Example use of TES data

Susan Kulawik, 2009

© 2009 California Institute of Technology
Government sponsorship acknowledged

The following shows how to find, browse plots of, download, plot, and compare TES data. We start with a sonde to match, locate the closest TES matching data, apply the TES observation operator to the sonde to see if the sonde and TES results are consistent, and plot the results. All examples use the IDL programming language.

1. Data you want to find

Say you have a temperature sonde you want to compare to TES result. This comparison analysis would also work for comparisons to model data. Here is the sonde location:

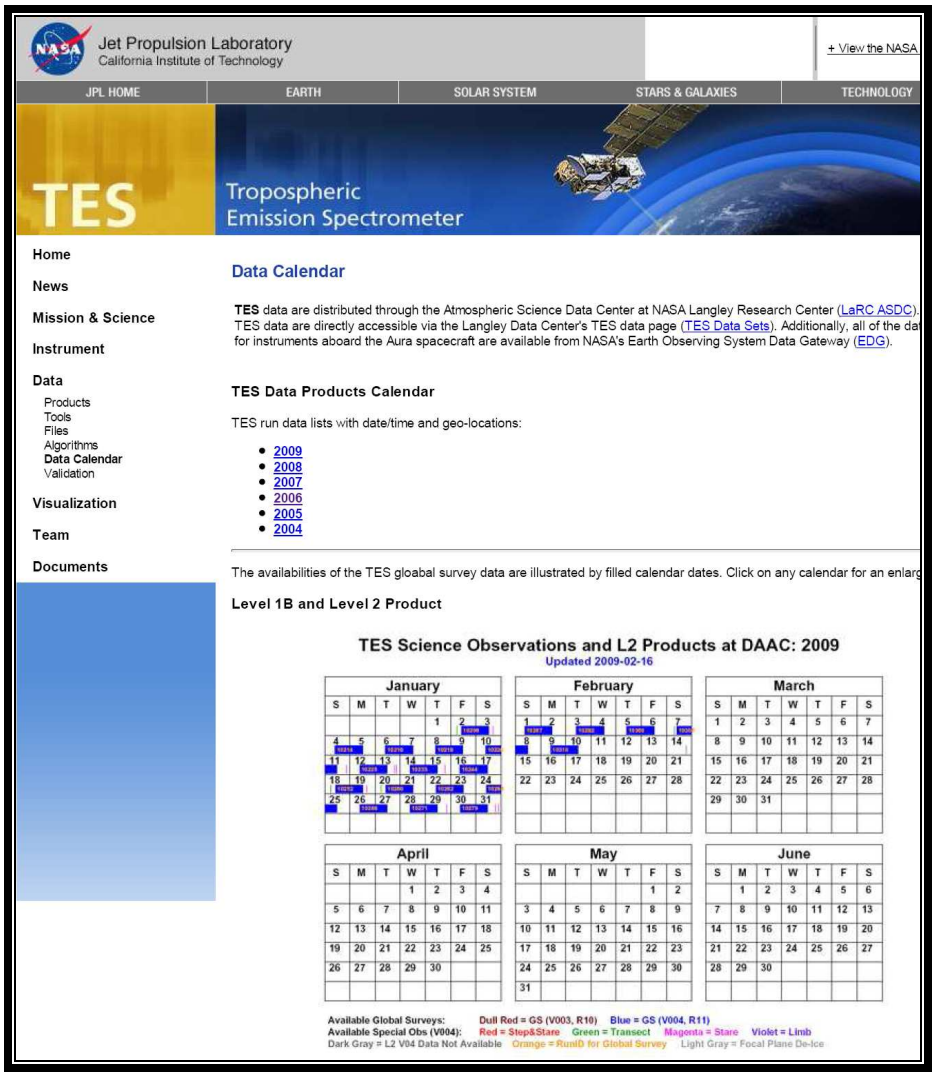
Latitude 21.9800
Longitude 200.650
Date/Time 2006-02-13 12:00:00 UTC

