FG03-95ER61986

SB3D USER MANUAL

Santa Barbara 3D Radiative Transfer Model

William O'Hirok (bill@icess.ucsb.edu)

01/99

Earth-Space Research Group

Institute for Computational Earth System Science

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Table of Contents

Introduction.....	3
Overview	3
SB3D files.....	7
input.root input.....	9
control.root file.....	15
Output files.....	19
Compiling SB3D	22
Running SB3D.....	23
Model notes.....	25

CAVEAT EMPTOR:

Almost any program of at least moderate complexity will contain some flaws. In most cases after a program has been tested through most of its configurations, it could be stated that the chance of encountering a flaw is "one in a million." Those odds are fine for most programs. However by the very nature of a stochastic based code, such as SB3D, the program is almost virtually guaranteed to encounter an existing bug even if the odds are more like "one in a billion". Hence, do not be surprised if at some time the program crashes by hitting one of those "impossible" bugs. Hopefully, most systematic errors have been found and eliminated. But, to sum up NOTHING is guaranteed!

INTRODUCTION

SB3D is a three-dimensional atmospheric and oceanic radiative transfer model for the solar spectrum. The microphysics employed in the model are the same as used in the model SBDART. It is assumed that the user of SB3D is familiar with SBDART and IDL. SB3D differs from SBDART in that computations are conducted on media in three-dimensions rather than a single column (i.e. plane-parallel), and a stochastic method (Monte Carlo) is employed instead of a numerical approach (Discrete Ordinates) for estimating a solution to the radiative transfer equation. Because of these two differences between SB3D and SBDART, the input and running of SB3D is more unwieldy and requires compromises between model performance and computational expense. Hence, there is no one correct method for running the model and the user must develop a sense to the proper input and configuration of the model.

As shown below, there are generally three methods for computing the solar flux for an atmosphere containing clouds. The plane parallel method (PPM) as employed in climate models (e.g. GCMs) and SBDART computes radiation assuming there is no horizontal variability in the atmospheric column. For instance, the flux for a partly cloud scene would be computed by deriving a mean cloud field, computing the flux for a plane-parallel cloud in an atmospheric column and computing the flux for the same column devoid of any cloud. The fluxes are weighted according to their horizontal coverage and combined to provide the total flux for the scene. The independent pixel approximation (IPM) also assumes plane-parallel clouds, but instead of performing the computation on a mean cloud field the cloud field is partitioned into many atmospheric columns. The flux is computed separately for each column and the results combined to provide a total domain average flux. For both the PPM and IPM, a photon can experience variations in optical thickness and atmospheric constituents microphysics in only the vertical direction. The three-

dimensional (3DM) method differs from the PPM and IPM by allowing photons to traverse horizontally. Hence, photons can encounter variations in optical depth and constituent microphysics in both the vertical and horizontal direction as done in nature. However, spatially the 3DM differs from nature, in that the model domain still has horizontal boundaries, and within the domain the atmosphere is discretized. To account for the horizontal boundary problem, photons exiting a model boundary returns at the opposing boundary with the same trajectory.

The Monte Carlo method is very simple in theory but can be difficult to employ when the media being investigated is multidimensional both in space and substance. Shown below is a diagram of the Monte Carlo method used in SB3D along with the governing equations.

Simply, a weighted photon (can be thought as a packet of photons) enters the model domain, travels a specified distance depending on the extinction coefficient, interacts with an atmospheric constituent (cloud, aerosol or gas), is dewighted by the single-scattering albedo of the constituent, and scattered along a trajectory defined by the phase function. This process is repeated until the photon exits the model domain or the weight of the photon is reduced to below a predefined threshold. Many more photons are needed until the model output converges to a solution within a certain tolerance (see convergence plot below). The total number of photons required depends on the spatial resolution of the desired solution (domain averages vs. column fluxes), the angular resolution (irradiance vs. radiance), the domain complexity (spatial resolution of the model domain), spectral resolution (number of wavelengths), wavelength and desired accuracy.

The time it takes to arrive at a solution is directly related to the number of photons processed and thus the factors mentioned above. While it may only take a matter of minutes to complete a run using SBDART, a run with SB3D can take anywhere from minutes to days to weeks. Along with time considerations, one must also be aware of memory requirements. Shown here is a representation of the model structure.

The model domain is referred to as the shell in the model code. The shell can be of any length, width or height. The shell is partitioned into what are called cells. Each cell represents an atmospheric volume which is completely homogeneous. Variability within the model is based on the distribution of these cells within the shell. Cells can also be of any length, width or height. The number of cells in the horizontal direction is unlimited, but limited to 511 in the vertical. The only limit on the number of cells, and thus the spatial resolution of the input field, is the random access memory available on the computer. The required memory is a multiple of the number of cells in the x direction * the y direction * the z direction. This multiple depends partially on the selected model output. Since there are many output options available the total number of cells cannot be directly determined. As a guide, a 128 x 128 x 64 shell can easily be run on a machine of 128 Mbytes.

There is no clear approach to setting the spatial structure of the model. Although, for example, it may be desirable to process a cloudy atmosphere using a structure of 200 x 200 100m cells in the horizontal and 60 layers in the vertical, the time for processing can be reduced to a sixteenth of the original processing time if a horizontal layer contains 50 x 50 400m cells. Additionally, processing time increases with the spatial size of the cell. Small cell sizes require a photon to traverse many cells between scattering events, thus increasing the computational time between scattering events and the eventual extinction of the photon. However, as processed in the code, adjacent cells that are homogeneous are skipped which reduces the above effect. Throughout the code many such algorithms are employed to reduce memory expense and computational time but these should be generally transparent to the user and only be a concern to those modifying the code.

An issue about the actual computer code that requires user interface is the setting of array sizes for the maximum number of cells in the x, y and z direction. The code is written in FORTRAN77 and does not employ dynamically allocated memory. It is impossible to account for every type of shell structure. For example, one run may require a cloud field of 100 x 100 x 33 cells while the other run may be simulating a flight transect with a cloud field of 4000 x 1 x 100 cells. Obviously, the array dimensions can not all be set for the maximum possible field size (i.e. 4000 x 4000 x 100). Hence, if the arrays are too small they must be modified by changing some values in the file 'sizefile.inc' and recompiling the source code. At the moment, the code can only be run on PCs and requires Microsoft FORTRAN PowerStation for compiling. The code does not need to be changed or modified for all runs, only those runs where the shell structure exceeds the default limits. The code will not run and will provide a warning if the shell structure is not properly set. More details on this issue follow in this document.

