CrIS Weather Product Test Bed:

AIRS simulated orbit and NOAA radiosonde data sets

(Sept. 21, 1998)

This document is intended to assist users of the IPO weather product test bed for the CrIS, MHS, and AMSU instruments. The topics covered in this document are:

Table of Contents

These Weather Product Test Bed (WPTB) databases can be used for the following applications

- verify data formats (retrievals, noise models, and statistics) and ability to communicate with the IPO,
- allow vendors to compare their results with the government results in a consistent manner,
- allow vendors to test the robustness of their algorithm for a variety of geophysical conditions,
- allow vendors to modify and generate scenes to isolate high risk areas in their algorithms.

The WPTB can not be used directly to address issues concerning

1. size of the CrIS field of view
2. co-location of instruments
3. vendor trade studies in OPD or effective APERTURE (noise model is fixed for radiance simulation)

The vendors can also utilize the WPTB databases to implement radiances with their own noise models; however, it is up to the vendors to supply the noise model in a format consistent with the radiance generator's format and to compute the radiances of interest. The IPO is working on provided additional RTA(s) to address item #3.

The vendor is encouraged to experiment with these datasets and to discuss any problems with the format or procedure with the IPO. To be useful with regard to the mid-November blind test, the files listed below should be returned to the government by the end of October. The government recognizes that vendor algorithm trade studies are not complete and that these files are not representative of the vendor's ability on these scan lines, nor do they represent capabilities of the vendor's instrumental design.

- Retrieval results, in the 100 level format, from the vendor retrieval algorithm for the five scans lines discussed on pages 7 and 8,
- the statistics output file for each of the five scan lines, generated from instructions on page 23,
- The vendor's noise model(s), prepared in the format discussed on pages 18 and 19.

This document assumes that the software has been provided with the following directory structure

```
~/doc/              this document and output from test runs
~/level1/           radiance files (level 1 files)
~/level2/           geophysical truth files and ancillary files
      us_std/       US Std. atmosphere files for comp_rad program
~/namelist/         namelist files used to compute level 1 files
~/noise/            noise model file(s)
~/RTA/              RTA coefficient files
 /solar/            solar irradiance files from UMBC
 /weight/           weight files for scan statistics

~/src/              government supplied software
     main/          main programs
     src_gsfc/      general subroutines
     mwrta_v4/      MIT microwave RTA
     umbc9803/      UMBC algorithm with GSFC ``wrapper''
     rta66/         re-issue of 66 layer RTA with new calling sequences
     src_jpl/       I/O and primitive functions
     simrad/        subroutines for simrad.F
     lib/           (empty) location for compiled library modules
```

This document will not reiterate the issues discussed in the document delivered in fiscal year 1997 ("Simulation of CrIS Radiances", updated: Oct. 9, 1997). Instead, this document will provide background information for the geophysical datasets and provide instructions on how to use the software. A simplified example of code (∼/src/main/comp_rad.F) is provided to document the compilation and use of these routines and to provide validation that the user can reproduce the radiances provided.

The vendor should not expect to be able to identically reproduce the radiances in the ∼/level1/ directory, since the random number generator used for noise computation is machine dependent. The difference of radiances should not exceed the Gaussian distribution of the noise model used, however. All the code used to produce these radiances has been provided for the purpose of documenting the assumptions and methodology used in computing these datasets.

The AIRS Orbital Geophysical Model

The AIRS orbital simulation is derived from a global simulation which is based on the output from a version of the operational GCM from NOAA NCEP. The fields in the GCM are 2.5 degree by 2.5 degree gridded data from a spectral model of NCEP (*e.g.*, Juang, H.-M. H., S.-Y. Hong and M. Kanamitsu, 1997, "The NCEP regional spectral model: an update." *Bulletin Amer. Meteor. Soc.* **78**, p.2125). The temperature, ozone and the liquid water are at 29 levels from 1015 mb to 3 mb. The water vapor profiles are at 12 levels from 1015 to 300 mb. A GOES IR image for Nov. 5, 1996 is shown in Figure 1.

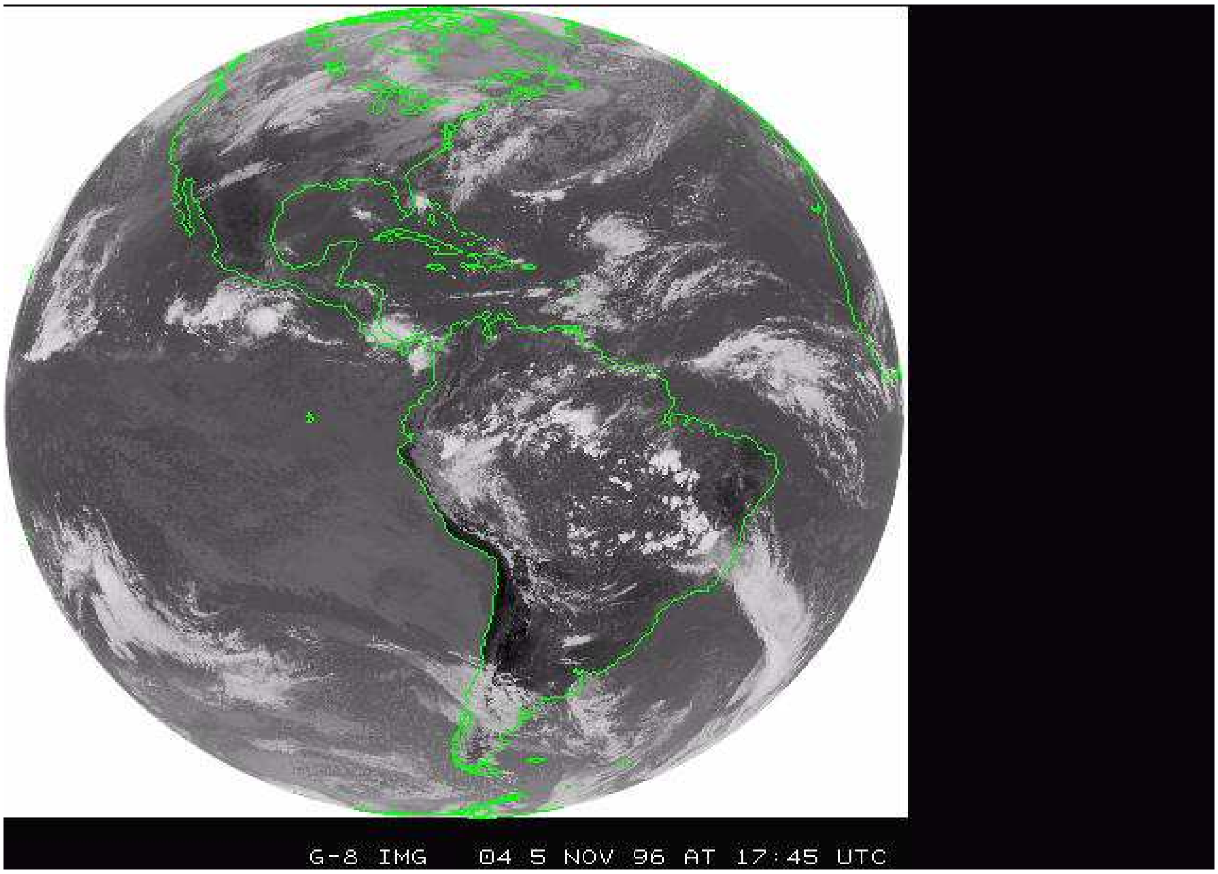

Figure 1: GOES IR image for Nov. 5, 1996

In order to generate a profile, the four surrounding grid points have to be found for the time period immediately preceding and the time period following the given time. A bilinear interpolation method is used to compute the forecast fields at the given location for the two time periods. Then the time interpolation of the forecast data is performed to get the data at the location and at the given time. A vertical interpolation of forecast fields was performed to generate the fields in the vertical layers of the AIRS retrieval software.

Note that this linear interpolation of 2.5 degree by 2.5 degree gridded data produces smooth fields. However the cloud clearing algorithm depends on the contrasts between AIRS/MHS footprints within an AMSU footprint. Therefore the cloud fraction was randomized at the end of the interpolation process. The artificially smooth fields, especially water vapor fields and surface parameters, will be studied and fixed later, if necessary.

The top of the forecast model data is at 300 mb for water vapor profile and 3 mb for others. The water vapor is extrapolated above 300 mb by multiplying the mixing ratio by (p/300)**3, where p is pressure in milli-bar, and then converted to molecular column density. The UARS climatology is appended to the temperature profile above 3 mb. The AIRS orbital dataset has the following salient features:

- implementation of surface topography and variable surface pressure
- daytime and nighttime conditions
- T(p), q(p), O3(p) from surface to 0.005 mB
- cloud liquid water profiles (only affects microwave)
- multiple level cloudy conditions with spectrally varying cloud emissivity and reflectivity, consistent with atmospheric conditions.
- variable surface skin temperature, surface emissivity, and surface reflectivity.
- variable land coverage, with coast-lines, islands, lakes, etc.
- orbital simulation with simulated scan lines with variable viewing angle and solar zenith angle.

  One simulation simplification has been made to help the vendors. In a real scan line of data, each of the 9 MHS and 9 CrIS spots that are co-located within a given AMSU-A footprint will have its own zenith angle. Some algorithms (*e.g.*, cloud clearing) need to "correct" the IR radiances to a single zenith angle. An algorithm to "correct" the group of 9 AIRS spots to the center AIRS footprint (*i.e.,* footprint# 5 of the 9 co-located footprints) has been developed by the AIRS Science Team for the AIRS instrument; however, we assume the vendors have not developed this algorithm. Even if the vendor algorithm existed, it would be for their CrIS footprint pattern and NOT the AIRS pattern that has been simulated here.

  For these reasons, the data in each file are grouped sequentially (*i.e*, 9 spots of the MHS file and 9 spots of CrIS file are associated with 1 AMSU spot). All 9 of the CrIS spots are simulated at the zenith angle of the central spot of CrIS (# 5/9) and all 9 of the MHS spots are simulated at the zenith angle of the MHS's central spot (which is identical to CrIS in this simulation). The AMSU-A will have one zenith angle which is not equal to the either CrIS or MHS. This is because AMSU-A steps at 3 1/3 degree and MHS steps at 1.1 degree.

In all cases, the geophysical water vapor and ozone profiles were converted from layer column densities back to mixing ratio (units of gram per gram of dry air) at the specified 100 levels. These mixing ratio files replace the original column density files as the truth. The new level.2 files contain the following information

level.2 header record (read with ~/src/src_gsfc/openl2.F)

- version: a 5 character version identification (a5). The mixing ratio files have a version name of "M2V1C".
- numpro: the number of profiles in this file (i10)
- nemis: number surface emissivity/reflectivity wavenumber hinge points (see **getemis1.f** for use of these hinge points) (i10)
- ncemis: number cloud emissivity/reflectivity wavenumber hinge points (i10)
- nl2lev: number of layers in the vertical profiles (i10)
- ndescr: number of description lines in header (i10)
- Pobs(L), L=1, nl2lev: pressure value at bottom level of layer (8f10.3). The top of the atmosphere is 0.005 mb.
- descr(i), i=1,ndescr: 80 character labels, (a80)

level.2 data record (read with ~/src/src_gsfc/readl2.F)

Each profile within the "M2V1C" version of the level.2 file has the following format

- idprof: 8 character profile identification, usually 1^{st} 4 characters are descriptive and the last 4 characters are a number (a8).
- latitude (±90) and east longitude (±180) in degrees (2f9.3)
- year, month, day, hour, minute, second: (i6,4i3,f7.3). NOTE: the year is now 4 digits and the sec is now a real*4 value.
- pland: fraction of land (0 = ocean, 1 = land) (f10.5)
- Psurf: surface pressure in millibar (f10.2)
- Tsurf: skin temperature (f10.2)
- emismw: microwave emissivity (f10.5)
- topog: topography parameter (not used) (f10.2)
- ncld: number of cloud layers (i3) NOTE: The format of ncld has changed from (i4) to (i3).
- pcldtop(i), cldfrc(i): cloud top pressure and cloud fraction for i=1,ncld (f10.2,f10.5)
- tair(L), gpg_w(L), gpg_o(L), gpg_l(L), ciw(L) for L=1,numlev: (f9.3,3e15.5,i2)
  - tair(L): temperature in Kelvin at Pobs(L)
  - gpg_w(L): water mixing ratio in units of gram of water per gram of dry air at Pobs(L).
  - gpg_o(L): ozone mixing ratio in units of gram of ozone per gram of dry air at Pobs(L).
  - gpg_l(L): the liquid water mixing ratio in units of gram of liquid water per gram of dry air at Pobs(L).
  - ciw(L): ice/liquid flag for gpg_l(L) (0=water, 1=ice)
- freqemis(i), emisir(i), rhoir(i), i=1,nemis: the IR surface emissivity and reflectivity defined at frequency hinge points (f10.3,2f10.5)
- cldfreq(i,k), cldemis(i,k), cldrho(i,k), i=1,ncemis, k=1,ncld: the IR cloud emissivity and reflectivity defined at frequency hinge points (f11.3,2f10.5)
- nspr, nspi: number of spare real and integer values (2i3)
- ispare(nspi): integer spare values (8i8)
- rspare(nspi): real spare values (4e15.5)

We have provided a subroutine (~/src/src_util/mr2cd.F) to convert the mixing ratio values to layer column density (molecules/cm^2), which is the required input for the government version of the 100 layer UMBC RTA. A description and derivation of the methodology for the conversion is discussed in Appendix A, at the end of this document.

The new AIRS Science Team dataset is quite challenging, especially with regard to clouds, and we have provided training and examples to exercise the vendors' algorithms. In this delivery we have provided the following:

1. **Training ensemble, ~/level2/TR.trn**

   The AIRS Science Team simulated a full orbit of data (6h forecast for Nov. 6, 1996) for the 705 km EOS PM-1 orbit. This orbit consists of 22,248 AMSU footprints (741.6 AMSU scan lines) and 200,232 AIRS and HSB footprints (2224.8 AIRS scan lines).

   For training purposes, given the smooth nature of the horizontal structure in this simulation, the full orbit is not necessary. To make the data file and training process more manageable, we have selected the center CrIS (AIRS) footprint (footprint # 5 of the 9 CrIS footprints co-located within the AMSU footprint) of every 5^{th} AMSU scan line from the orbit. Therefore, there are 30 CrIS footprints per AMSU scan line or 4470 profiles in total.

