```
    ============================================
    Descriptive notes for the SARTA v1.08 package
    ============================================
    The Stand Alone AIRS-RTA package (rtpV201 variant)
    Organization: University of Maryland Baltimore County (UMBC)
    Programmer: Scott Hannon (email: umbc.edu)
    AIRS Team Member: L. Larrabee Strow (email: strow@umbc.edu)
    SARTA web site: http://asl.umbc.edu/pub/packages/sarta.html
    Last update: 21 November 2011


Introduction:
------------
The SARTA program is a "stand alone" implementation of the AIRS-RTA,
our fast foward model for AIRS radiative transfer calculations.  The
SARTA program uses a special purpose HDF format called RTP (Radiative
Transfer Profile) for the input and output profile processing.  To
compile and run SARTA package you will first need to build the RTP
library from the RTP package. Most of SARTA's processing instructions
come from the input RTP data file, with some optional processing info
(such as file names) coming from command-line arguments.


Platform-Specific Considerations and Limitations:
------------------------------------------------
The user will need a FORTRAN 77 compiler capable of reading in big
endian data FORTRAN binary data files (the fast model coefficents).
In addition, the FORTRAN code uses the non-standard (but common
extension) "STRUCTURE" variable type.


Performance:
-----------
Typical processing speed is on the order of 5-10 profiles per second
with a ~ 3 GHz CPU.  This assumes all 2378 AIRS channels and a large
number of profiles will be processed.


Command-line arguments:
----------------------
Command line arguments are optional, but most users will probably want
to use the "fin" and "fout" arguments all the time.  Each command line
argument is an entry of the form
   <variable>=<value>.
Each entry must be 80 characters or less. The <variable> names are
not case sensitive.  It is not necessary to enclose <value> strings
in quotes unless they contain blanks.  The recognized command-line
variables are:

    fin : name of input file.  The default is 'sarta_in.rtp'.

    fout : name of output file.  The default is 'sarta_out.rtp'.

    lrhot : logical (true/false or T/F) switch used to force the
       reflected downwelling thermal reflectivity to
          rho=(1-emis)/pi
       instead of RTP input rho.  The default is false.

    listp : list of desired profile numbers (all other profiles will
       be ignored).  The default is to process all profiles.

       The listp profile numbers may be specified either as a
       sequence of integers separated by a comma, or alternately as
       a quoted string containing integers separated by a blank space.
       Examples:
          listp=1,2,3,4,5
          listp='1 2 3 4 5'
       Due to the 80 character limit, the maximum number of entries
```

in listp is limited. (Eg 15 four digit numbers, or
25 two digit numbers. MAXPRO is the hardcoded limit.)

Hint: If you want to process a large number of profiles,
you should probably run a subsetting program to read in
the original RTP file and create a new RTP file containing
only the desired profiles.

Known Limitations:
---

1) The calculation for the reflected solar radiance contribution breaks down at large solar zenith angles (ie when the sun gets near to the horizon). This does not cause any numeric/computation problems, but be aware the accuracy of for angles > 80 degrees is probably not very good.
2) The linear perturbation model used to model variable N2O is not an accurate approximation over the large range of possible (plausible) N2O variability. In particular, the N2O adjustment in the 2180 to 2260 cm^-1 region should be regarded as suspect.

Dual Frequency Variant:
---

The SARTA v1.08 package includes a "dual frequency" variant of the standard v1.08 code. This variant can be used to calculate spectra with the channel frequencies slightly shifted from their nominal value. To do this, the code reads in two databases with slightly different channel frequencies, and does a weighted sum (ie a linear interpolation) of the two to perform calculations at some desired frequency.

The channel frequencies for each profile are specified in terms of the AIRS grating model Y-offset. The nominal Y-offset is -14.0 um, but over each orbit it may drift back and forth 0.5 um. The channels also drift back and forth 0.5 um over the yearly seasonal cycle. And AIRS also has a long term drift of about 0.1 um per year towards more negative values. Over the expected lifetime of AIRS, the range of variation of Y-offset is estimated as roughly -13 to -15 um. Each 1 um change in Y-offset corresponds to a freq/120000 shift of the channel frequencies (ie ~ 8.3 ppm).