The best way of arriving at an optimum shell structure size is through some trial and error. The maximum number of cells is not an exact quantity and requires some user iteration. The model will not run if there is not enough memory available. However, as noted above, one may not want to run the maximum allowable model size since the computational time will likely be excessive. Rather a compromise must be made between size and processing time requiring experimentation to determine if the length of time until model convergence is acceptable. For very simple Monte Carlo simulations the number of photons required for predicting a desired level of error can be estimated through Bernoulli probability. However, such a determination is virtually impossible for most model runs and one must rely on the convergence criteria. The number of photons required for convergence (and thus processing time) is proportional to the square root of the number of photons used. Hence, one must carefully choose an minimum acceptable level of random error. Finally, the speed of convergence depends greatly on the wavelength being computed. For wavelengths with little absorption the processing time will be much less than for wavelengths where there is high gaseous or droplet absorption. In summation, the model should be "played" with in order to gather a sense to proper model configurations, convergence criteria and selection of inputs.

Many of the microphysical inputs to the model are the same as used in SBDART. Thus documentation for namelist INPUT is not provided here but can be found in document rt.doc. What follows is a list of the files required for running SB3D, the inputs not found in rt.doc, instructions on how to compile and run SB3D.

SB3D Files

To run SB3D a set of executable files, namelist control files, input files and a data file are required. The italicized files represent filenames that are named by the user.

Executable Files:

- Mcshell.exe* Program loops through a set of parameters (e.g. wavelength, solar zenith and azimuth angle and computational mode) calling the executable files *Mcprep* and *Mcrun* at each step.
- Mcprep.exe* Program preprocess model microphysics using same library routines as SBDART. Program must be run one wavelength at a time.
- Mcrun.exe* Program computes fluxes using Monte Carlo method from *Mcprep.exe* output. Program must be run one wavelength at a time.

Data Files:

- Miestorb.dat* Binary file contains pre-computed Mie scattering data for cloud droplets for use in *Mcprep.exe*. This file is not required, but it greatly speeds processing.

Namelist Files:

- Control.root* Namelist file for controlling model execution.
- Input.root* Namelist file for setting microphysics and model structure.

User Input Files:

- zscale.root* ASCII file containing scaling data along vertical axis. The length of the file corresponds to the number of model layers and the values are the thickness of each layer in kms. The lowest layer represents the surface and normally has a thickness of .001 km. This file will be generated if it does not exist and filled with a standard atmosphere scaling. The format is a single column of NZ length.
- mapset.root* ASCII file containing altitude (kms) of layers for output of flux data. See input variable NS documentation for details. This file will be generated if it does not exist. The format is a single column of length NS.
- radset.root* ASCII file containing radiance bin zenith and azimuth angle and cone size. See input variable NR for details. This file will be generated if it does not exist. The IDL format of the array indexing is `dum(3,NR)`. The columns represent zenith angle (deg.) azimuth angle (deg.) and cone size (deg.).
- wl.root* Optional ASCII file containing wavelengths for processing. This file is accessed if standard wavelength looping is not used. See variable MCWLSTRT for details. The format is a single column with the length equal to the absolute value of MCWLSTRT when it is set to a negative value.
- profile* Optional ASCII file containing an atmospheric profile. See variable PRFLEFNM for details. In IDL format the array indexing is `dum(4, NZ+1)`. The columns represent pressure (mb), temperature (K), water vapor (gm^{-3}) and ozone (gm^{-3}) in that order.

- clouda* Optional ASCII file containing 3D cloud distribution of liquid water or ice in units of concentration (gm^{-3}) or optical thickness. Up to 5 different files can be used. See variable NZS1 and example for details. In IDL format the array indexing is `dum(NZT1,NX,NY)`. The horizontal dimensions must match that of the entire model.
- cldrea* Optional ASCII file containing 3D cloud distribution of cloud effective radius (μm). For liquid water the numbers are positive and can vary internally from 2 to 128 μms . For ice, the number is negative and the size must be a constant single size throughout all clouds. The actual size is specified by the variable ICE (see namelist MCINPUT) regardless of the value contained in this file. Up to 5 different files can be used. See variable NZS1 and example for details. In IDL format the array indexing is `dum(NZT1,NX,NY)`. The horizontal dimensions must match that of the entire model.
- cloudb* Optional ASCII file containing 3D distribution of ice or drizzle in units of concentration (gm^{-3}) or optical thickness. Up to 5 different files can be used. See variable NZS2 for details. In IDL format the array indexing is `dum(NZT2,NX,NY)`.
- image* Optional file containing grid of surface reflectance values. In IDL format the array is `dum(NX,NY)`.

INPUT

input.root input

The default namelist for the file 'Input.root' is shown below in namelist INPUT and MCINPUT. Parameters in the namelist INPUT are the same as used in SBDART and the documentation can be found in SBDART's rt.doc.

```

&INPUT
  IDATM =          4
  PBAR =        -1.000000
  SCLH2O =       -1.000000
  UW =         -1.000000
  UO3 =        -1.000000
  ZCLOUD =    0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00
  TCLOUD =    0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00
  LWP =       0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00
  NRE =        8.000000      8.000000      8.000000      8.000000      8.000000
  RHCLD =     -1.000000
  XRSC =        1.000000
  XN2 =        -1.000000
  XO2 =        -1.000000
  XCO2 =       -1.000000
  XCH4 =       -1.000000
  XN2O =       -1.000000
  XCO =        -1.000000
  XNO2 =       -1.000000
  XSO2 =       -1.000000
  XNH3 =       -1.000000
  XNO =        -1.000000
  XHNO3 =     -1.000000
  ZAER =       0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00
  TAERST =     0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00    0.000000E+00
  JAER =        0            0            0            0            0
  IAER =        0
  V =          23.000000
  RHAER =     -1.000000
  TBAER =     -1.000000
  ABAER =       0.000000E+00
  WBAER =       0.000000E+00
  W3BAER =      0.000000E+00
  GBAER =       0.000000E+00
  ISALB =        0
  ALBCON =     7.000000E-02
  SC =         2.500000E-01    2.500000E-01    2.500000E-01    2.500000E-01
  /
&MCINPUT
  NX =          1
  NY =          1
  NZ =         33
  XSCALE =      1.000000
  YSCALE =      1.000000
  PRFLEFNM = -
  ALBGAS =      5.000000E-01
  NZS1 =        -1            -1            -1            -1            -1
  NZT1 =        -1            -1            -1            -1            -1
  MOD1C = -      -            -            -            -            -
  MOD1S = -      -            -            -            -            -
  NZS2 =        -1            -1            -1            -1            -1
  NZT2 =        -1            -1            -1            -1            -1
  MOD2C = -      -            -            -            -            -
  MOD2S =         0            0            0            0            0
  KMODE =         1
  EXTLIM =      1.000000E-05
  MIE =          T
  CLDTYPE =         1
  ICE =       -108.000000
  DRZ =       128.000000
  IMGFILE = -
  NR =          5