   In Figure 2, we illustrate the distribution of the selected scan lines within the orbit. The daytime profiles are shown in red and nighttime profiles are shown in blue. Table 1 shows the distribution of day/night and land/ocean profiles.

   No radiances will accompany this ensemble of data. We have provided all the code (*e.g.*, see ~/src/main/comp_rad.F or ~/src/main/simrad.F) and data-files necessary to compute radiances consistent with the data described in item # 2, below.

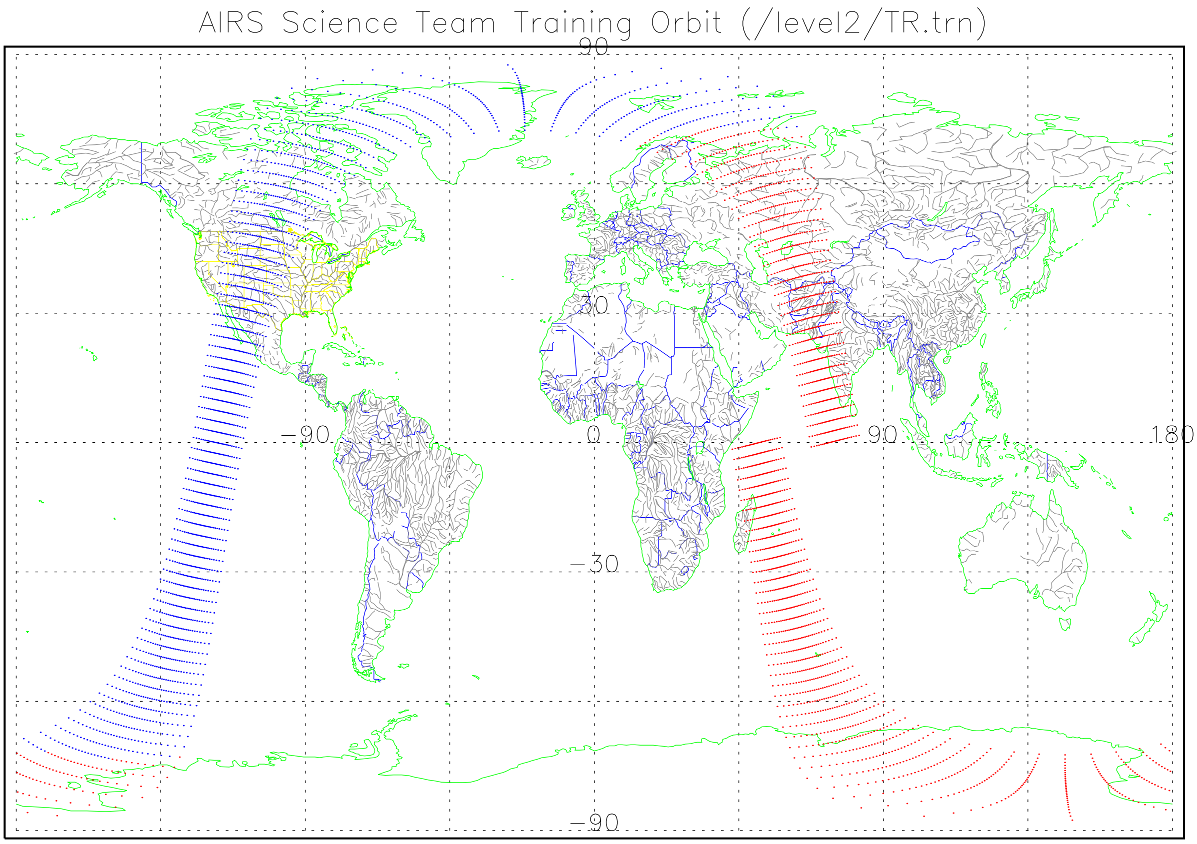

Figure 2: Distribution of profiles in the training dataset.

The distribution of profiles within the training dataset file, ~/level2/TR.trn, is shown in Table 1. There are a total of 4470 profiles distributed throughout the orbit shown in Figure 1. The column labeled "total/ocean+land" shows the distrubution of all profiles as a function of latitude. The table further divides the profiles into ocean and land and daytime or nightime profiles. For example, between 30.0 to 39.9 northern latitude there are 253 profiles, of which 2 are ocean (1 day and 1 night) and 251 are land (125 day and 126 night).

Table 1: Distribution of Training Profiles

		total	ocean			land		
latitude		ocean+land	total	day	night	total	day	night
-90	-81	83	22	.	22	61	.	61
-80	-71	332	126	38	88	206	.	206
-70	-61	282	247	140	107	35	.	35
-60	-51	262	262	131	131	.	.	.
-50	-41	255	252	128	124	3	.	3
-40	-31	250	250	124	126	.	.	.
-30	-21	256	254	128	126	2	.	2
-20	-11	250	239	125	114	11	.	11
-10	-1	264	264	127	137	.	.	.
0	9	265	261	127	134	4	.	4
10	19	252	221	125	96	31	1	30
20	29	253	108	77	31	145	48	97
30	39	253	2	1	1	251	125	126
40	49	255	12	.	12	243	128	115
50	59	262	21	19	2	241	113	128
60	69	281	86	65	21	195	76	119
70	79	332	216	181	35	116	114	2
80	90	83	57	57	.	26	26	.
total		4470	2900	1593	1307	1570	631	939

2. **selected AMSU scan lines**

Sixteen AMSU scan lines of geophysical data, selected from an orbit on the day prior to the training ensemble (6h forecast for Nov. 5, 1996 from 8:30 to 9:54), are provided for the vendors use. The 16 selected AMSU scan lines are arranged in pairs starting with #1 and #2 in India. The location and some properties of the scan lines are shown in Table 2 and Table 3. The index # within the orbit (orb#) is shown in the tables for reference to the original AIRS Science Team orbital data file.

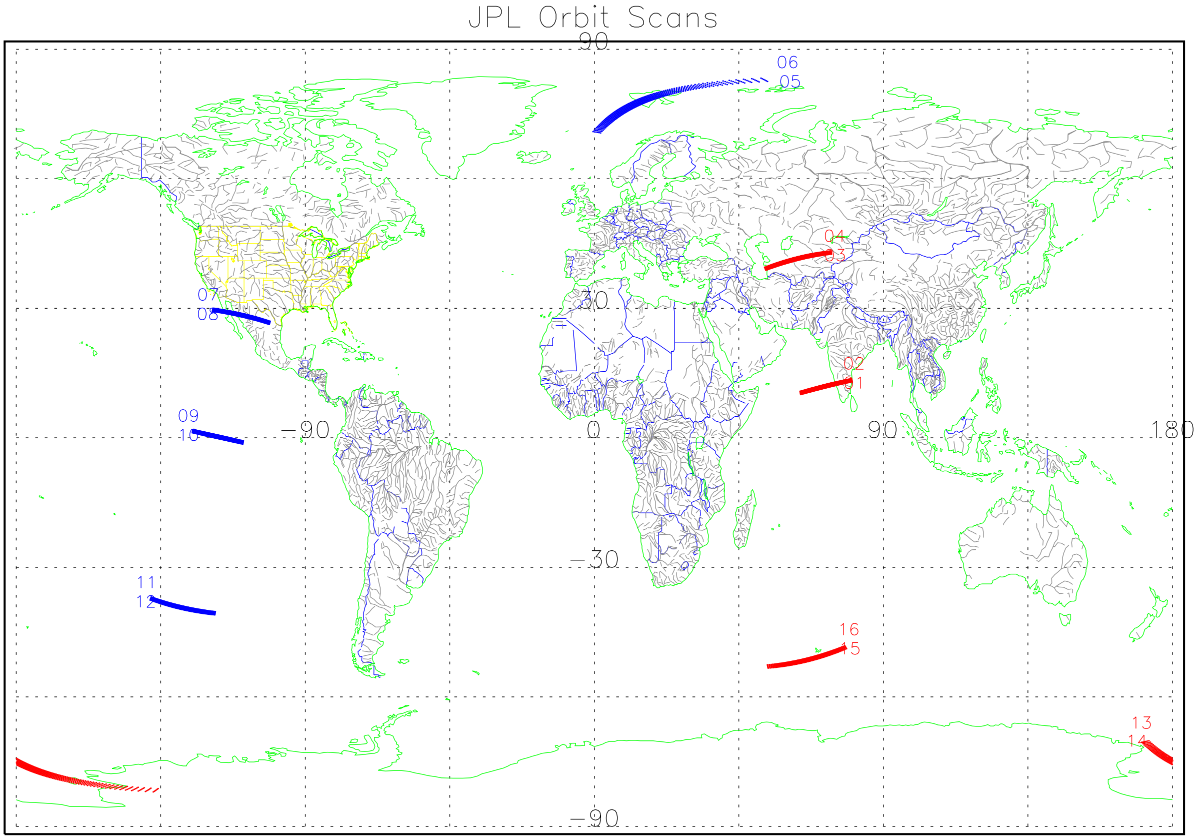

Figure 3: Location of 16 profiles

Radiances have been generated for four of these (#4, #8, #11, #14) scan lines. In addition, we have generated radiances for scan line # 11 without any clouds. The CrIS radiances are provided as Hamming apodized radiances with the nominal CrIS instrument Hamming apodized noise included. Simulated AMSU-A and MHS brightness temperatures are also provided. These data were generated assuming that there are 9 CrIS and 9 MHS footprints co-located with each AMSU. We will use the AIRS footprint pattern at this time - since CrIS scanning patterns are not firm. For each AMSU scan line of data there are 3 files (270 CrIS footprints, 270 MHS footprints, 30 AMSU-A footprints).

The radiance data files are located in ~/level1/ and have the following naming convention:

- the scan ID number, "sc04_", "sc08_", "sc11_", "sc14_", or "sc11c_"
- the instrument: "am", "mh", or "ir" for AMSU, MHS, and InfraRed, respectively. Also, the truth level.2 file is echoed in "tr".
- the type of file: ".asc"

For example, the radiance for the CrIS instrument for scan # 4 is given in ~/level1/sc04_ir.asc. The first nine records of this file are associated with the AMSU footprint, which is in the first record of ~/level1/sc04_am.asc, and the first 9 records of the MHS file, given in ~/level1/sc04_mh.asc. The ancillary file (description given in # 4, below) is ~/level.2/AUX.04. Other footprints are read in sequentially.

The program ~/src/main/simrad.F was used to generate these radiances from namelist files given in ~/namelist/sim_sc04_cris.nl and ~/namelist/io_sc04_cris.nl. The truth file was ~/level2/TR.04 and the scan pattern was specified in ~/level2/L1scan04.dat.

The radiances provided are Hamming apodized radiances (units of milli-Watts/meter^2/steradian/cm^{-1}). We chose Hamming radiances because

- It is a reasonable apodization function to use in retrievals and
- It can be easily converted to radiances of other apodization functions.

The Hamming apodized radiances can be converted to unapodized radiances using COSINE transforms and dividing the interferogram by the Hamming apodization function. Each band needs to be transformed separately. This routine is not provided; however, it is trivial to implement. The steps of the conversion are

- isolate the N radiances, $R_A(n)$, associated with each band
- apply an inverse COSINE transform for the N points

$$\begin{aligned} I_A(k) &= \mathrm{DCT}^{-1}\left(R_A(n)\right) && (1.1) \\ &= \frac{R_A(1) + R_A(N)\cdot\cos(\pi\cdot k)}{2\cdot N} + \frac{1}{N}\sum_{j=2}^{N-1} R_A(n)\cdot\cos\left(\pi\cdot j\cdot k/N\right) && (1.2) \end{aligned}$$

- divide the result by the Hamming apodization function

$$I_U(k) = \frac{I_A(k)}{0.54 + 0.46\cdot\cos(\pi\cdot k/N)} \tag{1.3}$$

- (optional) Multiply the unapodized interferogram by the desired apodization function
- apply the COSINE transform to obtain radiances with the new (unapodized, if optional step is ignored) apodization function.

$$\begin{aligned} R_U(n) &= \mathrm{DCT}\left(I_U(k)\right) && (1.4) \\ &= R_A(1) + R_A(N)\cdot\cos(\pi\cdot k) + 2\cdot\sum_{j=2}^{N-1} R_A(n)\cdot\cos\left(\pi\cdot j\cdot k/N\right) && (1.5) \end{aligned}$$

A COSINE transform can be done with FFT's; however, these transforms need to be done on the exact number of points within each band (*i.e.*, 761, 481, 161). Extreme care must be taken with zero filling. It is suggested that the direct COSINE transform, discussed above, be implemented to avoid errors. The unapodized radiance result can easily be tested with the Hamming matrix operator

$$R_A(n) = 0.54\cdot R_U(n) + 0.23\cdot\left(R_U(n-1) + R_U(n+1)\right) \tag{1.6}$$

Table 2: Location of AMSU Scans									
scan	orb	Day or	# of	starting		ending			
#	#	Night	land	Lat	Long	Lat	Long	Ts	Location
1	25	D	68	10	80	13	80	300	Indian Ocean/India
2	26	D	69	10	80	13	80	300	Indian Ocean/India
3	87	D	255	38	53	42	73	290-275	Kazakh (former USSR)
4	88	D	251	38	53	42	73	290-275	Kazakh (former USSR)
5	171	N	66	70	1	82	53	277-266	N.polar, Svalbard Isl.
6	172	N	45	70	1	82	53	277-266	N.polar, Svalbard Isl.
7	311	N	206	-27	-101	-29	-118	295-291	Baha, Mexico
8	312	N	188	-27	-101	-29	-118	295-291	Baha, Mexico
9	370	N	0	0	-109	1	-125	295-297	Eastern Pacific
10	371	N	0	0	-109	1	-125	295-297	Eastern Pacific
11	453	N	0	-40	-118	-37	-138	284-287	S.E. Pacific
12	454	N	0	-40	-118	-37	-138	284-287	S.E. Pacific
13	539	D	48	-81	-135	-70	+170	257-249	Antarctica
14	540	D	42	-81	-135	-70	+170	257-249	Antarctica
15	633	D	0	-53	54	-48	78	274-277	S.In.Ocean, Kerguelen Isl.
16	634	D	0	-53	54	-48	78	274-277	S.In.Ocean, Kerguelen Isl.

