Data Assimilation of MLS/Aura radiance data

1. Preface

After the delivery of Version 0.05 (June 30, 2010) of the MLS Callable Forward Model (CFM) by the MLS/JPL project [1], we started to interface the MLS L1B radiance data (geolocation files L1BOA and Ghz channel L1BRADG data /O3, T, and H2O/) and CFM-software into the GSI observer for GEOS-5.5.1. This report presents a summary of modifications in GSI-modules and description of new modules that read the MLS radiance data in the GEOS-5 system (section 2). The data format for 3Dvar analysis with 6-hour time window is described in section 3. Section 4 reports the location of modules and radiance data on NCCS Linux-Discover cluster at NASA/GSFC.

2. GSI modules associated with the L1B MLS radiance data in GEOS-5.5.1

MLS L1B limb radiance data represent the new observational data class and corresponding modifications of GSI subroutines and introduction of new modules have been performed for GEOS-5.5.1. Below the brief summary of modified subroutines and new modules is documented.

2.1   mls_radinfo.f90 (new)

```
module mlsrad_info
!$$$ module documentation block
!          .      .     .                                       .
! module:   mlsrad_info
!   prgmmr: V Yudin     org: GMAO               date: 2010-07-01
!
! abstract:  This module contains variables and routines related
!            to the assimilation of MLS radiances (T, H2O, O3 channels)
!
!
! Subroutines Included:
!   subroutine init_mlsrad       - set ozone related variables to 
defaults
!   subroutine mlsradinfo_read   - read in ozone info
!
! Functions Included:
!
! Variable Definitions:
!   def diag_mlsrad     - logical to turn off or on the diagnostic ! !
!                         ozone file (true=on)
!   def jpch_mlsrad        - number of (levels+1) !number of satellites
!   def mype_mlsrad        - task id for writing out rad-diagnostics
!   def pob_mlsrad         - pressure level of observation (hPa)
!   def gross_mlsrad       - gross error limit
!   def error_mlsrad       - observation error
!   def nusis_mlsrad       - sensor/intrument/satellite id (1=MLS-Aura)
!   def nulev_mlsrad        - integer level of ozone observation
!   def iuse_mlsrad        - integer flag to control usage of ozone !
```

```
!
! attributes:
!   language: f90
!   machine:  Linux/Discover
!
!$$$ end documentation block
!
! contains
!          subroutine mlsradinfo_read(mype)
!          subroutine init_mlsrad
!          subroutine mlsradinfo_read(mype)
!
! reads     mlsradinfo.txt
! sens/instr/sat lev  use pressure gross   obs    b_oz  pg_oz
!                                  error  error variational qc
! mlsrd_toh        1    1    0.240  1.000  1.000 10.000  0.000

! ====================================================================
```

2.2 read_mlsrad.f90 (new)

```
subroutine read_mlsrad(nread,ndata,nodata,jsatid,infile,gstime,&
           lunout, obstype,twind,sis,ithin,rmesh)
!
!
!
!$$$  subprogram documentation block
!                .     .    .                                    .
! subprogram:    read_mlsrad                   read mls radaince data
!
!=============================================================
! abstract:  This routine reads MLS-L1B radiance data   Both layer
!            and total column values are read in.  The routine has
!            ability to read both text(formatted) and f77-unformatted
!            6-hr chunk of MLS-orbits (L1BOA -Geolocation file)
! L1BRADG -GHZ-radiances (temperature, ozone and water vapour channels)
!
! Quality Control:   as described in the MLS_documentation
!=============================================================
!
!
! program history log:
!                      2010-07-01  valery yudin SAIC/GMAO
!
!
!   input argument list:
!     obstype  - observation type to process
!     jsatid   - satellite id to read
!     infile   - unit from which to read ozone data
!     gstime   - analysis time in minutes from reference date
!     lunout   - unit to which to write data for further processing
!     obstype  - observation type to process
!     twind    - input group time window (hours)
!     sis      - satellite/instrument/sensor indicator
!     ithin    - flag to thin data
!     rmesh    - thinning mesh size (km)
```

```
!
!   output argument list:
!     nread    - number of sbuv/omi ozone observations read
!     ndata    - number of sbuv/omi ozone profiles retained for further
!                processing
!     nodata   - number of sbuv/omi ozone observations retained for !
!                 further processing
!
! remarks:
!                 L1B-MLS radiance data => h5-format
!                 two types of files GEO-locations + RAD-values
!                 Main GHz channels:  TP and O3 and H2O
! attributes:
!                       language: f90
! machine:  Linux/Discover NCCS
!
```

2.5 setup_mlsrad.f90 (new)

```
subroutine setup_mlsrad(lunin,&
              mype,aivals,stats,nchanl,nreal,nobs,&
              obstype,isis,is,rad_diagsave,init_pass,last_pass)
```

TBD: Needs extra-work including some fixes in the CFM-MLS JPL software for Jacobians and multiple-calls of MLS forward models for 6-hour chunk of the MLS radiance data.

