$, O_2^+, N_2^+, N^+, NO^+, $O^+(^2D)$, $O^+(^2P)$, He^+)
  - Neutral winds - Bulk horizontal velocities, individual vertical velocities
    - Full vertical momentum equation solved for each species
    - Vertical winds include friction terms between the different species
  - Bulk ion velocities (parallel + perpendicular), electron velocity (ExB drift)
  - Neutral, (full) Ion, and (full) Electron temperatures



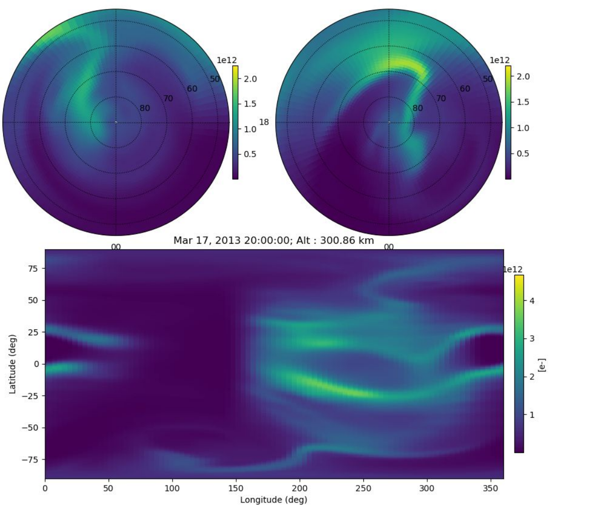

GITM
GOCE
Mass Density (10^{-11} kg/m^3)
20
15
10
5
0
10
14
18
22
Mass Density (10^{-11} kg/m^3)
10
0
-10
GOCE - GITM
(nRMS: 24.0%, -4.5% Difference)
Mar 17, 2013 UT Hours



Snapshot of mass density from GITM + GITM-GOCE comparisons during March 2013 storm

Equation Sets

- Neutrals are modeled with the Navier-Stokes equations (mostly)
  - Many terms are bulk and not individual
  - Lots of source terms
    - Chemistry, Coriolis, ion drag, geometry effects, eddy diffusion, EUV, aurora, frictional heating, heat transfer, chemical heating, (NO, CO2) radiative cooling
  - Full momentum equation allows capture of acoustic waves
    - Limits time-step to a few seconds to capture wave speeds
- Ions are modeled with a modified equation set
  - Continuity can include divergence term, but it is on a switch
    - Vertical boundary conditions for ions are painful
    - Field-lines extend above domain and really need a reservoir of plasma
    - Divergence brings need to message pass field-aligned ion velocities
    - Need to advance field-aligned ion velocity implicitly
  - Momentum equation is solved both along the field and across the field
    - Bulk only, which is a disadvantage; especially along field line
  - Fully self-consistent electrodynamics dynamo in low-latitude region
    - Implemented from Richmond paper
  - Ion and electron temperatures are "full" equations and not steady-state approximations
    - Electron heating/cooling is extremely complicated and "fast"
    - Both frictional heating (Vi-Vn)2 and heat transfer (Ti-Tn) included both ions and neutrals

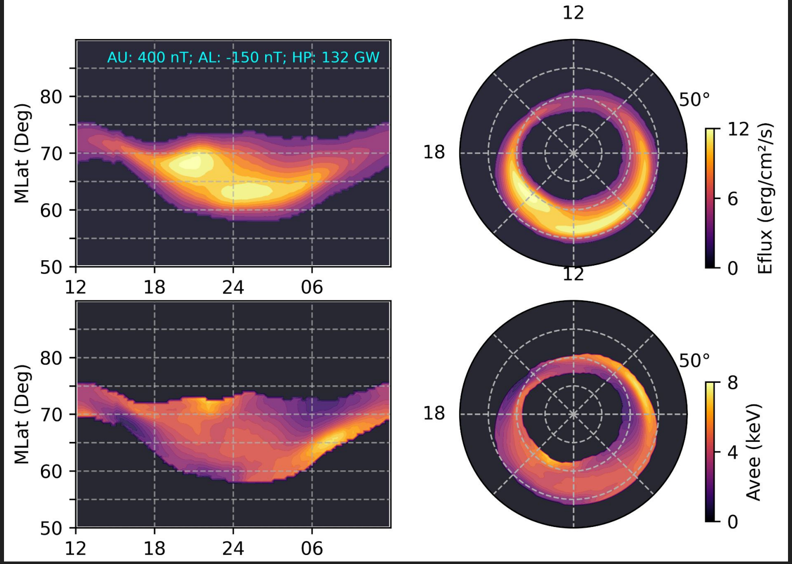


Electron densities at 300 km on March 17, 2013 20 UT

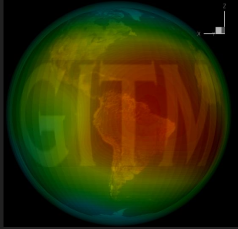
High Latitude Drivers

- Need to set the electric field at high latitudes, lots of ways to do this
  - Empirical models, such as Weimer
  - Output from other models (SWMF)
  - Data assimilation models like AMIE
- This is applied as the electric potential at all cell centers
  - Gradient to derive the electric field
  - Zero out electric field along field-line
    - Not doing this caused a lot of problems for many years
- Aurora specified at top of the atmosphere
  - Empirical models
    - FTA Model - AE driven
  - Output from other models (SWMF)

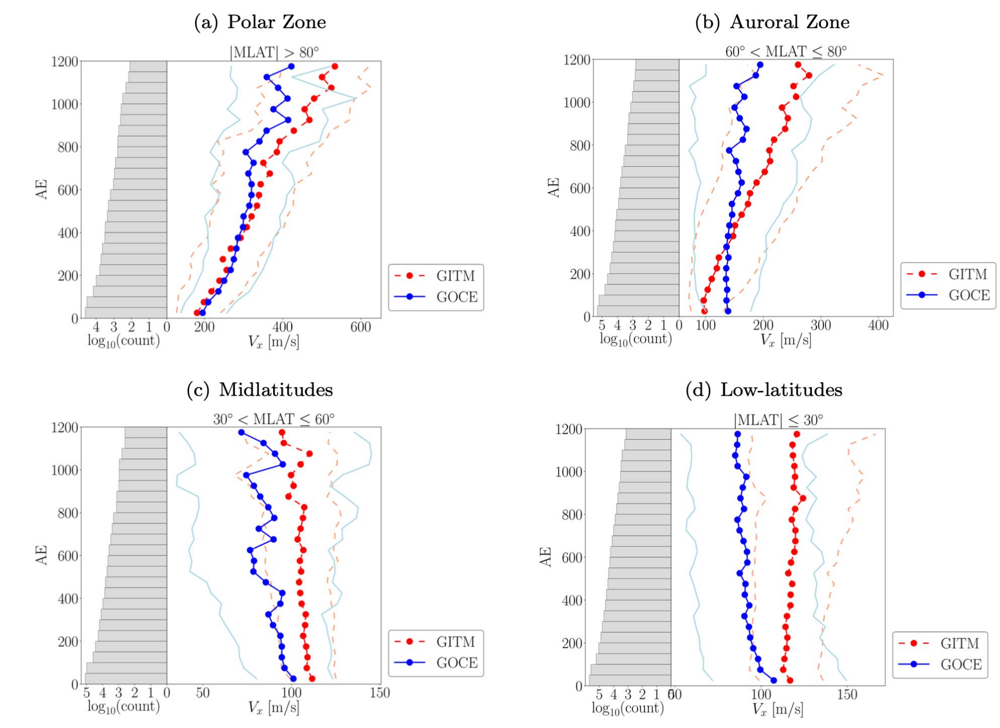
- Auroral energy deposition code computes ionization vs altitude
  - a bunch of mono-energetic beams
  - Similar to EUV energy deposition
  - Fang electron and ion deposition