Table 3: Cloud and Surface Pressure Properties of AMSU Scans													
scan	orb	cloud fraction (%)			pcld(1) (mb)			pcld(2) (mb)			Psurf (mb)		
#	#	min	max	avg	min	max	avg	min	max	avg	min	max	avg
1	25	1.0	52.7	26.0	99	129	117	672	672	672	937	1018	994
2	26	1.2	51.6	21.8	101	130	116	672	673	672	942	1021	995
3	87	0.0	90.1	28.8	213	447	318	365	700	492	843	1022	977
4	88	0.0	92.7	32.3	215	447	321	362	700	478	871	1020	983
5	171	56.0	96.2	87.3	239	343	301	477	556	513	970	991	980
6	172	45.9	94.3	86.8	241	331	297	483	559	512	969	993	978
7	311	31.2	68.5	48.7	137	164	151	882	884	883	831	1011	922
8	312	26.2	67.3	43.2	134	165	151	882	884	883	826	1013	926
9	370	7.7	40.7	21.3	90	159	126	806	876	829	1005	1010	1008
10	371	2.0	41.9	18.3	90	162	122	809	877	844	1006	1010	1008
11	453	68.6	91.4	78.9	250	288	271	349	368	353	1012	1021	1017
12	454	67.8	92.8	81.5	249	288	271	348	374	352	1013	1021	1018
13	539	3.8	86.1	54.4	195	510	359	452	716	507	911	983	960
14	540	23.7	90.8	60.5	189	521	364	453	716	498	930	986	962
15	633	7.7	81.4	53.1	237	790	426	419	773	610	983	992	986
16	634	5.5	82.1	49.0	240	790	426	421	772	609	983	992	986

3. **NOAA 1988 radiosonde dataset**

We have re-issued the NOAA-88 geophysical dataset on 100 levels. The file is now called "∼/level2/noaa88b.asc". This dataset still includes the old surface and cloud models and has no liquid water. The temperature, moisture, and ozone profiles were extended to 0.005 mb in a reasonable manner. No radiance is provided for this product; however, the vendors can compute radiances themselves.

This file is a subset of the original data file because not all profiles could be co-located to rocketsondes. The total number of profiles is now 7457 of the original 8344 profiles. The distribution of these profiles is shown in Table 4, for ocean profiles, and Table 5, for land profiles.

The subset ensembles used in the 1997 IGS study for the IPO, consisting of 65 profiles in the winter ensemble and 81 profiles in the summer ensemble, are affected by the missing profiles. In the winter ensemble there are now 54 profiles and 74 in the summer ensemble.

For the noaa88b.asc dataset the level.2 file utilizes some of the spare fields

(a) rspare(1) is the SBUV co-location distance in degrees; if this value is -999.9 then the previous profile's SBUV profile was used,

(b) rspare(2) is the SBUV co-location time difference in hours

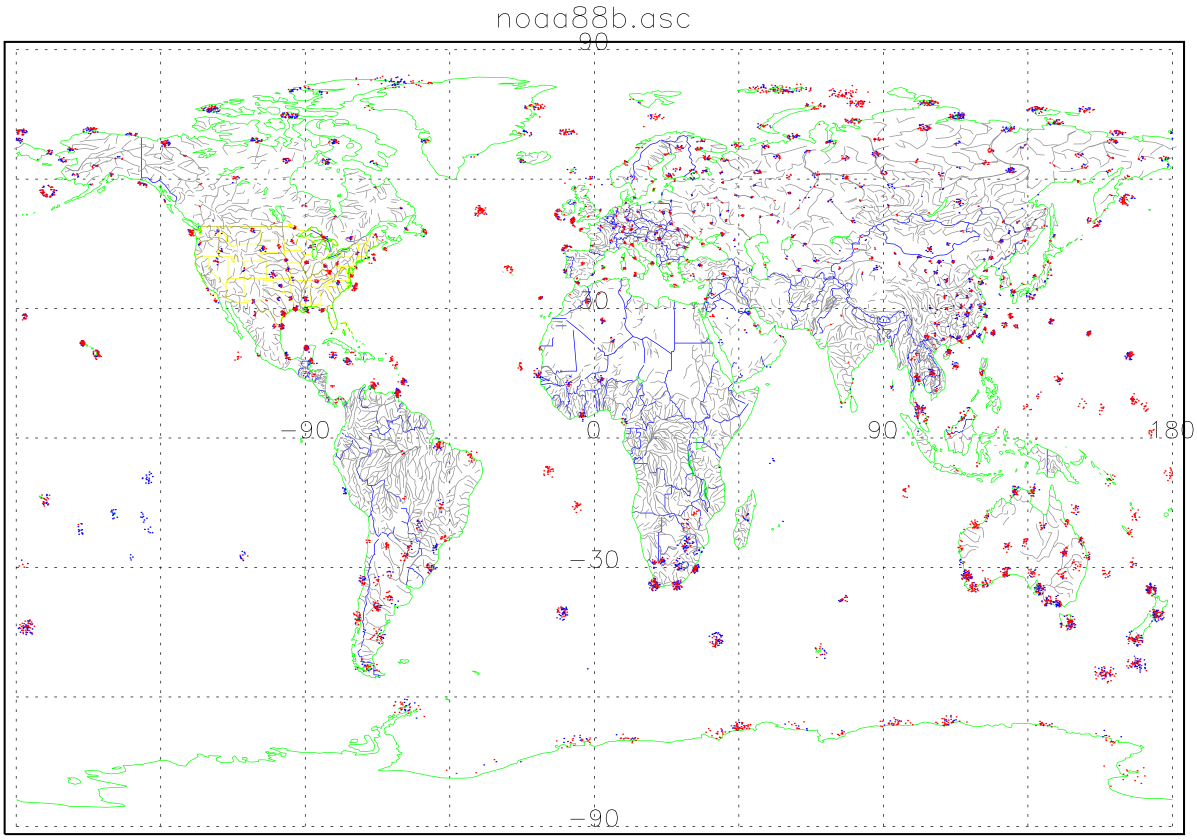

Figure 4: NOAA 1988b dataset

In the following Tables, "D" are daytime profiles, as determined from the date, time, and location, and "N" are nighttime profiles. "total" refers to "ocean" plus "land" profiles.

Table 4: NOAA88b, profiles over OCEAN													
latitude		total	ocean	ocean	ocean	winter		spring		summer		fall	
lower	upper	all	D+N	N	D	N	D	N	D	N	D	N	D
-90	-81	2	1	.	1	.	.	.	.	.	.	.	1
-80	-71	41	9	2	7	.	4	2	.	.	.	.	3
-70	-61	244	58	20	38	5	20	12	.	2	.	1	18
-60	-51	148	117	59	58	24	31	23	.	4	.	8	27
-50	-41	404	149	81	68	12	41	33	.	18	3	18	24
-40	-31	735	128	65	63	18	29	21	4	9	8	17	22
-30	-21	420	60	37	23	16	7	6	5	4	3	11	8
-20	-11	280	104	42	62	12	22	12	14	9	13	9	13
-10	-1	124	49	12	37	4	12	5	12	1	9	2	4
0	9	236	55	8	47	2	11	2	12	2	12	2	12
10	19	584	176	67	109	18	25	16	34	10	15	23	35
20	29	650	67	30	37	16	10	3	6	4	17	7	4
30	39	597	75	19	56	4	6	4	16	5	24	6	10
40	49	746	103	28	75	11	6	1	31	3	28	13	10
50	59	554	255	85	170	38	18	1	77	2	62	44	13
60	69	852	7	5	2	3	.	.	1	.	1	2	.
70	79	754	156	40	116	22	1	.	56	.	56	18	3
80	90	176	26	8	18	7	.	.	8	.	10	1	.
total		7547	1595	608	987	212	243	141	276	73	261	182	207

Table 5: NOAA88b, profiles over LAND													
latitude		total	land	land	land	winter		spring		summer		fall	
lower	upper	all	D+N	N	D	N	D	N	D	N	D	N	D
-90	-81	2	1	.	1	.	1	.	.	.	.	.	.
-80	-71	41	32	5	27	.	26	4	.	.	.	1	1
-70	-61	244	186	43	143	7	110	30	.	4	.	2	33
-60	-51	148	31	7	24	.	18	5	.	2	.	.	6
-50	-41	404	255	113	142	23	84	42	1	30	7	18	50
-40	-31	735	607	273	334	53	154	97	29	98	43	25	108
-30	-21	420	360	143	217	26	85	45	24	62	48	10	60
-20	-11	280	176	62	114	7	44	29	32	23	29	3	9
-10	-1	124	75	18	57	3	18	4	11	9	22	2	6
0	9	236	181	59	122	39	53	8	20	2	33	10	16
10	19	584	408	184	224	69	83	34	52	34	40	47	49
20	29	650	583	248	335	104	91	40	89	36	88	68	67
30	39	597	522	230	292	115	87	6	88	11	58	98	59
40	49	746	643	268	375	156	80	5	125	11	124	96	46
50	59	554	299	134	165	81	21	1	72	.	57	52	15
60	69	852	845	392	453	239	18	.	179	1	227	152	29
70	79	754	598	284	314	182	6	.	187	.	108	102	13
80	90	176	150	73	77	47	.	.	47	.	28	26	2
total		7547	5952	2536	3416	1151	979	350	956	323	912	712	569

4. **Ancillary information**

   A file containing ancillary information for each of the 16 scan lines is provided. For each AMSU footprint the following information is provided

   - a 8 character tag (same as "idprof" in level.2 and level.1 files) to identify the AMSU footprint
   - the 18 hour NCEP forecast surface pressure
   - the fraction of land coverage within each of the 9 co-located CrIS or MHS footprints

   These files are stored in the ~/level2/ directory with the name "AUX." and the scan number suffix.

5. **Statistics program**

   A program to compute statistics (~/src/main/simstat.F) between two files is provided. We will require that the vendor produce their product on the 100 levels of the new UMBC algorithm. This program reads the truth and retrieval file in our L2 format (openl2.f and readl2.f) and produces an output file of IPO statistics. This program is backward compatible to the 66 level files. It also handles all formats of the readl2 routines and converts mixing ratio files to layer column densities for computation of the statistics.

6. **RTA and radiance simulation code**

   We have re-issued the 66 layer RTA and radiative transfer code with upgrades for the 100 level code. This code should give a similar answer to the previous code (some minor differences exist because of order of multiplication) but uses the new calling sequences. In this way, the user can validate their code modifications against existing retrievals.

   The UMBC 100 level RTA for CrIS is also provided with a "wrapper" to make it compatible with retrieval codes.

   A routine to generate noise free radiance (~/src/main/comp_rad.F) is provided to document the proper calling sequences and to give the user a simple program to validate compilation of these routines.

   The routine (~/src/main/simrad.F) and namelist files used to simulated the AMSU scan line radiances is provided. This routine is a complicated routine. A terse description of the namelist variables is given later. The user can modify/utilize this routine for the following purposes:

   (a) to understanding how the government produces the simulated scan lines,
   (b) allow the vendor to produce radiances for all 16 scan lines of data,
   (c) to be utilized with the AIRS Science Team training dataset or the NOAA 1988 dataset to produce radiance with or without noise in a manner consistent with the scan line radiances.

7. **Blind test**

   (proposed for mid-November) The blind test will consist of $\approx$ 4 AMSU scan lines (each with 270 CrIS footprints, 270 MHS footprints, 30 AMSU-A footprints) of simulated CrIS radiances with the ITS nominal instrument noise included. The ancillary file with surface pressure and fractional land coverage will be provided for these scans. In the blind test, the geophysical files will not be provided.

   In preparation for the blind test, the vendors should avoid using any information from the level.2 files supplied with item # 2. In particular, the routine getemis.F for microwave emissivity and the IR surface spectral frequency hinge points within the level.2 file should not be used within the vendor's retrieval.

Rapid Transmittance Algorithm for notional CrIS instrument

This set of radiances utilized the latest version of the UMBC rapid transmittance algorithm (RTA) for the notional CrIS instrument. It has the following features

- 3 bands with the OPD and band limits given in table # 1. A band limiting filter has been applied to the radiances before transformation, so the channels at the edge of the band roll off. They are usable for retrievals; however, the noise model does not include the effects of this filter.

Table 6: definition of bands used in simulation

band	L (maximum OPD)	wavenumber range	# channels
1	0.79942	630.05 - 1105.08	761
2	0.39997	1205.09 - 1805.13	481
3	0.20008	2104.15 - 2503.99	161

- 100 levels from 0.005 to 1100.0 mb.
- improved spectroscopy
- off nadir SOLAR angles
- off-nadir and altitude dependent secant angle
- latitude and altitude dependent gravity function

In addition, the radiative transfer code has been upgraded to include

- the UMBC IR thermal down-welling model
- reflected solar radiation computed consistent with solar zenith angles

We have provided a new "wrapper" for the UMBC code which simplifies the use of these coefficients. The code to load and utilize this RTA, along with some examples of radiance computation is provided.

Compatibility with other RTA's

This version of the rapid transmittance algorithm (RTA) implementation separates the state variables (T(p), q(p), O3(p), etc.) from RTA variables (coefficients, predictors, flags, etc.). The calling sequences are intended to be RTA independent. The 66 layer RTA has been re-issued with the RTA independent call structure. We expect that all future RTA's to be "plug-in" compatible with this delivery. The government's radiance simulator and retrieval code are completely upward and downward compatible with many other RTA's. A single variable, iphys, is used within the code to select and/or verify that the RTA is loaded correctly and is consistent with physics or assumptions (both RTA and radiative transfer (RT)). The current levels of iphys currently supported within the government routines are

1. early version of the RTA without topography
2. version used for the AIRS Science Team track data
3. same as # 2, except solar reflection term NOT divided by pi
4. 12/96 version of the UMBC algorithm for AIRS. Secant angle is not altitude dependent, no microwave background, solar bidirectional transmittance is a kludged value.
5. 5/98 version of the UMBC algorithm for AIRS + CrIS
6. same as 5, except transmissive and overlapping clouds are allowed (in development)

The RTA variables are broken into two classifications and stored in common blocks. The dimensions, declarations, and coefficients are defined in rta_stat.com and all variables that are profile dependent (flags, predictors, trace gas column densities, etc.) are stored in rta_prof.com. In a vectorized environment the common block rta_prof could be converted into a structure and passed from load_ir routine to the other routines via the upper level routines. This would violate the upward and downward compatibility of the code and require that the calling routines know the dimensionality of the arrays, therefore, the use of the rta_prof common block is preferred at this time.