Here are the raw sonde values:

```
sonde=[ 296.150,     295.350,     288.750,     288.350,     284.250,
281.750,     280.550,     280.350,     278.350,     277.950,     276.750,
266.550,     263.450,     262.750,     258.550,     249.350,     248.550,
247.150,     235.650,     225.950,     215.250,     204.350,     203.950,
203.750,     207.650,     211.450,     210.550,     208.750,     205.850,
205.350,     203.550,     208.550,     209.950,     207.750,     210.550,
209.750,     209.950,     212.350,     217.050,     215.650,     214.350,
223.450,     222.350,     223.550,     231.950,     233.950,     234.950,
233.150]

psonde =[1016.00,     1000.00,     925.000,     916.000,     850.000,
817.000,     788.000,     766.000,     731.000,     712.000,     700.000,
593.000,     540.000,     534.000,     500.000,     418.000,     400.000,
380.000,     300.000,     250.000,     200.000,     154.000,     150.000,
142.000,     122.000,     119.000,     100.000,     95.0000,     72.0000,
70.0000,     66.0000,     50.0000,     45.0000,     38.0000,     34.0000,
30.0000,     27.0000,     26.0000,     21.0000,     20.0000,     17.0000,
12.0000,     10.0000,     9.00000,     7.00000,     6.00000,     5.00000,
4.00000]
```

2. Checking what TES data is available

http://tes.jpl.nasa.gov/data/datacalendar/

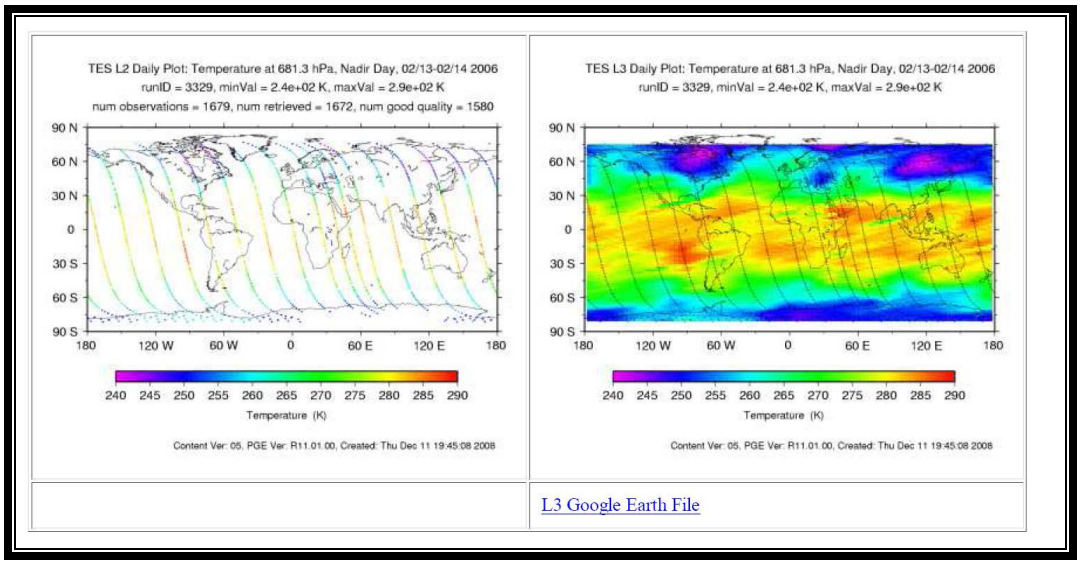


Click on 2006, and see one nadir option:

3329 Global Survey - 16 orbits 2006-02-13T09:55:26 2006-02-14T12:15:25

3. Quick look plots

Go to http://tes.jpl.nasa.gov/visualization/SCIENCE_PLOTS/TES_L3_Daily.htm and enter the date and species desired (TATM). The following plots are displayed, where the left shows the results for each target, and the right has interpolation between TES observations:



4. Downloading the TES data

Go to the Langley Atmospheric Sciences Data Center (ASDC) webpage:

http://eosweb.larc.nasa.gov/PRODOCS/tes/table_tes.html

Hit the "Level 2 Global Survey Standard Products" link. Select "TES/Aura L2 Atmospheric Temperatures Nadir" v004 (most recent) data. We are in luck, this has been processed in v004. Download file "TES-Aura_L2-ATM-TEMP-Nadir_r0000003329_F05_05.he5". It only takes 15 minutes!

