Latest Developments

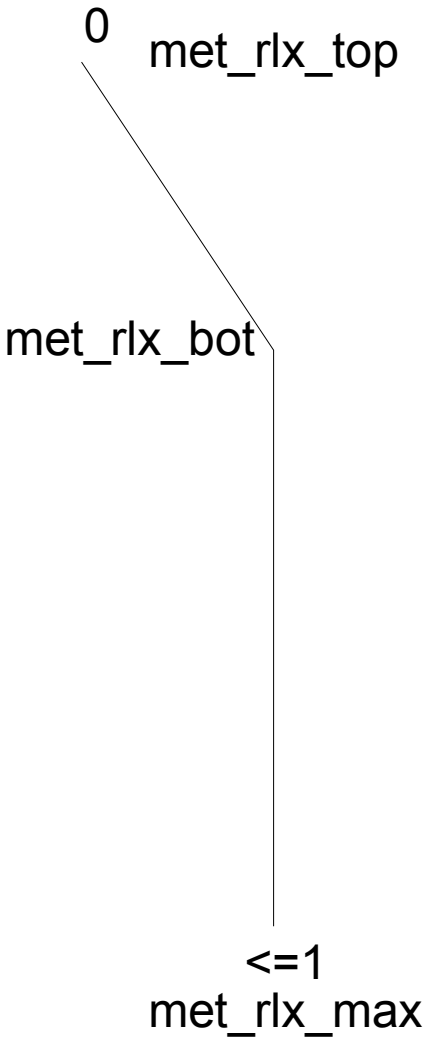
- Specified Dynamics Compets for CAM-chem/ WACCM
  - F_SD_CAMCHEM, F_SD_BAM, F_SD_WACCM_TSLT_GEOS5
- History sampling observation tracks
  - Satellite track
  - Aircraft flight path
- Local time averaging history output (C Bardeen)
- Time dependent 3D chemistry sources
  - Aircraft emissions
  - Potentially be used for SPE/GCR NOx sources

Specified Dynamics

- Offline meteorology dynamical data (T,U,V) used to drive the model
  - No longer need to have number of levels in met data files match the number of model levels
    - Uses bottom portion of meteorology data
  - Added CAM configure option -offline_dyn
  - 1.9x2.5 GEOS5 (2004-2010)
  - 1.9x2.5 MERRA (1979-2010)

Specified Dynamics for WACCM

- Don't have meteorology data for WACCM upper region
- Specify prescribed T,U,V in lower region
- Blind free running predicted winds in upper region with the prescribed winds in lower region with a vertical weighting function



