COSP user's manual
Version 1.3

A. Bodas-Salcedo
Met Office Hadley Centre
FitzRoy Rd., Exeter, EX1 3PB, United Kingdom

June 21, 2010

Contents

1 Introduction

The Cloud Feedback Model Intercomparison Project (CFMIP) Observation Simulator Package (COSP) is a modular piece of software whose main aim is to enable the simulation of data from several satellite-borne sensors from model variables. It is written almost entirely in Fortran 90 and it is conceptually divided into three steps. First, the gridbox-mean profiles are broken into subcolumns. Then, the vertical profiles of individual subcolumns are passed to individual instrument simulators (e.g. lidar forward model, ISCCP similator). Finally, a statistical module gathers the outputs from all the instruments and builds statistics that can be compared to similar statistics from observations.

The scheme that we use to break the grid-box mean profiles of cloud water contents is the Subgrid Cloud Overlap Profile Sampler (SCOPS), a technique developed for the International Satellite Cloud Climatology Project (ISCCP) simulator [*Klein and Jakob*, 1999; *Webb et al.*, 2001]. SCOPS uses a pseudo-random sampling process, fully consistent with the maximum, random and maximum/random cloud overlap assumptions used in many models [e.g. *Pincus et al.*, 2005]. Maximum overlap is applied to the convective cloud, and maximum/random is used for large-scale cloud. *Zhang et al.* [2010] have developed a simple algorithm that provides sub-grid distribution of precipitation fluxes compatible with the cloud distribution output by SCOPS and the gridbox mean precipitation fluxes simulated by the model.

The current version of COSP includes simulators for the following instruments: CloudSat radar [*Haynes et al.*, 2007], CALIPSO lidar [*Chepfer et al.*, 2008], ISCCP [*Klein and Jakob*, 1999; *Webb et al.*, 2001], the Multiangle Imaging SpectroRadiometer (MISR), and the Moderate Resolution Imaging Spectroradiometer (MODIS). The fast radiative transfer code RTTOV [*Saunders et al.*, 1999] can also be linked to COSP to produce clear-sky brightness temperatures for many different channels of past and current infrared and passive microwave radiometers. The Climate Model Output Rewriter (CMOR) library is used to write the ouputs to NetCDF files that comply with the Climate and Forecast (CF) Metadata Convention and fulfill the requirements of the climate community's standard model experiments. The Coupled Model Intercomparison Project Phase 5 (CMIP5) has requested COSP outputs to be included into a subset of CMIP5 experiments[1]. COSP is open source software and can be downloaded from the CFMIP website without charge[2].

The document is organised as follows. Section 2 provides information on the namelists that are used to configure COSP. Section 3 discusses how to set up the microphysical settings. Section 4 gives some details on the configuration of COSP for CFMIP-2 experiments. Appendix A shows the structure of the NetCDF input data files. This document is still under development, and therefore is not complete, although I hope it will still be useful in its current form. It is encouraged to read the README.txt file that is included with COSP, along with this user's manual.

2 Configuration: setting the COSP namelists

The user interaction with COSP is done via namelists. This section provides information on the namelists that are used to configure COSP.

2.1 COSP_INPUT namelist

This namelist is located in file `cosp_input_nl.txt`, and it contains the input arguments for COSP and all the simulators. Table 1 contains a description of the variables in this namelist.

---

[1] http://cmip-pcmdi.llnl.gov/cmip5/experiment_design.html

[2] http://www.cfmip.net

For details on RTTOV variables, please refer to RTTOV documentation.

Table 1: COSP_INPUT namelist.