2.6 obsmod.F90 (modified)

```
module obsmod
!$$$ module documentation block
!           .      .    .                                       .
! module:   obsmod
!   prgmmr: derber      org: np23                date: 2003-09-25
!
! abstract: This module contains variables and arrays pertinent for
!           observational data.
!   2010-06-28 yudin/GMAO - added MLS radiances(mlsrad)observation type

!
!mlsvay
!
  type mlsrad_ob_type
     sequence
     type(mlsrad_ob_type),pointer :: llpoint => NULL()
     type(mlsrad_diags), dimension(:), pointer :: diags => NULL()
     real(r_kind),dimension(:),pointer :: res => NULL()
                                      !  mlsrad residual
     real(r_kind),dimension(:),pointer :: err2 => NULL()
                                      !  mlsrad error squared
     real(r_kind),dimension(:),pointer :: raterr2 => NULL()
                  !  square of ratio of final obs error
                                      !  to original obs error
     real(r_kind)    :: time          !  observation time in sec
     real(r_kind),dimension(:,:),pointer :: wij => NULL()
```

```
                                  !  horizontal interpolation weights
     real(r_kind),dimension(:),pointer :: prs => NULL()
                                      !  pressure levels
     integer(i_kind),dimension(:),pointer :: ipos  => NULL()
     integer(i_kind) :: nl_mlsrad   ! number of levels for this profile
     integer(i_kind) :: ij(4)         !  horizontal locations
     logical         :: luse   !  flag indicating if ob is used in pen.

     integer(i_kind) :: idv,iob  ! device id and obs index for sorting
  end type mlsrad_ob_type

  type mlsrad_ob_head
     integer(i_kind):: n_alloc=0
     type(mlsrad_ob_type),pointer :: head => NULL()
  end type mlsrad_ob_head
!
!mlsvay
!
```

2.7 read_obs.F90 (modified)

```
! mlsvay
!+++++++++++++++++++++++++++++++++++++++++++++++++++++
!         Process MLS-L1B radiance data
!+++++++++++++++++++++++++++++++++++++++++++++++++++++
      else if(obstype == 'limb')then
          print *, mype, mype_root
          print *, 'mlsrad before call file:', infile
          print *
call gsi_inquire(lenbytes,lexist, infile,myprint*, ditype(i), infile,
'read_obs.F90 ', mype, ' MLS mpi-proc/exist-file ', lexist
!
             if (lexist) then
          print *, '  read_mlsrad '
call read_mlsrad(nread,npuse,nouse,platid,infile,gstime,&
       lunout,obstype,twind,sis,ithin,rmesh)
!           print *, 'mlsrad_to3h2o after  call '
! mlsvay
!+++++++++++++++++++++++++++++++++++++++++++++++++++++
!         Finish with MLS-L1B radiance data
!+++++++++++++++++++++++++++++++++++++++++++++++++++++
```

2.8 setup_rhsall.F90 (modified)

```
subroutine setuprhsall(ndata,mype,init_pass,last_pass)

!$$$  subprogram documentation block
!                .      .    .                                       .
! subprogram:  setuprhsall   sets up rhs of oi
!   prgmmr: derber           org: np23                date: 2003-05-22
!
! abstract: This routine sets up the right hand side (rhs) of the
!           analysis equation.  Functions performed in this routine
```

```
!include:
!a) calculate increments between current solutions and obs,
!b) generate statistical summaries of quality control and innovations,
!c) generate diagnostic files (optional), and
!d) prepare/save information for use in inner minimization loop

!2010-07-01 vay: yudin- start adding call for MLS radiance data.

       if(ditype(is) == 'limb)then

           call setup_mlsrad(lunin,&
              mype,aivals,stats,nchanl,nreal,nobs,&
              obstype,isis,is,rad_diagsave,init_pass,last_pass)
       endif
```

3. Formats of MLS radiance data for GSI 3DVar data analysis (6-hour window)

The size of original L1BOA and L1BRADG data is rather large (one day of data exceeds ~1.5 Gb-storage). These data contain extra information (other minor species and instrument-related info). Therefore for the computationally manageable data storage and manipulation, the original data have been rewritten and shortened for the planned data analysis of ozone, temperature, and water vapor. The Fortran read-statements and allocation of corresponding arrays (see portion of read_mlsrad.f90 below) explain the data format for combined L1BOA and L1BRADG data for 3Dvar applications with the standard 6-hour window. The original L1B radiance data and the "baseline" radiance offset (correction) were also combined following MLS CFM recommendations. The "negative" precision radiance data (according to the MLS data quality control statements) were excluded from the joint data L1BOA and L1BRADG data files.