```

```

NMR =      0
NS =      5
OCNTYPE =  0
WIND =    5.000000
FOAMREF =  5.500000E-01
CCMICE =   F
CCMAER =   F
/

```

Namelist MCINPUT parameters:

NX designates the number of cells along the x axis. The minimum value is 1 and the maximum value must be 1 less than the parameter xsize listed in the compile instructions.

NY See NX.

NZ designates the number of layers along the vertical axis. The minimum value is 2 and the maximum value is 511. Compared to SBDART, the normal amount of layers is 34 since the surface is considered a layer of minimal thickness.

XSCALE is the length of all cells in the x dimension (km).

YSCALE is the length of all cells in the y dimension (km).

PRFLEFNM is the file name containing atmospheric profile data. The number of layers in the file must match the variable NZ. This file is read in when IDATM = -1.

ALBGAS is a pseudo single-scattering albedo for absorbing gas. Since gaseous absorption by definition has a single scattering albedo of 0., ALBGAS is set to a value (nominally .5) to reduce statistical variance in regions which may be surrounded by high gas concentrations. The direction of travel by the photon is not changed after a gas interaction. If ALBGAS is greater than 0., the gas extinction coefficient is rescaled by $1/(1 - \text{ALBGAS})$.

NZS1 is a five element array that designates the lowest layer for each 3D block of liquid water content (or ice) or optical thickness that is contained in up to five data files described by input file type 'clouda'. Five different data sets can be used as input to provide the means for creating five separate cloud decks. The actual filenames of these data sets are stored in the array MOD1C as shown below. If two cloud data sets have layers that overlap, the data set last read will be the only one used for the overlapped layer. Layers which are not overlapped remain the same. The data files are read in order of the left to right in the array MOD1C. A value of -1 in NZS1 means no data.

NST1 is a five element array that designates the thickness of each 3D block of liquid water (or ice) or optical thickness. The thickness is given number of layers. A value of -1 means no data.

MOD1C is a five element string array that contains the filename for each 3D block of liquid water or ice (in terms of concentration (gm^{-3}) or optical thickness) represented by file type 'clouda'. Filename is limited to 16 characters. A dash mark means no data files.

MOD1S is a five element string array that contains the filename for each 3D block of cloud droplet size distributions (file type 'cldrea') that correspond to the above cloud concentration. For each cloud concentration file there must exist a corresponding droplet size distribution file with the same dimensions. Filename is limited to 16 characters. A dash mark means no data files.

CLDTYPE designates the cloud concentration units contained in the MOD1C files. CLDTYPE =1 for optical thickness. CLDTYPE =2 for liquid water content (gm^{-3}).

Example: Stratus with overlying cirrus.

```

NX      = 10
NY      = 8
NZS1    =          2          12          -1          -1          -1
NZT1    =          3          2          -1          -1          -1
MOD1C   = 'fldlw.strat' 'fldlw.cir'      -      -      -
MOD1S   = 'fldre.strat' 'fldlw.cir'      -      -      -
CLDTYPE =2

```

This example shows that there are two separate data files. The first file 'fldlw.strat' contains a block of liquid water concentrations (gm^{-3}) that has the dimensions of 10 x 8 x 3 cells. The block will fill model layers 2, 3 and 4. The corresponding size distribution for each cell is contained in file 'fldre.strat'. The second file 'fldlw.cir' in MOD1C contains a second block of liquid water concentrations (gm^{-3}) that has the dimensions of 10 x 8 x 2 cells. This block will fill layers 12 and 13. The corresponding size distributions for each cell is contained in the file fldre.cir'.

NZS2 is a five element array that designates the lowest layer for each 3D block of ice or drizzle (in units of concentration (gm^{-3}) or optical thickness) that is contained in up to five data files described by input file type 'cloudb'. This input provides a means for mixing ice or drizzle into the cloud layers produced by file type 'clouda' above. However, this input allows only ice or drizzle of a single effective radius and allows no variation. If two cloud data sets produced by file type 'cloudb' having layers that overlap, the data set last read will be the only one used for the overlapped layer. The data files are read in order of the left to right in the array MOD2C. A value of -1 in NZS2 means no data.

NST2 is a five element array that designates the thickness of each 3D block of ice or drizzle. The thickness is given number of layers. A value of -1 means no data.

MOD2C is a five element string array that contains the filename (filetype 'cloudb') for each 3D block of ice or drizzle in units of concentration (gm^{-3}) or optical thickness. Filename is limited to 16 characters. A dash mark means no data files.

MOD2S is a five element integer array that designates if the corresponding file in MOD2C is drizzle or ice. A positive 1 equals drizzle. A negative 1 equals ice. A zero means no data.

ICE sets the effective radius for ice. This value should be a negative number. Default is -108 μm .

DRZ sets the drizzle drop size. This value is positive. Default is 128 μm .

Example: Cumulus-Nimbus with drizzle and cirrus shield.

```

NX      = 10
NY      = 8
NZS1   =          3          -1          -1          -1          -1
NZT1   =          11          -1          -1          -1          -1
MOD1C  =      'fldlw.nim' -          -          -          -          -
MOD1S  =      'fldre.nim' -          -          -          -          -
CLDTYPE = 2
NZS2   =          2          10          -1          -1          -1
NZT2   =          2          4          -1          -1          -1
MOD2C  =      'fldlw.drz' 'fldlw.ice' -          -          -          -
MOD2S  =          1          -1          0          0          0
ICE    = -108
DRZ    = 128

```

This example shows that there is a single type 'clouda' data file and two type 'cloudb' data files. The first file 'fldlw.nim' in MOD1C contains a block of liquid water concentrations (gm^3) that has the dimensions of 10 x 8 x 11 cells. The block will fill model layers 3 through 13. The corresponding size distribution for each cell is contained in file 'fldre.nim'. The first file in MOD2C 'fldlw.drz' contains a block of drizzle concentrations (gm^3) that has the dimensions of 10 x 8 x 2 cells. This block will mix in drizzle with an RE equal to 128 μm into layers 2, 3 and 4. The second file 'fldlw.ice' in MOD2S will mix in ice with an RE equal to 108 into layers 10, 11, 12 and 13.