The dual frequency SARTA code uses profile field "udef1" to specify the desired Y-offset; this variable is ignored by the standard code. NOTE: header field "vchan" is always set to the (mean) frequency of the database(s); it is independent of prof.udef1.

AIRS PGEv6 variant:
---

The SARTA v1.08 package includes a "AIRS_PGEv6" variant of the standard v1.08 code. This code is approximately eqivalent to the RTA used inside the PGE. The code is based on the "dual frequency" code explained above, but allows for 17 separate module array Yoffsets along with a Doppler shift. These values are imported into SARTA using the non-standard but common "UNION" extension. This allows allow using the "prof.pnote" 80 character field as 4 character plus 19 REAL*4 values. The code attempts to read the 18 real values only if the first 4 characters of pnote are "YOFF". The 19 values are:

- 1-17 = Yoffset for modules arrays 1-17 (um)
  (ie modules 12,11,10,9,8,7,6,5,4d,4c,3,4b,4a,2b,1b,2a,1a)
- 18 = the Doppler frequency shift (ppm)
- 19 = the orbit phase (degrees)

The code attempts to determine if the REAL*4 data is big or little endian by checking if the value of the first Yoffset is within the expected range of variation.

The code tracks A/B weight changes using a lookup table (see the database "Coef/ABweight_table*txt) and "prof.rtime". A new table

entry should be added each time the A/B weight tables used by AIRS
are changed. These have only been a few such changes over the
mission so far, and only a few more are expected before end of
mission. Another A/B weight change is expected circa December 2011.

---

Summary of required variables for SARTA's input file:

The following variables should be properly filled out in the input
file read in by SARTA. If any of these variables are left blank,
SARTA will be lacking some vital piece of data or instructions.
Depending upon what is missing, SARTA will abort and write out an
error message (to unit number IOERR), or it will attempt to process,
in which case you should expect screwy output.

Hint: if SARTA crashes or gives screwy output, the cause is almost
always an error in the input RTP. Determine which profile in the
input RTP is the problem, and then check every field until you find
which one(s) you filled out incorrectly. If the error is not obvious
(ie a garbage value for some required field), make sure array length
fields nlevs, nemis, and nrho match their related arrays.

See the RTP v201 documentation for more info on the RTP variables.

header fields:
ptype = profile type; must be code value 1 or 2 (1=layer, 2=psuedo-level)
pfields = profile fields bit flags; must have PROFBIT set
nchan = number of channels
ichan = channel ID numbers
ngas = number of gases
glist = gas ID list; must include gases 1 (H20), 3(O3), 5 (CO), and 6 (CH4)
gunit = gas amount units; must be code value 1 or 2

