2-3. 2A25

2-3. 1. Objectives

The objectives of 2A25 are to correct for the rain attenuation in measured radar reflectivity and to estimate the instantaneous three-dimensional distribution of rain from the TRMM Precipitation Radar (PR) data. The estimated vertical profiles of attenuation-corrected radar reflectivity factor and rainfall rate are given at each resolution cell of the PR. The estimated rainfall rate at the actual surface height and the average rainfall rate between the two predefined altitudes (2 and 4 km) are also calculated for each beam position.

2-3. 2. Changes from V5 to V6

The major points of improvement in V6 are as follows:

(1) The effects of attenuation due to cloud liquid water, water vapor, and molecular oxygen are considered in the attenuation correction algorithm

(2) The attenuation between the nearSurfBin (the lowest range bin that is free from the mainlobe surface clutter) and the actual surface is estimated by assuming a given slope of dBZe and accounted for in the surface reference technique.

(3) The estimates of Ze, R and several other parameters that varies with the adjustment parameter (ε) of α are calculated as the expected values with respect to the posterior probability distribution function $p(\varepsilon)$. In V5, the maximum likelihood value of ε was used.

(4) The error estimates in the path-integrated attenuation by the surface reference technique and from the rain echoes are reevaluated

(5) The definition of the upper range of the surface clutter is changed for those cases in which the rain echo is undetected because of large attenuation.

(6) The value of the parameter that defines the height of nearSurfRain was changed in accordance with the change of clutterFreeBottom in 1B21 and 1C21.

(7) Several new output variables are introduced (See section 2.3-10). Some of them such as sigmaZero and freezH are exact copies of frequently used variables in 2A21 and 2A23.

(8) The input parameter files are copied in the 2A25 data file. They are written by using the following code. Refer to the TSDIS users manual to read them.

```
TKwriteFileInfo(&granuleHandle2A25,PARAM_FILE_GEN,"Parameters:General");
TKwriteFileInfo(&granuleHandle2A25,PARAM_FILE_CONV,"Parameters:Convective");
TKwriteFileInfo(&granuleHandle2A25,PARAM_FILE_STRAT,"Parameters:Stratiform");
TKwriteFileInfo(&granuleHandle2A25,PARAM_FILE_OTHER,"Parameters:Other");
```

```
TKwriteFileInfo(&granuleHandle2A25,ERROR_P_FILE,"Parameters:Errors");
```

2-3. 3. Algorithm Overview

2A25 basically uses a hybrid of the Hitschfeld-Bordan method and the surface reference method to estimate the vertical profile of attenuation-corrected effective radar reflectivity factor (*Ze*). (The hybrid method is described in Iguchi and Meneghini (1994).) The vertical rain profile is then calculated from the estimated *Ze* profile by using an appropriate *Ze*-*R* relationship. One major difference from the method described in the above reference is that in order to deal with the uncertainties in measurements of the scattering cross section of surface as well as the rain echoes, a probabilistic method is used. Since radar rain echoes from near the surface are hidden by the strong surface echo, the rain estimate at the lowest point in the clutter-free region is given as the near-surface rainfall rate for each angle bin.

2-3. 4. More Detailed Description of the Algorithm

The major input data to 2A25 are the measured radar reflectivity factor Z_m, the apparent decrease of the surface cross section ($\Delta\sigma^0$), its reliability, the rain type and miscellaneous height information. The algorithm first defines the region for processing: It processes only the data between the rain top and the lowest height above the surface that is free from the surface clutter. (The current algorithm does not use any data below the surface, i.e., the mirror image.) The bright-band height and climatological freezing height are used to define the regions of liquid (water), solid (ice), and mixed phase of precipitating particles. The initial values of the coefficients in the k-Z_e and Z_e-R relationships at different altitudes are accordingly defined.

The attenuation correction is, in principle, based on the surface reference method. This method assumes that the decrease in the apparent surface cross section is caused by the propagation loss in rain. The coefficient α in the k-Z_e relationship $k=\alpha Z_e^{\beta}$ is adjusted in such a way that the path-integrated attenuation (PIA) estimated from the measured Z_m-profile will match the reduction of the apparent surface cross section. The attenuation correction of Z_e is carried out by the Hitschfeld-Bordan method with the modified α. Since α is adjusted, we call this type of surface reference method the α-adjustment method. The α-adjustment method assumes that the discrepancy between the PIA estimate from $\Delta\sigma^0$ and that from the measured Z_m-profile can be attributed to the deviation of the initial α values from the true values which may vary depending on the raindrop size distribution and other conditions. It assumes that the radar is properly calibrated and that the measured Z_m has no error.

The surface reference method generally works rather well as long as the apparent decrease in surface cross section $\Delta\sigma^0$ is much larger than the fluctuations of the true surface cross section.

When the decrease is not significant, however, the relative error associated with this method in the estimates of rainfall rate becomes large since the fluctuation of surface cross section, which remains finite even when there is no rain, translates to the absolute error of the rain estimates.

In order to avoid inaccuracies in the attenuation correction when rain is weak, a hybrid of the surface reference method and the Hitschfeld-Bordan method is used [Iguchi and Meneghini, 1994]. In versions 5 and 6 of 2A25, the errors in these methods are treated in a probabilistic manner. Because the relationship between the error in α and that in the Hitschfeld-Bordan method changes substantially with the attenuation, the relative weight on the surface reference method to the Hitschfeld-Bordan method varies with the attenuation. When rain is very weak and the attenuation estimate is small, the PIA estimate from the surface reference is effectively neglected. With the introduction of the hybrid method, the divergence associated with the Hitschfeld-Bordan method is also prevented.

When the PIA estimate from the surface reference ($\Delta\sigma^{0}$) is unavailable, it is replaced by an equivalent $\Delta\sigma^{0}_{c}$ that would make the attenuation-corrected Z-profile near the surface nearly constant vertically if the correction by the surface reference method is applied with this equivalent $\Delta\sigma^{0}_{c}$. This does not imply that the final vertical profile near the surface after the attenuation correction becomes constant because of the use of the hybrid method. Note also that negative values of $\Delta\sigma^{0}_{c}$ are reset to zero. The use of $\Delta\sigma^{0}_{c}$ instead of $\Delta\sigma^{0}$ from the surface reference seldom occurs in V6.

The attenuation correction procedure requires two processing cycles. In the first cycle, the correction is made without taking the attenuation by cloud liquid water (CLW), water vapor (WV) and molecular oxygen (O_2) into account. From the attenuation-corrected profile, the rainfall rate at the surface is estimated. Based on this rainfall rate and the statistical relationship between the surface rain rate and the vertical profile of cloud liquid water, the attenuation of radar rain echo caused by CLW is estimated at each range bin. Similarly, the attenuation due to WV is estimated from the estimated surface temperature and by assuming the 90% relative humidity within the raining footprint and 70% outside the raining area. The attenuation due to O_2 is a simple function of the altitude. Then in the second cycle, the vertical profile of Z_m is corrected for the attenuation by CLW, WV and O_2, and this attenuation-corrected Z_m is corrected for the attenuation by rain.

The corrections for the non-uniform beam filling effect in the attenuation correction and the conversion from Ze to rainfall rate are not made in V6, although the non-uniformity of rain distribution, i.e., the low resolution variability of the PIA for a given angle bin is calculated from the PIAs at the angle bin in question and the eight surrounding angle bins in V6 as well as in V5.

The rainfall estimates are calculated from the attenuation-corrected Z_e -profiles by using a power law: $R=aZ_e^b$ in which the parameters a and b are both functions of the rain type, existence of bright-band, freezing height, storm height and absolute height. Effects of the difference in the raindrop size distribution by rain type, the phase state, the temperature, and the difference in terminal velocity due to changes in the air density with height are taken into account. The parameters a and b are expressed as a function of the adjustment parameter (ε) of α in the k-Z_e relation and adjusted in accordance with the α-adjustment in the attenuation correction. The final estimate of R is obtained as the expectation of all possible values of R with the probability $p(\varepsilon)$.

2-3. 5. Input data

Input files:
1C-21 HDF data file
2A-21 HDF data file
2A-23 HDF data file

Input swath data from 1C-21 which are used in 2A-25:
binClutterFreeBottom[][]
binEllipsoid[]
binStormHeight[][]
binSurfPeak[]
geolocation[][]
minEchoFlag[]
normalSample[][]
scanStatus.dataQuality
scanStatus.missing
scanStatus.prStatus2
scanTime
scLocalZenith[]
scRange[]

Input swath data from 2A-21 which are used in 2A-25:
pathAtten[]
reliabFlag[]
reliabFactor[]
sigmaZero[]

Input swath data from 2A-23 which are used in 2A-25:
rainType[]

warmRain[]
status[]
freezH[]
HBB[]

Input header data from 1C21 used in 2A-25:
rayHdr[].rayStart
rayHdr[].mainlobeEdge
rayHdr[].sidelobeRange[]

2-3. 6. Output data

Output files:
2A-25 HDF data file
VI file

Data format

The file content description for 2A25 can be found in the *Interface Control Specification (ICS) between the Tropical Rainfall Measuring Mission Science Data and Information System (TSDIS) and the TSDIS Science User (TSU) Volume 4: File Specification for TSDIS Products - Level 2 and 3 File Specifications*. It is available at: http://tsdis02.nascom.nasa.gov/tsdis/Documents/ICSVol4.pdf