Once the RTA is loaded, the user sets up the secant angles for that profile by calling the routine setang_ir(). The trace gas column densities (CO2, CH4, CO) are set by calling init_trace(). All the predictors are then initialized by calling init_rta(). After this is done the user can compute the RTA on a channel-by-channel basis with a call to ir_rta() for each channel. If the state variables change, then init_rta must be re-called. The routines setang_ir() and init_trace() only need to be called if those state variables have changed. There is a slight dependence of the secant angle to water vapor, which arises because the secant angle is a function of altitude. If the water profile changes substantially the altitude should be recomputed (comp_alt.F) and then setang_ir should be called.

A list of the files in the government RTA are given below. Some of these files have the same name as the original UMBC files; however, these files are different from the UMBC delivery. **DO NOT mix code from the Feb. 1998 UMBC delivery with this code.**

Program to convert UMBC files to the government format

The program cnv_umbc.F is a stand alone program which was used to convert the UMBC files to the format supplied in this dataset. The file names and directories are hardwired. This program is only provided to document the changes that were made to the UMBC coefficient files. The vendor does not need to compile or use this program.

- cnv_umbc.F (uses rta_stat.com)
  - load_tr0: driver for loading UMBC files
  - rdairscoef: loads UMBC coefficient files
  - rdprof: loads UMBC standard profiles
  - writ_tr0: writes output in government format
  - abort: subroutine to gracefully exit on error

Utility Programs

These programs are located in the ~/src/src_gsfc library.

- tai_to_utc: convert day/month/year to # of seconds since 1/1/1970
- sunang2: compute the solar zenith angle
- comp_alt: compute altitude as a function of pressure
- set_solar: read or initialize the solar irradiance array
- sun: compute the Earth-Sun distance
- set_orbit: use the Earth-Sun distance to convert solar irradiance to radiance
- airsrad1: radiative transfer for clear scenes
- cldy_rad2: radiative transfer for cloudy scenes
- airsrad2: subroutine to support cldy_rad2.F

umbc9803 library

- Makefile
- rta_stat.com: common block for static variables
- rta_prof.com: common block for profile dependent variables
- load_ir.F: module to load RTA file
- init_rta.F: module to set-up all predictors
- init_trace.F module to set trace gas abundances (CO2, CH4, CO)
- setang_ir: module to compute secant angles
- ir_rta.F: module to compute total clear column layer-to-space transmittance and two-path SOLAR layer-to-space transmittance
  - calt1.F: AIRS RTA module
    - calokw.F
  - calt2.F: AIRS RTA module
  - calt3.F: AIRS RTA module
    - calokw.F
  - calt4.F : AIRS RTA module
  - calt5.F: AIRS RTA module
  - calt6.F: AIRS RTA module
  - calt7.F: AIRS RTA module
  - calt8.F: CrIS RTA module (in UMBC delivery: calt1.F)
  - calt9.F: CrIS RTA module (in UMBC delivery: calt2.F)
  - calt10.F: CrIS RTA module (in UMBC delivery: calt3.F)
  - calt11.F: CrIS RTA module (in UMBC delivery: calt4.F)
  - qikexp.F: quick EXP function used by calt's
- comp_down.F: module to compute down-welling radiance for levels other than the surface. Used for cloudy radiance computations.

diagnostic subroutine – not required for normal use

- ir_rta3.F: module to compute FIXED, WATER, and OZONE transmittances separately (diagnostic and printout uses only)

State variable definitions and units

The user is responsible for providing the state variables, in the proper units, to the radiative transfer programs. The arrays must be dimensioned to ≥ 100. All the state variables are real*4.

NOTE: A layer column density is the column density between Pobs(L-1) and Pobs(L) and are given in units of molecules/cm^2.

- Pobs(L): for L = 1, 100, are the lower boundary pressures of the 100 layer model. This variable is not used, but it is verified to be consistent with the RTA file.
- temp(L): are the temperatures at Pobs(L)
- wcd(L): are the water vapor layer column densities
- wlcd(L): are the liquid water layer column densities (not used by IR RTA)
- ocd(L): are the ozone column densities
- cocd(L): (optional for AIRS RTA) are the carbon monoxide layer column densities
- ch4cd(L): (optional for AIRS RTA) are the methane layer column densities
- co2per: (optional for AIRS RTA) is the % change in CO2 abundance (co2per=0.0 $\rightarrow$ 363 ppm)
- Psurf: is the surface pressure
- Tsurf: is the surface skin temperature
- pland: is the fraction land coverage, 0.0=all water (ocean), 1.0=land
- emiss(n): is the surface emissivity for channel n, at frequency, freq(n)
- rho_solar or reflect(n): is the surface bi-directional reflectivity for channel n. In the radiative transfer code this value is multiplied by the solar radiance. This value should be solved for during daytime conditions. The value is (1-emiss(n))/3.14 for Lambertian surfaces and (1-emiss(n)) for specular surfaces.
- rho_therm: is the surface thermal reflectivity for channel n. This value is multiplied by the thermal down-welling radiation. It can either be solved for or set equal to (1-emiss(n))/3.14.
- viewang: is the instrument angle. 0 is directly down, varies +/- 49.5 degrees for this simulation (AIRS pattern)
- satheight: is the satellite altitude in km
- alat: is the latitude (degrees) at the center of the footprint
- alon: is the longitude (degrees) at the center of the footprint
- solzang: is the solar zenith angle (degrees). 0 is overhead, 90 is at the horizon.
- AU: is the Earth-Sun distance in Astronomical Units, 1.0 AU is 1.496E8 km.

Computational efficiency issues

An integer*4 bitwise flag, called "comprta", has been implemented to improve the performance of the new RTA. This flag can be set to -1 ensure that everything is computed. If computational enhancements are desired, this flag can direct init_rta() and ir_rta() to NOT recompute all the predictors (init_rta()) or transmittances (ir_rta()). This is particularly useful when finite difference Jacobians are computed or when approximate Jacobians are desired. The bit parameters are defined below.

At the present time, only the SOLAR and down-welling bits are used. Computation of the trace gases is automatically linked to the calls of load_ir(), setang_ir() and init_trace() - so the individual bits for CO_2, CH_4, and CO are not used at this time. In the future, the other bits may be implemented for water and ozone. To ensure upward compatibility the user should assume that if a bit is not set, then the predictors and transmittance associated with that state variable may NOT be updated from a previous computation.

```
    bit.1      1   secant etc.
    bit.2      2   T(p) terms
    bit.3      4   q(p) terms
    bit.4      8   O3(p) terms
    bit.5     16   CO2 terms
    bit.6     32   CO terms
    bit.7     64   CH4 terms
    bit.8    128   spare
    bit.9    256   spare
    bit.10   512   spare
    bit.11  1024   spare
    bit.12  2048   spare
    bit.13  4096   spare
    bit.14  8192   SOLAR surface transmittance
    bit.15 16384   DOWN-WELLING terms
    bit.16 32768   SOLAR profile terms

  integer*4 crta_A, crta_T, crta_D, crta_S, crta_SP
  integer*4 crta_Q, crta_O3, crta_CO2, crta_CO, crta_CH4
  parameter (crta_A=1, crta_T=2, crta_Q=4, crta_O3=8)
  parameter (crta_CO2=16, crta_CO=32, crta_CH4=64)
  parameter (crta_S=8192, crta_D=16384, crta_SP=32768)

  integer*4 crta_all
  parameter (crta_all=32767)
```

For example, to compute all the predictors and transmittances necessary for a cloud free radiance (all levels at the viewing angle and the solar bidirectional transmittance at the surface) the value of comprta would be

```
comprta = crta_all
```

For cloudy radiances we need the bidirectional transmittance at the solar+spacecraft secant angle for all levels. In this case, the flag should be set to

```
comprta = crta_all + crta_SP
```

If a finite difference Jacobian is needed, then the second call (w/ perturbation) to init_rta() and ir_rta() could save time by NOT re-computing predictors and transmittances of the unperturbed state. For example, for water Jacobians,

```
comprta = crta_Q + crta_S  + crta_D
```

For long-wave channels, additional time could be saved by not computing the bidirectional transmittance

```
comprta = crta_Q  + crta_D
```

The Notional CrIS noise model

1. No error estimate for the rapid transmittance algorithm (RTA) and spectroscopy errors has been included at this time. In practice this error can be minimized with systematic corrections ("tuning") and will be smaller than measured spectroscopy errors and RTA fitting errors. After down-select we will begin to evaluate tuning algorithm approaches.

2. A minimum instrument noise has been added to ensure that the noise models do not become unrealistically good. We have used a value of 0.1 NEΔT as a "floor" to the noise model. In practice, one would compute the noise model, NEΔN(i) for each channel i and then test this noise model against this threshold. If the noise model is in NEΔN units,

$$\text{threshold}(i) = 0.1 \cdot \left. \frac{\partial B(\nu(i))}{\partial T} \right|_{T=250} \tag{3.1}$$

$$\text{if} \quad \text{NE}\Delta\text{N}(i) < \text{threshold}(i) \quad \text{then} \quad \text{NE}\Delta\text{N}(i) = \text{threshold}(i) \tag{3.2}$$

   This feature can be disables be editing the "NEDT minimum" line within the noise file.

3. The vendor noise model file should be in the format of the noise model delivered in this package. We compute the instrument noise at a number of frequencies, f, and scene temperatures, T_s. Each scene temperature has a band integrated radiance, $R(b, Ts)$ for band=b in units of photons/s/cm^2/steradian. We then fit the NEΔN(f, b, Ts) to a polynomial. These values are used as input to the radiance simulated code and the government's retrieval algorithms.

   In the simulator program and the government's retrieval algorithm, the noise model is loaded with the routine ~/src/src_gsfc/load_noise.F. This program interprets the comments within the file, so the vendor's should ensure that all lines within the file are field formatted in compliance with the load_noise.F routine. The routine ~/src/src_gsfc/calc_phi.F computes band integrated radiance, $\phi(b)$, (given in photons/cm^2/steradian) from the simulated radiances (*i.e.*, observations, given in milli-Watts/cm^2/ steradian/cm^{-1}). Once the integrated radiance is known for each band then the noise model can be computed using a call to ~/src/src_gsfc/calc_nedn.F. See the program ~/src/simrad/addnoise.F for an example on how was done for the radiances computed within this dataset.

   The polynomial is used to compute NEΔN(i) from the coefficients data file and the scaled value of $\phi'(b) = \phi(n)/\text{scale}(b)$, where scale$(b)$ is read from the file.

$$\text{NE}\Delta\text{N}(i) = a_0(i) + a_1(i) \cdot \phi'(b) + a_2(i) \cdot \phi'(b)^2 \tag{3.3}$$

   and can be converted to NEΔT using the derivative of the Planck function.

   The data file for the model used in the enclosed radiance simulations is shown below. The Lincoln Laboratories model assumes a nominal CrIS instrument with an effective aperture of 4 cm and optical path difference, L, of 0.8, 0.4, and 0.2 cm for bands 1,2,and 3, respectively.

```
--------------- file = its_LL2.dat ---------------
     31     3
     620.0  2.61E-01 -2.49E-07  6.44E-07
     630.0  1.97E-01 -1.85E-06  4.26E-06
     640.0  1.84E-01 -1.12E-06  2.53E-06
     650.0  1.79E-01  2.75E-07 -3.43E-07
     660.0  1.76E-01  3.27E-07 -4.97E-07
     680.0  1.74E-01  1.62E-06 -3.72E-06
     700.0  1.75E-01 -6.69E-07  9.70E-07
     800.0  1.90E-01  3.41E-06 -7.21E-06
     900.0  2.13E-01  4.32E-07 -1.57E-06
```

```
     1000.0  2.40E-01 -1.77E-06  4.10E-06
     1095.0  2.70E-01  3.85E-06 -7.78E-06
     1205.0  3.44E-02 -1.91E-07  2.67E-07
     1210.0  3.45E-02  6.32E-08 -7.13E-08
     1290.0  3.54E-02 -2.07E-07  2.87E-07
     1370.0  3.68E-02  1.96E-07 -2.70E-07
     1450.0  3.86E-02 -9.87E-09  2.09E-08
     1540.0  4.08E-02 -1.00E-07  1.53E-07
     1620.0  4.32E-02  3.57E-09 -8.24E-09
     1710.0  4.60E-02 -8.39E-08  1.34E-07
     1775.0  4.81E-02 -3.53E-08  5.52E-08
     1805.0  4.93E-02 -2.30E-07  3.08E-07
     2104.0  5.74E-03  3.02E-09 -4.40E-08
     2155.0  5.81E-03 -7.37E-08  3.16E-07
     2225.0  5.96E-03 -1.86E-08  7.96E-08
     2300.0  6.17E-03  4.84E-08 -2.03E-07
     2375.0  6.36E-03  2.95E-09 -1.07E-08
     2450.0  6.60E-03 -1.32E-08  5.75E-08
     2525.0  6.84E-03  6.21E-08 -2.26E-07
     2600.0  7.08E-03 -1.51E-08  4.36E-08
     2700.0  7.42E-03 -8.03E-09  6.55E-08
     2800.0  7.78E-03 -5.21E-09  4.41E-08
   7
NEDN for ITS LL notional design w/ 90 K detector, D* and scattering improved
ITS  NEDN(i) = x0(i) + x1(i)*phi(band) + x2(i)*(phi(band)**2)
NEDT minimum = 0.100 K
NEDN(n) = interpolated/extrapolated from NEDN(i)
NEDT250(n) = NEDN(n)/dBdT(f(n),T=250)
phi(band) = integral(R*dW)/SCALE from Wlow to Whgh
 ib1 ib2  Wlow   Whgh      L    SCALE
   3   5
   0 11   630.0  1095.0 0.8000 1.0E18
  11 21  1210.0  1540.0 0.4000 1.0E17
  21 31  2155.0  2450.0 0.2000 1.0E16
2
determine NEDN from Lincoln Laboratories model, given the integrated
flux, phi(iband) and data from LL, Aug. 24, 1998
```

The unapodized noisy radiances are calculated as

$$R(n,i) = R_0(n,i) + r(n,i,k) \cdot \mathrm{NE\Delta N}(n, \vec{R}_0(i)) \tag{3.4}$$

A random number generator is used to provide the value of $r(n,i,k)$ for channel n, field-of-view i, and case k with a Gaussian distribution, which has a statistical average value of zero and RMS value of one for many cases.

$$\sum_{k=1}^{K} r(n,i,k) \rightarrow 0 \quad \text{as} \quad K \rightarrow \infty \tag{3.5}$$

$$\frac{1}{K}\sum_{k=1}^{K} r(n,i,k)^2 \rightarrow 1 \quad \text{as} \quad K \rightarrow \infty \tag{3.6}$$

Microwave RTA

The microwave RTA is provided in the directory ~/mwrta_v4. It has a similar calling sequence (the solar and UMBC down-welling model are not used) as the new IR RTA. The coefficients are loaded with open_mw.F and read_mw.F and the coefficient files are located in ~/RTA.