Validation with GOCE Winds

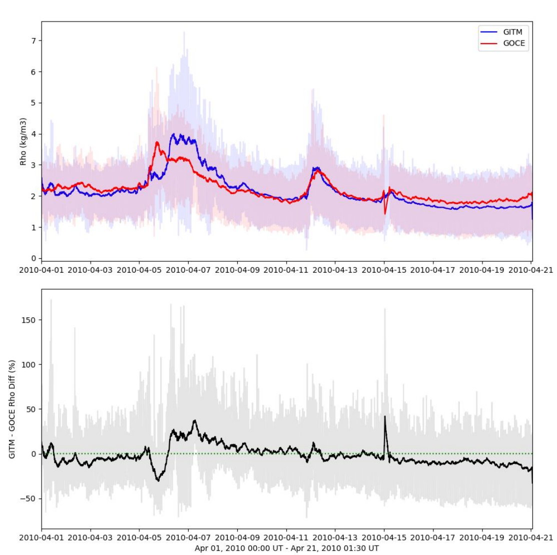


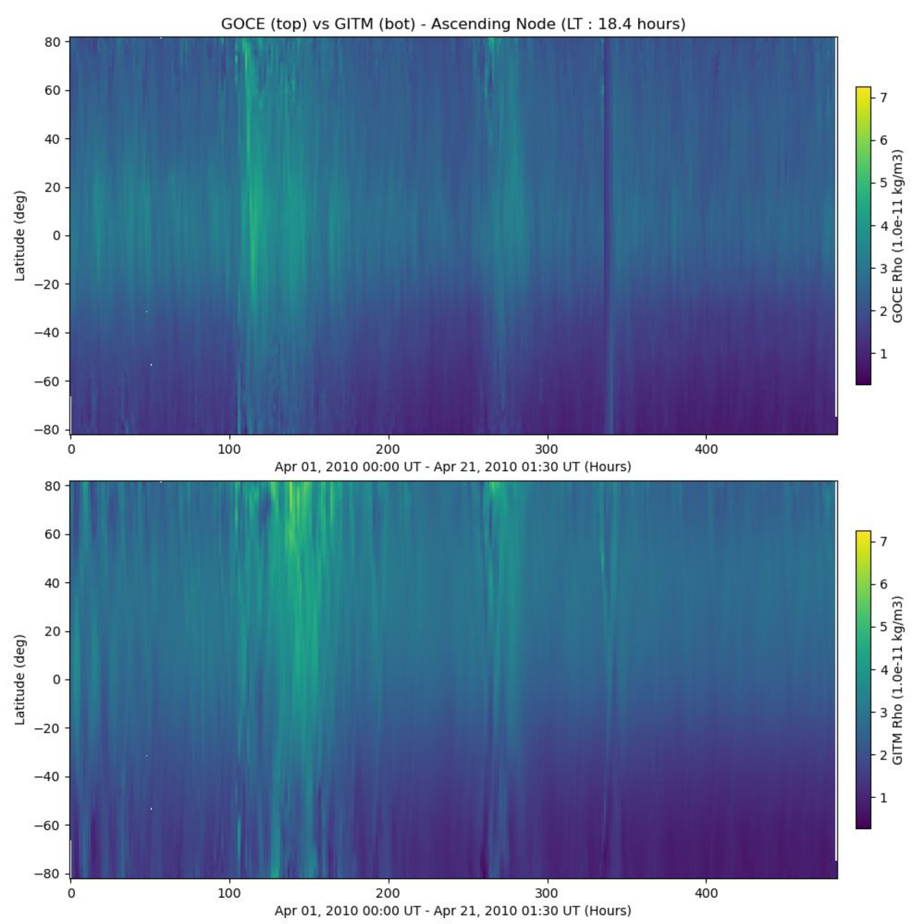
- Year long comparison between GITM and GOCE winds - 2013
- Published in https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2021SW002922
- Matches very well in polar cap
- Winds too strong at low and mid-latitudes
- Auroral zone winds too low at low activity, but too high at high activity
    - Consequences for frictional heating



Validation with GOCE Rho

- Year long comparison between GITM and GOCE mass densities - 2010
- Use this to test robustness of code and see that everything is working ok.

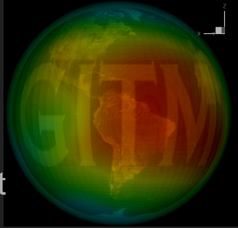




Aether - a new model of the ionosphere/thermosphere

- Currently in development
  - https://github.com/AetherModel/Aether
  - You can work on the development if you would like!
- C++
- Primary values:
  - Flexibility
  - Ease of use and development
  - Clean code that is “professionally” developed
  - Open development

The Team



- Aaron Ridley - UM
  - Project lead, Aether architecture and thermosphere development, school and resources development
- Jeff Anderson - NCAR
  - DART lead, Aether data assimilation, uncertainty quantification
- Jared Bell - GSFC
  - Aether numerical scheme and grid system development, Aether verification
- Alex Glocer - GSFC
  - Aether ionosphere and general model development, Aether validation and verification
- Angeline Burrell - NRL
  - Aether coding standards, Aether post-processing codes, Aether validation, teaching resources
- Meghan Burleigh - NRL
  - Aether architecture, ionosphere development, school and resources development
- Qusai Al Shidi - UM
  - Aether architecture, coding standards, school and resources development
- Ben Johnson - NCAR
  - DART / Aether interface development, data assimilation, uncertainty quantification
- Chen Wu - UM
  - Post Doc, working on Auroral models
- Undergrad Students @ UM
  - 2021: Rutvik Marathe, Olivia Doty, Maca Peralta, Keegan O’Connor
  - 2022: Michael Rinaldi, Ashwin Kumar, Kate Stamp