MIE defines the phase function used for cloud droplets. Mie equal to T uses Mie theory (Wiscombe, 1980). Mie equal to F uses a Henyey-Greenstein approximation. Normally, MIE should be set to T. Henyey-Greenstein is provided for making comparisons to other models. Henyey-Greenstein is always used for aerosols. Running with MIE set to T will not increase processing time when the data file miestorb.dat is found within the working directory. If not present, then computations can take considerably more time if many different effective radii are used for a cloud layer. The difference in irradiance between MIE and HG computations is not more than a few percent, but for radiance there can be a much larger difference.

KMODE defines the type of gaseous K-distribution method to be used. KMODE equal to 0 is the same as SBDART KDIST set to 1. In this mode there is a single three term K-distribution used (as done in SBDART). KMODE equal to 1 accounts for overlapping absorption lines. SBDART does not account for overlap and it is unclear if there is any added benefit to using the overlap since the LOWTRAN7 K-distribution is only an approximation. Overlaps between absorption lines occur throughout the solar spectrum, but are likely only significant near 2.7 μm . At this wavelength, the absorption is already maximized and the solar input is relatively low so any error associated with not employing the overlap is probably minimal for broadband computations. For comparison with spectral measurements, however, the error may be unacceptable. The problem with using the overlap is that it can dramatically slow down processing time at certain wavelengths by a factor of 3.

EXTLIM sets the tolerance by which adjacent cells are assumed to be equal in total extinction coefficient. Default value is 1.E - 05. Raising the value may speed processing time by smoothing the field in areas of minor variability that have little effect on domain average flux. This value should be kept at the default, unless experimentation shows the increase in speed does not bias the results.

IMGFILE is a string variable holding the name for an albedo image (file type 'image') that is used to compute surface reflectance. These values do not vary with wavelength. This albedo is used when ISALB is set to 11. The size of the image must be NX x NY in size. A dash means no data.

NR

is the number of radiance bins used for computing TOA radiance as a domain average. Radiance is computed by summing all the photon weight exiting the TOA within a cone centered on a radiance vector defined by a radiance zenith and azimuth angle. The vector origin is located at the final photon scattering location and points toward the TOA. The azimuth angle uses mathematical coordinates where 0 degrees is East rather than North. The radiance vector directed from the east towards the west is considered to have a azimuth angle of 0 degrees. From the north towards the south the azimuth angle is 90 degrees. The radiance bin cone size, zenith and azimuth angle are defined in file 'radset.root.' Since the radiance represents a subset of the TOA upwelling irradiance many more photons must be processed for radiance to produce the same level of random error. The number of photons decreases with an increase in cone size but at the expense of radiance resolution.

NMR

is the number of radiance bins used for computing TOA radiance for each atmospheric column. Thus the radiance output will be a map of radiance. The zenith, azimuth and radiance cone size are the same as used for NR, but only the bins $1 - \text{NMR}$ are computed. This subset reduces memory requirements. Mapping radiance is very computationally expensive since the radiance is now partitioned among $(N_X \times N_Y)$ cells. As shown below, the coordinates used for mapping is not the last photon scattering location (*) in the horizontal plane (a), but the location where the inverted radiance vector intersects the surface (b). Hence, the radiance is mapped to where a satellite would "observe" the interaction, for example, with an cloud.

NS

is the number of layers to output for mapping fluxes (both atmospheric and oceanic) for output. Rather than mapping the upwelling and downwelling irradiance and absorption for all cells in the model domain, only those cells listed in the file 'mapset.root' are stored. The layers are referenced in terms of altitude (km) and the model finds the layer top that is closest to that altitude. The upwelling and downwelling flux represents the photons that have passed through the top boundary of a specified layer. Absorption is summed between each specified layer. It is inclusive of the lower layer and exclusive of the higher layer. For example, below is a eight layer atmosphere and a three layer ocean. Altitude (as designated from the file *zscale.root*) and depth (fixed in code) boundaries are

represented by the layer and shown as solid lines. NS in this case is set to 4 and the values in 'mapset.root' are

[-1,0,0,5.5,16.0]

The output will produce a 4 layers of output data for (NX * NY) number of gridded cells. Upwelling and downwelling flux will be stored for depth -.1 km, and altitude 0.0 km, 6.0 km and 15.0 km. Three sets of absorption grids will also be store. The first gridded set will sum the absorption of layers -1, -2 and -3. The second set will be the absorption of layers 1. The third set will sum the absorption of layers 2, 3 and 4. The fourth set will sum the absorption of layers 5, 6 and 7.

20.0	_____
	8
15.0	_____
	7
10.0	_____
	6
8.0	_____
	5
6.0	_____
	4
4.0	_____
	3
2.0	_____
	2
0.001	_____
0.00	_____
-0.001	_____
	-1
	-2
	-3

-0.10	_____

OCNTYPE designates type of ocean surface. If ocntype equals 0 then ocean surface is Lambertian and albedo is from SBDART library. If ocntype equals 1 then ocean has 3-D wave surface and is completely absorbing once a photon penetrates the ocean surface. Scattering can still occur from foam and the air-ocean interface. There is no internal water column scattering. If ocntype equals 2 then ocean has a 3-D wave surface and internal scattering from hydrosols and phytoplankton. Yellow substance, chlorophyll and particulate matter coefficients can only be altered within the code. A setting of 2 is very time consuming and is primarily useful for remote sensing in the visible wavelengths. Currently, the ocean parameters are set for the Celtic Sea. Note: the ocean component is still experimental! Within the ocean component there is no horizontal transport of photons nor horizontal distribution of ocean constituents.

WIND is the wind velocity (m/s) for computing 3-D ocean surface waves. Greater wind increases the angular distribution of waves and generally enhances scattering. The relationship is not straight forward since photons in the model can scatter off one wave and be intercepted by another. Default is 5 m/s.

FOAMREF is the reflectance of ocean foam as a fraction. Default is 0.55.

CCMICE designates if set to T, the model uses the CCM3 ice microphysics for clouds.

CCMAER designates if set to T, the model uses the CCM3 aerosol parameterization. At the moment it is not used.