The methodology utilized within the code is documented in

- Rosenkranz, P.W. 1995. "A rapid atmospheric transmittance algorithm for microwave sounding frequencies." *IEEE Trans. Geosci. Remote Sens.* **33**, p.1135-1140.
- Rosenkranz, P.W. 1998. "Improved Rapid Transmittance Algorithm for Microwave Sounding Channels." presented at *International Geoscience and Remote Sensing Symposium* (IGARSS'98), 6-19 July 1998, Seattle, WA.

For the AMSU (channels 1 to 15) and MHS (channels 16 to 20) radiances, the radiance is given in temperature units, and the noise is given in NEΔT units. We used the instrument specification values for the noise, as given below, to generate noisy AMSU radiances. Microwave channels for the IPO cross-track microwave sounder still need to be defined; however, the channels provided in this simulated radiance dataset should provide a reasonable set of channels for sounding.

Table 7: AMSU-A, -B frequencies and noise estimates

channel	center frequencies (GHz)	NEΔT
1	23.8 +/- 0.0725	0.30
2	31.4 +/- 0.050	0.30
3	50.3 +/- 0.050	0.40
4	52.8 +/- 0.105	0.25
5	53.596 +/- 0.115	0.25
6	54.40 +/- 0.105	0.25
7	54.94 +/- 0.105	0.25
8	55.50 +/- 0.0875	0.25
9	ν_1 +/- 0.0875	0.25
	$\nu_1 = 57.290344$	
10	ν_1 +/- 0.217	0.40
11	ν_1 +/- 0.322139 +/- 0.048	0.40
12	ν_1 +/- 0.322139 +/- 0.022	0.60
13	ν_1 +/- 0.322139 +/- 0.010	0.80
14	ν_1 +/- 0.322139 +/- 0.045	1.20
15	89.0 +/- 1.0	0.50
16	89.0 +/- 0.9	0.60
17	157.0	0.60
18	183.311 +/- 1.00	0.80
19	183.311 +/- 3.00	0.80
20	190.311	0.80

How to use the government software

The government has provided a number of routines for use in simulating radiances and producing statistics files

<u>compiling the library</u>

```
cd ~/src/src_gsfc    library of useful utilities and radiative transfer code
make

cd ~/src/rta66       old RTA with new calling sequences
make

cd ~/src/umbc9803    UMBC RTA w/ government ``wrapper''
make

cd ~/src/mwrta_v4    new microwave RTA
make

cd ~/src/src_jpl     library of I/O and primitive functions
make

    NOTE 1: when compiling gauss.F the HPUX function DRAND48 is
            system specific and may require changes in compiler
            directives and function name(s).

cd ~/src/simrad      subroutines for simrad.F
make

cd ~/src/main
make comp_rad        program to validate radiance computation
make simstat         program to compute statistics

    NOTE 2: when compiling comp_rad and simstat two HPUX commands
            are used.  The function called iargc() fetches the
            number of command line arguments.  The subroutine
            igetarg() fetches a specific character argument.  These
            can be replaced with new system dependent calls or this
            section of code can be replaced with a small file and
            read statements.

    NOTE 3: The path strings (``path'' in comp_rad and ``path_tru''
            in simstat) should be edited for the vendor's system.
            Edit the strings and then change the character statement
            to the number of characters in the string.  The examples,
            as shown, should work without this modification.

make simrad103       program to compute cloudy radiance with noise, UMBC RTA

make comp_rad66      program to validate re-issue of old RTA model
make simrad66        program to compute cloudy radiance with noise, old RTA
```

using comp_rad

The comp_rad.F program has been provided to illustrate the use of the microwave and infrared rapid transmittance algorithm and government utility routines to simulate radiances. This program was used to produce noise free clear radiances for the US standard atmosphere dataset provided in the ~/level2/us_std/ directory and can also be used to compute cloudy radiances with the -cloud option.

Compile this program and run the following steps. The radiance file it produces can be compared to the file stored in ~/doc/, as follows:

```
cd ~/
src/comp_rad.exe level2/us_std/usstd100.asc rad.std -cris
diff rad.std doc/rad.std

NOTE: to test the re-issued 66-layer RTA the files
         HAMM48_v1.bi needs to
         ascii.trcoef.amsu.v1a
         ascii.trcoef.mhs.v1a
      were provided from the FY1997 IGS study's CDROM in ~/RTA/

src/comp_rad66.exe level2/us_std/ usstd066.asc rad66.std -cris iphys=2
diff rad66.std doc/rad66.std
```

The difference between the vendor's ASCII file and the file in ~/doc should not be significant. There are; however, significant differences between the 66 and 100 layer RTA's, as shown in Figure 5.

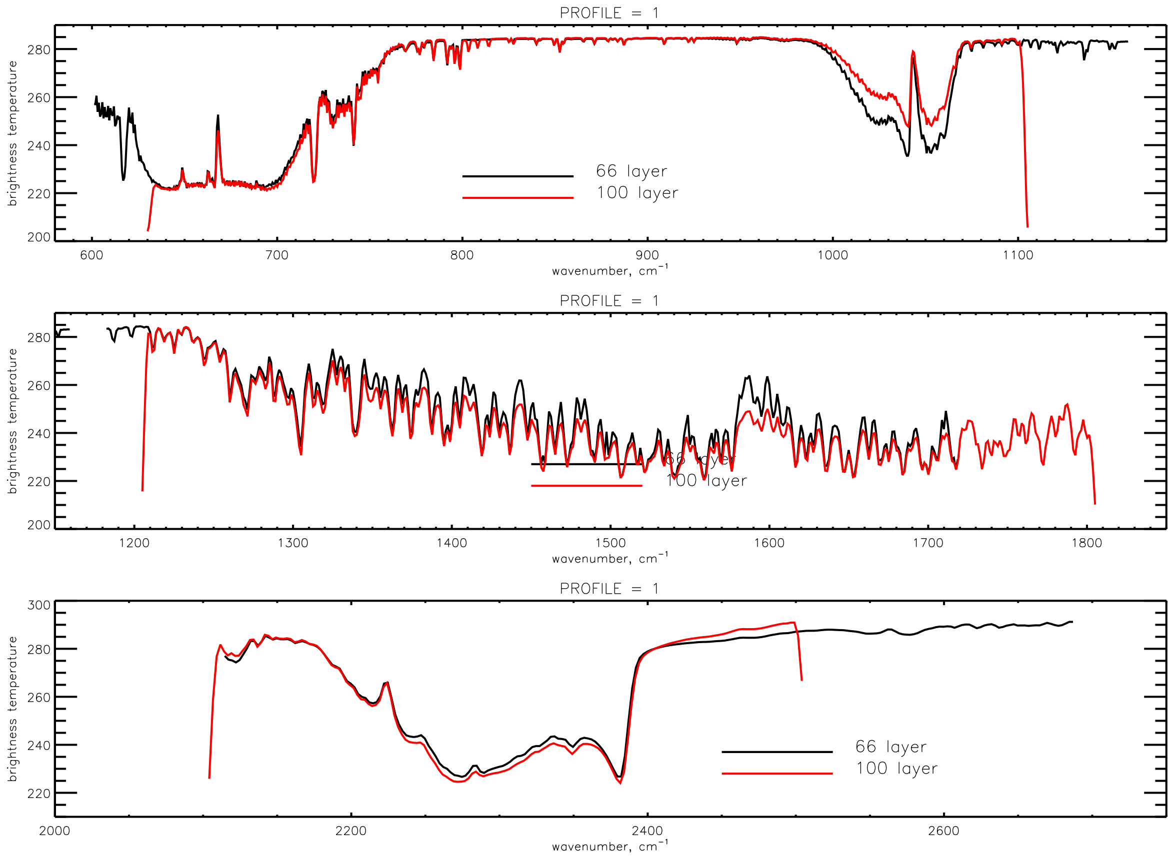

Figure 5: Comparison of 66 layer and 100 layer (UMBC) CrIS RTA's for the nominal moisture US Standard profile.

using the simstat program

The simstat program is a stand alone statistics program which reads the specified truth and retrieval files and writes a file of statistics. See the top of the program file (~/src/main/simstat.F) for command line format and options. This software has been provided

1. to document, in explicit detail, how the government computes statistics,
2. to provide a consistent statistical product which can be used to objectively compare an algorithms ability on these scans,
3. to provide a tool, which can be used to evaluate the EDR thresholds for temperature and moisture for the small ensembles presented here.

At this time, we are only addressing items #1 and #2. The statistics generated by simstat.F should only be used for relative comparisons and not absolute EDR thresholds. These scan lines are small ensembles and may not be representative of the EDR thresholds for a global ensemble. A 1998 IGS interferometer study, to be completed by Dec. 31, 1998, will publish the governments ability with these sixteen scan lines, focusing on the 5 AMSU scan lines provided (#4, #8, #11, #14, and #11c). In addition, the government will specify, at that time, how these profiles should be weighted to compute statistics relevant for the absolute interpretation with respect to the IPO thresholds.

We have provided an example retrieval for scan line #4. This retrieval is only an example for the program and **does not represent the governments ability** for this profile. The retrieval file, ~/doc/sc04.ret, and weight files, ~/weight/wgt_clr.04 and wgt_cld.04, were used to produce the results in files ~/doc/sc04_clr.stat and sc04_cld.stat, respectively. The vendor can test the compilation of their copy of simstat.F as follows:

```
cd ~/
src/simstat.exe level2/TR.04 doc/sc04.ret sc04_clr.stat -wgt weight/wgt_clr.04
diff sc04_clr.stat doc/sc04_clr.stat

src/simstat.exe level2/TR.04 doc/sc04.ret sc04_cld.stat -wgt weight/wgt_cld.04
diff sc04_cld.stat doc/sc04_cld.stat
```

We have provided weight files for all 16 scan lines. The weight files simply segregate the profiles on this basis of the "clear" threshold, in which truth AMSU cloud fraction is $\leq$ 50"cloudy" threshold, in which the truth AMSU cloud fraction is $>$ 50 %. In addition, the weight has a COSINE(latitude) dependence, which may be used in the future to weight one scan line relative to the other. A function along these lines may be useful to account for effects of overlapping orbits.

At this time, the vendors should run their retrieval algorithm on the five cases provided (#4, #8, #11, #14, and #11c) and return both the level.2 file and the output from simstat.F to the IPO. The vendors are free to alter the weight file in any manner they see fit, to produce the best statistic for both "clear" and "cloudy" ensembles. The vendor is also free to remove certain profiles from both "clear" and "cloudy", based on objective rejection criteria,if desired. The government will utilize the retrieval and simstat output files, provided by the vendors, to verify that the vendors have utilized the proper file formats to communicate their results. **To be useful with regard to the mid-November blind test, these files should be returned to the government by the end of October. The government recognizes that vendor algorithm trade studies are not complete and that these files are not representative of the vendor's ability on these scan lines, nor do they represent capabilities of the vendor's instrumental design**

using simrad

The program ~/main/simrad.F will produce radiances which are co-located with the footprints within an AMSU scan line. All the namelist options for this program are illustrated for the 5 cases in the directory "~/namelist/". Each scan line has its own namelist for the simulation variables (sim_sc04_cris.nl, etc.) and I/O variables (io_sc04_cris.nl, etc.). The input "truth" files are given in the ~/level2/ directory and the output is written into the ~/level1 directory (this includes the echo of the level.2 file). To run this program the output files must be renamed to avoid overwriting the existing files (**simrad** will print and error message if the output file exists).

For example, to regenerate the AMSU scan line # 4 the UNIX commands would be:

```
cd ~/level1
mv sc04_ir.asc sc04_ir.0
mv sc04_am.asc sc04_am.0
mv sc04_mh.asc sc04_mh.0
mv sc04_tr.asc sc04_tr.0

cd ~/namelist
run_gen sc04
```

simrad is a complicated program with many options used by the government. It is also upward and backward compatible with previous runs made by the government so that many of the variables are of historical importance only. The I/O namelist is also used be the governments retrieval program, so there are also options for that purpose which are ignored by the **simrad**.

We suggest that the program "comp_rad.F" be used to understand the calling sequences of the government software and, possibly, as a seed program for vendor applications. The **simrad** program can be used to generate radiances for the other AMSU scan lines with simple modification of the namelists. For example, the vendor can difference two of the existing scan line files to see the namelist variables which must be altered.

Also, the namelists can be altered easily to produce radiances for the training dataset with noise (iomode=4, addnoise=T) and without noise (iomode=4, addnoise=F). Clear conditions can be simulated with iomode=5 and utilizing the variables for pcld_top, ems_cld, rho_cld.

The vendor can review the namelists used to generate the radiances delivered with the original NOAA88 dataset and produce cloud-free or cloudy radiances with the new NOAA88b dataset. This is a more complicated task and is beyond the scope of this document to discuss. Some issues are

- Generation of simulated co-located footprints requires adding clouds to the level.2 file. In the 1997 IGS study this was accomplished by producing a file with 9 spots per profile used from the NOAA88 dataset. Each spot had its own clouds, specified by pcldtop and cldfrc, but the atmospheric state was the same. The clouds from the winter and summer ensembles, issued with the 1997 IGS study CDROM, could be merged with the existing profiles from the new NOAA88b dataset to produce new 100 layer ensembles.
- the new UMBC RTA does not have extra channels at the band edges and, therefore, the **simrad** method used to produce un-apodized radiances is not recommended. If un-apodized radiances are desired, we suggest producing Hamming apodized noisy radiances and then, using the method discussed on page 8, transform them to un-apodized radiances.
- some namelists variables may have changed since the 1997 delivery.