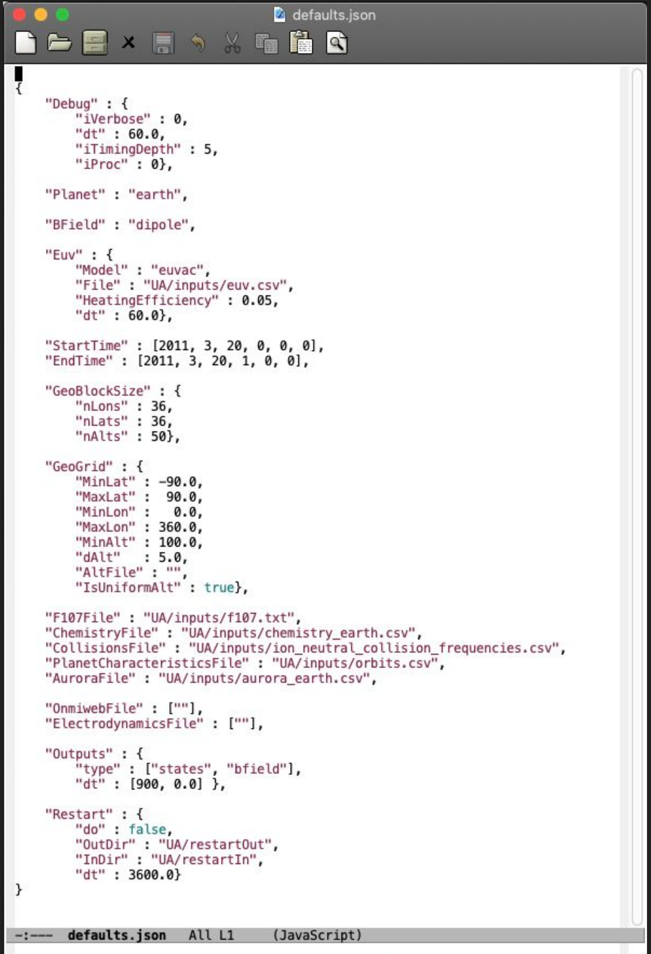
Collaborative workflow

- The Aether community uses a standard GitHub workflow:
    - New work is performed on `develop`, while stable code lives on `main`
        - Internal or external developers create new branches from `develop` and when ready, create pull requests that merge the code into `develop`
        - At least one approval is required before a pull request can be merged
        - Pull requests undergo automated acceptance testing using Github Actions for both MacOS and Linux is set up.
    - Dependencies are incorporated into the workflow using Docker
    - Questions, problems, and tasks are recorded as GitHub Issues
- Workflow resources:
    - Contributing guidelines are provided to encourage community contributions
        - Contributing.md (main directory)
        - Online documentation (https://aetherdeveloper.readthedocs.io/en/latest/contributing/contributors-guide.html)
    - Code style conventions have been adopted to improve clarity with multiple contributors
    - Standard linting resources are noted and the appropriate style enforced (AStyle!)
    - Commit messaging style ensures programming progress is easy to follow
    - In-code documentation, compatible with Sphinx and rST, is required
- Adaptability:
    - The workflow may be changed or clarified as issues or questions occur
    - To make changes, a consensus must be reached within the current development team
    - Community developers may bring up issues or questions through GitHub Issues

Flexibility

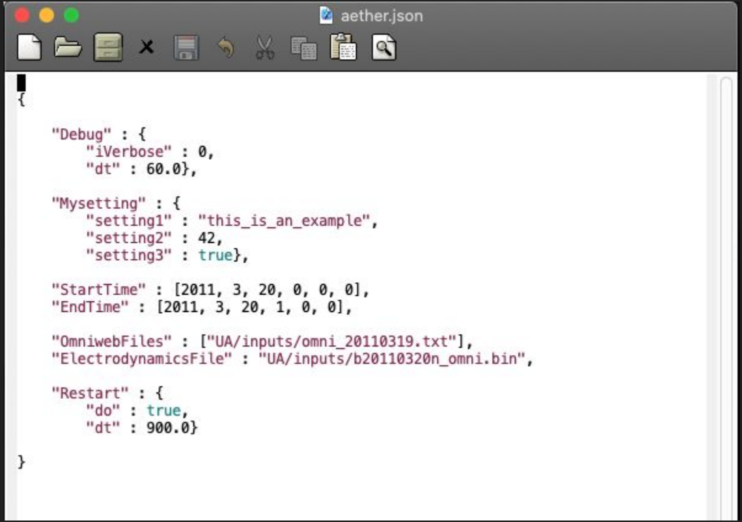
- Grid determined at run time (don’t need to recompile)
    - 1D, 2D, 3D possible
- Swappable features
    - Chemistry specified by CSV file
    - EUV specified by CSV file (i.e., number of wavelengths, cross sections, etc.)
    - Planetary characteristics specified by CSV file
        - Can run different planets without recompiling
        - Can change species to consider (neutrals and ions) without recompiling (goal)
    - Collision terms specified by CSV file
- json inputs
- Output flexibility
    - netCDF, binary
    - Developer puts output variables into generic holding place, and output system takes those and writes to requested formats

json inputs



```
{
    "Debug" : {
        "iVerbose" : 0,
        "dt" : 60.0,
        "iTimingDepth" : 5,
        "iProc" : 0},

    "Planet" : "earth",

    "BField" : "dipole",

    "Euv" : {
        "Model" : "euvac",
        "File" : "UA/inputs/euv.csv",
        "HeatingEfficiency" : 0.05,
        "dt" : 60.0},

    "StartTime" : [2011, 3, 20, 0, 0, 0],
    "EndTime" : [2011, 3, 20, 1, 0, 0],

    "GeoBlockSize" : {
        "nLons" : 36,
        "nLats" : 36,
        "nAlts" : 50},

    "GeoGrid" : {
        "MinLat" : -90.0,
        "MaxLat" :  90.0,
        "MinLon" :   0.0,
        "MaxLon" : 360.0,
        "MinAlt" : 100.0,
        "dAlt"   : 5.0,
        "AltFile" : "",
        "IsUniformAlt" : true},

    "F107File" : "UA/inputs/f107.txt",
    "ChemistryFile" : "UA/inputs/chemistry_earth.csv",
    "CollisionsFile" : "UA/inputs/ion_neutral_collision_frequencies.csv",
    "PlanetCharacteristicsFile" : "UA/inputs/orbits.csv",
    "AuroraFile" : "UA/inputs/aurora_earth.csv",

    "OnmiwebFile" : [""],
    "ElectrodynamicsFile" : [""],

    "Outputs" : {
        "type" : ["states", "bfield"],
        "dt" : [900, 0.0] },

    "Restart" : {
        "do" : false,
        "OutDir" : "UA/restartOut",
        "InDir" : "UA/restartIn",
        "dt" : 3600.0}
}
```

```
-:---  defaults.json   All L1     (JavaScript)
```



```
{

    "Debug" : {
        "iVerbose" : 0,
        "dt" : 60.0},

    "Mysetting" : {
        "setting1" : "this_is_an_example",
        "setting2" : 42,
        "setting3" : true},

    "StartTime" : [2011, 3, 20, 0, 0, 0],
    "EndTime" : [2011, 3, 20, 1, 0, 0],

    "OmniwebFiles" : ["UA/inputs/omni_20110319.txt"],
    "ElectrodynamicsFile" : "UA/inputs/b20110320n_omni.bin",

    "Restart" : {
        "do" : true,
        "dt" : 900.0}

}
```

json is very similar to python dictionaries, with keys and values within the code, e.g.:

settings["Planet"] = "earth"

settings["Euv"]["Model"] = "euvac"

json is a standard format, with libraries available to read/write

Use a default.json file to set defaults in code

Use aether.json to perturb defaults

Developers can add variables without having to add code at all!