The read-statements in *read_mlsrad.f90*

```
!
! Read MLS-radiance and geolocation data
!
    print *, ' inside   read_mlsrad.f90 '
     print *, 'before  mockup_in_gsi '
!
    call mockup_in_gsi
!
     print *, 'after  mockup_in_gsi '
    open(unit=lunin, file=Infile, form='formatted')
    print *, ' infile ', infile, ' was opened '
!
    read(lunin, '(i4, i4, i4, i4)') Npix, M_nch, M_nghz, M_nscan
    print*, Npix, M_nch, M_nghz, M_nscan
!
        Allocate(MLSRAD_out(Max_index, Npix))
```

```
!
    do kpix=1, Npix
     read(lunin, '(i4, i4, i4, i4, i4, f6.3, 6E10.3)')  &
            MLS_yyyy, MLS_MM, MLS_DD, MLS_Hr, MLS_Min, Thr, &
             A_lst, A_sza, A_lon, A_lat,  A_phi, A_losg
!
     read(lunin, '(120E10.3)') RadT
     read(lunin, '(120E10.3)') RadTer
     read(lunin, '(120E10.3)') RadOz
     read(lunin, '(120E10.3)') RadOzer
     read(lunin, '(120E10.3)') RadH2o
     read(lunin, '(120E10.3)') RadH2oer
!
    print *, kpix,  MLS_yyyy, MLS_MM, MLS_DD, MLS_Hr, MLS_Min, Thr
    enddo
   close(lunin)
    return
!
!    Convert observation time to relative time
     idate5(1) = MLS_yyyy             !year
     idate5(2) = MLS_MM               !month
     idate5(3) = MLS_dd               !day
     idate5(4) = Mls_hr               !hour
     idate5(5) = Mls_min              !minute
!
! check DAS  time-window
!
     call w3fs21(idate5,nmind)
     t4dv=real((nmind-iwinbgn),r_kind)*r60inv
     if (l4dvar) then
       if(t4dv<zero .OR. t4dv>winlen) goto 110
     else
       sstime=real(nmind,r_kind)
       tdiff=(sstime-gstime)*r60inv
!
       if(abs(tdiff) > twind) goto 110
     end if
!
!
           slons0=A_lon
           slats0=A_lat
           if(slons0< zero) slons0=slons0+r360
           if(slons0>=r360) slons0=slons0-r360
           dlat_earth = slats0 * deg2rad
           dlon_earth = slons0 * deg2rad
!
           if(regional)then
              call tll2xy(dlon_earth,dlat_earth,dlon,dlat,outside)
!              if(outside) cycle
           else
              dlat = dlat_earth
              dlon = dlon_earth
              call grdcrd(dlat,1,rlats,nlat,1)
              call grdcrd(dlon,1,rlons,nlon,1)
           endif
!
!data counts
```

```
!
           ndata=ndata+1
           nodata = nodata + 1
!
           mlsrad_out(1,ndata)=rsat
           mlsrad_out(2,ndata)=t4dv
           mlsrad_out(3,ndata)=dlon
           mlsrad_out(4,ndata)=dlat
           mlsrad_out(5,ndata)=dlon_earth*rad2deg
           mlsrad_out(6,ndata)=dlat_earth*rad2deg
           mlsrad_out(7,ndata)=A_sza                ! SZA (degrees)
           mlsrad_out(8,ndata)=A_lst                ! LST (hours)
           mlsrad_out(9,ndata)=A_phi                ! geod-lat (degrees
           mlsrad_out(10,ndata)=A_losg              ! geod-lat (degrees
           mlsrad_out(11,ndata)=A_galt              ! geod-alt (m)
            mlsrad_out(11,ndata)=A_lon
            mlsrad_out(11,ndata)=A_lat
!~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
!
! radiances: 25-channels 125-GHZ  6-types (values and errors)
!
! 2 options
!            treat 25x125x6 =18750 elements +11,  as MLS_max index
!            Npix= as horizontal points = ndata
!            treat 11 + 6 elements =17 as as MLS_maxindex
!            25x125x6xNpix as individual data points
!
! MLS_CFM: used 25x125= 3125 elements for each pixel and band
!
!    Current files contain 3-bands (T, O3, H2O)
!   (RadT, RadTer,RadOz, RadOzer, RadH2o, RadH2oer)
!~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
!
           mlsrad_out(itp:itp+Dim_mls, ndata)= Radt
           mlsrad_out(itper:itper+Dim_mls, ndata)= Radter
           mlsrad_out(ioz:ioz+Dim_mls, ndata)= Radoz
           mlsrad_out(iozer:iozer+Dim_mls, ndata)= Radozer
           mlsrad_out(ih2:ih2+Dim_mls, ndata)= Radh2o
           mlsrad_out(ih2er:ih2er+Dim_mls, ndata)= Radh2oer
           enddo

           close(lunin)
```