User interface

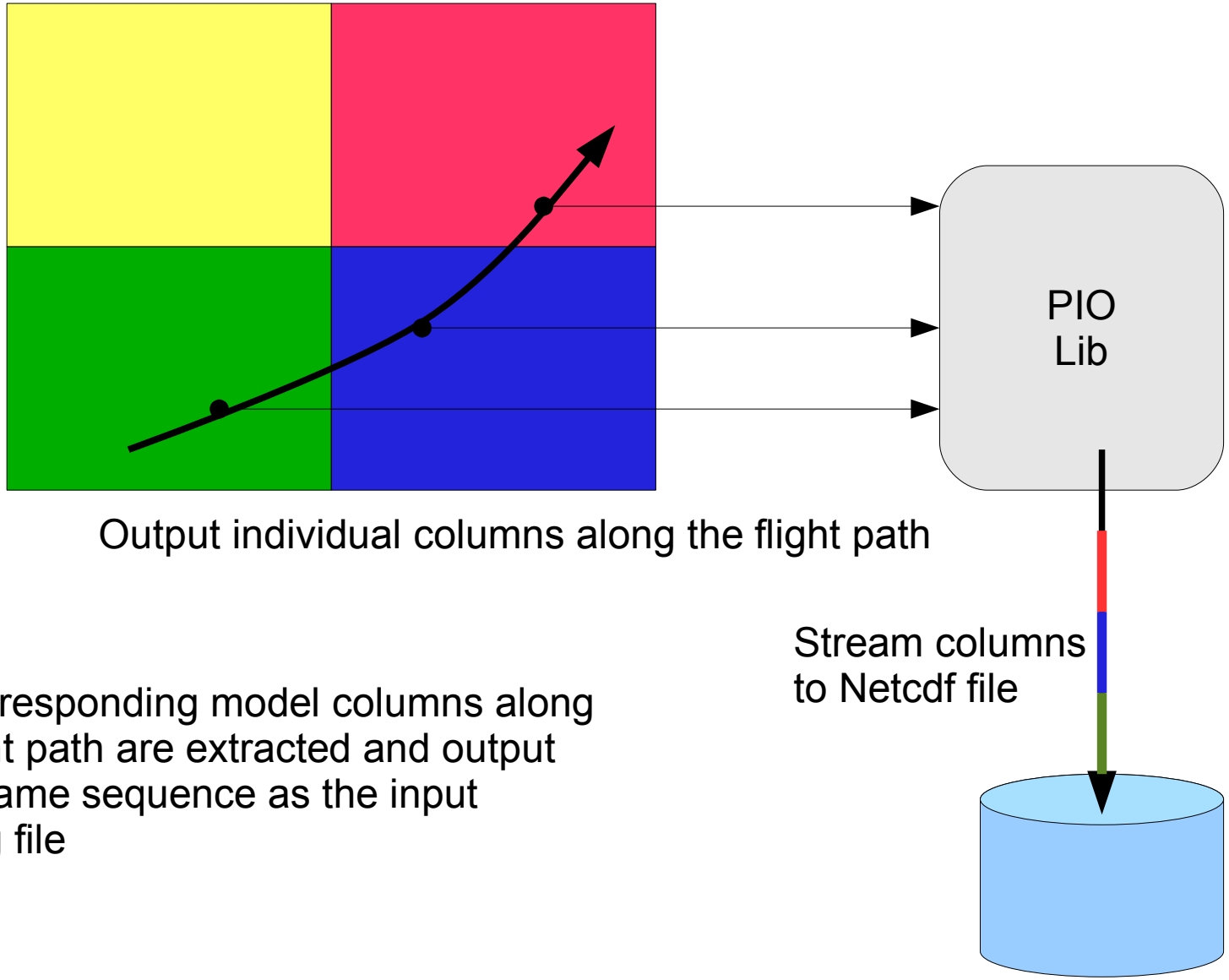
- CAM configure -offline_dyn -nlev nn
- Namelist variables

```
met_data_file      = '2005/GEOS5.1_19x2_20050101.nc'
met_data_path      = '$INPUTDATA/atm/cam/met/GEOS5'
met_filenames_list = '$INPUTDATA/atm/cam/met/GEOS5/filenames_list.txt'
met_fix_mass       = .true.
met_max_rlx        = 0.1 (0 to 1)
met_rlx_bot        = 50. (alt in km)
met_rlx_top        = 60. (alt in km)
bnd_topo  = '.../met/GEOS5/USGS-gtopo30_1.9x2.5_phys_geos5_c100929.nc'
```

History Column Sampling

- Sample the CAM history buffer along a specified path (satellite or aircraft)
  - All variables that have been added to cam history via addfld can be sampled
- Output stream of the nearest model columns to the specified sequence of coordinates
- Specify the sequence of coordinates via a tracking file
- Each model time step:
  - Read the tracking coordinates that are +/- half time step of the current model time
  - Find nearest model columns to those coordinates
  - Output those columns via PIO lib

Horizontal grid distributed across MPI tasks



Output individual columns along the flight path

The corresponding model columns along the flight path are extracted and output in the same sequence as the input tracking file

Namelist options:

```
sathist_fincl          = 'PS','Q','T','U','V','O3',...
sathist_track_infile   = '.../satellite_profilelist.nc'
sathist_mfilt          = 500000
sathist_hfilename_spec = '%c.cam2.sat.%y-%m-%d-%s.nc'
```

Tracking file:

```
Required variables:

    int time(profs) ;
        time:long_name = "time of day" ;
        time:units = "s" ;
    int date(profs) ;
        date:long_name = "date[yyyymmdd]" ;
        date:units = "yyyymmdd" ;
    float lat(profs) ;
        lat:long_name = "latitude" ;
        lat:units = "degrees" ;
    float lon(profs) ;
        lon:long_name = "longitude" ;
        lon:units = "degrees" ;
```

Get copied to the output file if present

```
Optional variables:
    int instr_num(profs) ;
        instr_num:long_name = "MLS=1, ACE-FTS=2, HIRDLS=3, SABER=4" ;
        instr_num:units = "1" ;
    float instr_sza(profs) ;
        instr_sza:long_name = "solar zenith angle" ;
        instr_sza:units = "degrees" ;
    float local_time(profs) ;
        local_time:long_name = "local solar time" ;
        local_time:units = "1 = sunrise, -1 = sunset, 0 = N/A" ;
    short occ_type(profs) ;
        occ_type:long_name = "type of occultation" ;
    int orbit_num(profs) ;
        orbit_num:long_name = "orbit number" ;
        orbit_num:units = "1" ;
    int prof_num(profs) ;
        prof_num:long_name = "profile number" ;
        prof_num:units = "1" ;
```