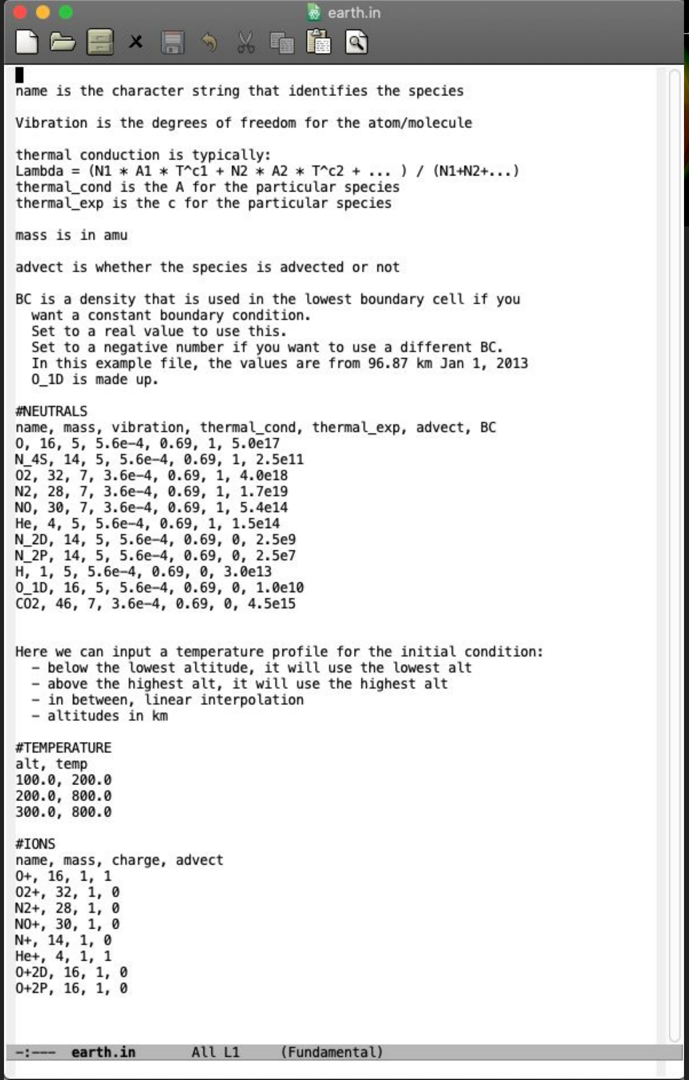
earth.in

This is still in the old-school method, so need to update.

Set species to model (string values are super important here, need to match with all other CSV files!), with masses, vibration, conduction, whether to advect, and lower boundary condition

Set initial temperature condition

Will swap out BCs and ICs with MSIS, but this is easy for now

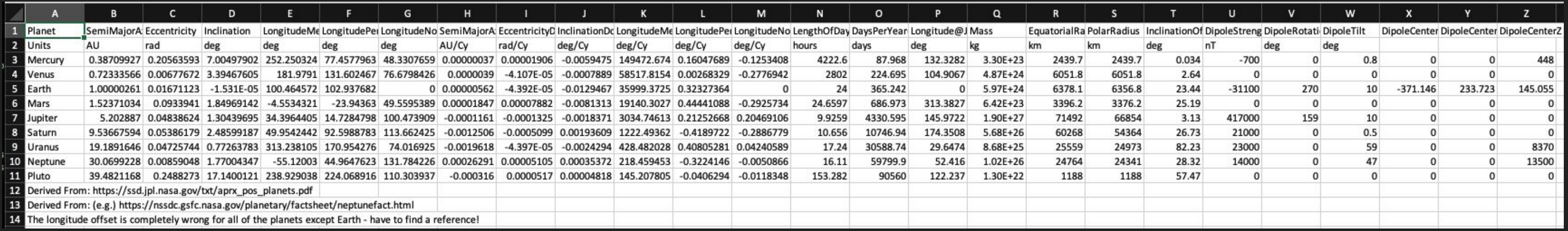


```
name is the character string that identifies the species

Vibration is the degrees of freedom for the atom/molecule

thermal conduction is typically:
Lambda = (N1 * A1 * T^c1 + N2 * A2 * T^c2 + ... ) / (N1+N2+...)
thermal_cond is the A for the particular species
thermal_exp is the c for the particular species

mass is in amu

advect is whether the species is advected or not

BC is a density that is used in the lowest boundary cell if you
  want a constant boundary condition.
  Set to a real value to use this.
  Set to a negative number if you want to use a different BC.
  In this example file, the values are from 96.87 km Jan 1, 2013
  O_1D is made up.

#NEUTRALS
name, mass, vibration, thermal_cond, thermal_exp, advect, BC
O, 16, 5, 5.6e-4, 0.69, 1, 5.0e17
N_4S, 14, 5, 5.6e-4, 0.69, 1, 2.5e11
O2, 32, 7, 3.6e-4, 0.69, 1, 4.0e18
N2, 28, 7, 3.6e-4, 0.69, 1, 1.7e19
NO, 30, 7, 3.6e-4, 0.69, 1, 5.4e14
He, 4, 5, 5.6e-4, 0.69, 1, 1.5e14
N_2D, 14, 5, 5.6e-4, 0.69, 0, 2.5e9
N_2P, 14, 5, 5.6e-4, 0.69, 0, 2.5e7
H, 1, 5, 5.6e-4, 0.69, 0, 3.0e13
O_1D, 16, 5, 5.6e-4, 0.69, 0, 1.0e10
CO2, 46, 7, 3.6e-4, 0.69, 0, 4.5e15


Here we can input a temperature profile for the initial condition:
  - below the lowest altitude, it will use the lowest alt
  - above the highest alt, it will use the highest alt
  - in between, linear interpolation
  - altitudes in km

#TEMPERATURE
alt, temp
100.0, 200.0
200.0, 800.0
300.0, 800.0

#IONS
name, mass, charge, advect
O+, 16, 1, 1
O2+, 32, 1, 0
N2+, 28, 1, 0
NO+, 30, 1, 0
N+, 14, 1, 0
He+, 4, 1, 1
O+2D, 16, 1, 0
O+2P, 16, 1, 0
```

Planetary Characteristics

CSV file sets: orbital characteristic about the sun, rotation (long -> local time needs work), mass, equatorial and polar radius, dipole characteristics