Figures 1-5 display plots of MLS radiance data for O_3 and T sensitive channels for Jan 21 and 25 of 2006 at two tangent heights (~18 km and ~ 45 km). These data and corresponding errors are accepted by GSI-GMAO software.

There are other configuration changes in "run"-directory needed to 'turn on' a new data observational class in the GSI analysis. The location of the MLS radiance data files (Section 4) was added to the *obsys*.rc* file with corresponding changes in *gsi.rc.tmpl* (ndat=66+1). A new entry was added to the *GSI_Gridcomp.rc* file so that the MLS radiance data files would be linked into the analysis "work" subdirectory.

4. Location of modules and radiance data at NCCS Linux-Discover platform.

4.1 The location of MLS radiance data for GSI-3Dvar (6-hr chunk)
/discover/nobackup/vyudin/arc_data/mls_rad/Y2006/M01

4.2 Original h5-formatted data (samples of L1BOA and L1BRADG files):
/archive/u/vyudin/rad_data/Y2006/M01

4.3 Set of initial conditions for b72-resolution of GEOS-5:
/discover/nobackup/vyudin/RS_B_060101

4.4 Run-directory*: /discover/nobackup/vyudin/mlsr_b72* (GEOS-5.5.1)

4.5 Source code (new and modified modules) and documentation notes:
/home/vyudin/MLS_FORT

4.6 Debugging of new modules in the GSIsa (stand-alone version):
/discover/nobackup/vyudin/Ana

The selected testbed period is January 2006. It is characterized by a long-term stratospheric warming event from January 15 through February 2006. The observed O_3 and H_2O profiles deviate significantly from climatological a priori for O_3 and H_2O in the L2 MLS retrievals. The initial guesses for O_3 and H_2O produced by GEOS-5 will be more representative for the perturbed stratospheric flows in the winter NH during the analyzed period.

Further debugging and testing

Two additional updates were provided by the MLS/JPL team in September 2010

1. A minor bug in the CFM was identified and fixed. Subsequently, we have verified that the Forward Model can now be called from within the GMAO code multiple times in a loop. This will add more flexibility to our implementation of the MLS radiance observation operator.
2. The capability to output Jacobian matrices was added to the CFM. This output will be essential for a full assimilation of the MLS radiances

In collaboration with the MLS team we have devised an efficient algorithm for calculating the Observation minus Forecast (O-F) statistics to be implemented in the subroutine *setup_mlsrad* (see Section 2.5). In particular, the MLS input files can be read only once per each 6 hour analysis window rather than multiple times as previously thought. As a simple test we successfully ran the CFM for two sets of temperature, ozone, and water vapor profiles at two separate observation times and geolocations.

Reference

[1] *Nguyen H.H, P.A. Wagner, and N.J. Livesey*, Aura Microwave Limb Sounder (MLS) Callable Forward Model, v.0.0.5, June 30, 2010