Output data (in alphabetical order):

attenParmAlpha[][]	k-Z parameter alpha at 5 nodes
attenParmBeta[]	k-Z parameter beta
correctZFactor[][]	attenuation-corrected Z factor in dBZ
epsilon[]	correction factor with the hybrid method
epsilon_0[]	correction factor with the SRT
errorRain[]	error estimate of rain rate near surface in dB
errorZ[]	error estimate of Z near surface in dB
e_SurfRain[]	estimated rain rate at the actual surface
freezH[]	freezing height from 2A23
geolocation[][]	geolocation
method[]	method used
navigate	navigation data
nearSurfRain[]	estimated rain rate near surface
nearSurfZ[]	estimated Z near surface
nubfCorrectFactor[][]	non-uniform beam filling correction factors
ParmNode[][]	bin numbers of 5 nodes for alpha, a and b
pia[][]	path-integrated attenuations from final Z, in surface clutter, and from 2A21

precipWaterParmA[][]	PWC-Z parameter a in PWC=a*Z^b at 5 nodes
precipWaterParmB[][]	PWC-Z parameter b in PWC=a*Z^b at 5 nodes
precipWaterSum[]	sum of PWC from rain top to surface
qualityFlag[]	quality flag
rain[][]	rainfall rate in mm/h.
rainAve[][]	average rainfall rate between 2 and 4 km
rainFlag[]	status flag for rainfall estimate
rainType[]	rain type from 2A23
rangeBinNum[][]	bin numbers of BB, storm top, etc.
reliab[][]	reliability of the output
scanStatus	scanStatus
scanTime	scanTime
scLocalZenith[]	spacecraft local zenith angle
sigmaZero[]	surface scattering cross section sigmaZero from 2A21
spare[][]	spare
thickThPIZ[]	range bin number where PIZ > threshPIZ
weightW[]	weight for the calculation of epsilonf
xi[][]	normalized standard deviation of PIA
zeta[][]	integral of alpha*Zm^beta
zeta_mn[][]	mean of zeta over 3x3 IFOVs
zeta_sd[][]	standard deviation of zeta
zmmax[]	maximum of Zm
ZRParmA[][]	Z-R parameter a in R=a*Z^b at 5 nodes
ZRParmB[][]	Z-R parameter b in R=a*Z^b at 5 nodes

For details, see section 2-3. 10.

2-3. 7. Interfaces with other algorithms

As described in "Input data" section, 2A25 reads data from 1C21, 2A21 and 2A23. The output data of 2A25 is used in 3A25 and 3A26.

2-3. 8. Caveats

1. 2A25 produces many output variables. Please read section 2-3. 10 carefully before using them. For example, negative numbers are stored in rain[][] and correctedZFactor[][] when the data are missing or in the possibly cluttered ranges.

   (IMPORTANT)

   If the input radar reflectivity factor Z_m is below the noise level, the corresponding rain estimate is set to 0. This procedure does not cause any serious problem except when the measured Z_m becomes smaller than the noise level by rain attenuation. In such a case, even if some heavy rain exists near the surface, and the actual rain rate there is rather large, the number in rain[][] is 0.

   To know whether such low radar reflectivity factors are caused by large attenuation or not, look at the fourth bit of 'reliab' and the fourth bit of 'rainFlag'. In V6, if Z_m becomes less than the noise level in the range bins near the surface with zeta larger than the threshold value, these range bins are regarded as cluttered bins and the bottom range bin to which the data are processed for rain profiling is raised from the clutter-free bottom defined in 2A23.

2. The error estimates in 'rain' and 'correctZFactor' are given in 'errorRain[]' and 'errorZ[]'. However, these estimates indicate only very crude estimates. (The estimation method is improved in V6 in which these errors are estimated based on the standard deviation of the final probability distribution.)

3. 2A25 processes data in 'rain certain' angle bins only. It processes all data downward from 1 km above the height at which the first 'certain' rain echo is detected. This new definition of the processing region in 2A25 is introduced in version V5 and retained in V6.

4. Over some area with a very high surface reflectivity, surface echoes picked up in antenna sidelobes may appear in the radar signal and they are sometimes misidentified as rain echoes. The sidelobe clutter rejection routine in 1B21 and 2A25 removes some of the sidelobe clutters internally, but not all sidelobe signals are completely removed.

5. 2A25 relies on the output of 1C21 to separate the surface cluttered ranges from the clutter free ranges. Because the clutter identification routine used in 1B21 is not perfect (it never can be), some surface clutter (mainlobe clutter) may be occasionally misidentified as rain echoes in 2A25, particularly in mountain regions. It is strongly suggested that you look at the vertical profile if the surface clutter seems present in the data.

6. The range bin numbers in the output of 2A25 are all relative to the Earth's ellipsoid (which is nearly equal to the mean sea level) with the ellipsoid range bin corresponding to 79. For example, if the range bin number is 75, its height from the ellipsoid is (79-75)*0.25 = 1.0 km. This number is NOT the height above the actual surface.

7. In V6, the value of alpha in the k-Z relationship (k = alpha * Z^beta) at 5 nodal points are given in attenParmAlpha[][]. The values in attenParmAlpha[][] are the initial values of alpha. To obtain the mean values of alpha used in the final attenuation estimation, the initial values must be multiplied by epsilon[].

8. The values of *a* and *b* given in ZRParmA[][] and ZRParmB[][] are the expected values of *a* and *b* in the *R-Z* relationship (*R* = *a* * *Z^b*), respectively. The values of *R* and *Ze* are calculated as the expected values, too. Therefore, you many not obtain the same value of *R* if you calculate *R* by using the formula *R* = *a* * *Z^b* with *a* and *b* given in ZRParmA[][] and ZRParmB[][]. In other words, <*R*> = <*a* * *Z^b*> which is not necessarily equal to <*a*>*<*Z_e*>^<*b*>.

2-3. 9. References

Iguchi, T., and R. Meneghini, “Intercomparison of Single Frequency Methods for Retrieving a Vertical Rain Profile from Airborne or Spaceborne Data,” *Journal of Atmospheric and Oceanic Technology*, 11, 1507-1516 (1994).

T. Kozu and T. Iguchi, “Nonuniform Beamfilling Correction for Spaceborne Radar Rainfall Measurement: Implications from TOGA COARE Radar Data Analysis,” *J. Atmos. Oceanic Technol.*, 16, 1722-1735 (1999).

Iguchi, T., T. Kozu, R. Meneghini, J. Awaka, and K. Okamoto, “Preliminary results of rain profiling with TRMM Precipitation Radar,” *Proc. of URSI-F International Triennial Open Symposium on Wave Propagation and Remote Sensing*, Aveiro, Portugal, pp.147-150, 1998.

T. Iguchi, T. Kozu, R. Meneghini, J. Awaka, and K. Okamoto, “Rain-Profiling Algorithm for the TRMM Precipitation Radar,” *Journal of Applied Meteorology,* Vol.39, No.12, pp.2038-2052, 2000.

R. Meneghini, T. Iguchi, T. Kozu, L. Liao, K. Okamoto, J. A. Jones, and J. Kwiatkowski, ”Use of the surface reference technique for path attenuation estimation from the TRMM Precipitation radar,” *Journal of Applied Meteorology,* Vol.39, No.12, pp.2053-2070, 2000.

2-3. 10. Detailed description of output variables

New Output Variables

```
float32   epsilon_0[49];
float32   e_SurfRain[49];
float32   freezH[49];
int16     parmNode[49][5];
float32   pia[49][3];
float32   precipWaterParmA[49][5];
float32   precipWaterParmB[49][5];
```

```
float32   precipWaterSum[49];
int16     rainType[49];
int16     rangeBinNum[49][7];
float32   scLocalZenith[49];
float32   sigmaZero[49];
```

New Definitions

```
int16     method[49]; (bit 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)
int16     qualityFlag[49]; (bit 2, 3, 4, 5, 11, 12, 13)
float32   rainAve[49][2]; (unit of rainAve[][1])
int16     rainFlag[49]; (bit 3, 4, 11)
float32   spare[49][2]; p(epsilon)'s area and standard deviation
```

Obsolete (No data in V6)

```
int16     attenParmNode[49][5]; moved to parmNode[49][5];
float32   pia2a25[49]; moved to pia[49][0]
float32   thickThPIZ[49];
float32   weightW[49]; replaced by epsilon_0[49]
float32   xi[49][2];
int16     ZRParmNode[49][5]; moved to parmNode[49][5];
```

List of 2A25 scan data

```
float32        attenParmAlpha[49][5];
int16          attenParmAlpha_scale[49][5];
float32        attenParmBeta[49];
int16          attenParmBeta_scale[49];
int16          attenParmNode[49][5];          Obsolete
float32        correctZFactor[49][80];
int16          correctZFactor_scale[49][80];
float32        epsilon[49];
float32        epsilon_0[49];                 New
float32        errorRain[49];
float32        errorZ[49];
float32        e_SurfRain[49];                New
float32        freezH[49];                    New
float32        geolocation[49][2];
int16          method[49];
NAVIGATION     navigate;
float32        nearSurfRain[49];
float32        nearSurfZ[49];
float32        nubfCorrectFactor[49][2];
int16          parmNode[49][5];               New
float32        pia[49][3];                    New
float32        pia2a25[49];                   Obsolete
float32        precipWaterParmA[49][5];       New
```

```
float32     precipWaterParmB[49][5];       New
float32     precipWaterSum[49];            New
int16       precipWaterSum_scale[49];      New
int16       qualityFlag[49];
float32     rain[49][80];
int16       rain_scale[49][80];
float32     rainAve[49][2];
int16       rainAve_scale[49][2];
int16       rainFlag[49];
int16       rainType[49];                  New
int16       rangeBinNum[49][7];            Partly new
int8        reliab[49][80];
PR_SCAN_STATUS scanStatus;
float64     scanTime;
float32     scLocalZenith[49];             New
float32     sigmaZero[49];                 New
float32     spare[49][2];
int16       thickThPIZ[49];                Obsolete
float32     weightW[49];                   Obsolete
int16       weightW_scale[49];             Obsolete
float32     xi[49][2];                     Obsolete
float32     zmmax[49];
float32     zeta[49][2];
float32     zeta_mn[49][2];
float32     zeta_sd[49][2];
int16       ZRParmA_scale[49][5];
float32     ZRParmA[49][5];
int16       ZRParmB_scale[49][5];
float32     ZRParmB[49][5];
int16       ZRParmNode[49][5];             Obsolete
```

Description of each output variable

----

attenParmAlpha

    float32   attenParmAlpha[49][5];
    REAL*4   attenParmAlpha(5,49)

Internally this quantity is stored as

int16 attenParmAlpha_scale[49][5] after multiplied (scaled) by 10000000.