All planets in system, can easily make more bodies



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	Planet	SemiMajorA	Eccentricity	Inclination	LongitudeMe	LongitudePe	LongitudeNo	SemiMajorA	EccentricityD	InclinationDc	LongitudeMe	LongitudePe	LongitudeNo	LengthOfDay	DaysPerYear	Longitude@J	Mass	EquatorialRa	PolarRadius	InclinationOf	DipoleStreng	DipoleRotati	DipoleTilt	DipoleCenter	DipoleCenter	DipoleCenterZ
2	Units	AU	rad	deg	deg	deg	deg	AU/Cy	rad/Cy	deg/Cy	deg/Cy	deg/Cy	deg/Cy	hours	days	deg	kg	km	km	deg	nT	deg	deg			
3	Mercury	0.38709927	0.20563593	7.00497902	252.250324	77.4577963	48.3307659	0.00000037	0.00001906	-0.0059475	149472.674	0.16047689	-0.1253408	4222.6	87.968	132.3282	3.30E+23	2439.7	2439.7	0.034	-700	0	0.8	0	0	448
4	Venus	0.72333566	0.00677672	3.39467605	181.9791	131.602467	76.6798426	0.0000039	-4.107E-05	-0.0007889	58517.8154	0.00268329	-0.2776942	2802	224.695	104.9067	4.87E+24	6051.8	6051.8	2.64	0	0	0	0	0	0
5	Earth	1.00000261	0.01671123	-1.531E-05	100.464572	102.937682	0	0.00000562	-4.392E-05	-0.0129467	35999.3725	0.32327364	0	24	365.242	0	5.97E+24	6378.1	6356.8	23.44	-31100	270	10	-371.146	233.723	145.055
6	Mars	1.52371034	0.0933941	1.84969142	-4.5534321	-23.94363	49.5595389	0.00001847	0.00007882	-0.0081313	19140.3027	0.44441088	-0.2925734	24.6597	686.973	313.3827	6.42E+23	3396.2	3376.2	25.19	0	0	0	0	0	0
7	Jupiter	5.202887	0.04838624	1.30439695	34.3964405	14.7284798	100.473909	-0.0001161	-0.0001325	-0.0018371	3034.74613	0.21252668	0.20469106	9.9259	4330.595	145.9722	1.90E+27	71492	66854	3.13	417000	159	10	0	0	0
8	Saturn	9.53667594	0.05386179	2.48599187	49.9542442	92.5988783	113.662425	-0.0012506	-0.0005099	0.00193609	1222.49362	-0.4189722	-0.2886779	10.656	10746.94	174.3508	5.68E+26	60268	54364	26.73	21000	0	0.5	0	0	0
9	Uranus	19.1891646	0.04725744	0.77263783	313.238105	170.954276	74.016925	-0.0019618	-4.397E-05	-0.0024294	428.482028	0.40805281	0.04240589	17.24	30588.74	29.6474	8.68E+25	25559	24973	82.23	23000	0	59	0	0	8370
10	Neptune	30.0699228	0.00859048	1.77004347	-55.12003	44.9647623	131.784226	0.00026291	0.00005105	0.00035372	218.459453	-0.3224146	-0.0050866	16.11	59799.9	52.416	1.02E+26	24764	24341	28.32	14000	0	47	0	0	13500
11	Pluto	39.4821168	0.2488273	17.1400121	238.929038	224.068916	110.303937	-0.000316	0.0000517	0.00004818	145.207805	-0.0406294	-0.0118348	153.282	90560	122.237	1.30E+22	1188	1188	57.47	0	0	0	0	0	0
12	Derived From: https://ssd.jpl.nasa.gov/txt/aprx_pos_planets.pdf																									
13	Derived From: (e.g.) https://nssdc.gsfc.nasa.gov/planetary/factsheet/neptunefact.html																									
14	The longitude offset is completely wrong for all of the planets except Earth - have to find a reference!																									

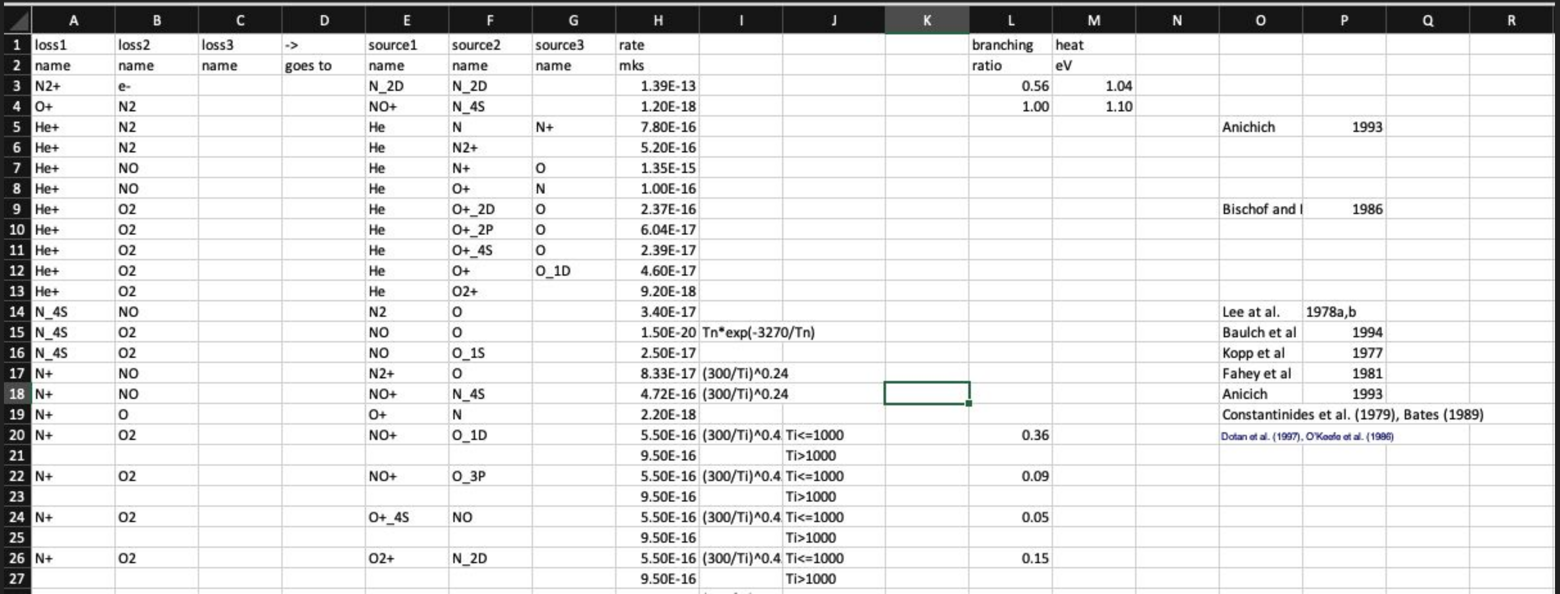
Chemistry

CSV file sets sources, losses, rates, branching ratios, heating

Working on incorporating temperature dependent rates (student)

Currently have two chemistry files of varying complexity

System for reporting chemical equations so user can see what is used



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	loss1	loss2	loss3	->	source1	source2	source3	rate				branching	heat					
2	name	name	name	goes to	name	name	name	mks				ratio	eV					
3	N2+	e-			N_2D	N_2D		1.39E-13				0.56	1.04					
4	O+	N2			NO+	N_4S		1.20E-18				1.00	1.10					
5	He+	N2			He	N	N+	7.80E-16							Anichich	1993		
6	He+	N2			He	N2+		5.20E-16										
7	He+	NO			He	N+	O	1.35E-15										
8	He+	NO			He	O+	N	1.00E-16										
9	He+	O2			He	O+_2D	O	2.37E-16							Bischof and I	1986		
10	He+	O2			He	O+_2P	O	6.04E-17										
11	He+	O2			He	O+_4S	O	2.39E-17										
12	He+	O2			He	O+	O_1D	4.60E-17										
13	He+	O2			He	O2+		9.20E-18										
14	N_4S	NO			N2	O		3.40E-17							Lee at al.	1978a,b		
15	N_4S	O2			NO	O		1.50E-20	Tn*exp(-3270/Tn)						Baulch et al	1994		
16	N_4S	O2			NO	O_1S		2.50E-17							Kopp et al	1977		
17	N+	NO			N2+	O		8.33E-17	(300/Ti)^0.24						Fahey et al	1981		
18	N+	NO			NO+	N_4S		4.72E-16	(300/Ti)^0.24						Anicich	1993		
19	N+	O			O+	N		2.20E-18							Constantinides et al. (1979), Bates (1989)			
20	N+	O2			NO+	O_1D		5.50E-16	(300/Ti)^0.4	Ti<=1000		0.36			Dotan et al. (1997), O'Keefe et al. (1986)			
21								9.50E-16		Ti>1000								
22	N+	O2			NO+	O_3P		5.50E-16	(300/Ti)^0.4	Ti<=1000		0.09						
23								9.50E-16		Ti>1000								
24	N+	O2			O+_4S	NO		5.50E-16	(300/Ti)^0.4	Ti<=1000		0.05						
25								9.50E-16		Ti>1000								
26	N+	O2			O2+	N_2D		5.50E-16	(300/Ti)^0.4	Ti<=1000		0.15						
27								9.50E-16		Ti>1000								

