

# USERS GUIDE: v1.0 - SAMI3-3.22

J.D. Huba (11/2/20)

## 1 Running SAMI3: quickstart

```
> cd ~
> mkdir sami3
> cd sami3
  (put 'sami3-3.22.tgz' in this directory)
> tar -zxvf sami3-3.22.tgz
```

This will create the directory 'sami3-3.22' which contains the sami3 code.

The code is written in Fortran77 and Fortran90. It requires a Fortran compiler and Open MPI. A description to install Open MPI is provided at the end of this document. The code has been tested on several Fortran compilers but only gfortran and the intel compiler ifort are included in the Makefile. The code uses the message passing interface and requires a multi-processor computer. The code will run on a system with as few as 4 processors (albeit more slowly than a system with more processors).

The files 'mpif90' and 'mpirun' need to be in your path. The test run uses openmpi-2.1.1 and is set up for 8 processors. Newer versions of openmpi (e.g., openmpi-4.0.5 require a 'hostfile'). It should take a few minutes ( $\sim 4 - 8$ ) to run.

```
> cd ~/sami3/sami3-3.22
> make sami3.x
> mpirun -np 8 sami3.x | tee sami3.out
```

The file 'sami3.out' captures the screen output of the run. An example of this file is given here.

### **sami3.out:**

The test run produces a file 'sami3.out' that should look something like this.

```
taskid =          4
  taskid          4
taskid =          3
  taskid          3
taskid =          7
  taskid          7
taskid =          2
```

```

taskid          2
taskid =        5
taskid          5
taskid =        0
taskid          0
taskid =        1
taskid          1
taskid =        6
taskid          6
d = 13326374.0      2.31058780E+10  4.53404467E+10  2.60522317E+09  113087104.
plon_ang, plon = 287.799866      5.02305555
                335
found start      480961          12          1
found end        492481          12          9
i_start,i_end    480961          492481
Completed the LOOP
h:m:s           23           59  0.00000000
By Bz           4.88999987      6.32000017
Vx n            -379.399994      3.55999994
ID of END                8
Data points read:      11520

Vavg (km/s), time shift (h) = 353.357819      0.00000000

```

RECALC\_08: RADIAL SOLAR WIND --> GSW SYSTEM IDENTICAL HERE  
TO STANDARD GSM (I.E., XGSW AXIS COINCIDES WITH EARTH-SUN LINE)

```

stuff           2002          335          0          15          0 -400.000000      0
finished initialization taskid = 0
max,ntmmax      1.11111114E-03      51
finished initialization taskid = 4
iwrk            4           51
in if parallel - iday,iday0      1          0
max flux_euvac  9.37951437E+09
finished initialization taskid = 5
iwrk            5           51
in if parallel - iday,iday0      1          0
max flux_euvac  9.37951437E+09
finished initialization taskid = 1
iwrk            1           51
in if parallel - iday,iday0      1          0
max flux_euvac  9.37951437E+09

```

```

finished initialization taskid =          6
iwrk          6          51
in if parallel - iday,iday0          1          0
max flux_euvac  9.37951437E+09
finished initialization taskid =          2
iwrk          2          51
in if parallel - iday,iday0          1          0
max flux_euvac  9.37951437E+09
finished initialization taskid =          3
iwrk          3          51
in if parallel - iday,iday0          1          0
max flux_euvac  9.37951437E+09
finished initialization taskid =          7
iwrk          7          51
in if parallel - iday,iday0          1          0
max flux_euvac  9.37951437E+09
calling potpphi  0.00000000          4.00000000
      1  4458129.50
      2  0.323069572
      3  0.100905530
      4  6.01667985E-02
      5  4.52680588E-02
      6  2.78896075E-02
      7  4.23603412E-03
hrutw2 = 0.00000000          0.500000000
nweimer          1
      22.5384808
istep =          0 ntm =          1
hr = 0.00000000          dt = 4.00000000
istep =          1 ntm =          2
hr = 1.11111114E-03 dt = 4.00000000
calling potpphi  60.000000          4.00000000
      1  0.494182825
      2  3.63136008E-02
      3  7.99512491E-03
      43.8139343
calling potpphi  60.000000          4.00000000
      1  1.88213363E-02
      44.7439919
calling potpphi  60.000000          4.00000000
      1  2.78082932E-03
      44.8808022

```

MASTER: All Done!