General configuration variables	
CMOR_NL	Name of CMOR namelist (Section 2.2)
NPOINTS	Number of gridpoints to be processed. This has to coincide with the number of points of the NetCDF input file in 2D mode (lat*lon). For 1D (curtain) mode, there is no restriction.
NPOINTS_IT	Maximum number of gridpoints to be processed in one iteration. This helps to reduce the amount of memory used by COSP. If you find memory faults, reduce this number.
NCOLUMNS	Number of subcolumns used for each profile.
NLEVELS	Number of levels. This must be the same number as in the input NetCDF file.
USE_VGRID	If .false., the outputs are written on model levels. If this is set to .true., then a vertical grid evenly spaced in altitude is used. If .true., then you need to define number of levels with Nlr.
NLR	Number of levels in statistical outputs (only used if USE_VGRID = .true.)
CSAT_VGRID	Set to .true. for CloudSat vertical grid. This is just a standard grid of 40 levels evenly spaced at CloudSat vertical resolution, 480 m. This only applies if USE_VGRID=.true.)
FINPUT	Input NetCDF file. This is the input file with all the input variables to that your COSP executable will read and process.
Inputs related to radar simulations	
RADAR_FREQ	Frequency (GHz) used in the radar simulations.
SURFACE_RADAR	Radar position. surface=1, spaceborne=0
use_mie_tables	Use a precomputed lookup table? yes = 1,no = 0
use_gas_abs	Include gaseous absorption? yes = 1,no = 0.
do_ray	Calculate/output Rayleigh refl = 1, not = 0. This should be set to 0, as the Rayleigh reflectivity is not output by COSP.
melt_lay	Melting layer model off = 0, on = 1
k2	Dielectric factor of water. -1 = use frequency dependent default.
use_reff	True if you want effective radius to be used by radar simulator (always used by lidar)
use_precipitation_fluxes	.true., ! True if precipitation fluxes are input to the algorithm
Inputs related to lidar simulations	
Nprmts_max_hydro	Max number of parameters for hydrometeor size distributions

Naero	Number of aerosol species (Not used)
Nprmts_max_aero	Max number of parameters for aerosol size distributions (Not used)
lidar_ice_type	Ice particle shape in lidar calculations (0 = ice-spheres ; 1 = ice-non-spherical)
OVERLAP	Overlap type: 1 = max, 2 = rand, 3 = max/rand
Inputs related to ISCCP simulations	
ISCCP_TOPHEIGHT	1 = adjust top height using both a computed infrared brightness temperature and the visible optical depth to adjust cloud top pressure. Note that this calculation is most appropriate to compare to ISCCP data during sunlit hours. 2 = do not adjust top height, that is cloud top pressure is the actual cloud top pressure in the model. 3 = adjust top height using only the computed infrared brightness temperature. Note that this calculation is most appropriate to compare to ISCCP IR only algortihm (i.e. you can compare to nighttime ISCCP data with this option)
ISCCP_TOPHEIGHT_DIRECTION	direction for finding atmosphere pressure level with interpolated temperature equal to the radiance determined cloud-top temperature. 1 = find the *lowest* altitude (highest pressure) level with interpolated temperature equal to the radiance determined cloud-top temperature. 2 = find the *highest* altitude (lowest pressure) level with interpolated temperature equal to the radiance determined cloud-top temperature. This is the default value since V4.0 of the ISCCP simulator. ONLY APPLICABLE IF top_height EQUALS 1 or 3
Inputs related to RTTOV simulations	
Platform	Satellite platform number
Satellite	Satellite
Instrument	Instrument
Nchannels	Number of channels to be computed
Channels	Channel numbers (please be sure that you supply Nchannels)
Surfem	Surface emissivity (please be sure that you supply Nchannels)
ZenAng	Satellite Zenith Angle (degrees)
CO2	Mixing ratio of CO_2
CH4	Mixing ratio of CH_4
N2O	Mixing ratio of N_2O

CO	Mixing ratio of CO

2.2 CMOR namelist

The CMOR2 library is used to write the ouputs to NetCDF files that comply with the CF Metadata Convention and fulfill the requirements of the climate community's standard model experiments for CMIP5. The namelist CMOR is used to passed all the metadata that the calls to the CMOR library require. This namelist is located in file `cmor/cosp_cmor_nl.txt`, and Table 2 details its variables. It is expected that this namelist will be expanded in COSPv1.3, to include all the attributes that are required by the CMIP5.

Table 2: CMOR namelist.