Control.root file

The control file executes the SB3D model. It provides for some simple looping of input parameters. Three loops are provided. The outside loop controls the wavelength, the middle loop controls the processing mode, and the inner loop the solar geometry.

Namelist CONTROL parameters:

```
&CONTROL
MCWLSTRT = 2.500000E-01
MCWLEND = 4.000000
MCWLINC = 5.000000E-03
MCSZA = 0.000000 -1.000000 -1.000000 -1.000000 -1.000000 -1.000000
MCAZMTH = 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
MCMODE = 1 0
TOACELL = 33
RADPRC = F
MAPRAD = F
SMAPFLX = F
MAPFLX = F
MAPPTH = F
PHTN = 10000000
CONVERGE = 1
CVRGSTEP = 10000
CVRGTOL = 1.000000E-01
CVRGX = 1
CVRGY = 1
CVRGR = 1
FILEOUT = 100000
SCRNOUT = 1000
IRAN = 12345678
WATTOUT = F
/
```

MCWLSTRT designates the initial wavelength (μm) for processing. By setting this number to a negative value, wavelengths in file *wl.root* are used for processing. The number of wavelengths processed in file *wl.root* is equal to the absolute value of MCWLSTRT. Valid wavelength range is 0.25 – 5.00 μm .

MCWLEND designates the value for the final wavelength (μm) processed when wavelength looping is employed.

MCWLINC is the increment (μm) used between MCWLSTART and MCWLEND.

MCSZA is a six element array containing the solar zenith angles (deg.) to be processed. The model will increment through the array and process a run for the solar zenith indexed. A -1 prevents further looping of the solar zenith angle.

MCAZMTH is a six element array containing solar azimuth angles (deg.) that correspond to the index used in MCSZA. Valid range is from 0 to 360 degrees. The azimuth angle is expressed in mathematical terms. The solar beam directed from the east towards the west is considered to have a azimuth angle of 0 degrees. From the north towards the south the azimuth angle is 90 degrees.

MCMODE is a two element array describing the operating mode. MCMODE set to 1 uses the horizontal transport mode (3DM) and MCMODE set to 2 uses the independent pixel approximation (IPM). A -1 in the second index prevents further processing of different modes. Both modes employ the Monte Carlo method. For 3DM, a photon can travel horizontally until it reaches the edge of the model shell. At this point, the photon exits the shell and returns at the opposing boundary with the same trajectory, thus creating a cyclic boundary. For the IPM, a photon reaching the edge of a cell returns into the same cell at the opposing boundary with the same trajectory. Here, the cyclic boundary is for each cell and not the entire shell as is the case for the 3DM. The PPM mode is not an option since there are a multitude of methods for describing the average microphysical properties of a cloud field. Thus, the user must produce the average cloud field for input. The equivalent PPM mode would then be obtained by setting NX and NY to 1 and running the model in either IPM or 3DM. For faster processing for PPM, the XSCALE and YSCALE should have large values (e.g. 10000).

Example: Broadband test. - root is named 'test'

```
MCWLSTRT = .25
MCWLEND = 4.00
MCWLINC = .005
MCSZA = 0.00      30.0      60.0      -1.      -1.      -1.
MAZMTH = 90.      90.      90.0      -1.000000  -1.000000  -1.000000
MCMODE = 1        -1
```

This example will compute 751 wavelengths from .25 to 4.00 μms in increments of .005 μms . For each wavelength, the mode will be the 3DM. Three different solar zenith angles will be processed at 0, 30 and 60 degrees. The azimuth angle will be 90 for each run. Output file names will be preceded by the root name plus solar zenith angle, processing mode and azimuth angle for identification. The first output file will have name test00H90, followed by test30I90 and test60I90.

Example: Spectral test - root is named 'spectral'
File wl.spectral contains [.5,.93,1.51]

```
MCWLSTRT = -3
MCWLEND = 4.00
MCWLINC = .005
MCSZA = 50.0      50.0      -1      -1.      -1.      -1.
MAZMTH = 90.      180.      0.0      -1.      -1.      -1.
MCMODE = 1        2
```

This example will compute 3 wavelengths at .50, .93 and 1.50 μms . For each wavelength, the mode will be the 3DM followed by IPM. A single solar zenith angle of 50.0 degrees. Two azimuth angles will be processed at 90 and 180 degrees. Output file names will be preceded by the root name plus solar zenith angle, processing mode and azimuth angle for identification. The first output file will have the root name spect50H90, followed by spect50H18, spect50I90 and spect50I180.

TOACELL is the layer that represents the top of the atmosphere. Normally, TOACELL is set to NZ.

RADPRC processes radiance if set to T. To save processing time set RADPRC to F if radiance is not required. The maximum number of radiance bins is 50. Parameter rsize in the source code must be changed to a value equal to or greater than NR if more bins are needed. See compiling SB3D instructions.

MAPRAD maps radiance for output if set to T. This option is memory intensive and reduces the maximum number of cells available in the model shell. To use MAPRAD, the parameters mrsize and mrysize in the source code must be set equal to or greater than NX and NY, respectively. Additionally, parameter mrsz in the source code should be equal to or greater than NMR. Default values for mrsize and mrysize is 1. See compiling SB3D instructions.