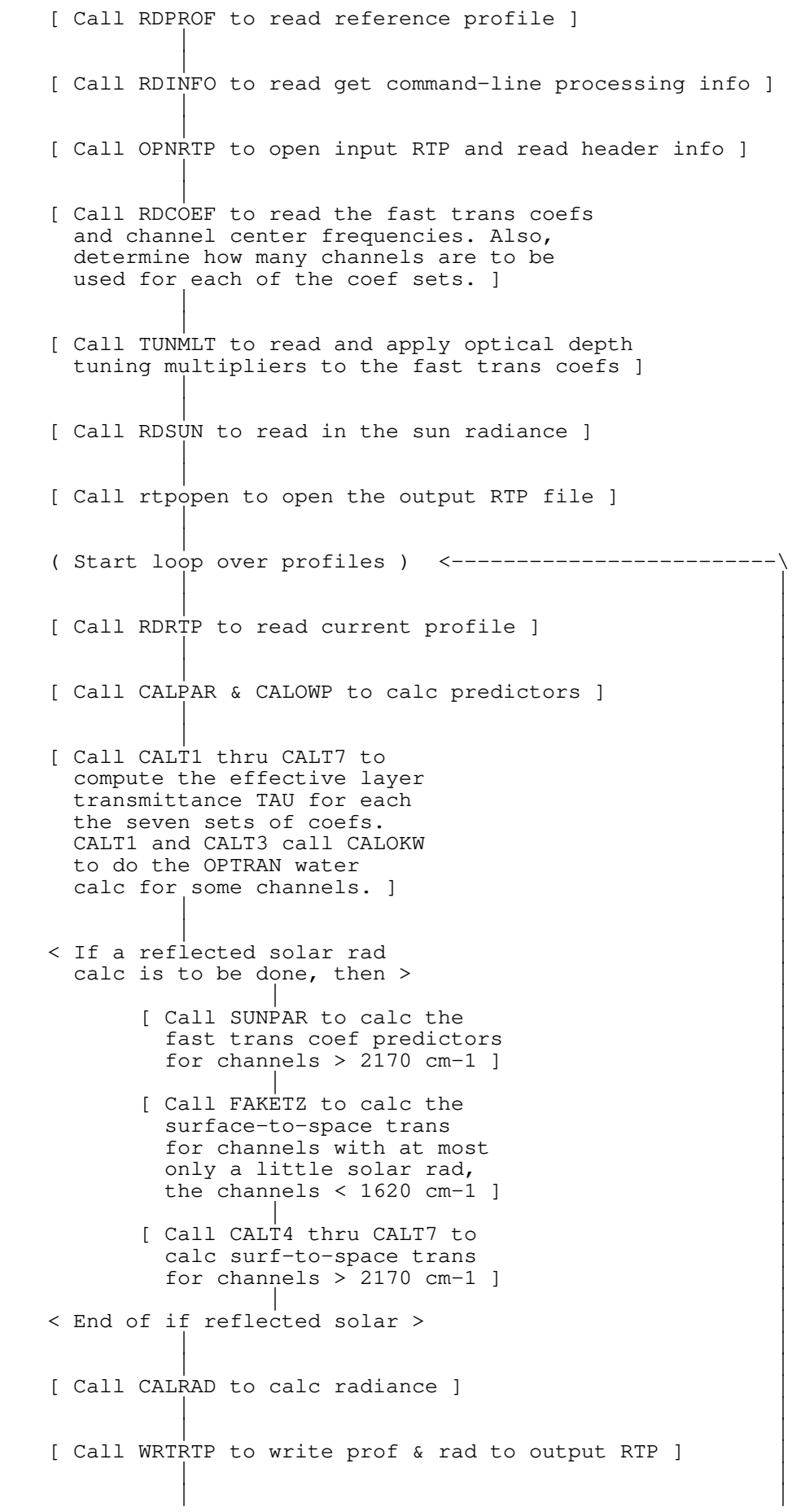
profile fields:
plat = profile latitude
nemis = number of emissivity (& rho) points
efreq = frequencies for emissivity (& rho) points
emis = emissivity points
rho = reflectivity points
stemp = surface temperature
spres = surface pressure
salti = surface altitude
nlevs = number of levels. NOTE: number of layers=nlevs-1
plevs = pressure levels
palts = level altitudes
ptemp = temperature profile
gas_1 = H2O (water vapor) amount
gas_3 = O3 (ozone) amount
gas_5 = CO amount
gas_6 = CH4 (methane) amount
co2ppm = mean CO2 mixing ratio {in parts per million}
satzen = satellite zenith angle *OR* scanang = scan angle.
solzen = solar zenith angle
gas_2 = CO2 amount : optional - otherwise uses co2ppm
gas_4 = N2O (nitrious oxide) amount : optional - otherwise uses ref amount
gas_9 = SO2 (sulfur dioxode) amount : optional - otherwise uses ref amount
gas_12 = HNO3 (nitric acid) amount : optional - otherwise uses ref amount