5. Reading in TES data

TES data is in HDF format, which is a standard format. The TES team has provided a reader for the TES files for the IDL programming language. See here:

http://eosweb.larc.nasa.gov/PRODOCS/tes/tools/read_software.html

Download "TES_L2_ReadSoftware_V5.tar (443 KB)" and unzip (e.g. using 7-zip). There is a Readme that tells how to call each program. For L2, we use the following call:

filename = "TES-Aura_L2-ATM-TEMP-Nadir_r0000003329_F05_05.he5"
read_l2_hdf, filename, species_data, geo_data

6. Quick look at TES data

help, species_data, /st

I've pulled out some of the more important components of species_data and geo_data:

```
IDL> help, species_data, /st
   SPECIES                        FLOAT   Array[67, 3394]
   ALTITUDE                       FLOAT   Array[67, 3394]
   AVERAGECLOUDEFFOPTICALDEPTH    FLOAT   Array[3394]
   AVERAGECLOUDEFFOPTICALDEPTHERROR FLOAT Array[3394]
   AVERAGINGKERNEL                FLOAT   Array[67, 67, 3394]
   CONSTRAINTVECTOR               FLOAT   Array[67, 3394]
   CLOUDTOPPRESSURE               FLOAT   Array[3394]
   DEGREESOFFREEDOMFORSIGNAL      FLOAT   Array[3394]
   O3_CCURVE_QA                   BYTE    Array[3394]
   OBSERVATIONERRORCOVARIANCE     FLOAT   Array[67, 67, 3394]
   PRESSURE                       FLOAT   Array[67, 3394]
   SPECIESRETRIEVALQUALITY        BYTE    Array[3394]
   UTCTIME                        STRING  Array[3394]

IDL> help, geo_data, /st
      LATITUDE                       FLOAT     Array[3394]
      LONGITUDE                      FLOAT     Array[3394]
      SCAN                           INT       Array[3394]
      SEQUENCE                       INT       Array[3394]
      SURFACETYPEFOOTPRINT           BYTE      Array[3394]
      TIME                           DOUBLE    Array[3394]
```

Geo_data.latitude and longitude have target footprint location. You can see that there are 3394 targets in this file. Species_data.species has the profile results. Each profile has 67 pressure levels. Note geo_data.time is the TAI time (starting 1/1/1993).

7. Finding a geolocation match to TES data and checking quality

To find each target's distance from the sonde location, use the IDL routine map_2points:

```
n = N_ELEMENTS(geo_data.latitude)
dist = FLTARR(n)
FOR ii = 0, n-1 DO dist[ii] = map_2points(200.650,21.98,
geo_data.longitude[ii], geo_data.latitude[ii],
/meters)/1000.d0
```

```
> print, min(dist, ind)
263.673
> print, ind
360
> print, species_data.utctime[360]
2006-02-13T12:41:57.946886Z
```

Note that I changed the domain for the sonde longitude to match TES, which ranges from -180 to 180. The closest TES point is 264 km from the sonde. The time matches within 40 minutes as well. This occurs at index 360. Before proceeding, check the quality of the TES result:

```
> print, species_data.SPECIESRETRIEVALQUALITY[360]
1
> print, species_data.O3_CCURVE_QA[360]
157
```

1 = good quality. 157 = fill. In v04 data, O3_CCURVE_QA is only populated for ozone data but it is always best to check both flags and remove any profiles where either have value 0.

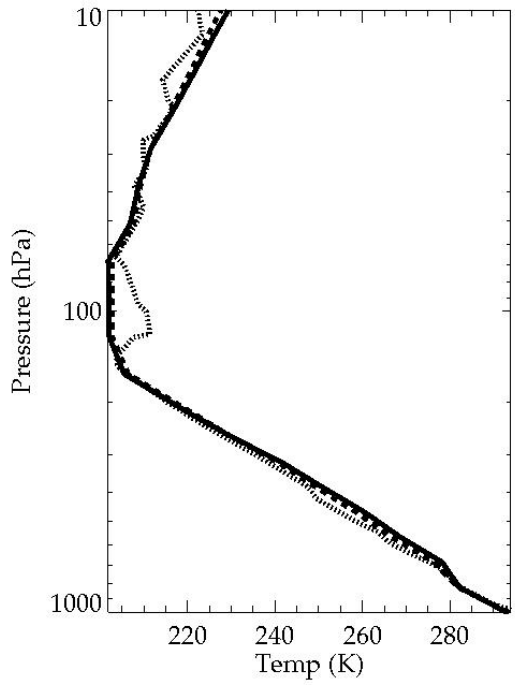
For aligning TES and model data, the procedure is different, and would involve deciding which model gridbox each TES profile belongs in, and deciding what to do when multiple TES profiles go into a single model gridbox.