|                        |            |             |
|------------------------|------------|-------------|
| done finalizing,taskid | 0          |             |
| timing 57.2304001      | 209.395187 | 152.164795  |
| done finalizing,taskid | 3          |             |
| timing 80.3911896      | 209.405426 | 129.014236  |
| done finalizing,taskid | 7          |             |
| timing 80.3980789      | 209.408463 | 129.010376  |
| done finalizing,taskid | 4          |             |
| timing 57.7652969      | 209.404724 | 151.639435  |
| done finalizing,taskid | 5          |             |
| timing 67.7987061      | 209.406464 | 141.6 07758 |
| done finalizing,taskid | 6          |             |
| timing 74.1990891      | 209.406219 | 135.207123  |
| done finalizing,taskid | 1          |             |
| timing 67.8977051      | 209.407791 | 141.510086  |
| done finalizing,taskid | 2          |             |
| timing 74.2042007      | 209.407684 | 135.203491  |

## 2 parameter\_mod.f90

```
module parameter_mod

!     total number of workers
integer,parameter :: numwork = 7

!     number of altitudes (e.g., field lines)

integer,parameter :: nf      = 124
integer,parameter :: nfp1 = nf + 1
integer,parameter :: nfm1 = nf - 1
integer,parameter :: nfm2 = nf - 2

!     number of grid cells along field line
!     nseg needs to be even

integer,parameter :: nz0 = 204, nextra = 0, nseg = 2, &
                    nze = nseg * nextra, nz = nz0+nze, &
                    nzp1 = nz + 1, &
                    nzm1 = nz - 1

!     number of grid cells in longitudinal direction per worker
```

```
integer,parameter :: nl    = 22, &
                    nlp1  = nl + 1, &
                    nlm1  = nl - 1
```

```
!      Total number of grid cells in longitudinal direction
```

```
integer,parameter :: nlt   = numwork*(nl - 2), &
                    nltp1  = nlt + 1, &
                    nltm1  = nlt - 1
```

The subroutine ‘parameter\_mod.f90’ has the variables that specify the number of processors, as well as the variable and grid indices. In general, the only parameters that should be changed are the following (shown above in the top section of parameter\_mod.f90)

- **numwork**: number of processors
- **nz**: number of grid points along the magnetic field
- **nf**: number of ‘field lines’
- **nl**: number of grid points in the zonal (longitudinal) direction for each processor

The total number of grid points in the zonal (longitudinal) direction is  $nlt = numwork \cdot (nl - 2)$ . Nominal values for these parameters are  $nz = 204 - 304$ ,  $nf = 80 - 124$ , and  $nlt = 96 - 144$ . For example, if  $numworker = 6$  and  $nl = 18$  then  $nlt = 96$ , or  $numworker = 12$  and  $nl = 10$  then  $nlt = 96$ . For a given value of  $nlt$ , the code runs faster for larger  $numwork$  and smaller  $nl$ . The minimum value of  $nl$  is 4. [note: to model the plasmasphere,  $nz = 304$  should be used.]

The labeling of the ion and neutral densities are also given in ‘parameter\_mod.f90’ as shown below.

```
!      ion densities
```

```
integer,parameter :: nion  = 7    ! number of ions
integer,parameter :: pthp  = 1    ! h+
integer,parameter :: pthep = 5    ! he+
integer,parameter :: ptnp  = 7    ! n+
integer,parameter :: ptop  = 2    ! o+
integer,parameter :: ptn2p = 6    ! n2+
integer,parameter :: ptnop = 3    ! no+
integer,parameter :: pto2p = 4    ! o2+
```

```
!      neutrals
```

```

integer,parameter :: nneut = 7    ! number of neutrals
integer,parameter :: pth  = 1    ! h
integer,parameter :: pthe = 5    ! he
integer,parameter :: ptn  = 7    ! n
integer,parameter :: pto  = 2    ! o
integer,parameter :: ptn2 = 6    ! n2
integer,parameter :: ptno = 3    ! no
integer,parameter :: pto2 = 4    ! o2