Attenuation parameter alpha at nodes.

k = alpha*Z^beta.

"alpha" is given at five nodal points. These numbers are initial values of alpha. The mean values of alpha used in the attenuation correction are obtained by multiplying attenParmAlpha[49][5] by epsilon[49].

The alpha values between the nodes are calculated by linear interpolation. The range bin numbers of the nodes are stored in ParmNode[49][5] (in version 5, they were stored in attenParmNode[49][5]).

----

attenParmBeta

float32 attenParmBeta[49];
REAL*4 attenParmBeta(49)

Internally this quantity is stored as

int16 attenParmBeta_scale[49] after multiplied (scaled) by 10000.

Attenuation parameter beta

k = alpha*Z^beta.

beta is given for each angle bin.

A constant beta is used for all ranges in one angle bin.

----

attenParmNode (Obsolete. No data in V6)

int16 attenParmNode[49][5];
INTEGER*2 attenParmNode(5,49);

This parameter gives the range bin numbers of the nodes where attanParmAlpha and attenParmBeta are given. The numbers are stored in parmNode[49][5] in V6.

----

correctZFactor

float32 correctZFactor[49][80];
REAL*4 correctZFactor(80,49)

Internally this quantity is stored as

int16 correctZFactor_scale[49][80] after multiplied (scaled) by 100.

Estimated effective Z-factor in dBZ at 13.8 GHz after attenuation correction.

If the input radar reflectivity factor Zm is below the noise level, or if the estimate is below 0 dB, correctZFactor is set to 0.0.

Everything else is the same as rain[49][80] (rain(80,49)).

----

epsilon

float32 epsilon[49];
REAL*4 epsilon(49)

The multiplicative correction factor to alpha in the k-Ze relation. The value is the mean of epsilon with the probability density function (pdf) of epsilon defined by the rain echo, surface echo and their uncertainties. The standard deviation of the pdf is given in spare[][1] and the area of the likelihood function is given in spare[][0].

----

epsilon_0 (New variable)

```
float32     epsilon_0[49];
REAL*4      epsilon_0(49)
```

The multiplicative correction factor to alpha in the k-Ze relation if the weight to the path-integrated attenuation (PIA) given by the surface reference technique is 100%. This output is given only when the PIA estimate from 2A21 is either reliable or marginally reliable. When it is not reliable, epsilon_0 is set to 0.

The exact formula used is as follows.

When pia[][0]>0, we define pia_ratio = (pia[][0]-pia[][1])/pia[][0]. If pia[][0]=0, pia_ratio=1.

```
att_f_bttm = pow(10.0,-(pia[][2] * pia_ratio)/10.0);
epsilon_0 = (1-pow(att_f_bttm,beta))/zeta;
```

I.e., the PIA estimate from 2A21 that represents the attenuation to the surface is converted to the attenuation to the bottom of clutter-free range and the latter is used for the calculation of epsilon_0.

----

errorRain

```
float32     errorRain[49];
REAL*4      errorRain(49)
```

Error estimate of rain rate near the surface expressed in dB

The error is calculated as the standard deviation of the probability distribution of rain rate derived from the pdf of epsilon.

----

errorZ

```
float32     errorZ[49];
REAL*4      errorZ(49)
```

Error estimate of correctZFactor near the surface expressed in dB.

The error is calculated as the standard deviation of the probability distribution of correctZFactor derived from the pdf of epsilon.

----

e_SurfRain (New variable)

```
float32     e_SurfRain[49];
REAL*4      e_SurfRain(49)
```

Estimated rainfall rate at the actual surface. e_SurfRain is calculated by assuming a constant slope of dBZe from the bottom of the valid (clutter-free) rain echo. The assumed slope is 0 dB/km for all rain types except for the stratiform rain over land where -0.5 dB/km toward the surface is assumed. Note that 0 dB/km in Ze corresponds to -0.17 dB/km in rainfall rate (decreases toward the surface).

----

freezH (New variable)

float32 freezH[49];
REAL*4 freezH(49)

Freezing height expressed in m estimated from the climatological surface temperature. This is a copy of freezH given in 2A23. (freezH in 2A23 is given as an integer, but it is stored as a float number in 2A25.)

Note that the phase transition height used in 2A25 is different from the freezing height given in freezH. In fact, when the bright band is detected, its height is the phase transition height, and in other cases the phase transition height is 1.2 times the height given in freezH.

----

geolocation

float32 geolocation[49][2];
REAL*4 geolocation(2,49)

The earth location of the center of the IFOV at the altitude of the earth ellipsoid. The first dimension is latitude and longitude, in that order. Values are represented as floating point decimal degrees. Off-earth is represented as -9999.9. Latitude is positive north, negative south. Longitude is positive east, negative west. A point on the 180° meridian is assigned to the western hemisphere.

----

method (New definition. bit 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)

int16 method[49];
INTEGER*2 method(49)

Method (rain model) used in the retrieval of vertical profiles of Z and R

The default value is 0 (including no rain case).

The following meanings are assigned to each bit in the 16-bit integer.

(See flag_mthd)

```
      0: (bit  1) no rain
   if rain
      0: (bit  1) over ocean
      1: (bit  1) over land
      2: (bit  2) over coast, river, etc.
      3: (bit  2) others (impossible)
     +4: (bit  3) PIA from constant-Z-near-surface assumption
     +8: (bit  4) spatial reference
    +16: (bit  5) temporal reference
    +32: (bit  6) global reference
    +64: (bit  7) hybrid reference
   +128: (bit  8) good to take statistics of epsilon.
   +256: (bit  9) HB method used, SRT totally ignored
   +512: (bit 10) very large pia_srt for given zeta
  +1024: (bit 11) very small pia_srt for given zeta
  +2048: (bit 12) no ZR adjustment by epsilon
```

+4096: (bit 13) no NUBF correction because NSD unreliable
+8192: (bit 14) surface attenuation > 60 dB
+16384: (bit 15) data partly missing between rain top and bottom

16th bit is currently not used.

The constant Z method is used only when the surface reference is unreliable. This routine calculates the average slope of the Zm profile (expressed in dBZ) near the bottom of radar echo and attributes the slope to the attenuation. If there are not enough valid data points in the profile, it returns with 0 attenuation. The constant Z method is seldom used in V6.

----

navigate

```
NAVIGATION      navigate;
```

Look at the TSDIS INTERFACE CONTROL SPECIFICATION (ICS) Vol. 3, Appendix B "Navigation"

----

nearSurfRain

```
float32      nearSurfRain[49];
REAL*4       nearSurfRain(49)
```

Internally this quantity is stored as
int16 nearSurfRain_scale[49][80] after multiplied (scaled) by 100.

Near-surface rainfall rate estimate

"Near-surface" is defined as the lowest point in the clutter free ranges in almost all cases. However, if Zm at this point is below the noise level and if zeta which corresponds to the estimated attenuation down to this point is larger than the zeta_th_L defined in the parameter file (it is currently set to 0.7 which approximately corresponds to 4 dB of attenuation), in other words,
if the first bit of reliab[][] = 0 and
if the forth bit of reliab[][] = 0,
then the lowest range bin at which Zm is above the noise threshold is chosen as the near-surface range bin. The actual value of this near-surface range bin is stored in rangeBinNum[][6] in V6.

Specifically,
nearSurfRain[n_anglebin] = rain[n_anglebin][rangeBinNum[n_anglebin][6]];
nearSurfRain(n_anglebin) = rain((rangeBinNum(7,n_anglebin)+1),n_anglebin)

----

nearSurfZ

```
float32      nearSurfZ[49];
REAL*4       nearSurfZ(49)
```

Internally this quantity is stored as

int16 nearSurfZ_scale[49][80] after multiplied (scaled) by 100.

Near-surface Z-factor

See nearSurfRain[] for the definition of "Near-surface".

nearSurfZ[n_anglebin] = correctZFactor[n_anglebin][rangeBinNum[n_anglebin][6]];

nearSurfZ(n_anglebin) = correctZFactor((rangeBinNum(7,n_anglebin)+1),n_anglebin)

----

nubfCorrectFactor

```
float32     nubfCorrectFactor[49][2];
REAL*4      nubfCorrectFactor(2,49)
```

'nubfCorrectFactor` is the non-uniform beam filling (NUBF) correction factor.

nubfCorrectFactor[][0] is the NUBF correction factor for the surface reference and its range is between 1.0 and 3.0.

nubfCorrectFactor[][1] is the NUBF correction factor for the Z-R relation and its range is between 0.8 and 1.0.