8. Plotting data

Let's plot the TES result and a priori vector:

```
indp=where(species_data.species[*,360] ge 0)
plot,species_data.species[indp,360],species_data.pressure[in
      dp,360],yrange=[1000,10],/ylog,xtitle='Temp
      (K)',ytit='Pressure (hPa)'
oplot,species_data.constraintvector[indp,360],species_data.p
      ressure[indp,360],linestyle=2
oplot, sonde, psonde,linestyle=1
```

Since the surface pressure is variable, TES data will have fill values (-999) at the bottom levels. Indp finds non-fill indices.



Although they look similar, the initial (dashed) and TES result profile (solid) differ by as much as 2K. The sonde (dotted line) has some differences from both. To properly compare to the sonde, we need to (a) to the TES pressure grid, and (b) account for TES sensitivities.

9. Mapping sonde data to the TES grid

This section is a very short guide on mapping from a fine grid to a coarser grid. One could use a straight interpolation, which would work if there are approximately similar levels for the sonde and TES. However, if the sonde is on a much finer pressure scale, the only information kept from the sonde would be the two points bracketing each TES level. To map from a fine grid to a coarse grid to create final values which take into account the sonde values at all of the intermediate levels, mapping must be done. For a step-by-step guide of this, see http://tes.jpl.nasa.gov/Ming/web.html (or see appendix).

(a) Calculate an interpolation map W which maps from TES pressures to sonde pressures, i.e. from LOG(species_data.pressure[indp2,360]) to LOG(psonde). This procedure is described here: http://tes.jpl.nasa.gov/Ming/get_map.txt . Note indp2 are pressures between sonde minimum and maximum pressure.

(b) Calculate an inverse of this map,W_inv = invert(transpose(W)##W)##transpose(W)

(c) Use this map to map the sonde to TES levels:

```
sonde_tes0 = W_inv ## sonde [note for ozone, use log(sonde)]
```

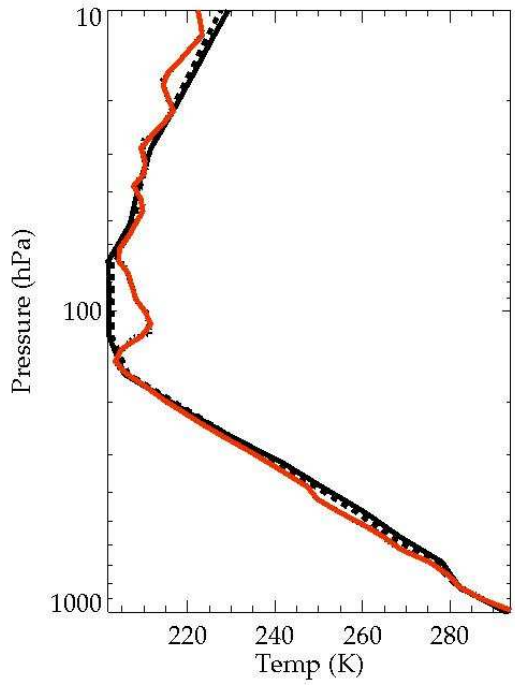
(d) Fill in the pressures outside of the sonde pressure range with the TES constraint vector:

sonde_tes = REFORM(species_`data.constraintvector[*,360])`
`sonde_tes[indp2] =` sonde_tes0

Here is the resultant, sonde mapped to the TES pressure grid (with the constraint vector filling up pressures where the sonde did not cover) :

```
sonde_tes = [    -999.000,    -999.000,     294.670,     295.064,
287.734,     282.215,     279.515,     275.339,     268.817,     264.883,
260.091,     254.636,     249.756,     247.425,     242.826,     238.176,
233.329,     228.190,     223.326,     218.807,     214.289,     210.420,
206.128,     203.453,     204.906,     209.417,     211.635,     210.220,
208.145,     207.260,     206.489,     204.275,     204.423,     206.544,
208.143,     209.772,     209.268,     207.629,     210.140,     210.265,
209.348,     211.727,     214.480,     216.750,     215.610,     214.423,
215.434,     218.261,     220.675,     223.336,     222.967,     222.364,
223.362,     226.540,     232.678,     235.031,     234.359,     243.807,
245.610,     250.123,     256.892,     260.402,     265.666,     259.057,
249.142,     240.075,     227.986]
```

You can add this to the previous plot and check that it is essentially the same as the original sonde (here interpolated result shown in red)



10. Application of the Observation operator

Again, this follows Ming Luo's guide http://tes.jpl.nasa.gov/Ming/web.html or see the TES Data User's Guide, section 7.1.