```

### 3 sami3-3.22.namelist

```

&go
maxstep = 50,
hrmax   = 24.,
dthr    = .25e-5,
hrpr    = 0.,
dt0     = 4.,
rmin    = 90.,
altmin  = 85.,
fbar    =157.664,157.848,157.920,158.015,157.863,157.726,157.537,157.300,
f10p7   =145.000,142.000,141.000,144.000,144.000,143.000,146.000,149.000,
ap      =18.0000,13.0000,11.0000,12.0000,9.00000,9.00000,18.0000,12.0000,
iyear   = 2002,
nday    = 335,336,337,338,339,340,341,342,
imonth_1 = 12,
iday_1   = 1,
imonth_2 = 12,
iday_2   = 9,
mmass    = 48 ,
nion1    = 1,
nion2    = 7,
hrinit   = 0.,
tvn0     = 1.,
tvexb0   = 1.,
ver      = 0.,
veh      = 0.,
vw       = 0.
gams     = .1,
gamp     = 2.5,
alt_crit_avg = 6370.
blat_min = 6.85,

```

```

blat_max = 89.,
snn      = 1.,1.,1.,1.,1.,1.,1.,
stn      = 1.,
denmin   = 1.e-6,
alt_crit = 150.,
cqe      = 6.e-14,
psmooth  = 1, ! smoother for vsi
hall     = .true.,
restart  = .false.,
storm_ti = 52.,
storm_tf = 60.,
vexb_max = 4e6,
lmadala  = .true.,
lcr      = .false.,
lvs      = .false.,
lweimer  = .true., ! set lvs = .false. if .true
decay_time = 1.,
pcrit    = 8.,
lhwm93   = .false.,
lhwm14   = .true.,
anu_drag0 = 4.e3,
kp       = 2,3,6,5,4,3,3,4,
xhardy   = 1.,
tphi     = 60.,
euv_fac  = 1.,
tmax     = 1.e4,
lfism    = .false.,
&end

```

The subroutine 'sami3-3.22.namelist' shown above has the variables that specify the run conditions. The code is designed to run a maximum of 7 days.

- **maxstep**: maximum number of time steps
  - for a normal run (e.g., 48 hrs), this is a large integer, e.g., maxstep = 10000000
  - for testing/debugging it can be set to a smaller number, e.g., maxstep = 10
- **hrmax**: runtime (hrs)
  - maximum runtime is hrmax = 168 (7 days)
  - nominal minimum runtime is hrmax = 36  
(it takes ~ 12 hrs for the ionosphere to develop)
- **dthr**: output cadence (hrs)
  - nominal output cadence is dthr = 0.25 (15 min)
  - for testing/debugging it can be reduced to output data more often

- **hrpr**: time data starts to be output
  - nominally  $hrpr = 0$  and data is output from beginning of the run
  - alternatively, if  $hrmax = 48$  and  $hrpr = 24$ , data is output starting after the first day (this reduces the size of the output files)
- **dt0**: maximum time step (sec)
  - should be in the range 2 - 8 sec
  - the larger  $dt0$ , the faster the code runs in general
  - however, depending on geophysical conditions and the grid, the code can give unphysical results if initially too large (e.g.,  $dt0 = 8$ )
- **rmin**: used to specify grid along the magnetic field
  - best to leave  $rmin = 90$
- **altmin**: used to specify grid along the magnetic field
  - best to leave  $altmin = 85$  km
- **fbar**: F10.7A
  - 81 day average of F10.7
- **f10p7**: F10.7
  - solar radio flux at 10.7 cm
  - used as a measure of solar activity
- **ap**: Ap index
  - The Ap index is a daily measure of magnetic activity
- **iyear**: year
- **nday**: day-of-year
  - the first value is the starting day-of-year
- **imonth\_1**: month of starting day-of-year
- **iday\_1**: day (in month) of starting day-of-year
- **imonth\_2**: month of last day-of-year
- **iday\_2**: day (in month) of last day-of-year
- **mmass**: mass number used in nrlmsise00
  - $mmass = 48$  (do not change)
- **nion1**: first ion specie
  - $nion1 = 1$  ( $H^+$ ) (do not change)