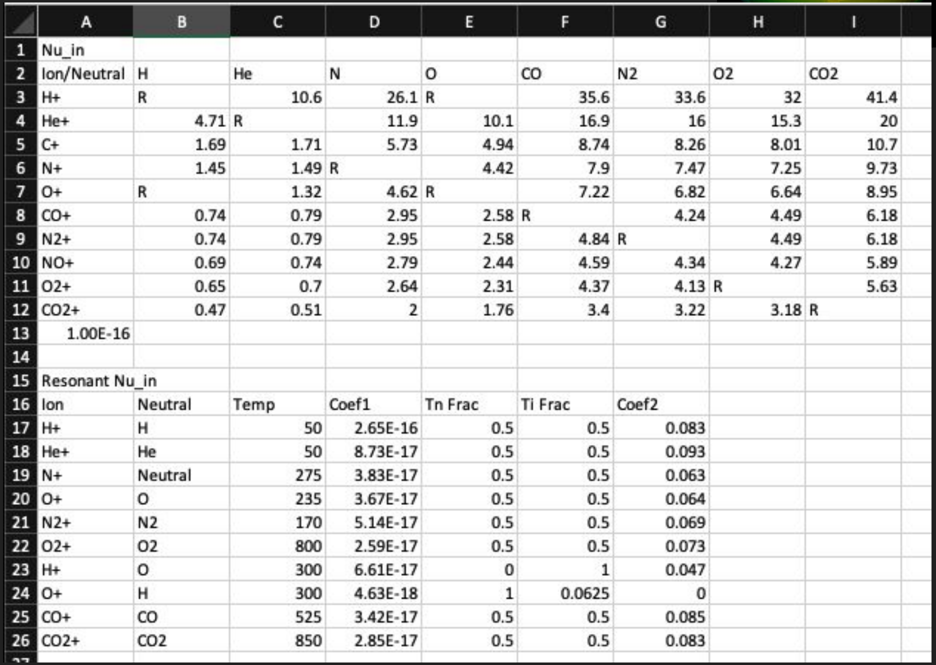
Collisions

Resonant and non-resonant ion-neutral collision frequencies

Resonant can be temperature dependent (since they are)

Code detects "R" and swaps in resonant collision frequencies

Used in ion drift calculations



	A	B	C	D	E	F	G	H	I
1	Nu_in								
2	Ion/Neutral	H	He	N	O	CO	N2	O2	CO2
3	H+	R	10.6	26.1	R	35.6	33.6	32	41.4
4	He+	4.71	R	11.9	10.1	16.9	16	15.3	20
5	C+	1.69	1.71	5.73	4.94	8.74	8.26	8.01	10.7
6	N+	1.45	1.49	R	4.42	7.9	7.47	7.25	9.73
7	O+	R	1.32	4.62	R	7.22	6.82	6.64	8.95
8	CO+	0.74	0.79	2.95	2.58	R	4.24	4.49	6.18
9	N2+	0.74	0.79	2.95	2.58	4.84	R	4.49	6.18
10	NO+	0.69	0.74	2.79	2.44	4.59	4.34	4.27	5.89
11	O2+	0.65	0.7	2.64	2.31	4.37	4.13	R	5.63
12	CO2+	0.47	0.51	2	1.76	3.4	3.22	3.18	R
13	1.00E-16								
14									
15	Resonant Nu_in								
16	Ion	Neutral	Temp	Coef1	Tn Frac	Ti Frac	Coef2		
17	H+	H	50	2.65E-16	0.5	0.5	0.083		
18	He+	He	50	8.73E-17	0.5	0.5	0.093		
19	N+	Neutral	275	3.83E-17	0.5	0.5	0.063		
20	O+	O	235	3.67E-17	0.5	0.5	0.064		
21	N2+	N2	170	5.14E-17	0.5	0.5	0.069		
22	O2+	O2	800	2.59E-17	0.5	0.5	0.073		
23	H+	O	300	6.61E-17	0	1	0.047		
24	O+	H	300	4.63E-18	1	0.0625	0		
25	CO+	CO	525	3.42E-17	0.5	0.5	0.085		
26	CO2+	CO2	850	2.85E-17	0.5	0.5	0.083		