The namelist options for the **simrad** program are:

```
=====================================
simunl: simulation namelist variables
=====================================

      iomode      = processing mode

    ..................... iomode definitions .......................

     1   creates numFOV radiances with clouds from namelist parameters
         each radiance is an average of spots.  For example, if
           numFOV=1 then 9-spots are averaged
           numFOV=2 then 4 spots are averaged
           numFOV=3 then 3 spots are averaged
           numFOV=9 then NO spots are averaged
         cloud spectral properties are set via namelist with EMS_CLD(iCLD)
             and RHO_CLD(iCLD)

         unif=F, cloudgen=T, anfac = 1.0/SQRT(numFOV)
         default randtype=1

     2   creates 9-spot fields from numFOV radiances, the radiances
         are duplicated within a FOV so that
           numFOV=1 all 9-spots are identical
           numFOV=2 then 4 spots are averaged
           numFOV=3 then 3 spots are averaged
           numFOV=9 then NO spots are averaged
         cloud spectral properties are set via namelist with EMS_CLD(iCLD)
             and RHO_CLD(iCLD)

         unif=F, cloudgen=T, anfac = 1.0/SQRT(numFOV)
         default randtype=1

     3   Spots are grouped by 9's.  Uses level.2 file, spot #5, to
         generate all aspects of radiances except cloud fraction and
         cloud liquid water which is derived from the individual spots
         cloud fraction.  The cloud liquid water is scaled by
         cldfrc(i)/cldfrc(5).

         unif=T, cloudgen=F, anfac=1.0, numFOV=1, numpro=nobs/9, numpro2=9
         default randtype=2
         in writ_output unif=T will write results as if numFOV=9

     4   level.2 file is used for all parameters, 1 spot at a time

         unif=F, numFOV=1, cloudgen=F, anfac=1.0
         default randtype=2

     5   level.2 file is used for all parameters except cloud parameters,
         which are taken from the namelist.  Cloud fraction and cloud
         height are generated via namelist parameters

         cloud spectral properties are set via namelist with EMS_CLD(iCLD)
```

```
   and RHO_CLD(iCLD)

   unif=F, numFOV=1, cloudgen=T, anfac=1.0
   default randtype=2

6  Spots are grouped by 9's.  Uses level.2 file, spot #5, to generate all
   aspects of radiances except cloud fraction and cloud liquid water
   The cloud fractions and cloud height are set according to the
   namelist, which should set values for all 9 spots (error occurs if
   the values are not set).

   The cloud liquid water is scaled by cldfrc(i)/cldfrc(5).

   unif=T, cloudgen=T, anfac=1.0, numFOV=1, numpro=nobs/9, numpro2=9
   default randtype=2
   in writ_output unif=T will write results as if numFOV=9

7  same as iomode=4 except AMSU is computed from average of 9 spots.
   The AMSU radiances are computed and averaged for the 9 spots.
   Only one AMSU is reported for 9 AIRS/MHS spots.

   level.2 file is used for all parameters, 1 spot at a time

   unif=F, numFOV=1, cloudgen=F, anfac=1.0
   default randtype=2

 ......................................................................

 trufile_in  = truth file name (character*80)
     simrad uses unit=logl2_out for L1 input, default=34)

 dosun      = T or F   sets SUN at nadir or horizon if forceSUN = T


 forceSUN  = T pays attention to DOSUN, cossun=1.0
             F used data+time to compute solzang, cossun
               and sets DOSUN=F if solzang > 89.0

 nadirflg = T  sets secants == 1


 satheight_set satheight in km's.
               default = 705km for INSTRUM(1)=0 and 824km otherwise

 viewang_set   viewang in degrees, default is variable (0,15,30,45,...)

 viewunit    = 38,  unit # for viewfile

 viewfile    = name of view angle and satellite height file.  This will
               override the variables satheight_set and viewang_set.


 comment     = comment for this run (character*24)
```

```
iprt        = printout level(s)

instrum     = instrument specification
     (1) = instrument type  0=AIRS,1=HIRS,2=ITS-apodized,3=ITS-unapodized
     (2) = noise model:     0=AIRS,1=HIRS,2=ITS-apodized,3=ITS-unapodized
     (3) = correlation flag
     (4) = noise mode sub-index

       e.g., AIRS: INSTRUM = 0, 0, 0, 1
             CrIS: INSTRUM = 2, 2, 2, 0

mhsnoise    0 = old set [0.5,0.5,0.5,0.5,0.5]
            1 = new set [0.6,0.6,0.8,0.8,0.8]

addnoise    T to add noise to radiances

randtype:   1=gasdev(), 2=gauss()
            default:  1 if iomode < 3, 2 if iomode >= 3

Nchlset    = number of AIRS channels in radiance output ,  -1 = all
freqchlset = frequency list (Nchlset > 0), any order

doamsu      default=T
domhs       default=T
unit_emismw default=F
unit_emisir default=F
oldemis     default=F for reading surface emissivity from file,
                    T = use level.2 file values

ntotfov    = total number of FOV's  (not used)

scalenoise = scale factor for AIRS noise
scaleMHS   = scale factor for MHS  noise

numFOV     = number of FOV's
numCLD     = number of clouds, default = numFOV-1 or 2 if numFOV=9


cloudtable = flag for generation of cloud sensitivity table,
     cloudtable >= 1 will write output to unit=89
     cloudtable <= 0 will NOT generate cloud table

npro_kernal = profile ID to generate kernal functions, ftn97
              -1 = none    use pl_kern.pro   or   pl_kern2.pro

wrprotect = T will call abort() if radiance (L1CLFILE_IN,
             L1FILE_IN, L1AMFILE_IN, L1MHFILE_IN) or truth
            (L2FILE_IN) file already exists.  F will allow
            overwriting these files.

l2ver_out  = version # for L2 output file, default = same as tru_file

    iver   code   description
      1    L2V1B  original format, has 1 frequency scale for
```

```
                cloud spectral properties
     2    L2P1C  upgrade to independent cloud spectral properties,
                 real*4 seconds, but L2P1C was an incorrect label.
                 This type == 3
     3    L2V1C  upgrade to independent cloud spectral properties,
                 real*4 seconds,
     4    M2V1C  same as iver=3, except water, ozone, and liq.water


if iomode sets cloudgen=T, then these variables specify the clouds
------------------------------------------------------------------

   pcld_top   = value for each cloud level, ncld = 1,numCLD
   ems_cld    = value for each cloud level, ncld = 1,numCLD
   rho_cld    = value for each cloud level, ncld = 1,numCLD
   alpha      = value for each FOV, of each cloud level, ncld = 1,numCLD

   randomcld   = F
   pcld_sdv    = 250.0
             pcld(iCLD) = pcld_top(iCLD) + pcld_sdv*2(RAN1-0.5)
   alpha_sdv   =   0.1
           alpha(iCLD,iFOV) = alpha(iCLD,iFOV) + alpha_sdv*2(RAN1-0.5)
```

```
==============================
ioairs: I/O namelist variables
==============================

   items needed for simrad
   -----------------------

      numpro =  number of profiles in output file,   -1 = all

      idpro      = profile list, if numpro > 0

      iphys       = physics model
             0 = 64 level model w/ variable emissivity in mw (NOT USED)
             1 = 64 level model w/  fixed   emissivity in mw, Psurf=1000
             2 = 66 level model for Science Team cloud studied
             3 = 66 level model w/ upgrades
                    a) fixed rho/pi problem
                    b) MW background
             4 = 100 level (12/95 version)
             5 = 100 level ( 3/98 version)

      noisefile   = noise file, ../noise/
         its_LL2.dat  new (8/99) Lincoln Lab's noise model

      solarfile   = solar irradiance file, ../solar/
         airs_solar_data.txt  for AIRS UMBC RTA (iphys=5)
         cris_solar_data.txt  for CrIS UMBC RTA (iphys=5)

      tc_airs     = transmittance coefficients for AIRS
          AIRS48_v1.asc 11/25/97 48 UMBC,  AIRS Gaussian-1
          HAMM48_v1.bi   9/26/97 L=0.80, 48 UMBC, extended chl's
          HAMM48_L12_v1.asc 11/25/97 L=1.20, 48 UMBC, extended chl's
          HAMM48_L12_v1.bi  11/25/97 L=1.20, 48 UMBC, extended chl's
          binary.trcoef.cris.v3a  UMBC 3/98 delivery

      tc_amsu     = transmittance coefficients for AMSU
           ascii.trcoef.amsu.v1a  f/iphys=2
           ascii.trcoef.amsu.v4-  f/iphys >= 4

      tc_mhs      = transmittance coefficients for MHS
           ascii.trcoef.mhs.v1a   f/iphys=2
           ascii.trcoef.hsb.v4-   f/iphys >= 4


      logl1_in    = 31
      l1file_in   = IR level.1 filename

      logl1am_in  = 32
      l1amfile_in = AMSU level.1 filename

      logl1mh_in  = 33
      l1mhfile_in = MHS level.1 filename

      logl1cl_in  = 30
```

```
  l1clfile_in = clear radiance level.1 filename (f/ simrad, NOT USED)

  logl2_in    = 25 = unit for L2 write/read
  l2file_in   = truth file (echoed by simrad, used by retrieval)


items included in I/O file for retrieval, but are NOT needed for simrad
-----------------------------------------------------------------------

  NPROTEST = 0,     # of profiles to do detailed printout
  idprotest = 0, 0, profile ID numbers (last 4 digits of idprof)
                    for detailed printout

  readlev1    = T  (NOT an option anymore)
  readlev2    = T  (NOT an option anymore)

  cli_file    = climatology filename
  regfile     = regression filename

  logl2fg_in  = 26
  l2fgfile_in = name of first guess input file (NOT USED f/ iphys=1)

  logmwfg_in  = 27
  l2mwfile_in = name of microwave only first guess input file

  l2error_unit = 36
  l2error_in   = filename for null and error estimate input (for retrieval)
  l2error_out  = filename for error estimate product (for retrieval)
  l2error_rms  = filename for null and error estimate output (for retrieval)

  noaaregeig   = NOAA IR regression radiance eigenvector file
  noaaregreg   = NOAA IR regression regression coefficient file
  noaaregfrq   = NOAA IR regression frequency file

  writeguess  = T if first guess retrieval (after AMSU #2) is desired
  logl2fg_out = 35
  l2fgfile_out = name of fg retrieval level.2 output file

  writelev2   = T if level.2 final output file is desired
  logl2_out   = 34 = unit for L2 output (simrad uses for L1 output)
  l2file_out  = name of level.2 retrieval output file

  logl1fg_ccr  = first guess clear column radiance unit #
  l1fgfile_ccr = first guess clear column radiance filename

  logl1_ccr    = final clear column radiance unit #
  l1file_ccr   = final clear column radiance filename
```

Appendix A: Conversion between Mixing Ratio and Column Density

This document describes the process of converting level.2 files from layer quantities (layer column densities) to level quantities (level mass mixing ratios). **We have borrowed heavily from concepts discussed in the UMBC documentation (specifically a document called "sci.txt" in the AIRS_LAYERS program) and we would like to acknowledge their contribution which has been imbedded within this document.** The level.2 level quantities will replace the original layer quantities and thus become the "truth" files. Any assumptions or approximations in converting layer quantities to level quantities will, therefore, become irrelevant.

Internally, to the code, the rapid transmittance algorithm (RTA) requires the layer quantities. This document also describes the methodology for converting the level quantities to layer quantities for the RTA.

The level quantities are the temperature, $T(L)$, in degree Kelvin at fixed pressure levels, $p(L)$, where L is an index for the level. The moisture (both vapor and liquid) and ozone are given in terms of mass mixing ratio, $q(L)$, which is the density of water molecules per density of dry air. The mixing ratio for species x (x is equal to water vapor, w, liquid water, l, or ozone, o) is given in dimension-less units (*e.g.*, gm/gm).

The total column density for species x, $C_x(p(L))$ is the total number of molecules/cm^2 from to top of the atmosphere to level $p(L)$. We will define an effective pressure level $p_{eff}(L)$ which can be used to define the total column density to the level of $p_{eff}(L)$, $C_x(p_{eff}(L))$. We will use a shorthand for the the layer column densities. For layer column densities between two pressure levels $p(L-1)$ and $p(L)$ we will write

$$\Delta C_x(p(L)) \equiv C_x(p(L)) - C_x(p(L-1)) \tag{A.7}$$

and between the two levels $p_{eff}(L-1)$ and $p_{eff}(L)$ we will write

$$\Delta C_x(p_{eff}(L)) \equiv C_x(p_{eff}(L)) - C_x(p_{eff}(L-1)) \tag{A.8}$$

The mixing ratio will only be defined at $p(L)$ and can be related to column densities as

$$q_x(p(L)) \equiv \frac{\mathrm{mw}_x \cdot \rho_x(p(L))}{\mathrm{mw}_d \cdot \rho_d(p(L))} = \frac{\mathrm{mw}_x \cdot \Delta C_x(p_{eff}(L))}{\mathrm{mw}_d \cdot \Delta C_d(p_{eff}(L))} \tag{A.9}$$

where the dry gas layer column density, $\Delta C_d(p_{eff}(L))$, is discussed later.

In Table A.1 and Table A.2 the definition of the 66 and 100 layer pressure systems are given in detail. To illustrate the relationship between $p(L)$ and $p_{eff}(L)$, an example of the levels is shown in Figure 6.

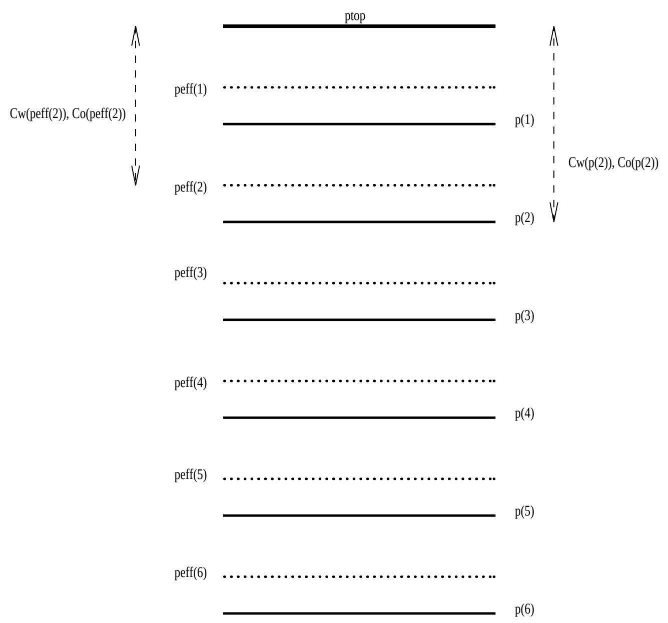

Figure 6: Example of 6 level system

Background information and Definition of terms

The relationship between distance, z, and pressure is given to a high degree of accuracy by the hydrostatic equilibrium equation,

$$\partial p = -\rho \cdot g(p) \cdot \partial z \tag{A.10}$$

where ρ is the total density in g/cm^3, and g is the acceleration of gravity which decreases with altitude and varies with latitude. The value of g is $\approx$ 1% lower at p =10 mb (31 km) and $\approx$ 2% lower at p =1 mb (48 km) compared to the surface. The variation with latitude is less than $\approx$ 0.5% and wind acceleration can change the effective value of $g \leq$ 0.2%. In the current government software (iphys=5) the latitude and altitude dependence of g is modeled by the UMBC routine ~/src/src_gsfc/grav.F.