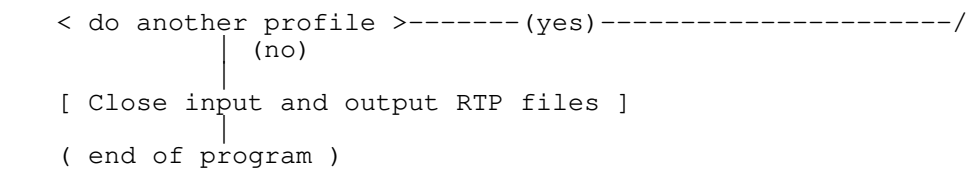
When using the "dual frequency" variant, then also profile field
freqcal = desired grating model Y-offset in um {-12.0 to -16.0}

Profile Units and Format:

The SARTA program uses RTP for both input and output profile I/O.
The KLAYERS program is recommended for processing raw profiles into
the type needed for use with SARTA. Briefly, the units are
All altitudes: meters

```
   All pressures: millibars {mb} = hectoPascals {hPa} [equivalent units]
   All temperatures: Kelvin
   All integrated layer absorber amounts: molecules/cm^2 or kilomoles/cm^2
   All angles: degrees
   All radiances: milliWatts per m^2 per cm^-1 per steradian
See the RTP documentation for further info.

Note: we recommend use of our KLAYERS program (available in a
separate package) to convert raw profiles into the type needed
for use with SARTA.

If ptype=2, the RTP input profile matches the definition
used by the AIRS PGE level2 support profile.  This differs
from ptype=1 in that the temperatures are pseudo-level
values at the lower boundary of the layers (except for
the special cases of the top layer and bottom layers).


-------------------------------------------------------------------------------
The rest of this file consists of a description of the AIRS layers and
an overview of the structure of SARTA.  Unless you are interested in the
inner workings of SARTA, feel free to stop reading here.
-------------------------------------------------------------------------------

The AIRS Layers:
---------------
The 100 AIRS layers are defined by 101 layer boundary pressure
levels.  For i=101 downto 1, Plev(i) (mb) =
     0.0050,    0.0161,    0.0384,    0.0769,    0.1370,
     0.2244,    0.3454,    0.5064,    0.7140,    0.9753,
     1.2972,    1.6872,    2.1526,    2.7009,    3.3398,
     4.0770,    4.9204,    5.8776,    6.9567,    8.1655,
     9.5119,   11.0038,   12.6492,   14.4559,   16.4318,
    18.5847,   20.9224,   23.4526,   26.1829,   29.1210,
    32.2744,   35.6505,   39.2566,   43.1001,   47.1882,
    51.5278,   56.1260,   60.9895,   66.1253,   71.5398,
    77.2396,   83.2310,   89.5204,   96.1138,  103.0172,
   110.2366,  117.7775,  125.6456,  133.8462,  142.3848,
   151.2664,  160.4959,  170.0784,  180.0183,  190.3203,
   200.9887,  212.0277,  223.4415,  235.2338,  247.4085,
   259.9691,  272.9191,  286.2617,  300.0000,  314.1369,
   328.6753,  343.6176,  358.9665,  374.7241,  390.8926,
   407.4738,  424.4698,  441.8819,  459.7118,  477.9607,
   496.6298,  515.7200,  535.2322,  555.1669,  575.5248,
   596.3062,  617.5112,  639.1398,  661.1920,  683.6673,
   706.5654,  729.8857,  753.6275,  777.7897,  802.3714,
   827.3713,  852.7880,  878.6201,  904.8659,  931.5236,
   958.5911,  986.0666, 1013.9476, 1042.2319, 1070.9170,
  1100.0000

These values come from the equation:

   Plev(x)=( a*x^2 + b*x + c )^power

where
   "power" = 7/2 = 3.50
   "x" is level number (an integer)
   and "a", "b", and "c" are solved for such that
      Plev(1)  = 1100.0 mb
      Plev(38) =  300.0 mb
      Plev(101)=    0.005 mb
   (the approximate values are: a=-1.55E-4, b=-5.59E-2, c=7.45)



Simplified flow chart of SARTA
------------------------------
   ( Start SARTA )
           |
```