EUV

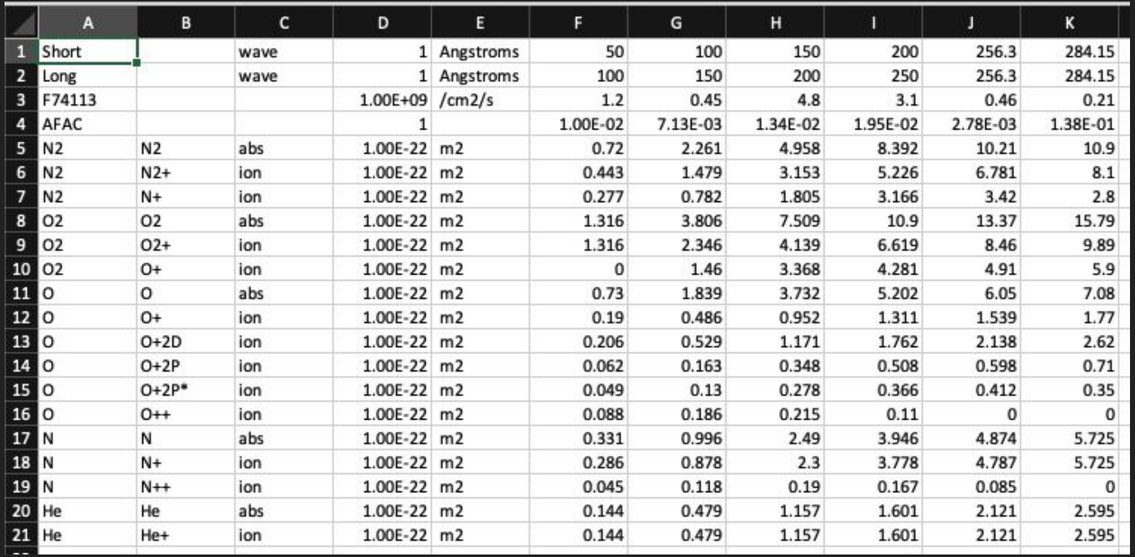
Wavelength ranges

Method to calculate spectra

Absorption cross sections

Ionization cross sections

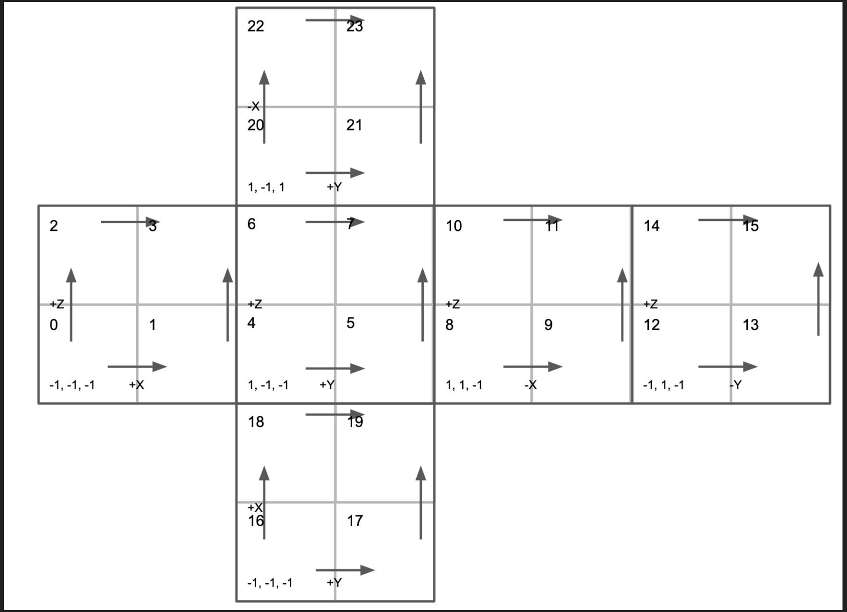
Still need more characteristics, but a good start (e.g., nighttime EUV, photoelectron augmentation to ionization)



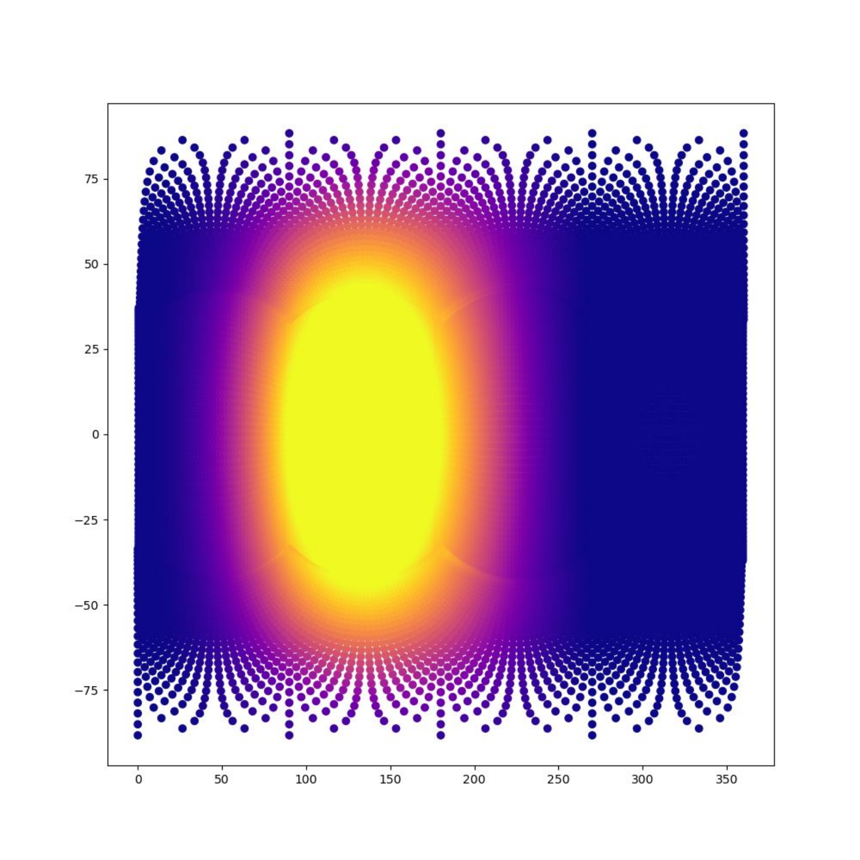
	A	B	C	D	E	F	G	H	I	J	K
1	Short		wave	1	Angstroms	50	100	150	200	256.3	284.15
2	Long		wave	1	Angstroms	100	150	200	250	256.3	284.15
3	F74113			1.00E+09	/cm2/s	1.2	0.45	4.8	3.1	0.46	0.21
4	AFAC			1		1.00E-02	7.13E-03	1.34E-02	1.95E-02	2.78E-03	1.38E-01
5	N2	N2	abs	1.00E-22	m2	0.72	2.261	4.958	8.392	10.21	10.9
6	N2	N2+	ion	1.00E-22	m2	0.443	1.479	3.153	5.226	6.781	8.1
7	N2	N+	ion	1.00E-22	m2	0.277	0.782	1.805	3.166	3.42	2.8
8	O2	O2	abs	1.00E-22	m2	1.316	3.806	7.509	10.9	13.37	15.79
9	O2	O2+	ion	1.00E-22	m2	1.316	2.346	4.139	6.619	8.46	9.89
10	O2	O+	ion	1.00E-22	m2	0	1.46	3.368	4.281	4.91	5.9
11	O	O	abs	1.00E-22	m2	0.73	1.839	3.732	5.202	6.05	7.08
12	O	O+	ion	1.00E-22	m2	0.19	0.486	0.952	1.311	1.539	1.77
13	O	O+2D	ion	1.00E-22	m2	0.206	0.529	1.171	1.762	2.138	2.62
14	O	O+2P	ion	1.00E-22	m2	0.062	0.163	0.348	0.508	0.598	0.71
15	O	O+2P*	ion	1.00E-22	m2	0.049	0.13	0.278	0.366	0.412	0.35
16	O	O++	ion	1.00E-22	m2	0.088	0.186	0.215	0.11	0	0
17	N	N	abs	1.00E-22	m2	0.331	0.996	2.49	3.946	4.874	5.725
18	N	N+	ion	1.00E-22	m2	0.286	0.878	2.3	3.778	4.787	5.725
19	N	N++	ion	1.00E-22	m2	0.045	0.118	0.19	0.167	0.085	0
20	He	He	abs	1.00E-22	m2	0.144	0.479	1.157	1.601	2.121	2.595
21	He	He+	ion	1.00E-22	m2	0.144	0.479	1.157	1.601	2.121	2.595