N.B. No NUBF correction is made in V6. As a result, both nubfCorrectFactor[][0] and nubfCorrectFactor[][1] are always set to 0 in V6.

----

parmNode (New variable)

```
int16        parmNode[49][5];
INTEGER*2    parmNode(5,49)
```

Range bin numbers of the nodal points at which

the attenuation parameter alpha and the Z-R parameters "a" and "b" are given in

attenParmAlpha[49][5] (attenParmAlpha(5,49)),

ZRParmA[49][5] (ZRParmA(5,49)), and

ZRParmB[49][5] (ZRParmB(5,49)), respectively.

For each angle bin, 5 nodal points are defined.

ParmNode[][] gives the range bin numbers of the 5 nodes at which the values of attenuation parameter "alpha" and the Z-R parameters "*a*" and "*b*" are given in ParmAlpha[][], ZRParmA[][] and ZRParmB[][], respectively. The values of alpha, *a* and *b* between the nodes are linearly interpolated. The range of ParmNode is between 0 and 79. (See the note for rangeBinNum.)

In no-rain angle bins, ParmNode[][] is set to 0.

Note that the definition of range bin number in 2A25 is not the same as 1B-21 or 1C-21. The bin number shows the position in the 80-element array so that it takes a number between 0 and 79 (inclusive). Bin number 79 corresponds to the surface of the ellipsoid which is approximately equal to the sea surface.

----

pia (New variable)

```
float32    pia[49][3];
REAL*4     pia(3,49)
```

pia[][0]

Path-integrated attenuation from rain top to surface.

This attenuation is calculated from the attenuation-corrected Z-profile in correctZFactor[][] and adjusted alpha.

The number represents the two-way attenuation to the actual surface.

pia[][1]

Path-integrated attenuation between the clutter-free bottom and the surface. This is the attenuation estimate in the range that is cluttered by the surface echo.

pia[][2]

Path-integrated attenuation to surface estimated by the surface reference technique in 2A21. This is an exact copy of pathAtten(49) in 2A21.

----

pia2a25 (Obsolete, No data in V6) moved to pia[49][0] in version 6.

```
float32    pia2a25[49];
REAL*4     pia2a25(49)
```

Path-integrated attenuation from the estimated Z profile

----

precipWaterParmA (New variable)

```
float32    precipWaterParmA[49][5];
READ*4     precipWaterParmA(5,49)
```

Coefficient *a* in the relation between the precipitation water content (PWC) and Ze at 5 nodal points. The unit of PWC is g/m^3 and that of Ze is mm^6/m^3.

PWC = *a**Z^*b*.

'a' is given at five nodal points.

These values are stored in the first 5 elements of the array.

'a' values between the nodes can be calculated by linear interpolation.

The range bin numbers of the nodes are stored in the ParmNode[49][5] (ParmNode(5,49)).

Note that the sum of PWC's calculated from Z=correctZFactor with formula PWC = *a**Z^*b* where *a* and *b* from the interpoloted values of precipWaterParmA and precipWaterParmB does not necessarily agree with precipWaterSum. The former is <*a*>*<*Z*>^<*b*> where as the latter is <*a**Z^*b*>. precipWaterSum also includes the precipitation water content in the surface clutter range.

----

precipWaterParmB (New variable)

float32 precipWaterParmB[49][5];

READ*4 precipWaterParmB(5,49)

Coefficient *a* in the relation between the precipitation water content (PWC) and Ze at 5 nodal points. The unit of PWC is g/m^3 and that of Ze is mm^6/m^3.

PWC = *a**Z^*b*.

'b' is given at five nodal points.

'b' values between the nodes can be calculated by linear interpolation.

The range bin numbers of the nodes are stored in the ParmNode[49][5] (ParmNode(5,49)).

----

precipWaterSum (New variable)

float32 precipWaterSum[49];

REAL*4 precipWaterSum(49)

Internally this quantity is stored as

int16 precipWaterSum_scale[49] after multiplied (scaled) by 1000.

Vertically integrated value of precipitation water content. The unit is g km/m^3 or equivalently kg/m^2.

"precipWaterSum" give the sum of the precipitation water content calculated from Ze at each range bin. The summation is from the rain top to the actual surface. The water content in the surface clutter range is estimated with the same assumption that is used in the attenuation correction. The sum includes both liquid and solid phase regions.

----

qualityFlag (New definition, bit 2, 3, 4, 5, 11, 12, 13)

int16 qualityFlag[49];

INTEGER*2 qualityFlag(49)

Quality flag for each angle bin data

The default value is 0.

In V6, the definitions of bits 2, 3, 4, 5 are changed, and bits 13 and 14 are added. The flags indicated by bits 4, 5 and 13 in V6 are moved from method flag in V5.

```
     0: normal
    +1: unusual situation in rain average
    +2: NSD of zeta (xi) calculated from less than 6 points
    +4: NSD of PIA calculated from less than 6 points
    +8: NUBF for Z-R below lower bound
   +16: NUBF for PIA above upper bound
   +32: epsilon not reliable, epsi_sig <= 0.0
   +64: 2A21 input data not reliable
  +128: 2A23 input data not reliable
  +256: range bin error
  +512: sidelobe clutter removal
 +1024: probability=0 for all tau
 +2048: pia_surf_ex <= 0.0
 +4096: const Z is invalid
 +8192: reliabFactor in 2A21 is NaN
+16384: data missing
```

16th bit (sign bit) is not used.

The contents are exact copy of internal variable flag_qlty.

----

rain

```
float32    rain[49][80];
REAL*4     rain(80,49)
```

Internally this quantity is stored as

int16 rain_scale[49][80] after multiplied (scaled) by 100.

rainfall rate in mm/h

49 elements in the 2-D array correspond to the angle bins

and

80 elements (first argument in FORTRAN convention) in the 2-D array correspond to the range bins.

If the estimated Z-factor is below 0 dBZ, the rain rate is always set to 0.

If the input radar reflectivity factor Zm is below the noise level, the corresponding rain estimate is set to 0. This procedure does not cause any serious problem except when the measured Zm becomes smaller than the noise level by rain attenuation. In such a case, even if some heavy rain exists near the surface, the number in this variable is 0. To know whether such low radar reflectivity factors are caused by large attenuation or not, look at the forth bit of 'reliab' and the forth bit of rainFlag.

80 range bins are filled with data from top to bottom in height. The last element corresponds to the ellipsoid height, i.e., 0 m high above the model ellipsoid (not the actual surface). The first element corresponds to the radar resolution cell about 20 km above in slant range along the beam from the footprint on the ellipsoid. The

range resolution is 250 m.

If the radar data is missing, MISSING value of -99.99 is stored. This situation may happen at range bins above 15 km high because NASDA only guarantees the data collection below 15 km. (The highest edge of the radar's receiving window comes down to nearly 15 km above the sea level near the equator.)

The bin number of the lowest range bin that contains valid rain data is (rangeBinNum[][6] - 1 in C) or (rangeBinNum(7,.) - 1 in FORTRAN). Below this level, CLUTTER value of -88.88 is stored.

If the estimated rainfall rate exceeds 300 mm/h, it is reset to 300 mm/h. In V6, this reset is made before the mean of the distribution is calculated so that the mean never reaches 300 mm/h. If the substantial part of the distribution exceeds this threshold, a flag in flagRain is set. (The rainfall rate that corresponds to the epsilon at which the pdf p(epsilon) becomes one tenth of the maximum of p(epsilon) is larger than 300 mm/h, the flag is set.)

----
rainAve (New definition)

```
float32     rainAve[49][2];
REAL*4      rainAve(2,49)
```

Internally this quantity is stored as

int16 rainAve_scale[49][2] after multiplied (scaled) by 100.

rainAve[][0] :

rainAve(1,*) : Average of rainfall rate between 2 and 4 km. The unit is mm/h.

If the lowest bin processed is higher than 2 km, the average is taken between the lowest altitude and 4 km. In this case, the 6th bit in rainFlag is set. If the lowest bin processed is higher than 4 km, the average is not calculated. In this case, 0 is stored, and the seventh bit of rainFalg is set.

rainAve[][1] :

rainAve(2,*) : Integrated rainfall rate from the rain top to the bottom.

The new unit in V6 is (cm/h)*km, and NOT (mm/h)*km. This odd unit is adopted to avoid the overflow when the sum is stored as int16 after scaled by 100. (It is not possible to use different scale factors for rainAve[][0] and rainAve[][1]. Cases with overflow were found in V5.)

----
rainFlag (New definition)

```
int16         rainFlag[49];
INTEGER*2     rainFlag(49)
```

Rain flag for each angle bin (See flag_rain in Appendix 2.)

The default value is 0.

The following meanings are assigned to each bit in the 16-bit integer.

0: (bit 1) no rain
+1: (bit 1) rain possible (this bit is set even when rain is certain)
+2: (bit 2) rain certain
+4: (bit 3) zeta > zeta_th(=0.7) (PIA larger than approximately 4 dB)
+8: (bit 4) zeta is too large (zeta > zeta_max=5.0)
+16: (bit 5) stratiform
+32: (bit 6) convective
+64: (bit 7) bright band is detected
+128: (bit 8) warm rain
+256: (bit 9) rain bottom above 2 km
+512: (bit 10) rain bottom above 4 km
+1024: (bit 11) large part of rain rate pdf is above upper limit
+16384: (bit 15) data partly missing between rain top and bottom

12th to 14th bits are currently not used.