[ Call RDPROF to read reference profile ]
[ Call RDINFO to read get command-line processing info ]
[ Call OPNRTP to open input RTP and read header info ]
[ Call RDCOEF to read the fast trans coefs
and channel center frequencies. Also,
determine how many channels are to be
used for each of the coef sets. ]
[ Call TUNMLT to read and apply optical depth
tuning multipliers to the fast trans coefs ]
[ Call RDSUN to read in the sun radiance ]
[ Call rtpopen to open the output RTP file ]
( Start loop over profiles )
[ Call RDRTP to read current profile ]
[ Call CALPAR & CALOWP to calc predictors ]
[ Call CALT1 thru CALT7 to
compute the effective layer
transmittance TAU for each
the seven sets of coefs.
CALT1 and CALT3 call CALOKW
to do the OPTRAN water
calc for some channels. ]
< If a reflected solar rad
calc is to be done, then >
[ Call SUNPAR to calc the
fast trans coef predictors
for channels > 2170 cm-1 ]
[ Call FAKETZ to calc the
surface-to-space trans
for channels with at most
only a little solar rad,
the channels < 1620 cm-1 ]
[ Call CALT4 thru CALT7 to
calc surf-to-space trans
for channels > 2170 cm-1 ]
< End of if reflected solar >
[ Call CALRAD to calc radiance ]
[ Call WRTRTP to write prof & rad to output RTP ]




Routine RDPROF:
---
This routine reads the reference profile data from a simple text file. After skipping over the header, the routine reads in the profile variables (altitude, presure, temperature, and CO2, H2O, O3, CO, and CH4 amounts). These are read in the order they occur in the file (that is, 1=lowest altitude to 100=highest altitude), but they are reversed in the arrays (ie element 1=highest altitude and 100=lowest altitude).

Routine OPNRTP:
---
This routine opens the input RTP file and performs some error checking to make sure the file contains the required info.

Routine RDCOEF:
---
The seven main binary data files containing the AIRS fast transmittance coefficients are opened and read one channel at a time. The channel list LSTCHN determines what channels to read/use. The seven sets of coefs are each stored in their own arrays. Next, the peturbation coefficients for the trace gases CO2,N2O,SO2,& HNO3 are read in from their corresponding data files. Next, the OPTRAN water coefficients (and OPTRAN water grid and raw predictor average values) are read in from the OPTRAN data file. Next, the downwelling thermal "F factor" coefficients read in from their file. Next, the "fixed gases" adjustment factor "fx" is read in from its file. Finally, the non-LTE coefficients are read in from their file.

Routine RDSUN:
---
Reads in a text file with solar radiance data for each AIRS channel. This is the solar radiance direct from the sun at the top of Earth's atmosphere. It asks for a file name, opens it, and reads it line by line, skipping any comment lines. There is solar radiance data for every channel.

Function VACONV and SACONV:
---
VACONV is used to convert the satellite viewing angle into the local path angle. SACONV does a similar angle conversion for the solar zenith angle. Both are based on The Law Of Sines, and the functions only account for geometry; no refractive effects are included. The local path angles for each layer varys slightly due to the curvature of the Earth and its atmosphere.

Routine CALPAR and CALOWP:
---
CALPAR calculates the main sets of predictors, which consist of various gas amount and temperature ratios and offsets relative to a reference profile. See the header of file "calpar.f" for details on the predictors. The routine also calculates an adjustment for the "fixed gases" optical depth based on the profile information. The equation used is a hybrid of analytic and empiric terms and should not be altered. CALOWP calculates the OPTRAN water predictors.

Routines CALT1 thru CALT7:
---
The fast trans coefficents and predictors are multiplied together and summed to calculate the effecttive layer transmittances. Each of the CALT1 thru CALT7 routines does this for one of the seven sets of coefs and predictors. The routines loop down over the layers, and check the fixed, water, ozone, carbon monoxide, and methane trans in each layer to be sure it's reasonable (0 < transmission < 1). For the bottom layer, the total layer optical depth is scaled by the bottom layer fraction BLMULT, and a surface-to-space trans TAUZ is output along with each of the LBOT layer transmittances. To help speed up the the exponential calculations, we use our own EXP(x) replacement function called QIKEXP which uses just the first few series expansion terms for exp(x) if x is suitably small.