x_result = x_tes_aPriori + AK_tes##(x_sonde - x_tes_aPriori)

x_result +- the observation error (found in the TES standard product) simulates what TES would observe IF **x_sonde** were the atmospheric true state. So, if **x_result** and **x_tes** are within the observation error, they are considered to be consistent with each other. Note: For all other species other than temperature, the above should be done in log, e.g.:

x_result = EXP(ALOG(x_tes_aPriori) + AK_tes##(ALOG(x_sonde) – ALOG(x_tes_aPriori)))

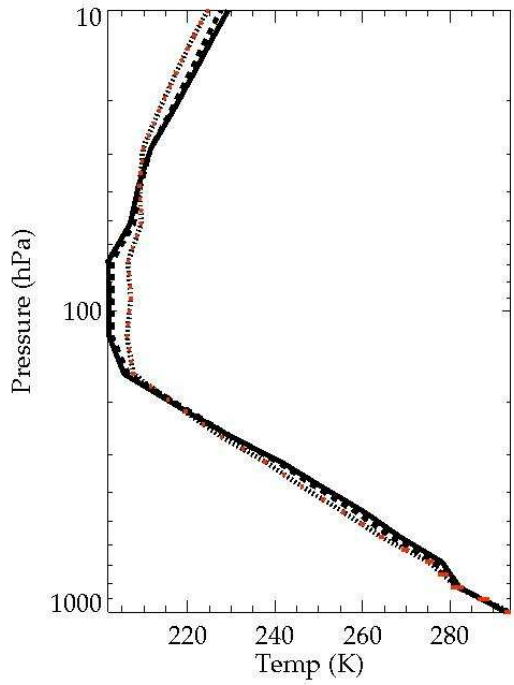
; get non-fill data:

```
indp=where(species_data.species[*,360] ge 0)
pressure = species_data.pressure[indp,360]
AK_TES = REFORM(species_data.AVERAGINGKERNEL[indp,indp,360])
```

```
x_tes_aPriori = species_data.CONSTRAINTVECTOR[indp,360]
x_tes = species_data.species[indp,360]
x_sonde = Sonde_tes[indp] ; pull out non-fill sonde data

; application of the observation operator
x_result = x_tes_aPriori + AK_tes##(x_sonde - x_tes_aPriori)

; get predicted error
obs_error = REFORM(species_data.OBSERVATIONERRORCOVARIANCE
[*,*,360])
obs_error = sqrt(obs_error[indp,indp])

print, obs_error
      1.11236      1.12449      1.19919      1.29392
1.09227     0.953536
     0.765534     0.655048     0.514325     0.447764
0.343971     0.335064
     0.290062     0.337749     0.312072     0.355580
0.331500     0.380160
[...]
```

11. Plot result +- predicted error and compare

```
; previously plotted
plot,species_data.species[indp,360],species_data.pressure[in
dp,360],yrange=[1000,10],/ylog,xtitle='Temp
(K)',ytit='Pressure (hPa)'
oplot,species_data.constraintvector[indp,360],species_data.p
ressure[indp,360],linestyle=2

oplot, x_result, pressure,linestyle=1
for jj = 0, N_ELEMENTS(x_result)-1 DO oplot, [x_result[jj]-
obs_error[jj],x_result[jj]+obs_error[jj]],
[pressure[jj],pressure[jj]],color=17 //error bars, red
```



Sonde with TES observation operator applied (dotted). This has markedly smoothed out the sonde in the stratosphere; however it is not consistent with either the TES result (solid) or TES initial guess from GMAO (dashed) to within the predicted observation error (red -- barely visible in plot).

Not all sondes agree well with TES results because of spatial or temporal differences in the air mass measured and statistics with many matches are used to validate TES results.

Appendix

Steps of making comparisons of TES nadir retrievals to your profiles with higher vertical resolution

Ming.Luo@jpl.nasa.gov, H. Worden and TES team
Updated March 20, 2007

http://tes.jpl.nasa.gov/Ming/web.html

TES nadir retrievals of tropospheric species profiles (and the integrated total or partial columns) are not the same as the true profiles (or the columns). Papers have been published to describe the differences and the relationship between the two, *e.g.*, Rodgers (2000), Rodger and Connor (2003), and comparison examples related to the TES data, *e.g.*, Worden *et al.* (2006), and Luo *et al.* (2007). The MOPITT team has also given advices and examples on the related topic, *e.g.*, Deeter (2002), and Emmons *et al.* (2003). This webpage gives detailed steps dealing with TES data and your data with higher vertical resolution, assuming that you understand the basics.