- SMAPFLX** if set to T, produces upwelling and downwelling irradiance and absorption output for each cell in a layer designated in file 'mapset.root'. This option can be very memory intensive and the output should generally be viewed as qualitative rather than quantitative unless care has been used for setting convergence criteria (see below). The maximum number of layers that can be stored is set by the parameter, ssize in the source code and must be set equal to or greater than NS. Default value for ssize is 10. See compiling SB3D instructions.
- MAPFLX** if set to T, produces upwelling and downwelling irradiance and absorption output for every cell within the model shell. This option is very memory intensive and the output should generally be viewed as qualitative rather than quantitative unless care has been used for setting convergence criteria (see below). To use MAPFLX the parameters, mfxsize and mfysize in the source code must be set equal to or greater than NX and NY, respectively. See compiling SB3D instructions.
- MAPPTH** if set to T, produces pathlength and photon weight output for every cell. This output is of limited use, but provided for diagnostics. This option is very memory intensive and reduces the maximum number of cells available in the model shell. To use MAPPTH, the parameters, mrxsize and mrysize in the source code must be set equal to or greater than NX and NY, respectively. See compiling SB3D instructions.
- CONVERGE** defines the type of convergence to be used for terminating the photon process at a given wavelength. Four modes are available depending on the desired output from the model. Mode 1 converges when domain averaged TOA upwelling, surface downwelling fluxes and atmospheric absorption all individually change by less than the value indicated by CVRGTOL over three consecutive photon count intervals designated by CVRGSTEP. Mode 2 converges when domain averaged TOA radiance changes by less than the value indicated by CVRGTOL over three consecutive photon count intervals designated by CVRGSTEP. This mode converges on radiance bins numbered 1 through the value given by parameter CVRGR. Mode 3 converges using the same criteria as mode 1 but for individual atmospheric columns or groups of columns. This mode is used in conjunction with the option SMAPFLX. Convergence is on the downwelling flux of the lowest atmospheric layer in mapset.root, the upwelling flux of the highest atmospheric layer in mapset.root and the absorption between these two layers. Atmospheric columns can be grouped together as superpixels using the parameters CVRGX and CVRGY. Mode 4 works the same as Mode 3 but for radiance as in Mode 2. The maximum number of radiance bins for testing convergence for Mode 4 is three.
- CVRGTOL** sets the tolerance level for convergence. If set to negative value, CVRGTOL varies inversely with the solar constant. Thus, the CVRGTOL is lowest in the visible spectrum and highest in the near-infrared. This procedure is primarily used for broadband computations to speed processing. If CVRGTOL is negative it varies according to
$$\text{CVRGTOL} = - ((\text{CVRGTOL} * \text{solar constant} (0.46 \mu\text{m})) / \text{solar constant} (\text{wl})).$$
- CVRGSTEP** The number of photons processed between checks for convergence. This value should be a multiple of (NX * NY). The proper settings of CVRGTOL and CVRGSTEP depends on the minimum acceptable level of random error. Suitable values for CVRGTOL and CVRGSTEP can usually be obtained through experimentation and monitoring of the model's output status screen. Make sure CVRGSTEP is large enough to prevent convergence from occurring at a local minima.

CVRGR	provides the number of bins required to converge before convergence is reached. If CVRGR is set to NR then all radiance bins must converge. By setting to CVRGR to a value less than NR, only the bins 1 through CVRGR must converge. This setting can be used to provide a qualitative sense to the radiance in bins that receive too few photons for reliable quantitative estimates, but are still of interest. Hence, the order of the bins in the file <i>radset.root</i> is important if CVRGR is less than NR.
CVRGX	is the number of columns grouped into a 'superpixel' along the x axis for using in convergence modes 3 and 4. CVRGX can range from 1 to NX but must be an even multiple of NX. By setting CVRGX to a value greater than 1, the time for convergence is decreased since the number of photons intercepting the superpixel is increased. The number of columns used in the 'superpixel' for analysis should be equal to or less than size number of columns that are averaged together from the output files.
CVRGY	see CVRGX.
PHTN	is the total number of fully weighted photons to be processed before termination. This value is normally set much higher than the number of photons required for convergence.
FILEOUT	is the number of photons processed between calls to output data to temporary storage files. The value should be large enough so that processing is not slowed down due to excessive file transfers. All output data are stored to file whenever convergence criteria is met or the number of photons processed equals PHTN regardless to the setting of FILEOUT. The main use for FILEOUT is to insure the storage of data for very time consuming jobs if the run needs to be stopped before completion.
SCRNOUT	is the number of photons processed between outputting model status to screen.
IRAN	is an integer seed for the random number generator. Set to a large odd integer. The seed value must be changed where runs are conducted on the same model input and the outputs combined to reduce random error.
WATTOUT	if set to T, produces output in $\text{Wm}^{-2}\mu\text{m}^{-1}$ for irradiance and $\text{Wm}^{-2}\mu\text{m}^{-1}\text{str}^{-1}$ for radiance. If set to F, all fluxes are in percent of TOA input.

Output files:

When SB3D is run, a command line root file name is entered. This root is at the end of all input files except the italicized files listed in the section INPUT FILES. For output files, the root is at the beginning of the file name and appended automatically with the solar geometry and processing mode. Subsequent runs with the same root name, solar geometry and processing mode have their output appended to the original file. The exception are the map files noted below. These files are appended with a wavelength. All files are in ASCII for internal viewing and to reduce platform crossing problems. All floating data (real) are written in single precision.

**.tmp* Temporary files of output data. Generally not used unless a model run is stop before being completed.

iran.tmp Contains internal model data for debugging.

screen.tmp File is a screen capture of model output status for remote viewing.

error.tmp File contains a log of model errors for a CONTROL file and indicates if a single model run (i.e. single wavelength) was not processed correctly.

profile.root File is produced by model and contains atmospheric profiles used in the computations. The output variables in column order are:

real	layer number
real	layer base (km)
real	layer top (km)
real	layer pressure (mb)
real	temperature (K)
real	water vapor content (gm^{-3})
real	saturated water vapor content (gm^{-3})
real	ozone (gm^{-3})

root.result File contains column list of domain averaged results. All fluxes are expressed in percent of TOA input. Hence, to compute irradiance at a single wavelength, the value must be divided by 100.0 and multiplied by the solar constant. The solar constant needs to be adjusted for the cosine of the solar zenith angle. Multiple runs of the same model inputs at different wavelengths are appended together to form rows. The output variables in column order are:

real	wavelength (μm)
real	process mode (1 = 3DM, 2 = IPM)
real	solar zenith angle (deg)
real	azimuth angle (deg.)
real	photon count
real	atmospheric absorption (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	surface absorption (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	toa albedo (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	transmission to surface (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	atmospheric absorption by gas (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	atmospheric absorption by cloud (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	total pathlength/total photon weight
real	number of voided photons (counts) -see screen sample for explanation.

root.prf File contains atmospheric profiles of irradiance, absorption and pathlength. Multiple runs of the same model inputs at different wavelengths are appended together. The output variables in column order are:

real	layer
real	atmospheric absorption (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	atmospheric absorption by gas (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	atmospheric absorption by cloud (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	layer top upwelling flux (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	layer top downwelling flux (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	layer pathlength/total photon weight
real	number of photons interacting with layer/total photon weight