16th bit (sign bit) is not used either.

----

rainType (New variable)

int16 rainType[49];

INTEGER*2 rainType(49)

This is an exact copy of rainType in 2A23.

----

rangeBinNum (New definition)

int16 rangeBinNum[49][7];
INTEGER*2 rangeBinNum(7,49)

rangeBinNum[][0]: range bin number at the top of the interval that is processed as meaningful data in 2A-25. This is 4 range bins (1 km) above the first (highest) rain-certain range bin.

rangeBinNum[][1]: range bin number at the top of the surface clutter defined in 1B21.

rangeBinNum[][2]: range bin number at the actual surface.

rangeBinNum[][3]: range bin number of the bright band if it exits. If not, the range bin number of the phase transition height (estimated 0C height) is stored. (Read the note for freezH.)

rangeBinNum[][4]: the range bin number at which the path-integrated Z-factor first exceeds the given threshold. If the path-integrated Z-factor does not exceed the threshold, it is set to 79.

rangeBinNum[][5]: the range bin number at which the measured Z-factor is maximum. If no rain, it is set to 79.

rangeBinNum[][6]: the range bin number at the bottom of the interval that is processed as meaningful data in 2A-25. The attenuation-corrected Ze and rainfall rate R at this range bin are defined as nearSurfZ and nearSurfRain. See note below.

All these range bin numbers are indexed vertically from top to bottom with 0 at the highest elevation and 79 at the earth ellipsoid.
(All negative bin numbers are set to 0, and numbers larger than 79 are set to 79.)

Exception: If the actual surface is lower than the ellipsoid, the number in rangeBinNum[][2] may be larger than 79. This situation happens occasionally, especially over Indian Ocean where the geoid surface is lower than the model ellipsoid. The same situation may happen at a very low place over land, for example, at Dead Sea where the surface is nearly 400 m below the sea level.

Range bin numbers are unitless.

Note 1: rangeBinNum[][1] contains the range bin number that is the top of the possibly surface cluttered ranges. This number is larger than the bin number for the bottom of the clutter-free ranges by one.
Note 2: rangeBinNum[][6] contains the range bin number that corresponds to the bottom of the interval that is processed as meaningful data in 2A-25. This range bin is one bin above the top of the region that is cluttered either by the surface echo in the antenna mainlobe or by noise. In the former case, this number is identical to (rangeBinNum[][1]-1), but in the latter case, rangeBinNum[][6] is different from (rangeBinNum[][1]-1). The latter case may happen when the rain echo near surface becomes lower than the noise level by very large attenuation due to heavy rain. When rain is light and the echo near surface is below the noise level, the data smaller than the noise level are treated as valid data and the rainfall rate at that bin is set to zero.

----
reliab
int8 reliab[49][80];
BYTE reliab(80,49)

Reliability parameter at each range bin

The default value is 0.

Each bit in the byte indicates the status shown below:

lowest (first) bit : 0 : measured signal below noise
lowest (first) bit : rain
second bit : rain certain
third bit : bright band
forth bit : large attenuation
fifth bit : weak return (Zm < 20 dBZ)
sixth bit : estimated Z < 0 dBZ
seventh bit : main-lobe clutter or below surface

eighth bit : missing data

For example, if the first bit is 0, i.e., if the number is an even number, then the measured signal in that range bin is below the noise level (noise threshold).

The large attenuation flag is set below the height at which the integral of 0.2*ln(10)*beta*alpha*Zm^beta first exceeds the given threshold. In the version 6, the threshold is chosen that approximately corresponds to the attenuation of 4 dB.

----
scanStatus

```
PR_SCAN_STATUS      scanStatus;
```

See the description of the 1B21 Scan Status in the TSDIS INTERFACE CONTROL SPECIFICATION (ICS) Volume 3.

----
scanTime

```
float64     scanTime;
REAL*8      scanTime
```

See the description of the 1B21 Scan Status in the TSDIS INTERFACE CONTROL SPECIFICATION (ICS) Volume 3.

----
scLocalZenith (New variable)

```
float32     scLocalZenith[49];
REAL*4      scLocalZenith(49)
```

Local zenith angle of the satellite at the center of the footprint. The number corresponds to the incidence angle of the radar beam to the surface (ellipsoid surface).

----
sigmaZero (New variable)

```
float32     sigmaZero[49];
REAL*4      sigmaZero(49)
```

Exact copy of sigmaZero from 2A21.

----
spare (New definition)

```
float32     spare[49][2];
REAL*4      spare(2,49)
```

spare[49][0] : area of the likelihood function of epsilon.

spare[49][1] : standard deviation of the probability distribution function (pdf) of epsilon.

----
thickThPIZ (Obsolete. No data in V6)
int16 thickThPIZ[49];
INTEGER*2 thickThPIZ(49)

----
weightW (Obsolete. No data in V6. Can be calculated from epsilon and epsilon_0.)
float32 weightW[49];
REAL*4 weightW(49)

Internally this quantity was stored as

int16 weightW_scale[49] after multiplied (scaled) by 1000.

Weighting factor in the calculation of epsilon (SRT correction factor) in the hybrid method. .The number is always between 0 and 1 (inclusive).

Note that in V6, epsilon_0 is output instead of weightW. The relationship among epsilon, epsilon_0 and weightW is as follows.
epsilon = 1 + weightW*(epsilon_0 - 1)

----
xi (Obsolete. No data in V6. Can be calculated from zeta_sd and zeta_mn.)
float32 xi[49][2];
REAL*4 xi(2,49)

Normalized standard deviation of zeta and PIA_est

xi[][0]: xi = zeta_sd/zeta_mn

xi[][1]: nsd_l = normalized standard deviation of pia_est

When zeta_mn is less than 0.01, xi[][0] is set to 0.

When pia_mn is less than 0.1, zi[][1] is set to 0.

xi is unitless.

----
zeta
float32 zeta[49][2];
REAL*4 zeta(2,49)

Integral of 0.2*ln(10)*beta*alpha*Z^beta from rain top to the clutter-free bottom.

zeta[][0]: zeta = Integral of 0.2*ln(10)*beta*alpha*Z^beta from the rain top to the bottom (lowest altitude processed).

zeta[][1]: PIA_est = -10*(log10(1-zeta_cr))/beta where zeta_cr is a corrected zeta (This zeta_cr is calculated by using the value of epsilon in the first cycle of processing which is different from the final estimate of epsilon.).

zeta is always between 0 and 100, typically between 0 and 2. When it is larger than 5, the 4th bit of rainFlag is set.

zeta is unitless.

----
zeta_mn
float32 zeta_mn[49][2];
REAL*4 zeta_mn(2,49)

Mean of zeta and PIA of 9 adjacent (3x3) beams.
At scan edges, the mean is calculated of 6 beams. At the scan edges of the first and last scans of the granule, the mean is calculated from only 4 beams.

zeta_mn[][0]: zeta_mn = mean of zeta
zeta_mn[][1]: PIA_mn = mean of PIA_est

The range of output value is the same as zeta itself.

zeta_mn is unitless.

----
zeta_sd
float32 zeta_sd[49][2];
REAL*4 zeta_sd(2,49)

Standard deviation of zeta and PIA_est in 9 adjacent (3x3) beams.
At scan edges, it is calculated in 6 beams. At the scan edges of the first and last scans of the granule, the mean is calculated from only 4 beams.

zeta_sd[][0]: zeta_sd = standard deviation of zeta
zeta_sd[][1]: PIA_sd = standard deviation of pia_est

zeta_sd is unitless.

----
zmmax
float32 zmmax[49];
REAL*4 zmmax(49)

zmmax is the maximum value of measured Z-factor expressed in dBZ at each IFOV.
The unit is dBZ or 10 log of mm^6/m^3. The range of the variable is between 0 and 100. (Typically between 10 and 60.)

ZRParmA

```
float32      ZRParmA[49][5];
REAL*4       ZRParmA(5,49)
```

Internally this quantity is stored as

int16 ZRParmA_scale[49][5] after multiplied (scaled) by 100000.

Z-R parameter 'a' at nodal points.

R = a*Z^b.

'a' is given at five nodal points.

These values are stored in the first 5 elements of the array.

'a' values between the nodes are calculated by linear interpolation.

The range bin numbers of the nodes are stored in ParmNode[49][5] (ParmNode(5,49)).

ZRParmB

```
float32      ZRParmB[49][5];
REAL*4       ZRParmB(5,49)
```

Internally this quantity is stored as

int16 ZRParmB_scale[49][5] after multiplied (scaled) by 10000.

Z-R parameter 'b' at nodal points.

R = a*Z^b.

'b' is given at five nodal points.

The nodal points are the same as those for alpha.

'b' values between the nodes are calculated by linear interpolation.

The range bin numbers of the nodes are stored in ParmNode[49][5] (ParmNode(5,49)).

ZRParmNode (Obsolete. No data in V6)

```
int16          ZRParmNode[49][5];
INTEGER*2      ZRParmNode(5,49);
```

This parameter is absorbed in parmNode[49][5].