The ideal gas law is the relationship between pressure and the atmospheric state variables of density and temperature

$$p = \rho \cdot R^{*} \cdot T/\mathrm{mw}_t \tag{A.11}$$

where R^* is the universal gas constant (8.31432 $\cdot 10^7$ erg/mole/K) and mw_t is the average molecular weight of the total (dry + water + ozone) composition of the layer. The ideal gas law is a reasonable representation of the equation of state of air. The computation of the average molecular weight will be discussed later. The thickness of the layer can be written in terms of the hydrostatic equilibrium equation as

$$\Delta z(L) = \int_{p(L)}^{p(L-1)} \frac{\partial p}{\rho(p) \cdot g(p)} \tag{A.12}$$

substituting the ideal gas law for $\rho(p)$ we have

$$\Delta z(L) = \int_{p(L)}^{p(L-1)} \frac{R^{*} \cdot T(p)}{\mathrm{mw}_t(p) \cdot g(p)} \cdot \frac{\partial p}{p} \tag{A.13}$$

For convenience we can define the atmospheric scale height, which is given by

$$H(p) \equiv \frac{R^{*} \cdot T(p)}{\mathrm{mw}_t(p) \cdot g(p)} \tag{A.14}$$

and substituting Eq. A.14 into Eq. A.13 yields the thickness of a layer

$$\Delta z(L) = \int_{p(L)}^{p(L-1)} H(p) \cdot \frac{\partial p}{p} \tag{A.15}$$

If the layer is thin, then we can assume that the scale height is a constant over the layer, $H(p) \simeq H(p_{eff}(L)) \equiv H(L)$, where p_{eff} is a pressure level within the layer. The total thickness of the layer can then be written as

$$\Delta z(L) \simeq H(L) \cdot \log_e\left(\frac{p(L-1)}{p(L)}\right) \tag{A.16}$$

and if $\Delta p(L) \ll p(L)$ then

$$\Delta z(L) \approx H(L) \cdot \frac{\Delta p(L)}{p(L)} \tag{A.17}$$

since the logarithm can be approximated by the first two terms of the expansion

$$\log_e(1+\epsilon) = 1 + \epsilon - \frac{1}{2}\epsilon^2 + \frac{1}{3}\epsilon^3 - \frac{1}{4}\epsilon^4 + \ldots \approx 1 + \epsilon \qquad \mathrm{for}|\epsilon| \ll 1 \tag{A.18}$$

The thin layer approximation allows a simple expression for the height associated with any pressure within the layer

$$z(p) = z(L) - H(L) \cdot \log_e \left(\frac{p}{p(L)} \right) \tag{A.19}$$

or equivalently find the pressure level corresponding to any height

$$p(z) = p(L) \cdot e^{-\left(\frac{z - z(L)}{H(L)}\right)} \tag{A.20}$$

The average pressure within a layer, $\overline{p}$ can be written in terms of $p(z)$ given in Eq. A.20, which is equal to

$$\overline{p}(L) \equiv \frac{\int_{z(L)}^{z(L-1)} p(z) \cdot dz}{\int_{z(L)}^{z(L-1)} dz} = \frac{-H(L) \cdot (p(L-1) - p(L))}{z(L-1) - z(L)} = \frac{\Delta p(L)}{\log_e \left(\frac{p(L)}{p(L-1)} \right)} \tag{A.21}$$

where $\Delta p(L) \equiv p(L) - p(L-1)$ and noting that

$$p(L) \int_{z_1}^{z_2} e^{-\left(\frac{z - z(L)}{H(L)}\right)} \cdot dz = -H(L) \cdot (p(z_2) - p(z_1)) \tag{A.22}$$

Therefore, the average pressure can be computed directly from the layer boundary levels. We define the effective pressure as the layer mean value, $p_{eff}(L) = \overline{p(L)}$. The value of $z(p_{eff}(L))$ is then given by Eq. A.19.

$$p_{eff}(L) \equiv \overline{p}(L) = \frac{\Delta p(L)}{\log_e \left(\frac{p(L)}{p(L-1)} \right)} \tag{A.23}$$

The average molecular weight of the layer

The value of $\text{mw}_t(L)$ is determined from number conservation and mass conservation within the layer. The total number of molecules must equal the sum of all molecules within the layer (sum of partial pressures or Dalton's Law). Therefore, at any given step within a retrieval the layer column density, $\Delta C_t(L)$, is equal to the layer column density of dry gases (all gases except water and ozone), $\Delta C_d(L)$, plus the layer column density of water vapor and ozone, $\Delta C_w(L) + \Delta C_o(L)$.

$$\Delta C_t(L) = \Delta C_d(L) + \Delta C_w(L) + \Delta C_o(L) \tag{A.24}$$

The mass within a layer must also be conserved. The total mass per cm^2 is given by solving Eq. A.10 for $\Delta p(L)/g(p)$. The total mass is related to the sum of the individual masses of the atmospheric composition.

$$\frac{\Delta p(L)}{g} \equiv \int_{p(L)}^{p(L-1)} \frac{\partial p}{g(L)} = \int_{z(L)}^{z(L-1)} \rho(z) \cdot \partial z \tag{A.25}$$

But the layer integral of density is the definition of layer column density

$$\int_{z(L)}^{z(L-1)} \rho(z) \cdot \partial z \equiv \frac{\text{mw}_t(L) \cdot \Delta C_t(p(L))}{N_a} \tag{A.26}$$

The total layer column density is related to the sum of the individual layer column densities via Eq. A.24, therefore, the pressure limits are proportional to the sum of the layer layer column densities.

$$\frac{\Delta p(L)}{g(p)} = \frac{1}{N_a} \cdot \left[\text{mw}_d \cdot \Delta C_d(p(L)) + \text{mw}_w \cdot \Delta C_w(p(L)) + \text{mw}_o \cdot \Delta C_o(p(L)) \right] \tag{A.27}$$

where N_a is Avogadro's number ($6.022169 \cdot 10^{23}$ molecules/mole). The molecular weights of water, $\mathrm{mw}_w = 18.016$, and dry air, $\mathrm{mw}_d = 28.9644$, are constants (dry air molecular weight varies due to composition changes with altitude, but the variation is insignificant between the surface and 1 mb). The dry layer column density, $\Delta C_d(L)$, can be found from known quantities using Eq. A.27,

$$\Delta C_d(p(L)) = \frac{\Delta p(L) \cdot N_a}{g(L) \cdot \mathrm{mw}_d} - \frac{\mathrm{mw}_w}{\mathrm{mw}_d} \cdot \Delta C_w(p(L)) - \frac{\mathrm{mw}_o}{\mathrm{mw}_d} \cdot \Delta C_o(p(L)) \tag{A.28}$$

which can also be written in terms of mixing ratios using Eq. A.9

$$\Delta C_d(p_{eff}(L)) = \frac{\Delta p_{eff}(L) \cdot N_a}{g(L) \cdot \mathrm{mw}_d \cdot (1 + q_w(p(L)) + q_o(p(L)))} \tag{A.29}$$

Equating Eq. A.25, A.26, and Eq. A.27 gives the result that the average molecular weight is given by a number weighted average of the water and dry molecular weights

$$\mathrm{mw}_t(p_{eff}(L)) = \frac{\mathrm{mw}_d \cdot \Delta C_d(p(L)) + \mathrm{mw}_w \cdot \Delta C_w(p(L))}{\Delta C_d(p(L)) + \Delta C_w(p(L))} \tag{A.30}$$

which can also be computed at $p(L)$ from the mixing ratios

$$\mathrm{mw}_t(p(L)) = \frac{\mathrm{mw}_d \cdot (1 + q_w(p(L)) + q_o(p(L)))}{1 + \frac{\mathrm{mw}_d}{\mathrm{mw}_w} \cdot q_w(p(L)) + \frac{\mathrm{mw}_d}{\mathrm{mw}_o} \cdot q_o(p(L))} \tag{A.31}$$

Recipe for converting Mixing Ratios to Column Density (f/ UMBC RTA)

The atmospheric state is specified with $p(L)$, $T(p(L))$, $q_w(p(L))$,$q_o(p(L))$, and $q_l(p(L))$. For the level.2 files the layers column densities will be defined within layers defined with boundaries at $p(L)$ and $p(L-1)$ such that $p(L-1) < p_{eff}(L) < p(L)$. In general, the pressure levels can be any system of levels.

The conversion between layer column densities and mass mixing ratios is well defined in terms of the layer column density of dry air (*i.e.,* all gases except for water and ozone).

Converting from mixing ratios to column densities requires computation of the dry layer column density. Given the dry layer column density, computed with Eq. A.29, the effective pressure levels, $\Delta C_d(p_{eff}(L))$, the layer column densities for species x can be computed with

$$\Delta C_x(p_{eff}(L)) = q_x(p(L)) \cdot \frac{\mathrm{mw}_d}{\mathrm{mw}_x} \cdot \Delta C_d(p_{eff}(L)) \tag{A.32}$$

Eq. A.32 assume that the mixing ratio values at $p_{eff}(L)$ are representative of the entire layer. This is not strictly true; however, it is very reasonable for thin layers. Also, during a retrieval the value of the dry layer column density changes on the isobaric grid as the water and ozone profiles change. The conversion from mixing ratio to column density must be re-computed whenever these mixing ratios are altered. This re-computation assures that the effective pressure level is still a reasonable choice.

$$C_x(p_{eff}(L)) = \sum_{j=1}^{L} \Delta C_x(p_{eff}(L)) \tag{A.33}$$

The column density at the RTA pressure levels, $p(L)$, can be found by interpolation.

$$C_x(p(L)) = \exp\left(\log_e(C_x(p_{eff}(L)) + r_{eff}(L) \cdot \log_e(C_x(p_{eff}(L)/C_x(p_{eff}(L-1))))\right) \tag{A.34}$$

where the interpolation factor, $r_{eff}(L)$, is given by

$$r_{eff}(L) = \frac{\log_e(p(L)/p_{eff}(L-1))}{\log_e(p_{eff}(L)/p_{eff}(L-1))} \tag{A.35}$$

The "recipe" for converting mixing ratio to layer column density is given by the following steps. These are documented in the routine ~/src/src_util/mr2cd.F.

1. compute $z(p)$
2. compute $g(z(p), \text{latitude})$
3. compute $\Delta C_d(p_{eff})$ using Eq. A.29
4. compute $\Delta C_w(p_{eff})$, $\Delta C_o(p_{eff})$, and $\Delta C_l(p_{eff})$ using Eq. A.32
5. compute $C_w(p_{eff})$, $C_o(p_{eff})$, and $C_l(p_{eff})$ using Eq. A.33
6. interpolate to $C_w(p)$, $C_o(p)$, and $C_l(p)$ using Eqs. A.34 and A.35
7. compute ΔC_w, ΔC_o, and ΔC_l

Recipe for converting level.2 Column Densities file to Mixing Ratio file

The atmosphere is initially specified by $p(L)$, $T(p(L))$, $\Delta C_w(p(L))$, $\Delta C_o(p(L))$, and $\Delta C_l(p(L))$. For conversion of column densities to mixing ratios the situation is equally sensitive to the knowledge of $C_d(p(L))$. Within a retrieval this computation should be avoided, since the meaning of the effective pressure level will have been altered as the column densities are changed. Within retrievals it is recommended that the column densities are continually re-computed using the recipe for converting mixing ratio's to column densities, given above.

If the file is a "truth" file, then the conversion is unambiguous since $C_d(p(L))$ can be computed directly from the column densities (see Eq. A.28). For converting truth files, we applied the following conversion from column densities to mixing ratios

$$C_x(p(L)) = \sum_{j=1}^{L} \Delta C_x(p(L)) \tag{A.36}$$

$$C_x(p_{eff}(L)) = \exp\left(\log_e(C_x(p(L-1)) + r(L) \cdot \log_e(C_x(p(L)/C_x(p(L-1))))\right) \tag{A.37}$$

where the interpolation factor, $r(L)$, is given by

$$r(L) = \frac{\log_e(p_{eff}(L)/p(L-1))}{\log_e(p(L)/p(L-1))} \tag{A.38}$$

and the mixing ratio is given by

$$q_x(p(L)) = \frac{\mathrm{mw}_d}{\mathrm{mw}_x} \cdot \frac{\Delta C_x(p_{eff}(L+1))}{\Delta C_d(p_{eff}(L+1))} \tag{A.39}$$

The "recipe" for converting layer column density to mixing ratio is given by the following steps. These are documented in the routine ~/src/src_util/cd2mr.F.