At the end of the profiles for each wavelength these data in row order are added:

real	wavelength (μm)
real	process mode (1 = 3DM, 2 = IPM)
real	solar zenith angle (deg.)
real	solar azimuth angle (deg.)
real	photon count

root.ocn File contains oceanic profiles of irradiance and absorption. Multiple runs of the same model inputs at different wavelengths are appended together. For each wavelength, the data shown for the end of the profile output is attached. The output variables in column order are:

real	ocean layer
real	ocean layer base depth from surface (m)
real	ocean layer top depth from surface (m)
real	oceanic absorption (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	layer top oceanic upwelling flux (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)
real	layer top oceanic downwelling flux (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$)

root.rad File contains radiance output in percent and the number of photons in a radiance bin. For this count (not the radiance) the weight of a photon is not considered. The counter is incremented by 1 each time a photon enters the radiance bin. However, for the radiance output (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}\text{str}^{-1}$), the weight of the photon is applied as is done for the downwelling and upwelling fluxes and absorption outputs. In IDL format the output array is (NR,2) with the first set of values (second index = 0) being radiance and the second set of values (second index = 1) being count. For each wavelength, the data shown for the end of the profile output is attached.

root.rmap File contains a mapped radiance output for the TOA in percent and the number of kilo-photons in a radiance bin. For this count the weight of photon is not considered. The counter is incremented by 1 each time a photon enters the radiance bin. However, for the radiance output (percent $\text{Wm}^{-2}\mu\text{m}^{-1}\text{str}^{-1}$), the weight of the photon is applied as is done for the downwelling and upwelling fluxes and absorption outputs. In IDL format the output array is (NX,NY,NR,2) with the first set of values (second index = 0) being radiance and the second set of values (second index = 1) being count. For each wavelength, the data shown for the end of the profile output is attached.

root.smap File contains downwelling and upwelling flux and absorption for the layers indicated in file 'mapset.root' when SMAPFLX is set to T. In IDL format the output array is (NX,NY,NS,4), with the first set of values (fourth index = 0) being absorption (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$), the second set (fourth index = 1) being upwelling flux at the top boundary of a cell (percent or

$\text{Wm}^{-2}\mu\text{m}^{-1}$), the third set (fourth index = 2) being downwelling flux at the top boundary of a cell (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$), and the fourth set (fourth index = 3) being the number of kilophotons processed in a cell. For this count the weight of photon is not considered. The counter is incremented by 1 each time a photon enters a cell. However, the weight of the photon is applied for the downwelling and upwelling fluxes and absorption. For each wavelength, the data shown for the end of the profile output is attached.

root.fmap File contains absorption data (total, gas and cloud), and upwelling and downwelling flux for every cell within the model shell when MAPFLX is set to T. Since this file can be very large for a single wavelength, a separate file is created for each wavelength processed. In IDL format the output array is (NX,NY,NZ,5), with the first set of values (fourth index = 0) being total absorption (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$), the second set (fourth index = 1) being absorption by gas (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$), the third set (fourth index = 2) being absorption by cloud droplet or ice (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$), the fourth set (fourth index = 3) being upwelling flux at the top boundary of a cell (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$), and the fifth set (fourth index = 4) being downwelling flux at the top boundary of a cell (percent or $\text{Wm}^{-2}\mu\text{m}^{-1}$). For each wavelength, the data shown for the end of the profile output is attached.

root.pmap File contains pathlength and photon number for every cell within the model shell when MAPPTH is set to T. Photon number is the total weight of all photons entering a cell. Since this file can be very large for a single wavelength, a separate file is created for each wavelength processed. In IDL format the output array is (NX,NY,NZ,2), with the first set of values (fourth index = 0) being pathlength (km), and the second set (fourth index = 1) being photon count. For each wavelength, the data shown for the end of the profile output is attached.

COMPILING SB3D

For most applications the only time the source code needs to be compiled is when the size parameters are changed to account for large model domains or for mapping fluxes. Size parameters are set in the include file 'sizefile.inc'. Once values are changed the files mcprep.f and mcrun.f need to be recompiled. Compiling is done through Microsoft PowerStation 4.0. Below are listed the files necessary for compiling the source code. These should all be contained in a single directory. Although the mclib.lib contains the same library routines as SBDART, a few modifications were required to the library in Miev0.f and taugas.f - so do not use the SBDART library. The modifications are listed in the source code. Additional source code includes mcshell.f for producing Mcshell.exe and mkmie.f for generating the Mie data set file 'miestorb.dat'.

Source Code files:

SB3D files [mcshell.f, mcprep.f, mcshell.f, mkmie.f]

SBDART library [atms.f, cloudpar.f, errpack.f, miev0.f, refice.f, refwat.f, runmie.f, sbwk.f, solirr.f, suralb.f, tauaero.f, taugas.f]

Library - mclib.lib

Makefiles

[mclib.mak, mclib.mdp, mcshell.mak, mcshell.mdp, mcprep.mak, mcprep.mdp, mcrun.mak, mcrun.mdp, mcmie.mak, mcmie.mdp]

Data file [sizefile.inc, angle.dat]

Changing size parameters and compiling:

Start Microsoft Powerstation 4.0

- 1) Under the File heading Open sizefile.inc, edit required size parameters, save and close file.
- 2) Under the File heading Open Workspace 'mcprep.mdp'.
- 3) Under the Tools heading choose Rebuild All.
- 4) Under the File heading Close Workspace.
- 5) Under the File heading Open Workspace 'mcrun.mdp'.
- 6) Under the Tools heading choose Rebuild All.
- 7) Under the File heading Close Workspace.
- 8) Copy mcprep.exe and mcrun.exe in the Release directory to the directory used for processing.

RUNNING SB3D

SB3D can be run under any PC using windows NT or windows 95/98. See instructions below for using ESRG mc computer farm.

1) To run SB3D for the first time, copy the files listed below to a working directory.

mcshell.exe
mcprep.exe
mcrun.exe

2) Open an MSDOS window and type mcshell.exe *root* (*root* is any name less than 8 characters)

If SB3D does not find the file input.*root* it will create the files listed below in the same directory. It will then stop.

input.*root*
control.*root*
zscale.*root*.
mapset.*root*.
radset.*root*.
wl.*root*.

4) Edit these files according to the SB3D documentation and create field input files (if required) *clouda*, *cloudb*, *cldre*, *profile* and *image*.

5) Delete all *.tmp files by typing del *.tmp. If the file 'error.tmp' is present the job will not run. This is equivalent to a 'Are you sure?' message but a bit more demanding.

5) To run a job type mcshell.exe *root*

If an error occurs immediately it is likely a problem related to the file input.*root* with the format of the namelist being corrupted. Check around the cloud file input area in namelist MCINPUT as the most likely source of the problem. If the problem cannot be corrected a new input.*root* file will need to be generated.