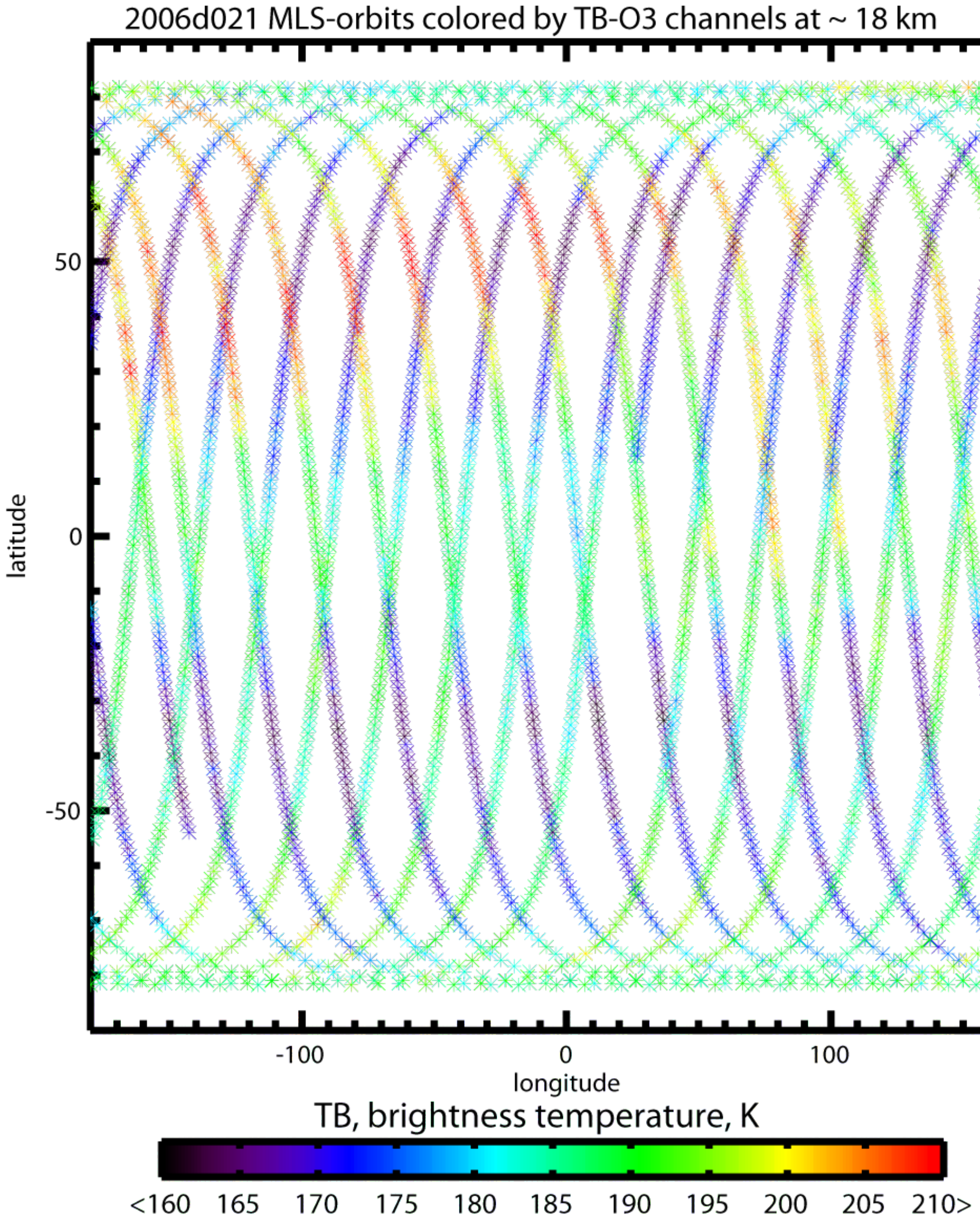

Figure 1. 21-01-2006, typical MLS brightness temperatures of ozone-sensitive channels at ~18 km.