Appendix 1. Output data structure defined in the toolkit

```
/****************************************/
/* Define L2A-25 data structure.        */
/****************************************/

typedef struct
{
int8       mainlobeEdge;
int8       sidelobeRange[3];
} CFLAGS;

typedef struct
{
CFLAGS       clutFlag[CLUTFLAG_TBL_SIZE];
} CLUTTER_FLAGS;

typedef struct
{
float64       scanTime;
float32       geolocation[49][2];
PR_SCAN_STATUS scanStatus;
NAVIGATION    navigate;
float32       scLocalZenith[49];
int16         rain_scale[49][80];
float32       rain[49][80];
int8          reliab[49][80];
int16         correctZFactor_scale[49][80];
float32       correctZFactor[49][80];
int16         attenParmNode[49][5];
int16         attenParmAlpha_scale[49][5];
float32       attenParmAlpha[49][5];
int16         attenParmBeta_scale[49];
float32       attenParmBeta[49];
int16         ZRParmNode[49][5];
int16         parmNode[49][5];
float32       precipWaterParmA[49][5];
float32       precipWaterParmB[49][5];
int16         ZRParmA_scale[49][5];
float32       ZRParmA[49][5];
int16         ZRParmB_scale[49][5];
float32       ZRParmB[49][5];
float32       zmmax[49];
int16         rainFlag[49];
int16         rangeBinNum[49][7];
int16         rainAve_scale[49][2];
float32       rainAve[49][2];
int16         precipWaterSum_scale[49];
float32       precipWaterSum[49];
int16         weightW_scale[49];
float32       weightW[49];
float32       epsilon_0[49];
int16         method[49];
float32       epsilon[49];
float32       zeta[49][2];
```

```
    float32         zeta_mn[49][2];
    float32         zeta_sd[49][2];
    float32         xi[49][2];
    float32         sigmaZero[49];
    float32         freezH[49];
    int16           thickThPIZ[49];
    float32         nubfCorrectFactor[49][2];
    int16           qualityFlag[49];
    float32         nearSurfRain[49];
    float32         nearSurfZ[49];
    float32         pia2a25[49];
    float32         e_SurfRain[49];
    float32         pia[49][3];
    float32         errorRain[49];
    float32         errorZ[49];
    float32         spare[49][2];
    int16           rainType[49];
  } L2A_25_SWATHDATA;
```

Scale factors:

```
#define L2A25_RAIN              100
#define L2A25_CORRECTZFACTOR    100
#define L2A25_ATTENPARMALPHA    10000000
#define L2A25_ZRPARMA           100000
#define L2A25_ZRPARMB           10000
#define L2A25_RAINAVE           100
#define L2A25_LIQWATERSUM       1000
#define L2A25_ATTENPARMBETA     10000
#define L2A25_WEIGHTW           1000
#define L2A25_NEARSURFRAIN      100
#define L2A25_NEARSURFZ         100
```

Appendix 2. Parameters defined in "param_general_6.61.dat"

```
 1  /* parameter file for v6.4 of 2A25. Oct. 9 2002 */
 2  /* Do not delete empty lines. */
 3  /* Do not add or delete lines. Absolute line numbers are important. */
 4  1.0000   vratio[0]    /* Terminal velocity ratio at 0 km */
 5  1.0396   vratio[1]    /* Terminal velocity ratio at 1 km */
 6  1.0817   vratio[2]    /* Terminal velocity ratio at 2 km */
 7  1.1266   vratio[3]    /* Terminal velocity ratio at 3 km */
 8  1.1745   vratio[4]    /* Terminal velocity ratio at 4 km */
 9  1.2257   vratio[5]    /* Terminal velocity ratio at 5 km */
10  1.2806   vratio[6]    /* Terminal velocity ratio at 6 km */
11  1.3394   vratio[7]    /* Terminal velocity ratio at 7 km */
12  1.4026   vratio[8]    /* Terminal velocity ratio at 8 km */
13  1.4706   vratio[9]    /* Terminal velocity ratio at 9 km */
14  1.5440   vratio[10]   /* Terminal velocity ratio at 10 km */
15  1.6234   vratio[11]   /* Terminal velocity ratio at 11 km */
16  1.7283   vratio[12]   /* Terminal velocity ratio at 12 km */
17  1.8404   vratio[13]   /* Terminal velocity ratio at 13 km */
18  1.9597   vratio[14]   /* Terminal velocity ratio at 14 km */
19  2.0867   vratio[15]   /* Terminal velocity ratio at 15 km */
```

```
20   2.2219    vratio[16]      /* Terminal velocity ratio at 16 km */
21   2.3658    vratio[17]      /* Terminal velocity ratio at 17 km */
22   2.5189    vratio[18]      /* Terminal velocity ratio at 18 km */
23   2.6819    vratio[19]      /* Terminal velocity ratio at 19 km */
24   2.8554    vratio[20]      /* Terminal velocity ratio at 20 km */
25   0.075     lprate          /* 1/20 of the lapse rate per 250 m */
26   1.2       fhcf            /* Freezing height correction factor */
27   0.3       nsd_cnv[0]      /* Conv. factor of NSD for stratiform: was 0.3 */
28   0.5       nsd_cnv[1]      /* Conv. factor of NSD for convective: was 0.5 */
29   0.4       nsd_cnv[2]      /* Conv. factor of NSD for default: was 0.4 */
30   0.0000    nubf_cf[0]      /* Conv. coefficients for nubfCFs: was 1.30 */
31   0.0000    nubf_cf[1]      /*     nubfCFs = 1 + nubf_cf[0]*NSD^2      */
32   0.00      nubf_cf[2]      /*                   nubf_cf[1]*NSD^2*PIA_a   */
33   0.00      nubf_cf[3]      /* Conv. coefficients for nubfCFzr: was 0.17 in V5 */
34   0.10      zeta_min        /* Threshold value of zeta for SRT */
35   5.00      zeta_max        /* Threshold for zeta to judge something wrong. */
36   0.70      zeta_th_L       /* Threshold for zeta to judge large attenuation. */
37   0.00      z_offset        /* Offset to be added to 1C21 Z factor in dB     */
38   0.00      z_slope[0][0] /* Slope for ocean strat in clutter. + for larger Z toward surf. */
39   0.00      z_slope[0][1] /* Slope of dBZ/km for ocean conv. in cluttered range */
40   0.00      z_slope[0][2] /* Slope of dBZ/km for ocean others in cluttered range */
41 -0.50       z_slope[1][0] /* Slope of dBZ/km for land strat in clutter. */
42   0.00      z_slope[1][1] /* Slope of dBZ/km for land conv. in cluttered range */
43   0.00      z_slope[1][2] /* Slope of dBZ/km for land others in cluttered range */
44   1.00      epsi_init[0][0] /* initial offset factor of epsilon for ocean stratiform rain */
45   1.00      epsi_init[0][1] /* initial offset factor of epsilon for ocean convective rain */
46   1.00      epsi_init[0][2] /* initial offset factor of epsilon for ocean other rain */
47   1.00      epsi_init[1][0] /* initial offset factor of epsilon for land stratiform rain */
48   1.00      epsi_init[1][1] /* initial offset factor of epsilon for land convective rain */
49   1.00      epsi_init[1][2] /* initial offset factor of epsilon for land other rain */
50   0.08      atten_O2_surf     /* PIA due to O2 at 0 m from geoid */
51   7.70      scale_h_O2          /* scale height of O2 in km */
52   90.0      r_humid_in_rain /* relative humidity inside the raining area */
53   70.0      r_humid_out_rain   /* relative humidity outside the raining area */
54   2.00      scale_h_H2O        /* scale height of H2O in km */
```

Appendix 3. Parameters defined in "param_error_6.61.dat"

Note that many of the parameters defined in this file are not used in V6. The parameters marked with "n" are not used.

```
01 0.7       d_Zm_typ          /* Typical error (offset) in Zm in dB */                  o
02 0.5       d_alpha_typ       /* Typical error in alpha in dB */                        n
03 0.05      d_beta_typ        /* Typical error in beta in dB */                         n
04 0.6       d_zra_typ         /* Typical error in a in dB */                            o
05 0.05      d_zrb_typ         /* Typical error in b in dB */                            o
06 1.5       d_surf_typ        /* Typical error in sigma^0 in dB */                      n
07 1.0       d_nubfCf_s        /* Typical error in NUBF correct. factor for surf.*/      n
08 0.2       d_nubfCf_zr       /* Typical error in NUBF correct. factor for ZR */        o
09 -0.2      dv_dh             /* Typical change of velocity per km in dB*/              n
10 1.0       height_err_0C     /* 0C height error in km */                               o
11 0.3       height_err_BB     /* BB height error in km when it exists */                o
12 -0.07     dalpha_dT         /* Change of alpha per 1 degree in dB */                  n
13 0.01      dbeta_dT          /* Change of beta per degree in dB */                     n
```

```
14 0.02       da_dT              /* Change of a in ZR per degree in dB */                   o
15 -0.005   db_dT              /* Change of b in ZR per degree in dB */                   o
16 1.3       d_alpha_id       /* Error in alpha in dB caused by wrong ident. */          n
17 -0.05     d_beta_id        /* Error in beta in dB caused by wrong ident. */           n
18 2.00      d_zra_id         /* Error in a in dB caused by wrong identification */      o
19 0.02      d_zrb_id         /* Error in b in dB caused by wrong identification */      o
20 0.4     stddev_epsi_strat    /* nominal error of epsilon for strat in linear unit */   o
21 0.3     stddev_epsi_conv     /* nominal error of epsilon for conv in linear unit */    o
22   0.7      stddev_SRT_O     /* nominal error of SRT over ocean in dB */               o
23   2.2      stddev_SRT_L     /* nominal error of SRT over land in dB */                o
24   5.0      stddev_SRT_N     /* nominal error of SRT in const-Z method in dB */        o
```