(1) TES data

TES standard product file (.he5) consists following data fields per profile (*e.g.*, iObs) that you need for your inter-comparison: **Pressure[67, iObs]**, **speciesVMR[67, iObs]** (*e.g.*, TATM, O_3), **ConstraintVector[67, iObs]**, **AveragingKernel[67,67,iObs]**, where **iObs** is your selected TES profile, **67** is the number of TES standard pressure levels.

In most cases, the 1st a couple (a few) pressure levels are filled with **-999** (index = 0, 1, ...). The 1st non **-999** pressure value represent the surface. You need to 'reform' (in IDL) the TES vectors (**Pressure, speciesVMR and ConstraintVector**) and the **AveragingKernel** to smaller number of levles starting from the surface level (numLevel < 67).

(2) The standard equation for mimicing TES retrievals assuming your profile is the 'true' (see the above listed references)

x_result = x_tes_aPriori + AK_tes##(x_your_profile - x_tes_aPriori)

where the corresponding TES data is

x_tes_aPriori= *alog*(**ConstraintVector**), and
AK_tes = **AveragingKernel**

You probably noticed that **x_your_profile** in the equation must be on the same pressure grids as the TES pressure grids for the corresponding TES species profile. The next section gives you suggestions on how to map and extend (if necessary) your profile to all the TES pressure levels.

Here we use an IDL notation "##" to match the way that the values in TES '**AveragingKernel**' matrices are stored.

It is important to know that the above **x_tes_aPriori** and **x_your_profile** need to be '*alog*(VMR)' (IDL notation), so '*exp*(**x_result**)' is what you want (except atmospheric temperature). Again, this is uniqe to TES due to that TES does volume mixing ratio (VMR) retrievals in *alog*(VMR).

(3) Map your profile at fine levels to TES standard pressure levels (number < 67)

There are a few things that you need to consider before doing the mapping.

1. Your species profile needs to be a function of pressure in hPa.
2. In most cases, your species profile needs to cover entire TES pressure range, surface to 0.1 hPa. If your profile covers only a small portion of the atmosphere and do not have anything above or below handy, one suggestion is to append the shifted TES a priori profile down or upward of your profile. This extension offers reasonably realistic atmosphere where you don't have measurements.
3. An interpolation of your profile to very fine levels may be necessary.

Now you are ready to map your profile at fine levels to the TES pressure levels.

There is a way described in Rodgers 2000, section 10.3.1.

> **x** = the species profile vector on your fine grids; **z** = your species profile vector mapped to TES levels.
> $\mathbf{x} = \mathbf{W}\mathbf{z}$, or $\mathbf{z} = \mathbf{W}^{*}\mathbf{x}$, where **W** is the map matrix.

The following IDL codes show you how to do this with eq (10.7) in Rodgers 2000, $\mathbf{W}^{*} = (\mathbf{W}^{T}\mathbf{W})^{-1}\mathbf{W}^{T}$:
The IDL pseudo code for calculating the map, **W**, and the IDL pseudo code for obtain your maped profile, **x_your_profile** (or **z** in the equation above).

Final note: it is important to check your interpolated, extended, and mapped profiles via, for example overlaying with the original.

(4) References

Deeter (2002), Calculation and Application of MOPITT Averaging Kernels, available at http://mopitt.eos.ucar.edu/mopitt/data/avg_krnls_app.pdf .

Emmons *et al.*, (2004), Validation of Measurements of Pollution in the Troposphere (MOPITT) CO retrievals with aircraft in situ profiles, *J. Geophys. Res., 109*, D03309, doi:10.1029/2003JD004101.

Luo *et al.*, (2007), Comparison of carbon monoxide measurements by TES and MOPITT the influence of a priori data and instrument characteristics on nadir atmospheric species retrievals, *J. Geophys. Res.*, 112, D09303, doi:101029/2006JD007663.

Rodgers, C. D., *Inverse Methods for Atmospheric Sounding: Theory and Practice*, World Sci., River Edge, N. J., 2000.

Rodgers, C. D., and B. J. Connor (2003), Intercomparisons of remote sounding instruments, *J. Geophys. Res., 108*, 4116, doi:10.1029/2002JD002299.

Worden *et al.*, (2007), Comparisons of Tropospheric Emission Spectrometer (TES) ozone profiles to ozonesondes: Methods and initial results, *J. Geophys. Res.*, 112, D03309, doi:10.1029/2006JD007258.