- **nion2**: last ion specie
  - nion1 = 7 (N<sup>+</sup>) (do not change)
- **hrinit**: starting time (UT hrs)
  - nominal value hrinit = 0 (i.e., midnight GMT) - best not to change
- **tvn0**: modifies the magnitude of the neutral wind speed
  - nominally tvn0 = 1
  - use caution when changing, e.g., tvn0 = 2 will double the neutral wind speed
- **tvexb0**: modifies the magnitude of the E × B velocity
  - nominally tvexb0 = 1
  - use caution when changing, e.g., tvexb0 = 0 will set the E × B drift to 0
- **ver**: not used (leave 0)
- **veh**: not used (leave 0)
- **vw**: constant vertical wind (cm/s)
  - best to leave vw = 0
- **gams**: adjusts grid spacing along the magnetic field line
  - best to leave gams = .1
- **gamp**: adjusts grid spacing in latitude
  - best to leave gams = 2.5
- **alt\_crit\_avg**: used to specify values at high latitude boundary
  - best to leave alt\_crit\_avg = 6370.
- **blat\_min**: latitude of the base of the first field line
  - best to leave blat\_min = 6.85
- **blat\_max**: latitude of the base of the highest field line
  - nominal values 60° - 89°
  - maximum value is 89°
- **snn**: multiplier for neutral densities
  - the 7 entries correspond to the 7 neutral species in ‘parameter\_mod.f90’
  - for example, if snn = 1.,2.,1.,1.,1.,1.,1., then the oxygen neutral density is doubled
- **stn**: multiplier for the neutral temperature
  - this will change the ion temperatures because of ion-neutral coupling
  - it will not change the scale height of the neutrals
- **denmin**: minimum ion density allowed (cm<sup>-3</sup>)
  - best to leave denmin = 1.e-6

- **psmooth**: smoothing factor (not used)
- **hall**: Hall conductivity
  - hall = .true. (include the Hall and Pedersen conductances in the potential solver)
  - hall = .false. (include only the Pedersen conductance in the potential solver)
  - by toggling this switch the influence of the Hall conductance on the electric field is elucidated
  - generally, hall = .true. should be used
- **restart**: restart a run
  - restart = .false. (do not restart)
  - restart = .true. (restart a run; needs rst files)
- **storm\_ti**: for Volland-Stern potential (not used)
- **storm\_tf**: for Volland-Stern potential (not used)
- **vexb\_max**: maximum  $E \times B$  velocity (cm/s)
  - nominally use 40 km/s; doesn't affect ionosphere dynamics
- **lmadala**: potential solver
  - lmadala = .true. (solves for the neutral wind dynamo field)
  - lmadala = .false. (does not solve for the dynamo field:  $E \times B$  drift is zero)
- **lcr**: corotation potential
  - lcr = .true. (adds the corotation potential)
  - lcr = .false. (does not add the corotation potential)
  - best to leave lcr = .false.
- **lvs**: Volland-Stern high latitude potential
  - best to leave lvs = .false.
- **lweimer**: Weimer high latitude potential
  - lweimer = .true. (use Weimer high latitude potential)  
needs OMNI.txt file linked to year of Weimer data file
  - lweimer = .false. (do not use Weimer high latitude potential)
- **decay\_time**: enhanced loss of  $H^+$  and  $He^+$ 
  - invoked at distance greater than  $pcrit \cdot R_E$  where  $R_E$  is the earth radius
- **pcrit**: used for enhanced loss (above)
  - nominally, pcrit = 8
- **lhwm93**: HWM93 neutral wind model
  - lhwm93 = .true. (use HWM93)
  - lhwm93 = .false. (do not use HWM93)
  - nominally use lhwm93 = .false.

- **lhwm14**: HWM14 neutral wind model
  - lhwm14 = .true. (use HWM14)
  - lhwm14 = .false. (do not use HWM14)
  - nominally use lhwm14 = .true.
- **anu\_drag0**: anomalous ion drag
  - nominally, use anu\_drag0 = 4.e3
  - used to prevent high velocity nighttime flows into the ionosphere from the plasmasphere
- **kp**: Kp index
  - used in the Hardy model
  - valid values are 1 - 6
- **xhardy**: modifies ionization associated with high latitude precipitation
  - nominal value xhardy = 1
- **tphi**: how often the potential equation is solved (sec)
  - nominal value tphi = 60
- **euv\_fac**: modifies EUVAC spectrum
  - nominal value euv\_fac = 1
  - euv\_fac < 1 decreases EUV intensity
  - euv\_fac > 1 increases EUV intensity
- **tmax**: maximum electron temperature (K)
  - nominal range  $10^4$  -  $10^5$
- **lfism**: FISM EUV spectrum
  - lfism = .true. (use FISM daily spectrum)  
needs fism linked to FISM year data (up to 2018)
  - lfism = .false. (use EUVAC spectrum)

## 4 Running SAMI3: redux

The code does not have to be recompiled if changes are only made to ‘sami3-3.22.namelist’. However, changes to the file ‘parameter\_mod.f90’ require the code to be recompiled.

```
> make clean
> make sami3
```