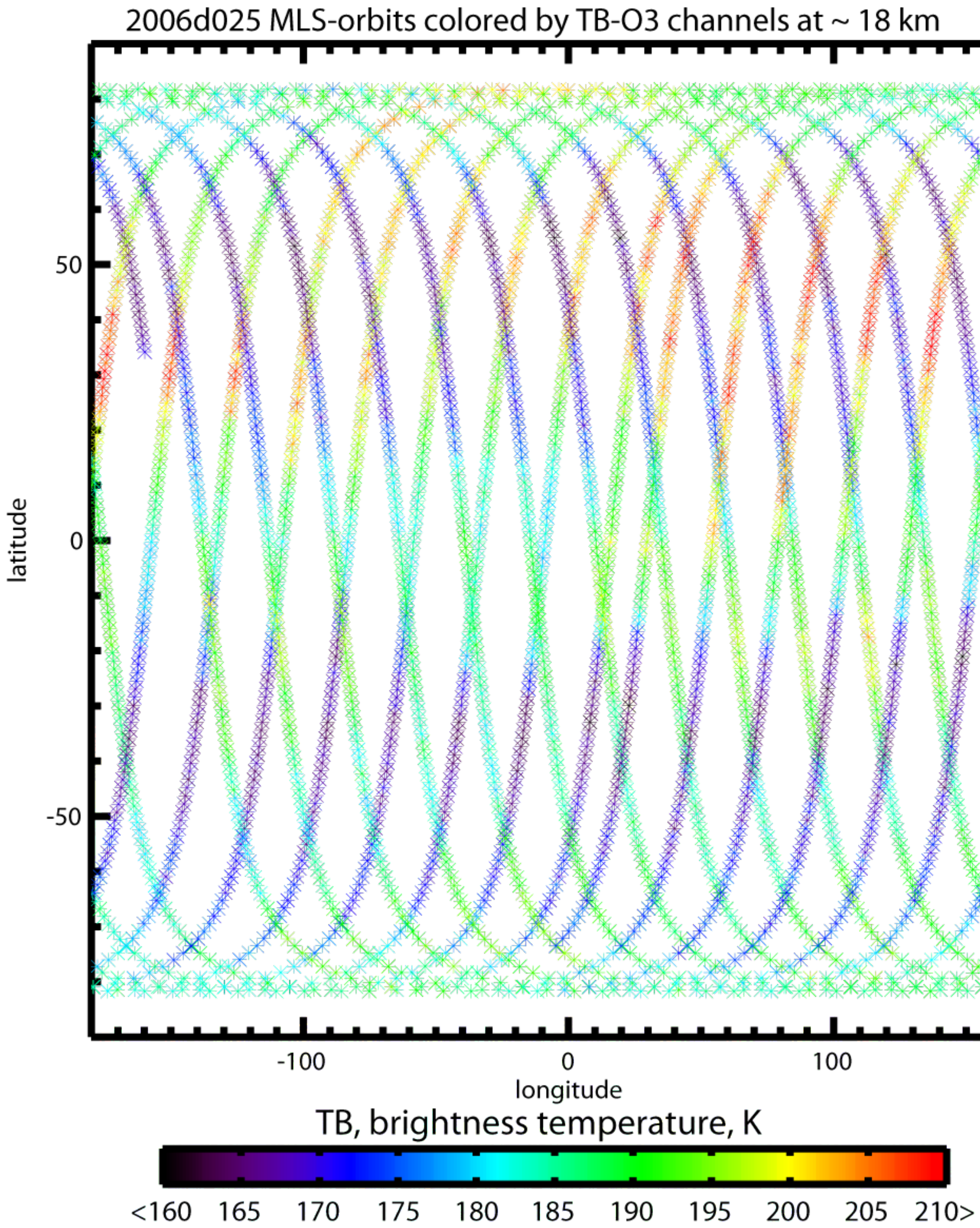

Figure 2. 25-01-2006, typical MLS brightness temperatures of ozone-sensitive channels at the tangent height ~ 18 km.