Appendix 4. Parameters defined in "param_strat_1.dat"

```
 1   /* parameter file for v6 of 2A25. June 13, 2001 */
 2   /* Do not delete empty lines. */
 3   /* Do not add or delete lines. Absolute line numbers are important. */
 4   /* stratiform parameters */
 5   /* coefficients for R-Z relationship: R = a * Z^b
 6   log10(a) = zr_a_c0 + zr_a_c1*x + zr_a_c2*x*x
 7   log10(b) = zr_b_c0 + zr_b_c1*x + zr_b_c2*x*x
 8   where x = log10(alpha_final/alpha_initial)   */
 9   /* initial coefficients of k-Z relationship: k = alpha * Z^beta */
10   0.0000861 alpha_init[0][0]   /* alpha for low density snow, bb.011, strat */
11   0.0001084 alpha_init[0][1]   /* alpha for high density snow, bb.017, strat */
12   0.0004142 alpha_init[0][2]   /* alpha for bright band peak, bb.17, strat */
13   0.0002822 alpha_init[0][3]   /* alpha for rain (stratiform) 0C   */
14   0.0002851 alpha_init[0][4]   /* alpha for rain (stratiform) 20C */
15   0.79230     beta_init[0]         /* beta for stratiform column, strat */
16 /* R-Ze coefficients */
17 -1.8545 zr_a_c0[0][0]     /* stratiform, bb.011,    a'= 251.0 */
18 -1.8985 zr_a_c0[0][1]     /* stratiform, bb.017,    a'= 304.3 */
19 -2.3448 zr_a_c0[0][2]     /* stratiform, bb.17,     a'=1648.4 */
20 -1.6969 zr_a_c0[0][3]     /* stratiform, 0C ,        a'= 284.3 */
21 -1.6416 zr_a_c0[0][4]     /* stratiform, 20C,        a'= 276.1 */
22
23   1.6263 zr_a_c1[0][0]     /* stratiform, bb.011 */
24   1.6041 zr_a_c1[0][1]     /* stratiform, bb.017 */
25   1.4259 zr_a_c1[0][2]     /* stratiform, bb.17 */
26   0.9367 zr_a_c1[0][3]     /* stratiform, 0C */
27   0.9567 zr_a_c1[0][4]     /* stratiform, 20C */
28
29 -0.2734 zr_a_c2[0][0]     /* stratiform, bb.011 */
30 -0.2797 zr_a_c2[0][1]     /* stratiform, bb.017 */
31 -0.4191 zr_a_c2[0][2]     /* stratiform, bb.17 */
32 -0.7720 zr_a_c2[0][3]     /* stratiform, 0C */
33 -1.9319 zr_a_c2[0][4]     /* stratiform, 20C */
34
35 -0.1119 zr_b_c0[0][0]     /* stratiform, bb.011, b=0.7729, 1/b=1.294 */
36 -0.1167 zr_b_c0[0][1]     /* stratiform, bb.017, b=0.7644, 1/b=1.308 */
37 -0.1374 zr_b_c0[0][2]     /* stratiform, bb.17,   b=0.7288, 1/b=1.372 */
38 -0.1601 zr_b_c0[0][3]     /* stratiform, 0C ,      b=0.6917, 1/b=1.446 */
39 -0.1722 zr_b_c0[0][4]     /* stratiform, 20C,       b=0.6727, 1/b=1.487 */
40
41 -0.1040 zr_b_c1[0][0]     /* stratiform, bb.011 */
```

```
-0.0907 zr_b_c1[0][1]     /* stratiform, bb.017 */
-0.0235 zr_b_c1[0][2]     /* stratiform, bb.17 */
+0.0996 zr_b_c1[0][3]     /* stratiform, 0C */
+0.1116 zr_b_c1[0][4]     /* stratiform, 20C */

  0.1327 zr_b_c2[0][0]     /* stratiform, bb.011 */
  0.1275 zr_b_c2[0][1]     /* stratiform, bb.017 */
  0.1118 zr_b_c2[0][2]     /* stratiform, bb.17 */
  0.2811 zr_b_c2[0][3]     /* stratiform, 0C */
  0.4095 zr_b_c2[0][4]     /* stratiform, 20C */
/* LWC-Ze coefficients (log10 of a and b in LWC = a Ze^b) */
-2.4161 zl_a_c0[0][0]     /* stratiform, bb.011, a=0.00383613 */
-2.4881 zl_a_c0[0][1]     /* stratiform, bb.02,   a=0.00325046 */
-3.1290 zl_a_c0[0][2]     /* stratiform, bb.17,   a=0.00074301 */
-2.6994 zl_a_c0[0][3]     /* stratiform, 0C ,       a=0.00199806 */
-2.6502 zl_a_c0[0][4]     /* stratiform, 20C ,      a=0.00223787 */

  1.5422 zl_a_c1[0][0]     /* stratiform, bb.011 */
  1.8509 zl_a_c1[0][1]     /* stratiform, 0C */
  1.8344 zl_a_c1[0][2]     /* stratiform, 0C */
  1.5283 zl_a_c1[0][3]     /* stratiform, 0C */
  1.5422 zl_a_c1[0][4]     /* stratiform, 20C */

-0.2365 zl_a_c2[0][0]     /* stratiform, bb.011 */
-0.2254 zl_a_c2[0][1]     /* stratiform, 0C */
-0.3571 zl_a_c2[0][2]     /* stratiform, 0C */
-0.5889 zl_a_c2[0][3]     /* stratiform, 0C */
-1.6158 zl_a_c2[0][4]     /* stratiform, 20C */

-0.1471 zl_b_c0[0][0]     /* stratiform, bb.011, b=0.71266 */
-0.1520 zl_b_c0[0][1]     /* stratiform, bb.02,   b=0.70472 */
-0.1768 zl_b_c0[0][2]     /* stratiform, bb.17,   b=0.66564 */
-0.2122 zl_b_c0[0][3]     /* stratiform, 0C,        b=0.61342 */
-0.2243 zl_b_c0[0][4]     /* stratiform, 20C,       b=0.59658 */

-0.1056 zl_b_c1[0][0]     /* stratiform, bb.011 */
-0.0915 zl_b_c1[0][1]     /* stratiform, 0C */
-0.0442 zl_b_c1[0][2]     /* stratiform, 0C */
  0.0630 zl_b_c1[0][3]     /* stratiform, 0C */
  0.0751 zl_b_c1[0][4]     /* stratiform, 20C */

  0.1453 zl_b_c2[0][0]     /* stratiform, bb.011 */
  0.1357 zl_b_c2[0][1]     /* stratiform, 0C */
  0.1265 zl_b_c2[0][2]     /* stratiform, 0C */
  0.1913 zl_b_c2[0][3]     /* stratiform, 0C */
  0.4320 zl_b_c2[0][4]     /* stratiform, 20C */
```