If ‘lweimer = .true.’ then the appropriate OMNI data file needs to be linked to ‘OMNI.txt’. For example, is ‘iyear = 2014’ then

```
> ln -sfm ./weimer/2014/OMNI_HRO_1MIN_2014.txt OMNI.txt
```

If 'lfism = .true.' then the appropriate FISM daily data file needs to be linked to 'fism'. For example, is 'iyear = 2014' then

```
> ln -sfm ./fism/2014O fism
```

In general, the command to run the code is

```
> mpirun -np N sami3.x | tee sami3.out
```

where  $N = \text{numwork} + 1$  and the hostfile needs to be edited where 'slots = N' (if a hostfile is needed for the version of Open MPI being used).

## 5 Output files

The output files are written as unformatted Fortran files (except 'time.dat'). The grid data is written out in the file 'sami3-3.22.f90' and the remaining data files are written out in 'output.f90'. The following data files are output:

|            |  |
|------------|--|
| deneu.dat  | electron density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt) |
| deni1u.dat | H+ density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)       |
| deni2u.dat | O+ density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)       |
| deni3u.dat | NO+ density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)      |
| deni4u.dat | O2+ density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)      |
| deni5u.dat | He+ density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)      |
| deni6u.dat | N2+ density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)      |
| deni7u.dat | N+ density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)       |
| denn1u.dat | H density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)        |
| denn2u.dat | O density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)        |
| denn3u.dat | NO density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)       |
| denn4u.dat | O2 density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)       |
| denn5u.dat | He density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)       |
| denn6u.dat | N2 density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)       |
| denn7u.dat | N density ( $\text{cm}^{-3}$ ) (nz,nf,nlt,nt)        |
| teu.dat    | electron temperature (K) (nz,nf,nlt,nt)              |
| ti1u.dat   | H+ temperature (K) (nz,nf,nlt,nt)                    |
| ti2u.dat   | O+ temperature (K) (nz,nf,nlt,nt)                    |
| ti5u.dat   | He+ temperature (K) (nz,nf,nlt,nt)                   |
| vs1u.dat   | H+ velocity parallel to B (cm/s) (nz,nf,nlt,nt)      |
| vsi2u.dat  | O+ velocity parallel to B (cm/s) (nz,nf,nlt,nt)      |
| u1pu.dat   | meridional E x B velocity (cm/s) (nz,nf,nlt,nt)      |
| u3hu.dat   | zonal E x B velocity (cm/s) (nz,nf,nlt,nt)           |

|           |  |
|-----------|--|
| u1u.dat   | zonal neutral wind velocity (cm/s) (nz,nf,nlt,nt)        |
| u2u.dat   | meridional neutral wind velocity (cm/s) (nz,nf,nlt,nt)   |
| baltu.dat | altitude in magnetic coordinates (km) (nz,nf,nlt)        |
| blatu.dat | latitude in magnetic coordinates (degree) (nz,nf,nlt)    |
| blonu.dat | longitude in magnetic coordinates (degree) (nz,nf,nlt)   |
| zaltu.dat | altitude in geographic coordinates (km) (nz,nf,nlt)      |
| glatu.dat | latitude in geographic coordinates (degree) (nz,nf,nlt)  |
| glonu.dat | longitude in geographic coordinates (degree) (nz,nf,nlt) |
| phiu.dat  | electrostatic potential (statvolt) (nnx,nny)             |

Other variables can be output by uncommenting them in the ‘output.f90’ file. The number of time steps is the number of lines in the ‘time.dat’ file (e.g., can be found by ‘> wc time.dat’ or ‘> tail time.dat’ - kinda ugly).

## 6 Utility Files

### 6.1 Interpolation

#### 6.1.1 TEC

In the utility directory there is a sub-directory ‘TEC’ that contains the programs:

- interp-3d-mapB.f
  - calculates the mapping from the SAMI3 grid in magnetic coordinates to a regular lat/lon/alt grid in magnetic coordinates
- interp-3d-dene\_nB.f
  - interpolates the electron density to the regular lat/lon/alt grid in magnetic coordinates
- tec\_nmf2\_hmf2-3dB.f
  - calculates the TEC/nmF2/hmF2 based on the interpolated grid data
- param\_diag\_blank.inc
  - common file that defines the indices
    - nz/nf/nl (SAMI3 grid)
    - nx/ny/nl (interpolated grid)
- run\_interp\_tecB.py
  - python script that compiles and runs above fortran files