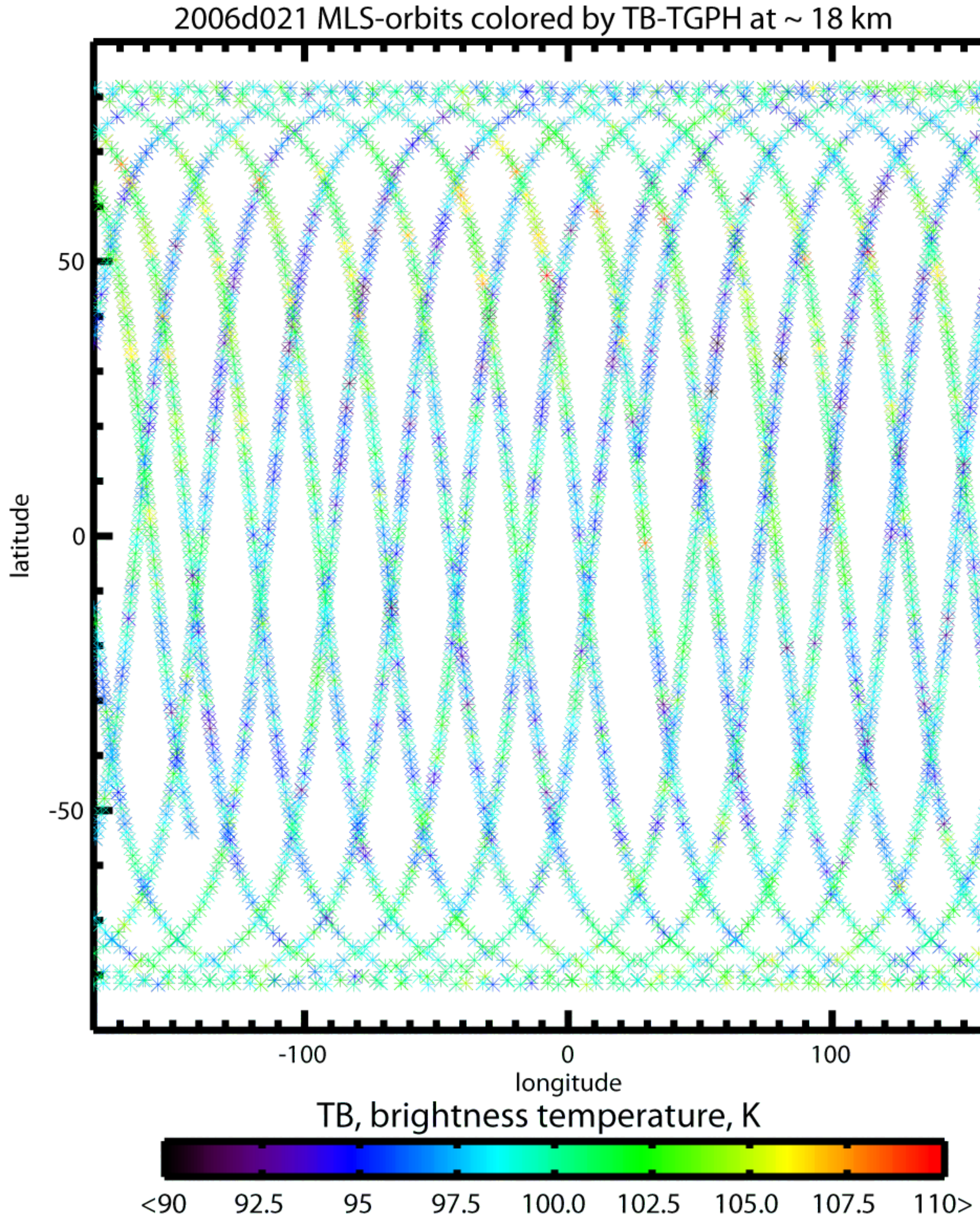

Figure 3. 21-01-2006, typical MLS brightness temperatures of temperature-sensitive channels at the tangent height ~ 18 km.

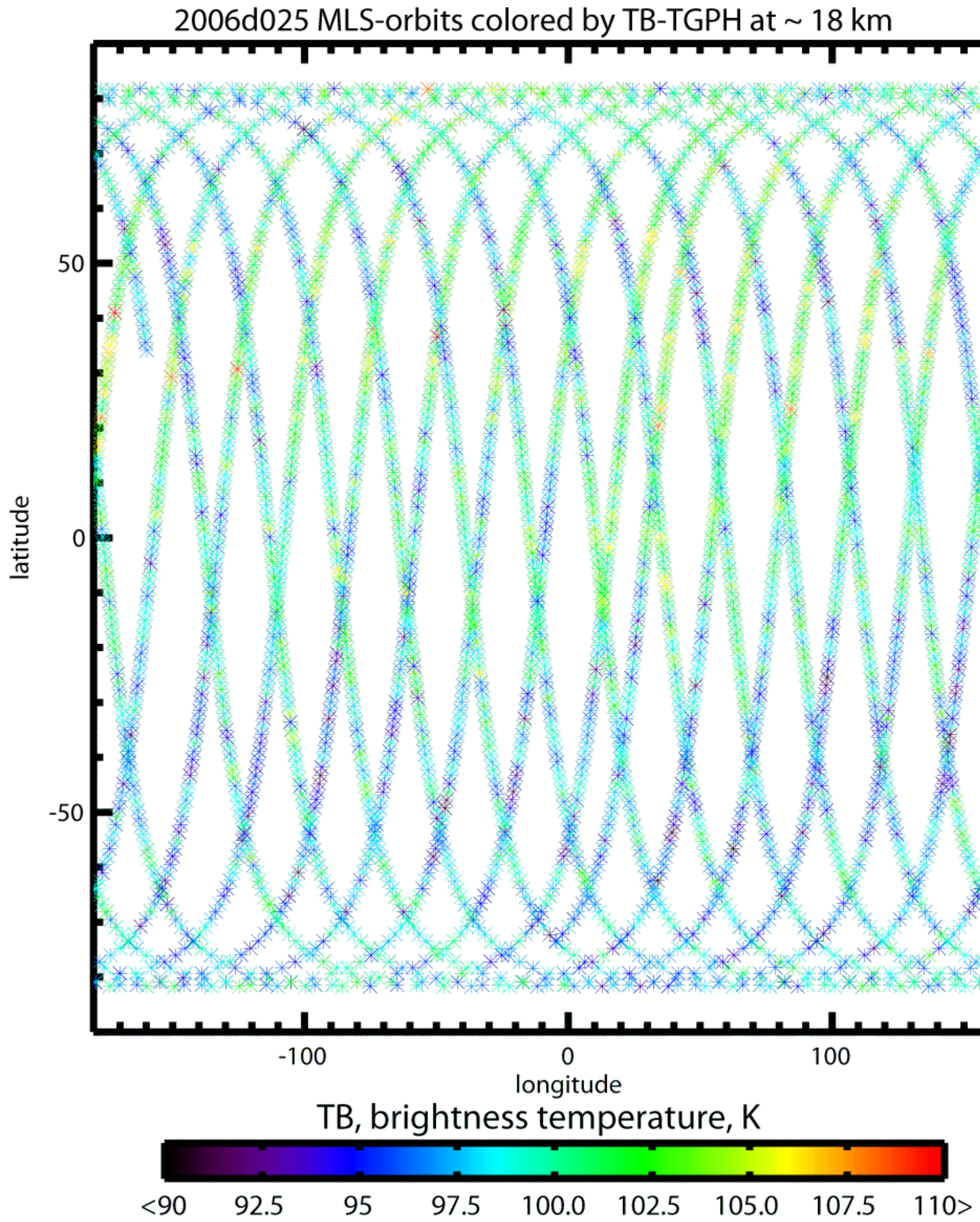

Figure 4. 25-01-2006, typical MLS brightness temperatures of temperature-sensitive channels at the tangent height ~ 18 km.