Appendix 5.   Parameters defined in "param_conv_1.dat"

```
/* parameter file for v6 of 2A25. June 13, 2001 */
/* Do not delete empty lines. */
/* Do not add or delete lines. Absolute line numbers are important. */
/* convective rain parameters */
/* coefficients for R-Z relationship: R = a * Z^b
log10(a) = zr_a_c0 + zr_a_c1*x + zr_a_c2*x*x
```

```
log10(b) = zr_b_c0 + zr_b_c1*x + zr_b_c2*x*x
where x = log10(alpha_final/alpha_initial)   */
/* initial coefficients of k-Z relationship: k = alpha * Z^beta */
 0.0001273   alpha_init[1][0]    /* alpha for low density snow */
 0.0004109   alpha_init[1][1]    /* alpha for rain (convective) 0C   */
 0.0004109   alpha_init[1][2]    /* alpha for rain (convective) 0C   */
 0.0004109   alpha_init[1][3]    /* alpha for rain (convective) 0C   */
 0.0004172   alpha_init[1][4]    /* alpha for rain (convective) 20C */
 0.7713      beta_init[1]        /* beta for convective column, conv */
/* R-Ze coefficients */
-1.6932 zr_a_c0[1][0]    /* convective, bb.011, a=0.02027, a'= 174.09 */
-1.4579 zr_a_c0[1][1]    /* convective, 0C ,     a=0.03484, a'= 159.44 */
-1.4579 zr_a_c0[1][2]    /* convective, 0C ,     a=0.03484, a'= 159.44 */
-1.4579 zr_a_c0[1][3]    /* convective, 0C ,     a=0.03484, a'= 159.44 */
-1.3953 zr_a_c0[1][4]    /* convective, 20C ,    a=0.04024, a'= 147.43 */

 1.8122 zr_a_c1[1][0]    /* convective, bb.011 */
 0.8745 zr_a_c1[1][1]    /* convective, 0C */
 0.8745 zr_a_c1[1][2]    /* convective, 0C */
 0.8745 zr_a_c1[1][3]    /* convective, 0C */
 0.9377 zr_a_c1[1][4]    /* convective, 20C */

-0.5919 zr_a_c2[1][0]    /* convective, bb.011 */
-1.2688 zr_a_c2[1][1]    /* convective, 0C */
-1.2688 zr_a_c2[1][2]    /* convective, 0C */
-1.2688 zr_a_c2[1][3]    /* convective, 0C */
-2.5559 zr_a_c2[1][4]    /* convective, 20C */

-0.1217 zr_b_c0[1][0]    /* convective, bb.011, b=0.7556, 1/b=1.3234 */
-0.1792 zr_b_c0[1][1]    /* convective, 0C,       b=0.6619, 1/b=1.5108 */
-0.1792 zr_b_c0[1][2]    /* convective, 0C,       b=0.6619, 1/b=1.5108 */
-0.1792 zr_b_c0[1][3]    /* convective, 0C,       b=141   10.6619, 1/b=1.5108 */
-0.1915 zr_b_c0[1][4]    /* convective, 20C,      b=0.6434, 1/b=1.5542 */

-0.1235 zr_b_c1[1][0]    /* convective, bb.011 */
+0.0977 zr_b_c1[1][1]    /* convective, 0C */
+0.0977 zr_b_c1[1][2]    /* convective, 0C */
+0.0977 zr_b_c1[1][3]    /* convective, 0C */
+0.0986 zr_b_c1[1][4]    /* convective, 20C */

 0.1535 zr_b_c2[1][0]    /* convective, bb.011 */
 0.2375 zr_b_c2[1][1]    /* convective, 0C */
 0.2375 zr_b_c2[1][2]    /* convective, 0C */
 0.2375 zr_b_c2[1][3]    /* convective, 0C */
 0.4773 zr_b_c2[1][4]    /* convective, 20C */
/* LWC-Ze coefficients (log10 of a and b in LWC = a Ze^b) */
-2.2070 zl_a_c0[1][0]    /* convective, bb.011, a=0.00620868 */
-2.4070 zl_a_c0[1][1]    /* convective, 0C ,     a=0.00391752 */
-2.4070 zl_a_c0[1][2]    /* convective, 0C ,     a=0.00391752 */
-2.4070 zl_a_c0[1][3]    /* convective, 0C ,     a=0.00391752 */
-2.3522 zl_a_c0[1][4]    /* convective, 20C ,    a=0.004444 */

 2.0441 zl_a_c1[1][0]    /* convective, bb.011 */
 1.5269 zl_a_c1[1][1]    /* convective, 0C */
 1.5269 zl_a_c1[1][2]    /* convective, 0C */
 1.5269 zl_a_c1[1][3]    /* convective, 0C */
 1.5766 zl_a_c1[1][4]    /* convective, 20C */
```

```
64
65 -0.5818 zl_a_c2[1][0]     /* convective, bb.011 */
66 -1.0761 zl_a_c2[1][1]     /* convective, 0C */
67 -1.0761 zl_a_c2[1][2]     /* convective, 0C */
68 -1.0761 zl_a_c2[1][3]     /* convective, 0C */
69 -2.2027 zl_a_c2[1][4]     /* convective, 20C */
70
71 -0.1618 zl_b_c0[1][0]     /* convective, bb.011, b=0.68902 */
72 -0.2377 zl_b_c0[1][1]     /* convective, 0C,        b=0.57855 */
73 -0.2377 zl_b_c0[1][2]     /* convective, 0C,        b=0.57855 */
74 -0.2377 zl_b_c0[1][3]     /* convective, 0C,        b=0.57855 */
75 -0.2500 zl_b_c0[1][4]     /* convective, 20C,       b=0.56232 */
76
77 -0.1259 zl_b_c1[1][0]     /* convective, bb.011 */
78   0.0533 zl_b_c1[1][1]     /* convective, 0C */
79   0.0533 zl_b_c1[1][2]     /* convective, 0C */
80   0.0533 zl_b_c1[1][3]     /* convective, 0C */
81   0.0545 zl_b_c1[1][4]     /* convective, 20C */
82
83   0.1724 zl_b_c2[1][0]     /* convective, bb.011 */
84   0.2681 zl_b_c2[1][1]     /* convective, 0C */
85   0.2681 zl_b_c2[1][2]     /* convective, 0C */
86   0.2681 zl_b_c2[1][3]     /* convective, 0C */
87   0.5077 zl_b_c2[1][4]     /* convective, 20C */
```

Appendix 6. Parameters defined in "param_other_1.dat"

```
 1  /* parameter file for v6 of 2A25. June 13, 2001 */
 2  /* Do not delete empty lines. */
 3  /* Do not add or delete lines. Absolute line numbers are important. */
 4  /* parameters for other type of rain */
 5  /* coefficients for R-Z relationship: R = a * Z^b
 6  log10(a) = zr_a_c0 + zr_a_c1*x + zr_a_c2*x*x
 7  log10(b) = zr_b_c0 + zr_b_c1*x + zr_b_c2*x*x
 8  where x = log10(alpha_final/alpha_initial)    */
 9  /* initial coefficients of k-Z relationship: k = alpha * Z^beta */
10   0.0001273   alpha_init[2][0]     /* alpha for low density snow, 0.011, conv */
11   0.0001598   alpha_init[2][1]     /* alpha for low density snow, 0.017, others */
12   0.0004109   alpha_init[2][2]     /* alpha for rain (others) 0C    */
13   0.0004109   alpha_init[2][3]     /* alpha for rain (others) 0C    */
14   0.0004172   alpha_init[2][4]     /* alpha for rain (others) 20C */
15   0.7713        beta_init[2]           /* beta for others column, others */
16
17 -1.6932 zr_a_c0[2][0]     /* others, bb.011    */
18 -1.7280 zr_a_c0[2][1]     /* others, bb.017 */
19 -1.4579 zr_a_c0[2][2]     /* others, 0C */
20 -1.4579 zr_a_c0[2][3]     /* others, 0C */
21 -1.3953 zr_a_c0[2][4]     /* others, 20C */
22
23   1.8122 zr_a_c1[2][0]     /* others, bb.011 */
24   1.7697 zr_a_c1[2][1]     /* others, bb.017 */
25   0.8745 zr_a_c1[2][2]     /* others, 0C */
26   0.8745 zr_a_c1[2][3]     /* others, 0C */
27   0.9377 zr_a_c1[2][4]     /* others, 20C */
28
```

```
-0.5919 zr_a_c2[2][0]     /* others, bb.011 */
-0.6085 zr_a_c2[2][1]     /* others, bb.017 */
-1.2688 zr_a_c2[2][2]     /* others, 0C */
-1.2688 zr_a_c2[2][3]     /* others, 0C */
-2.5559 zr_a_c2[2][4]     /* others, 20C */

-0.1217 zr_b_c0[2][0]     /* others, bb.011, */
-0.1274 zr_b_c0[2][1]     /* others, bb.017 */
-0.1792 zr_b_c0[2][2]     /* others, 0C */
-0.1792 zr_b_c0[2][3]     /* others, 0C */
-0.1915 zr_b_c0[2][4]     /* others, 20C */

-0.1235 zr_b_c1[2][0]     /* others, bb.011 */
-0.1085 zr_b_c1[2][1]     /* others, bb.017 */
+0.0977 zr_b_c1[2][2]     /* others, 0C */
+0.0977 zr_b_c1[2][3]     /* others, 0C */
+0.0986 zr_b_c1[2][4]     /* others, 20C */

 0.1535 zr_b_c2[2][0]     /* others, bb.011 */
 0.1520 zr_b_c2[2][1]     /* others, bb.017 */
 0.2375 zr_b_c2[2][2]     /* others, 0C */
 0.2375 zr_b_c2[2][3]     /* others, 0C */
 0.4773 zr_b_c2[2][4]     /* others, 20C */
/* LWC-Ze coefficients (log10 of a and b in LWC = a Ze^b) */
-2.2070 zl_a_c0[2][0]     /* others, bb.011, a=0.00620868 */
-2.2699 zl_a_c0[2][1]     /* others, bb.02,   a=0.00537114 */
-2.4070 zl_a_c0[2][2]     /* others, 0C ,      a=0.00391752 */
-2.4070 zl_a_c0[2][3]     /* others, 0C ,      a=0.00391752 */
-2.3522 zl_a_c0[2][4]     /* others, 20C ,     a=0.004444 */

 2.0441 zl_a_c1[2][0]     /* others, bb.011 */
 1.9998 zl_a_c1[2][1]     /* others, bb.02 */
 1.5269 zl_a_c1[2][2]     /* others, 0C */
 1.5269 zl_a_c1[2][3]     /* others, 0C */
 1.5766 zl_a_c1[2][4]     /* others, 20C */

-0.5818 zl_a_c2[2][0]     /* others, bb.011 */
-0.5713 zl_a_c2[2][1]     /* others, bb.02 */
-1.0761 zl_a_c2[2][2]     /* others, 0C */
-1.0761 zl_a_c2[2][3]     /* others, 0C */
-2.2027 zl_a_c2[2][4]     /* others, 20C */

-0.1618 zl_b_c0[2][0]     /* others, bb.011, b=0.68902 */
-0.1675 zl_b_c0[2][1]     /* others, bb.02,   b=0.68004 */
-0.2377 zl_b_c0[2][2]     /* others, 0C,        b=0.57855 */
-0.2377 zl_b_c0[2][3]     /* others, 0C,        b=0.57855 */
-0.2500 zl_b_c0[2][4]     /* others, 20C,       b=0.56232 */

-0.1259 zl_b_c1[2][0]     /* others, bb.011 */
-0.1099 zl_b_c1[2][1]     /* others, bb.02 */
+0.0533 zl_b_c1[2][2]     /* others, 0C */
+0.0533 zl_b_c1[2][3]     /* others, 0C */
+0.0545 zl_b_c1[2][4]     /* others, 20C */

 0.1724 zl_b_c2[2][0]     /* others, bb .011 */
 0.1662 zl_b_c2[2][1]     /* others, bb .017 */
 0.2681 zl_b_c2[2][2]     /* others,   0C */
```

86 0.2681 zl_b_c2[2][3] /* others, 0C */
87 0.5077 zl_b_c2[2][4] /* others, 20C */

END