Here, ‘nx’ is the index for the interpolated magnetic latitude grid, ‘ny’ is the index for the interpolated altitude grid, and ‘nl’ is the index for the interpolated magnetic longitude grid (which is the same as the native SAMI3 grid). To run the above do the following. First, set up a soft link for ‘data\_dir’ to the directory that contains the SAMI3 data.

```
> ln -sf /sami3_data_directory data_dir
```

Edit the indices  $nz/nf/nl$  in the ‘param\_diag\_blank.inc’ file so that they agree with those used in the SAMI3 simulation. The interpolated indices  $nx/ny$  are nominally set to 100; these can be changed. Run the python script:

```
> python run_interp_tecB.py
Number of time steps:
    (enter the number of output times of the simulation)
baltmin0,baltmax0:
    (enter the min/max altitude of the interpolated grid; nominal values are 90 2200)
input altmax for TEC:
    (enter the max altitude to calculate TEC; nominal value is 2000)
```

The above will generate the following files in the SAMI3 (data) directory

```
blat0.dat: interpolated (magnetic) latitude: blat0(nx)
balt0.dat: interpolated altitude: balt0(ny)
blon0.dat: interpolated (magnetic) longitude: blon0(nl)
dene0B.dat: interpolated electron density: dene0B(nx,ny,nl,nt)
tecuB.dat: TEC(nx,nl,nt)
nmf2uB.dat: nmf2(nx,nl,nt)
hmf2uB.dat: hmf2(nx,nl,nt)
```

Note: Two other files are generated but not used: blat0p.dat and balt0p.dat.

### 6.1.2 BtoG

In the utility directory there is a sub-directory ‘BtoG’ that contains the program ‘btog\_apex.f90’ which maps the interpolated magnetic latitude and longitude grids to a geographic latitude and longitude grid using the Richmond apex model (i.e., IGRF). Prior to running the program, make sure that the parameters ‘ $nx$ ’ and ‘ $ny$ ’ are the same as defined in the ‘param\_diag\_blank.inc’ file when calculating interpolated quantities.

Set up a soft link for ‘data\_dir’ to the directory that contains the SAMI3 data, as well as the file ‘parameter\_mod.f90’

```
> ln -sf /sami3_data_directory data_dir
> ln -sf /sami3_data_directory/parameter_mod.f90 .
```

Compile and run the code

```
> make btog_apex.x > btog_apex.x
```

This will generate the following files in the SAMI3 data directory

```
zalt0B.dat: altitude grid: zalt0B(ny)
glon0B.dat: geographic longitude grid: glon0B(nx,ny,nl)
glat0B.dat: geographic latitude grid: glat0B(nx,ny,nl)
```

The difference in the magnetic and geographic coordinate systems is highlighted in Fig. 1 which shows a representative TEC contour plot (near the winter solstice in December) using magnetic coordinates (blat0/blon0) in the left panel (a) and geographic coordinates (glat0B/glon0B) in the right panel (b). The underlying continents in Fig. 1a are not correct but the point is that the longitudinal behavior of the TEC in Fig. 1b captures the variation of the magnetic equator by the ‘dip’ in the ionization crests over the Atlantic/American sectors.

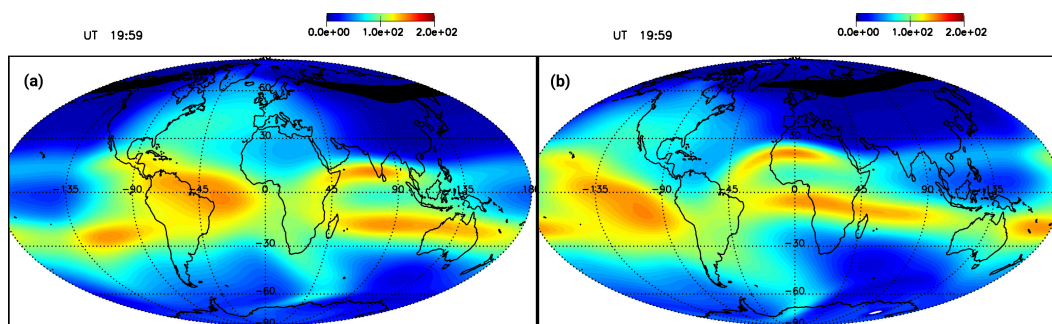


Figure 1: Representative TEC contour plot in magnetic coordinates (a) and geographic (b).