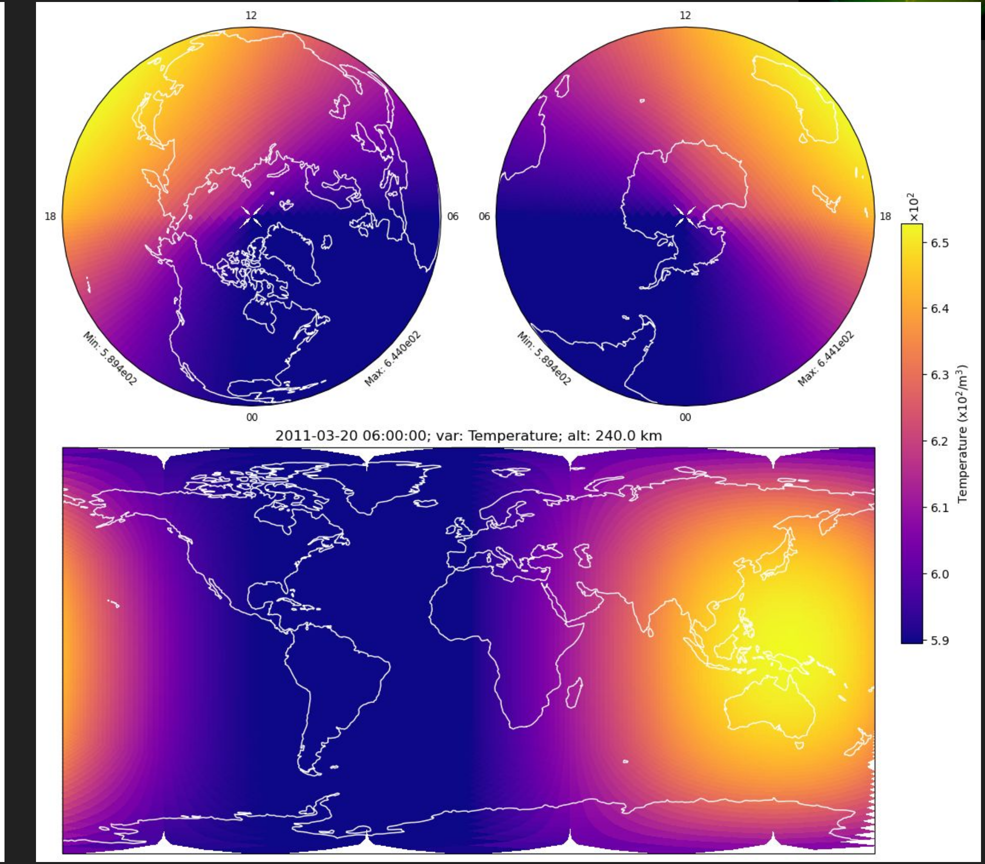
Grid Systems

- Both spherical and CubedSphere grids implemented
- Quad-tree domain decomposition implemented
  - Very simply at this point
  - Only uniform grid allowed now
- Outputs in blocks so plotting is more complicated

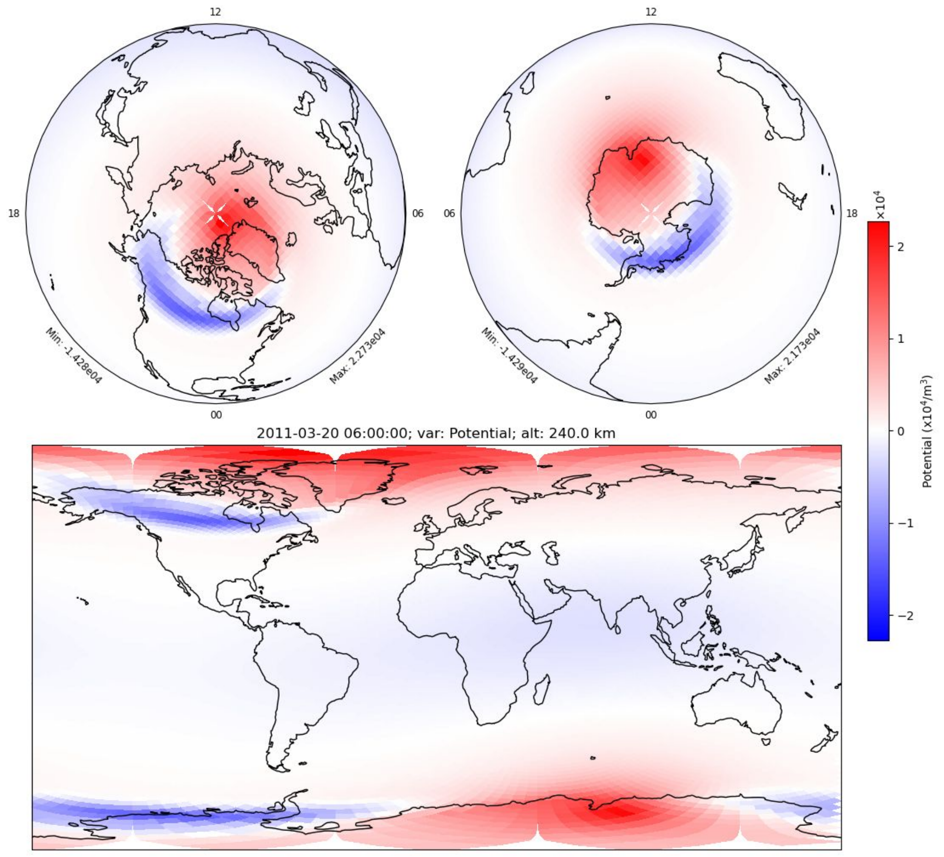


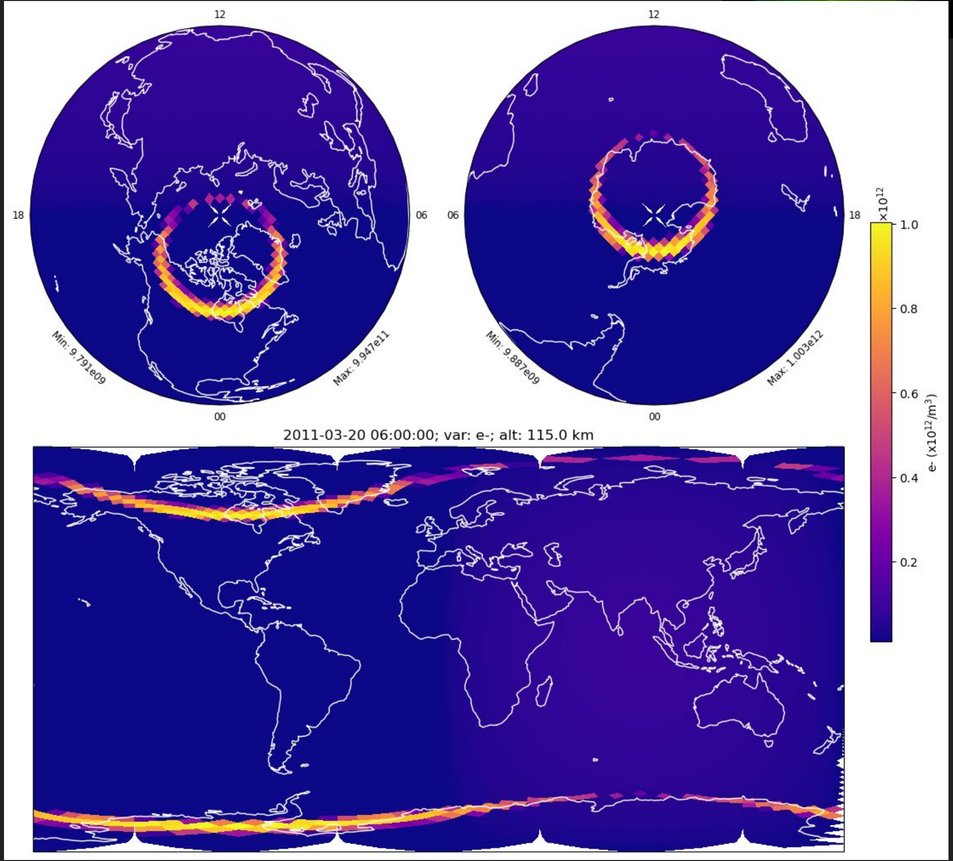
Some Examples - 2° resolution





Potential and Aurora





Summary

- GITM
  - Have made a few changes to increase stability and improve performance
  - Latest version on github is up-to-date
- Aether
  - New model of thermosphere / ionosphere
  - Extreme flexibility in physics and specifications
  - Spherical grid and CubedSphere grids available
  - Still in development