INPATH	Directory where the MIP table is located.
OUTPATH	Directory where the outputs will be written.
START_DATE	Experiment start date.
MODEL_ID	String with your model name or id.
EXPERIMENT_ID	Type of experiment. This has to be one of those listed in the variable `expt_id_ok` in the MIP table.
INSTITUTION	Your institution.
SOURCE	Data source (e.g. model version, id of your model run).
CALENDAR	Calendar type used by the model.
REALIZATION	Realisation within an ensemble of runs for a given experiment.
CONTACT	Contact details.
HISTORY	What CMOR has done to the user supplied data (e.g., transforming its units or rearranging its order to be consistent with the MIP requirements). You can live this blank.
COMMENT	Extra comments that may help the interpretation of the data.
REFERENCES	Papers or other references describing the model.
TABLE	Name of the MIP table. This has to be consistent with the mode used to run COSP, which is defined by the input file. Different tables are needed for 1D and 2D models. The current list of table distributed with COSP are: CMIP5_cf3hr: MIP table for 1D mode. This is a modified version (with extra variables) of the CMIP5_cf3hr distributed with the CMOR2 library for the off-line CFMIP2 experiments. CMIP5_cf3hr.cmor1: the same as CMIP5_cf3hr, but to be used when linking COSP with CMOR1.3. COSP_table_2D: table to be used in 2D mode. COSP_table_2D.cmor1: same as COSP_table_2D, but to be used when linking COSP with CMOR1.3.

MAXTSTEPS	Maximum number of records that can be recorded to the output files. CMOR will issue an error and stop if you try to write more records.

2.3 COSP_OUTPUT namelist

This is the namelist that sets up output-related variables (see Table 3). It controls the instrument simulators that are run and the list of variables to be written to file. If a simulator is switched off, then none of its outputs are written out, independently of the status of the logical flags of the output variables associated with that particular simulator.

Table 3: COSP_OUTPUT namelist.

Logical flags that control which simulators are run	
Lradar_sim	
Llidar_sim	
Lisccp_sim	
Lmisr_sim	
Lmodis_sim	
Lrttov_sim	
Flags for ISCCP simulator outputs	
Lalbisccp	ISSCP Mean Cloud Albedo
Lboxptopisccp	Cloud Top Pressure in Each Column as Calculated by the ISCCP Simulator
Lboxtauisccp	Optical Depth in Each Column as Calculated by the ISCCP Simulator
Lclisccp	ISSCP Cloud Area Fraction
Lcltisccp	ISSCP Total Cloud Fraction
Lmeantbclrisccp	Mean clear-sky 10.5 micron brightness temperature as calculated by the ISCCP Simulator
Lmeantbisccp	Mean all-sky 10.5 micron brightness temperature as calculated by the ISCCP Simulator
Ltauisccp	Mean Optical Depth as Calculated by the ISCCP Simulator
Lpctisccp	ISSCP Mean Cloud Top Pressure
Flags for CALIPSO simulator outputs	
Latb532	Lidar Attenuated Total Backscatter (532 nm)
LcfadLidarsr532	CALIPSO Scattering Ratio CFAD
Lclcalipso2	CALIPSO Cloud Fraction Undetected by CloudSat
Lclcalipso	CALIPSO Cloud Area Fraction
Lclhcalipso	CALIPSO High Level Cloud Fraction
Lcllcalipso	CALIPSO Low Level Cloud Fraction
Lclmcalipso	CALIPSO Mid Level Cloud Fraction
Lcltcalipso	CALIPSO Total Cloud Fraction