1. compute $z(p)$
2. compute $g(z(p), \text{latitude})$
3. compute $\Delta C_d(p)$ using Eq. A.28
4. compute $C_d(p), C_w(p), C_o(p), C_l(p)$ using Eq. A.36
5. compute $C_d(p_{eff}), C_w(p_{eff}), C_o(p_{eff}), C_l(p_{eff})$ by interpolation using Eqs. A.37 and A.38
6. compute $q_w(p)$, $q_o(p)$, and $q_l(p)$ using Eq. A.39

Table A.1: 66 layer pressure grid						
L	$p(L-1)$	$p(L)$	$p_{eff}(L)$	$\log_e\left(\frac{p(L)}{p(L-1)}\right)$	$\frac{\Delta p(L)}{p(L)}$	$r(L)$
1	0.0050	1.0000	0.1878	5.29832	0.99500	0.31565
2	1.0000	2.0000	1.4427	0.69315	0.50000	0.47123
3	2.0000	3.0000	2.4663	0.40547	0.33333	0.48313
4	3.0000	4.0000	3.4761	0.28768	0.25000	0.48802
5	4.0000	5.0000	4.4814	0.22314	0.20000	0.49071
6	5.0000	6.0000	5.4848	0.18232	0.16667	0.49241
7	6.0000	7.0000	6.4872	0.15415	0.14286	0.49358
8	7.0000	8.0000	7.4889	0.13353	0.12500	0.49444
9	8.0000	9.0000	8.4902	0.11778	0.11111	0.49509
10	9.0000	10.0000	9.4912	0.10536	0.10000	0.49561
11	10.0000	15.0000	12.3315	0.40547	0.33333	0.48313
12	15.0000	20.0000	17.3803	0.28768	0.25000	0.48802
13	20.0000	30.0000	24.6630	0.40547	0.33333	0.48313
14	30.0000	40.0000	34.7606	0.28768	0.25000	0.48802
15	40.0000	50.0000	44.8142	0.22314	0.20000	0.49071
16	50.0000	60.0000	54.8481	0.18232	0.16667	0.49241
17	60.0000	70.0000	64.8716	0.15415	0.14286	0.49358
18	70.0000	80.0000	74.8887	0.13353	0.12500	0.49444
19	80.0000	90.0000	84.9019	0.11778	0.11111	0.49509
20	90.0000	100.0000	94.9122	0.10536	0.10000	0.49561
21	100.0000	110.0000	104.9206	0.09531	0.09091	0.49603
22	110.0000	120.0000	114.9275	0.08701	0.08333	0.49638
23	120.0000	130.0000	124.9333	0.08004	0.07692	0.49667
24	130.0000	140.0000	134.9382	0.07411	0.07143	0.49692
25	140.0000	150.0000	144.9426	0.06899	0.06667	0.49712
26	150.0000	160.0000	154.9461	0.06454	0.06250	0.49732
27	160.0000	170.0000	164.9495	0.06062	0.05882	0.49748
28	170.0000	180.0000	174.9522	0.05716	0.05556	0.49763
29	180.0000	190.0000	184.9549	0.05407	0.05263	0.49776
30	190.0000	200.0000	194.9571	0.05129	0.05000	0.49788
31	200.0000	220.0000	209.8411	0.09531	0.09091	0.49603
32	220.0000	240.0000	229.8549	0.08701	0.08333	0.49638
33	240.0000	260.0000	249.8665	0.08004	0.07692	0.49667
34	260.0000	280.0000	269.8763	0.07411	0.07143	0.49692
35	280.0000	300.0000	289.8852	0.06899	0.06667	0.49712
36	300.0000	320.0000	309.8922	0.06454	0.06250	0.49732
37	320.0000	340.0000	329.8990	0.06062	0.05882	0.49748
38	340.0000	360.0000	349.9044	0.05716	0.05556	0.49763
39	360.0000	380.0000	369.9097	0.05407	0.05263	0.49776
40	380.0000	400.0000	389.9142	0.05129	0.05000	0.49788

Table A.1: 66 layer pressure grid (continued)						
L	$p(L-1)$	$p(L)$	$p_{eff}(L)$	$\log_e\left(\frac{p(L)}{p(L-1)}\right)$	$\frac{\Delta p(L)}{p(L)}$	$r(L)$
41	400.0000	425.0000	412.3737	0.06062	0.05882	0.49747
42	425.0000	450.0000	437.3805	0.05716	0.05556	0.49763
43	450.0000	475.0000	462.3871	0.05407	0.05263	0.49776
44	475.0000	500.0000	487.3928	0.05129	0.05000	0.49788
45	500.0000	525.0000	512.3988	0.04879	0.04762	0.49795
46	525.0000	550.0000	537.4025	0.04652	0.04545	0.49809
47	550.0000	575.0000	562.4079	0.04445	0.04348	0.49813
48	575.0000	600.0000	587.4114	0.04256	0.04167	0.49822
49	600.0000	625.0000	612.4155	0.04082	0.04000	0.49828
50	625.0000	650.0000	637.4189	0.03922	0.03846	0.49834
51	650.0000	675.0000	662.4209	0.03774	0.03704	0.49845
52	675.0000	700.0000	687.4247	0.03637	0.03571	0.49847
53	700.0000	725.0000	712.4272	0.03509	0.03448	0.49852
54	725.0000	750.0000	737.4302	0.03390	0.03333	0.49855
55	750.0000	775.0000	762.4324	0.03279	0.03226	0.49861
56	775.0000	800.0000	787.4346	0.03175	0.03125	0.49865
57	800.0000	825.0000	812.4359	0.03077	0.03030	0.49872
58	825.0000	850.0000	837.4386	0.02985	0.02941	0.49872
59	850.0000	875.0000	862.4388	0.02899	0.02857	0.49882
60	875.0000	900.0000	887.4395	0.02817	0.02778	0.49890
61	900.0000	925.0000	912.4425	0.02740	0.02703	0.49888
62	925.0000	950.0000	937.4449	0.02667	0.02632	0.49887
63	950.0000	975.0000	962.4452	0.02598	0.02564	0.49895
64	975.0000	1000.0000	987.4451	0.02532	0.02500	0.49903
65	1000.0000	1025.0000	1012.4495	0.02469	0.02439	0.49893
66	1025.0000	1050.0000	1037.4508	0.02410	0.02381	0.49896

where, the interpolation factor is given by

$$r(L) = \frac{\log_e\left(p_{eff}(L)/p(L-1)\right)}{\log_e\left(p(L)/p(L-1)\right)} \tag{A.40}$$

Table A.2: 100 layer pressure grid						
L	$p(L-1)$	$p(L)$	$p_{eff}(L)$	$\log_e\left(\frac{p(L)}{p(L-1)}\right)$	$\frac{\Delta p(L)}{p(L)}$	$r(L)$
1	0.0050	0.0161	0.0095	1.16938	0.68944	0.45182
2	0.0161	0.0384	0.0257	0.86924	0.58073	0.46401
3	0.0384	0.0769	0.0554	0.69445	0.50065	0.47118
4	0.0769	0.1370	0.1041	0.57748	0.43869	0.47600
5	0.1370	0.2244	0.1771	0.49345	0.38948	0.47948
6	0.2244	0.3454	0.2806	0.43127	0.35032	0.48206
7	0.3454	0.5064	0.4208	0.38262	0.31793	0.48408
8	0.5064	0.7140	0.6043	0.34356	0.29076	0.48570
9	0.7140	0.9753	0.8379	0.31186	0.26792	0.48702
10	0.9753	1.2972	1.1286	0.28522	0.24815	0.48812
11	1.2972	1.6872	1.4837	0.26286	0.23115	0.48905
12	1.6872	2.1526	1.9105	0.24361	0.21620	0.48986
13	2.1526	2.7009	2.4164	0.22691	0.20301	0.49055
14	2.7009	3.3398	3.0091	0.21233	0.19130	0.49116
15	3.3398	4.0770	3.6962	0.19945	0.18082	0.49169
16	4.0770	4.9204	4.4855	0.18803	0.17141	0.49217
17	4.9204	5.8776	5.3848	0.17776	0.16286	0.49260
18	5.8776	6.9567	6.4020	0.16856	0.15512	0.49298
19	6.9567	8.1655	7.5450	0.16021	0.14804	0.49333
20	8.1655	9.5119	8.8216	0.15263	0.14155	0.49364
21	9.5119	11.0038	10.2397	0.14570	0.13558	0.49393
22	11.0038	12.6492	11.8074	0.13935	0.13008	0.49419
23	12.6492	14.4559	13.5325	0.13351	0.12498	0.49444
24	14.4559	16.4318	15.4228	0.12812	0.12025	0.49467
25	16.4318	18.5847	17.4862	0.12312	0.11584	0.49487
26	18.5847	20.9224	19.7305	0.11848	0.11173	0.49507
27	20.9224	23.4526	22.1634	0.11416	0.10789	0.49524
28	23.4526	26.1829	24.7927	0.11013	0.10428	0.49541
29	26.1829	29.1210	27.6259	0.10635	0.10089	0.49557
30	29.1210	32.2744	30.6707	0.10281	0.09771	0.49571
31	32.2744	35.6505	33.9345	0.09949	0.09470	0.49585
32	35.6505	39.2566	37.4246	0.09636	0.09186	0.49598
33	39.2566	43.1001	41.1484	0.09341	0.08918	0.49611
34	43.1001	47.1882	45.1133	0.09062	0.08663	0.49622
35	47.1882	51.5278	49.3262	0.08798	0.08422	0.49633
36	51.5278	56.1260	53.7942	0.08548	0.08193	0.49644
37	56.1260	60.9895	58.5241	0.08310	0.07974	0.49654
38	60.9895	66.1253	63.5228	0.08085	0.07767	0.49663
39	66.1253	71.5398	68.7971	0.07870	0.07569	0.49672
40	71.5398	77.2396	74.3533	0.07666	0.07379	0.49680

Table A.2: 100 layer pressure grid (continued)						
L	$p(L-1)$	$p(L)$	$p_{eff}(L)$	$\log_e\left(\frac{p(L)}{p(L-1)}\right)$	$\frac{\Delta p(L)}{p(L)}$	$r(L)$
41	77.2396	83.2310	80.1980	0.07471	0.07199	0.49688
42	83.2310	89.5204	86.3375	0.07285	0.07026	0.49697
43	89.5204	96.1138	92.7780	0.07107	0.06860	0.49704
44	96.1138	103.0172	99.5256	0.06936	0.06701	0.49711
45	103.0172	110.2366	106.5862	0.06773	0.06549	0.49717
46	110.2366	117.7775	113.9654	0.06617	0.06403	0.49725
47	117.7775	125.6456	121.6692	0.06467	0.06262	0.49730
48	125.6456	133.8462	129.7026	0.06323	0.06127	0.49738
49	133.8462	142.3848	138.0715	0.06184	0.05997	0.49742
50	142.3848	151.2664	146.7809	0.06051	0.05872	0.49747
51	151.2664	160.4959	155.8357	0.05923	0.05751	0.49752
52	160.4959	170.0784	165.2408	0.05799	0.05634	0.49759
53	170.0784	180.0183	175.0012	0.05680	0.05522	0.49765
54	180.0183	190.3203	185.1216	0.05565	0.05413	0.49767
55	190.3203	200.9887	195.6060	0.05454	0.05308	0.49772
56	200.9887	212.0277	206.4589	0.05347	0.05206	0.49778
57	212.0277	223.4415	217.6846	0.05243	0.05108	0.49783
58	223.4415	235.2338	229.2872	0.05143	0.05013	0.49785
59	235.2338	247.4085	241.2697	0.05046	0.04921	0.49792
60	247.4085	259.9691	253.6371	0.04952	0.04832	0.49793
61	259.9691	272.9191	266.3918	0.04861	0.04745	0.49796
62	272.9191	286.2617	279.5373	0.04773	0.04661	0.49801
63	286.2617	300.0000	293.0774	0.04688	0.04579	0.49803
64	300.0000	314.1369	307.0145	0.04605	0.04500	0.49806
65	314.1369	328.6753	321.3516	0.04524	0.04423	0.49809
66	328.6753	343.6176	336.0912	0.04446	0.04349	0.49814
67	343.6176	358.9665	351.2365	0.04370	0.04276	0.49816
68	358.9665	374.7241	366.7887	0.04296	0.04205	0.49822
69	374.7241	390.8926	382.7514	0.04224	0.04136	0.49824
70	390.8926	407.4738	399.1261	0.04154	0.04069	0.49825
71	407.4738	424.4698	415.9144	0.04086	0.04004	0.49827
72	424.4698	441.8819	433.1172	0.04020	0.03940	0.49834
73	441.8819	459.7118	450.7375	0.03956	0.03878	0.49838
74	459.7118	477.9607	468.7764	0.03893	0.03818	0.49841
75	477.9607	496.6298	487.2360	0.03832	0.03759	0.49839
76	496.6298	515.7200	506.1142	0.03772	0.03702	0.49846
77	515.7200	535.2322	525.4158	0.03714	0.03646	0.49845
78	535.2322	555.1669	545.1392	0.03657	0.03591	0.49846
79	555.1669	575.5248	565.2853	0.03601	0.03537	0.49847
80	575.5248	596.3062	585.8550	0.03547	0.03485	0.49848

Table A.2: 100 layer pressure grid (continued)

L	$p(L-1)$	$p(L)$	$p_{eff}(L)$	$\log_e\left(\frac{p(L)}{p(L-1)}\right)$	$\frac{\Delta p(L)}{p(L)}$	$r(L)$
81	596.3062	617.5112	606.8472	0.03494	0.03434	0.49854
82	617.5112	639.1398	628.2631	0.03443	0.03384	0.49858
83	639.1398	661.1920	650.1041	0.03392	0.03335	0.49856
84	661.1920	683.6673	672.3672	0.03343	0.03287	0.49860
85	683.6673	706.5654	695.0546	0.03294	0.03241	0.49858
86	706.5654	729.8857	718.1617	0.03247	0.03195	0.49868
87	729.8857	753.6275	741.6927	0.03201	0.03150	0.49869
88	753.6275	777.7897	765.6464	0.03156	0.03107	0.49863
89	777.7897	802.3714	790.0154	0.03112	0.03064	0.49876
90	802.3714	827.3713	814.8073	0.03068	0.03022	0.49873
91	827.3713	852.7880	840.0157	0.03026	0.02980	0.49873
92	852.7880	878.6201	865.6404	0.02984	0.02940	0.49873
93	878.6201	904.8659	891.6791	0.02943	0.02901	0.49875
94	904.8659	931.5236	918.1295	0.02903	0.02862	0.49882
95	931.5236	958.5911	944.9920	0.02864	0.02824	0.49883
96	958.5911	986.0666	972.2646	0.02826	0.02786	0.49880
97	986.0666	1013.9476	999.9442	0.02788	0.02750	0.49877
98	1013.9476	1042.2319	1028.0234	0.02751	0.02714	0.49891
99	1042.2319	1070.9170	1056.5110	0.02715	0.02679	0.49882
100	1070.9170	1100.0000	1085.3951	0.02679	0.02644	0.49883

where, the interpolation factor is given by

$$r(L) = \frac{\log_e\left(p_{eff}(L)/p(L-1)\right)}{\log_e\left(p(L)/p(L-1)\right)} \tag{A.41}$$