## 6.2 NETCDF

Note: The NETCDF library needs to be installed to use this section. In the SAMI3 data directory make a directory ‘netcdf’ and then copy the files in the ‘utility/netcdf’ directory to this directory. The essential files are ‘sami3\_dene\_tec.f’ and the makefile ‘Makefile.’ The ‘Makefile’ may need to be edited to specify the locations of the NETCDF libraries and include files. They can be found by running

```
> nf-config -all
```

The file ‘sami3\_dene\_tec.f’ needs to be edited to specify the parameters NZ/NF/NL/NT that correspond to those in the SAMI3 data directory. Following this, compile and run the code.

```
> make wr.x
> wr.x
```

This will generate a netcdf file 'sami3\_dene\_tec.nc' which, as an example, will contain something like

```
netcdf sami3_dene_tec {
dimensions:
    nz = 204 ;
    nf = 124 ;
    nl = 96 ;
    nt = UNLIMITED ; // (193 currently)
    nx = 100 ;
    ny = 100 ;
variables:
    float alt(nl, nf, nz) ;
        alt:units = "km" ;
    float lat(nl, nf, nz) ;
        lat:units = "degrees" ;
    float lon(nl, nf, nz) ;
        lon:units = "degrees" ;
    float dene(nt, nl, nf, nz) ;
        dene:units = "cm-3" ;
    float alt0(ny) ;
        alt0:units = "km" ;
    float lat0(nl, ny, nx) ;
        lat0:units = "degrees" ;
    float lon0(nl, ny, nx) ;
        lon0:units = "degrees" ;
    float dene0(nt, nl, ny, nx) ;
        dene0:units = "cm-3" ;
    float tec(nt, nl, nx) ;
        tec:units = "m-2" ;
    float nmf2(nt, nl, nx) ;
        nmf2:units = "cm-3" ;
    float hmf2(nt, nl, nx) ;
        hmf2:units = "km" ;
    float time(nt) ;
        time:units = "hrs" ;
```

Here, the variables 'alt0/lat0/lon0' correspond to 'zalt0B/glat0B/glon0B' discussed above.

The file 'sami3\_dene\_tec.f' can serve as a template to be edited to output other variables to the netcdf file.



### 6.3 HDF5

Note: The HDF5 library needs to be installed to use this section. In the SAMI3 data directory make a directory 'hdf5' and then copy the files in the 'utility/hdf5' directory to this directory. The essential files are 'sami3\_dene\_tec.f' and the makefile 'Makefile.' The 'Makefile' may need to be edited to specify the locations of the HDF5 libraries and include files. The files 'write\_hdf5\_3D\_time\_dene.f90' and 'write\_xdmf\_time\_dene.py' need to be edited to specify the parameters nx/ny/nl/nt that correspond to nz/nf/nl/nt in the SAMI3 data directory [NOTE: this will be fixed so not to confuse with nx/ny used in the interpolation files.] Following this, compile and run the code.

```
> make dene.x  
> dene.x
```

This will generate the hdf5 file 'dene.h5'.

Next run the python script

```
> python write_xdmf_time_dene.py
```

This will generate the xmf file 'dene.xmf'. This file can be opened in the graphics software Paraview or Visit for visualization.

The files 'write\_hdf5\_3D\_time\_dene.f90' and 'write\_xdmf\_time\_dene.py' can serve as templates to be edited to output other variables to the hdf5/xmf files.

## 7 Installing Open MPI

Open MPI can be downloaded from <https://www.open-mpi.org/>. As an example, we will consider the package `openmpi-4.0.5.tgz` and build it for `gfortran`.

```
> cd ~
> mkdir openmpi
> cd openmpi
  (put 'openmpi-4.0.5.tgz' in this directory)
> tar -zxvf openmpi-4.0.5.tgz
> cd openmpi-4.0.5
> export FC=gfortran
> ./configure --prefix=~/.openmpi/openmpi-4.0.5
> make
> make install
```

There should be the files `'mpif90'` and `'mpirun'` in the directory `'~/.openmpi/openmpi-4.0.5/bin/'` These files will be use to compile and run SAMI3. Note: to build Open MPI with intel fortran use `'export FC=ifort'`. Additionally, this version of Open MPI requires a `'hostfile'`; some earlier versions do not require this file.