Routine CALOKW:
---
This routine uses the OPTRAN water fast transmittance coefficients and predictors to calculate the water optical depth for a subset of the channels in sets 1 and 3. The routine passes the computed AIRS layer water optical depths back to the calling routine (either CALT1 or CALT3).

Routine FAKETZ:
---
Calculates a "fake" surface-to-space transmittance for an arbitrary angle by scaling the optical depth aong the viewing angle (ln(TAUZ)) by the ratio of the angle secants. The exact form of the calc is:

```
   TAUZFK = EXP( LN(TAUZ) * SECFAK/SEC )
```

This is a crude approximation of the correct value.

Routine SUNPAR:
---
This is essentially the same as CALPAR, except it only calcs the predictors for sets 4 thru 7, which are the sets with channels where the reflected solar radiance can potentially be rather large.

Routine CALRAD:
---
The radiance is calculated for each channels in turn. The radiance is a sum of four components: surface, upwelling (non-reflected) atmospheric, reflected downwelling atmoshperic thermal, and reflected solar. Currently there is no scattering.

First it computes black body emissions for each layer using the Planck equation:

```
   planck = c1*v^3/( exp(c2*v/T) - 1 )
```

where c1 and c2 are the radiation constants, T is the temperature TP, and v is the frequency FREQ.

We assume the layers emit radiances of

```
  rad_layer = (1 - tau)*planck
```

where tau is the layer transmittance TAU.

The routine loops over the layers, starting at the bottom surface. The total radiance leaving the bottom surface and going upward is the surface emission and reflected solar and thermal. The reflected solar and thermal are handled as separate terms added to radiance arriving at the satellite, while the surface emission is handled along with the atmospheric layer emissions by propogating thru each layer in turn.

```
   rad_surface =  e*planck
```

where e is the bottom surface emissivity EBOT, and the surface is at temperature TBOT.

We trace the upward radiance thru the atmosphere and determine the

total radiance leaving the top layer (and then reaching the satellite:
   the sum L=L_bot downto 1 of { rad(L-1)*tau(L) + rad(L) }
where rad(L_bot-1) = rad_surface, and rad(1) = RAD.

The reflected sun term is based on an approximation suggested by
Joel Susskind et al. The reflected solar radiance reaching the
satellite is given by
   Rsun = rho_s * omega * TAUZSN * Hsun
where omega is the solid angle of the sun as seen from Earth,
Hsun is the (non-reflected) solar radiace at the top of the
atmosphere, and TAUZSN is (surface) layer-to-space transmittance
of a path along an effective total angle defined as
   secant_eff = secant_view + secant_sun
Note that this requires a separate transmittance calculation at the
effective sun angle. Hsun is passed to this routine (it is close to
planck for 5800 K in the important 2100-2700 cm-1 region), while omega
is computed using the distance of the Earth from the sun DISTES
   omega = pi * radius_sun / distance_Earth_sun

The reflected downwelling thermal calculates a downward radiance the
same way we do the upward thermal emission, but modifys it with a
fudge factor to account for the use of inappropriate layer effective
transmittances (ie derived from layer-to-space transmittances instead
of layer-to-surface) and the integration over all angles. This is
then multiplied by the surface reflectivity and the upward view angle
surface-to-space transmittance to get the radiance contribution at
the satellite. The parameterized fudge factor "F" was determined
by regression with the training set using Lambertian reflectance.
The reflected thermal term is thus most accurate when using
Lambertian reflectivity.

Routine CALNTE:
---
During daytime, the absorption of solar energy by some molecules in
the very thin air at the top of the atmosphere can pump certain
vibrational states of these molecules faster than collisions between
air molecules can redistribute the energy amoung the air molecules
in accordance with the air's overall temperature. Under these
conditions, the air is said to be in non-local thermaldynamic
equilibrium (non-LTE), and the radiance seen at the top of the
atmosphere differs from that seen under LTE conditions. This
routine approximates that change in radiance (non-LTE vs LTE),
which can then be added to the usual LTE radiance computed by
CALRAD.

--- end of file ---