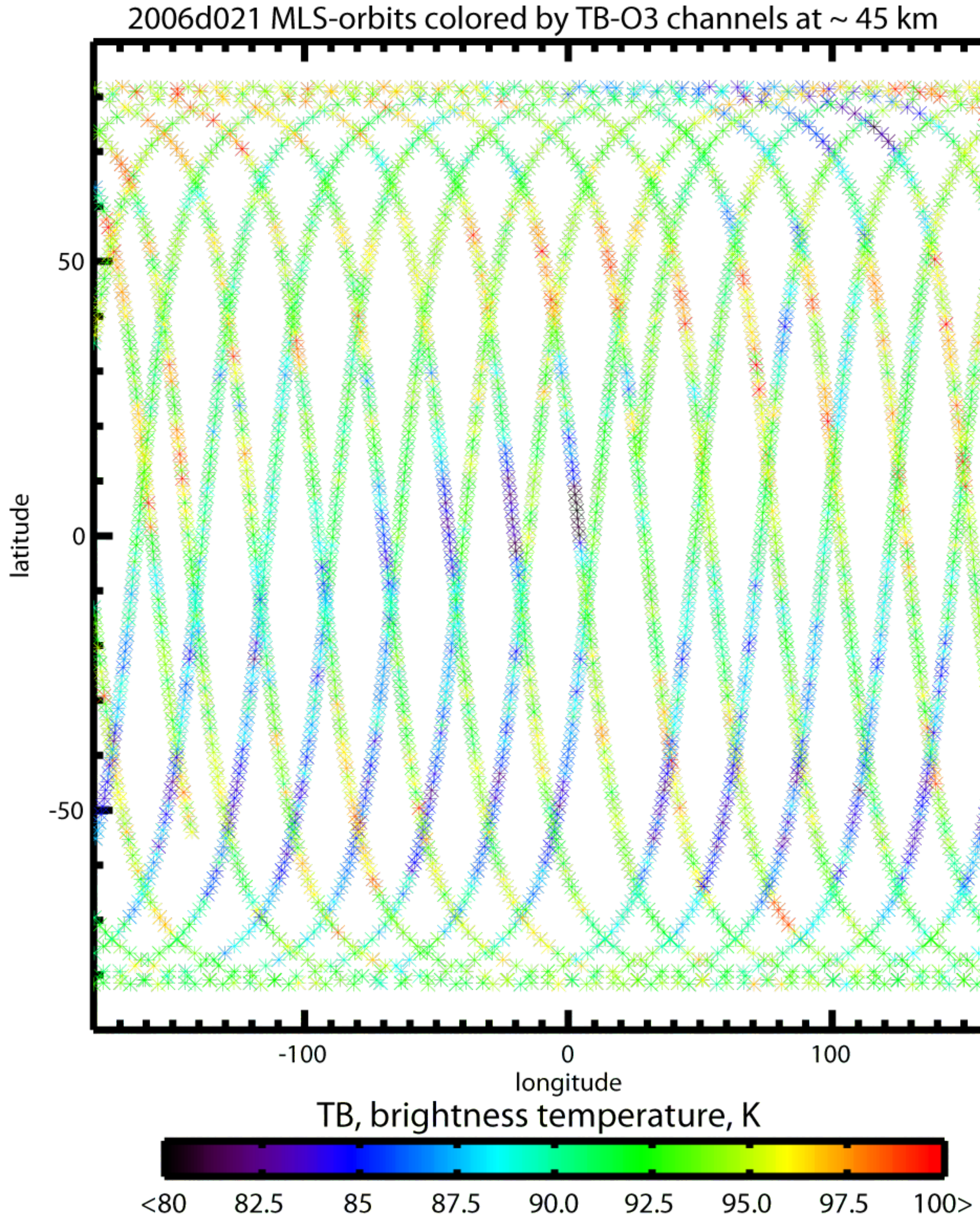

Figure 5. 21-01-2006, typical MLS brightness temperatures of ozone-sensitive channels at the tangent height ~ 45 km.