On screen the status of the run should appear. Be patient, the speed by which the screen updates depends on the settings in 'control.*root*' and the complexity of the simulation. The first set of screen output will contain information about the program mcprep.exe. Once that program is completed, status of mcrun.exe is shown. The screen for mcrun.exe also appears in the file 'screen.tmp'. If repeating a job and the original files are not needed, make sure these output files have been deleted, otherwise new output will be appended to the old data files. If the screen output seems to be quickly moving view the file 'error.tmp' in another window to check for errors. If the job needs to be halted do a CTRL-C in the running window. This may need to be done a few times since it must halt the program running (mcprep.exe and mcrun.exe) and the program mcshell.exe. A sample screen is shown on the following page.

Sample screen: Below is an example of the output provided on screen.

```
-----
3DM evaluation (W/m2/um)
Wavelength      =      .550
Photons processed =    194000.
Percent completed =      1.940
Photons voided   =        0.
atm abs   srf abs   toa upw   srf dnw
107.206   680.642  1095.252   741.440

First four TOA radiance (W/m2/um/str)
385.669   382.059   390.125   385.361
First four TOA radiance counts (k)
13.563    12.990    13.052    11.526
```

The screen shows that the model is being run in the 3DM mode at a wavelength of .55 μm . 194K photons (complete weight) have been processed thus far, representing 1.9% of the maximum number of photons allowed for this run. There have been no trapped photons as indicated by the number of photons voided. A photon is terminated if the number of scattering events for the photon exceeds a specified limit in the code in order to prevent infinite looping. This occurrence is extremely rare. Generally, number of photons voided will be zero or a very small number. If greater than, for example, 0.01% of the photon count, there may be a problem with the input or code. In $\text{Wm}^{-2} \mu\text{m}^{-1}$, the atmospheric absorption, surface absorption, TOA upwelling irradiance and downwelling irradiance to the surface are shown in the next line. If RADPRC is set to T for processing radiance then the next two output lines are shown. The first line shows the TOA radiance in $\text{Wm}^{-2} \mu\text{m}^{-1} \text{str}^{-1}$ for the first 4 radiance bins provided in the file 'radset.root'. The following line provides the number of photons that have been collected in the radiance bins in units of 1000. These counts represent every contribution (both partial and fully weighted photons) to the radiance bin. If convergence is accomplished the line below is shown.

```
Converge at count 3.000E+06 count 2.750E+06 and count 2.500E+06 within .500%
```

If no convergence is achieved a the message below is printed to screen.

```
'Process complete - no convergence'
```

Running on ESRG Mcfarm.

Six PCs are located in the computer room dedicated to processing Monte Carlo simulations. To use these computers a working directory and account must be set up on each machine. The simplest method for moving files between these machines and a local PC is through NT NETWORK NEIGHBORHOOD. Since these machines are NT CLIENTS they can be access for executing jobs only through TELNET using the ATAMAN software installed on each machine. The machines are called MC1, MC2, MC3, MC4, MC5 and MC6.

To run a job copy all required files from a local PC into the working directories of each of the machines. Open a TELNET window on the local PC for each MC# machine to be used. TELNET into a machine, provide account name and password and change to the working directory. Start the job as described above in running SB3D. Status screens should appear in each telnet window. Do not close the TELNET window. It can be shrunk to an ICON but not closed, otherwise the job will be terminated. A work around to this problem is to make the job a SERVICE, but this is very complicated. File 'screen.tmp' can be viewed from a remote location to check the status of a job.

MODEL NOTES

Error messages

Error messages are self-explanatory. To preserve memory, the model is run in single precision except in sensitive areas such as calls to SBDART libraries, SBDART libraries and for accumulating photon statistics. There are consistency checks in place to check for some precision problems. Although unlikely, if a consistency error occurs, the model skips the present wavelength and writes to an error log. A fix to the error may require modifying some source code parameters. A couple of hints on what to change is given at the beginning of mcrun.f and in the error message. An error which may occur at the start of a run relates to array sizes. Normally, this problem can be fixed by editing the include file 'sizefile.inc' and recompiling.

Convergence

As noted before some of the control inputs are subjective and require a 'sense' on their proper values. Compromises must be made on model resolution and the acceptable level of random error. This requires experience and observation on how the model converges through some monitoring of the output screen. Do not set the variable CVRGSTEP too low otherwise the model may seem to converge but is actually at a local minima and may drift to a different solution.

SBDART

Model results should be checked at times with runs from SBDART using a similar input. Obviously, the results will not be the same since SBDART cannot use a 3D cloud field input, but a comparison can check for gross errors. If gross errors occur, check both inputs carefully. Make sure SBDART has KDIST set to 1. Additionally differences can occur since the phase function in SB3D has more detailed structure in the forward scattering, it has K-distribution overlap, different counting methods in regards to fluxes and model layering than SBDART, and SB3D use a stochastic method that contains random error.

Cloud layers

Cloud layers entered in namelist INPUT will overwrite clouds entered in MCINPUT.

Cloud saturation

SB3D uses the same RHCLD parameter as SBDART but differs on its application. Rather than saturating an entire atmospheric layer, only those cells occupied by a cloud are saturated to the level indicated by RHCLD. To maintain the same total water vapor amount for the entire model domain, the cells outside of the cloud are dried to compensate for the increased water vapor in the clouds. The water vapor in the dry cells will never be reduced by more than 50% of their original value. To maintain consistency in this case, the water vapor in the cloudy cells will also be reduced until the original water vapor amount is achieved. Additionally, clouds are only allowed to saturate below the tropopause.

Radiance

Radiance computations are very sensitive to the number of photons processed and can take a very long time to converge. The smaller the radiance bin, the greater the number of photons required. For some viewing angles (high zenith angles) and radiance bin sizes, it may be computationally prohibitive to compute radiance. For most applications, however, a useful radiance can be produced as long as these problems are taken into account.

Aerosols

Aerosols are sensitive to the humidity of the air. For cloudy cells, the air is assumed to be saturated to a level designated by RHCLD. Hence, within a single layer that is partly cloudy, the aerosol microphysics may vary although the horizontal distribution of the aerosols is a constant. This procedure differs from SBDART where the water vapor content in a layer is a constant value.

Ocean

The ocean component is still experimental and care should be taken in its use.

Rayleigh Scattering

For proper operation, the rayleigh scattering adjustment XRSC should be greater than .0000001.