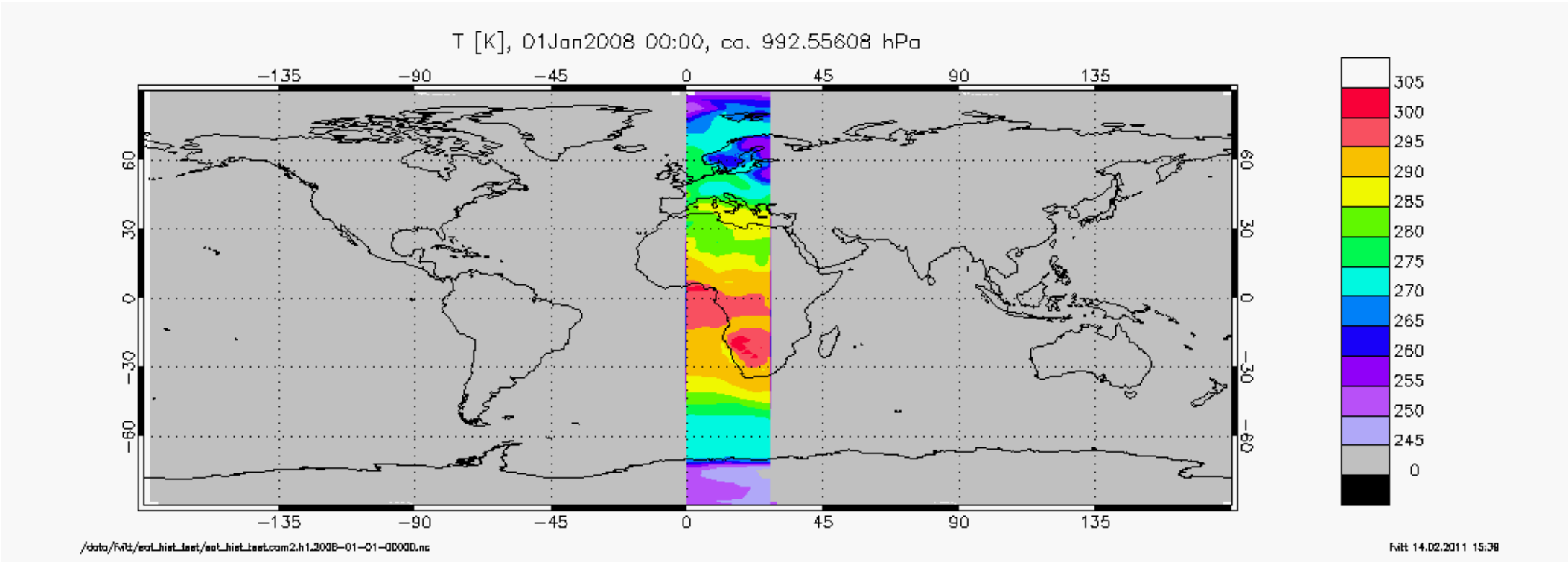
The output file is unstructured with column sequenced in the same order as the input tracking file:

```
dimensions:
        lev = 26 ;
        ilev = 27 ;
        ncol = UNLIMITED ; // (100018 currently)
variables:
        double lat(ncol) ;
                lat:long_name = "latitude" ;
                lat:units = "degrees_north" ;
        double lon(ncol) ;
                lon:long_name = "longitude" ;
                lon:units = "degrees_east" ;
        double lev(lev) ;
                lev:long_name = "hybrid level at midpoints (1000*(A+B))" ;
                lev:units = "level" ;
                lev:positive = "down" ;
                lev:standard_name = "atmosphere_hybrid_sigma_pressure_coordinate" ;
                lev:formula_terms = "a: hyam b: hybm p0: P0 ps: PS" ;
        double time(ncol) ;
                time:long_name = "time" ;
                time:units = "days since 2005-01-01 00:00:00" ;
                time:calendar = "noleap" ;
        int date(ncol) ;
                date:long_name = "current date (YYYYMMDD)" ;
        int datesec(ncol) ;
                datesec:long_name = "current seconds of current date" ;
        float Q(ncol, lev) ;
                Q:units = "kg/kg" ;
                Q:long_name = "Specific humidity" ;
        ....
```

Local time averaging

Namelist settings:

```
avgflag_pertape = 'A','L'
fincl2          = 'Q','T','PS'
lcltod_start    = 0,0
lcltod_stop     = 0,7200
```



Time-Dependent Aircraft Emissions

In the chemistry preprocessor input file declare which species have forcings from external data sets:

```
Ext Forcing
   NO<-dataset, NO2<-dataset, CO<-dataset, Op, O2p, Np, N2p, N2D, N, e, OH
End Ext Forcing
```

These source/loss terms are then used by the chemistry solvers

We can now specify the aircraft emissions as time-dependent 3D input dataset:

```
ext_frc_specifier = 'NO2 -> .../IPCC_emissions_RCP45_aircraft_NO2_2000-2100_1.9x2.5.nc'
ext_frc_type      = 'SERIAL'
```

END