LparasolRefl	PARASOL Reflectance
LlidarBetaMol532	Lidar Molecular Backscatter (532 nm)
Flags for CloudSat simulator outputs	
Lcfaddbze94	CloudSat Radar Reflectivity CFAD
Ldbze94	CloudSat Radar Reflectivity
Flags for CALIPSO-CloudSat combined outputs	
Lcltlidarradar	Lidar and Radar Total Cloud Fraction
Flags for other outputs	
Lfracout	Subcolumn output from SCOPS
Flags for MODIS simulator outputs	
Lclhmodis	MODIS High Level Cloud Fraction
Lclimodis	MODIS Ice Cloud Fraction
Lcllmodis	MODIS Low Level Cloud Fraction
Lclmmodis	MODIS Mid Level Cloud Fraction
Lclmodis	MODIS Cloud Area Fraction
Lcltmodis	MODIS Total Cloud Fraction
Lclwmodis	MODIS Liquid Cloud Fraction
Liwpmodis	MODIS Cloud Ice Water Path
Llwpmodis	MODIS Cloud Liquid Water Path
Lpctmodis	MODIS Cloud Top Pressure
Lreffclimodis	MODIS Ice Cloud Particle Size
Lreffclwmodis	MODIS Liquid Cloud Particle Size
Ltauilogmodis	MODIS Ice Cloud Optical Thickness (Log10 Mean)
Ltauimodis	MODIS Ice Cloud Optical Thickness
Ltautlogmodis	MODIS Total Cloud Optical Thickness (Log10 Mean)
Ltautmodis	MODIS Total Cloud Optical Thickness
Ltauwlogmodis	MODIS Liquid Cloud Optical Thickness (Log10 Mean)
Ltauwmodis	MODIS Liquid Cloud Optical Thickness
Flags for RTTOV outputs	
Ltbrttov	Mean clear-sky brightness temperature as calculated by RTTOV

3 Microphysical settings

This section discusses how to set up the COSP microphysical settings. This is is particularly important for the computation of the radar reflectivities as they are strongly dependent on the paricle size. This section should be read in conjunction with Section 4 of the *QuickBeam* User's Guide[3]. In the following discussion, let's assume that the particle size distribution (PSD), $n_x(D)$,

---

[3] http://reef.atmos.colostate.edu/haynes/radarsim/userguide.pdf

for a particle of diameter D, is defined as a gamma function:

$$n_x(D) = n_{ox} D^{\alpha_x} e^{-\lambda_x D}, \tag{1}$$

where n_{0x} is the intercept parameter, λ_x is the slope parameter, α_x is the constant shape parameter (x can be either R for rain, a for aggregates, c for ice crystals or g for graupel). For a single moment scheme, the intercept parameter is assumed constant or a simple function of λ_x

$$n_{ox} = n_{ax} \lambda_x^{n_{bx}} \tag{2}$$

where n_{ax} and n_{bx} are constants.

The terminal fall velocity of a precipitating particle, $V_x(D)$ can be expressed as a function of diameter:

$$V_x(D) = c_x D^{d_x} \left(\frac{\rho_0}{\rho} \right)^{\mathcal{G}_x} \tag{3}$$

where c_x, d_x, h_x and $\mathcal{G}_x$ are constants, ρ is the air density $[kg/m^3]$ and ρ_0 is a reference density of 1.29.

We assume a power law relating the mass of the particle to the diameter:

$$M_x(D) = a_x D^{b_x}. \tag{4}$$

The mass-diameter relation for rain simply assumes a spherical drop with a density equal to that for liquid water, 1000 kg m^{-3}.

3.1 Effective radius

COSP requires effective radius as input for CALIPSO and CloudSat. Default values can be used, although it is recommended to use values that are consistent with the model's microphysics. You can use the default values by setting to zero the input array of effective radii. The defaults are 30 μm for the lidar, and the values defined in `HCLASS_P1` for CloudSat (see details below). In order to compute the effective radius it is necessary to be able to infer the particle size distribution. This requires to being able to obtain the parameter λ_x from the model variables (specific humidities or precipitation fluxes).

The ith moment of the PSD is given by:

$$\mu_x^i = \int_0^\infty D^i n_x(D) dD = n_{ox} \frac{\Gamma(\alpha_x + i + 1)}{\lambda_x^{\alpha_x + i + 1}}. \tag{5}$$

When the hydrometeor mixing ratio is available, the value of λ_x is given by:

$$\lambda_x = \left(\frac{n_{ax} a_x \Gamma(b_x + 1 + \alpha_x)}{\rho q_x} \right)^{\frac{1}{b_x + 1 + \alpha_x - n_{bx}}}. \tag{6}$$

For precipitation fluxes, the flux can be related to the PSD by:

$$F_x = \int_0^\infty M_x(D) V_x(D) n_x(D) dD. \tag{7}$$

Using Eqs. (1, 2, 4), and solving this integral for λ_x gives:

$$\lambda_x = \left(\frac{a_x \; c_x \left(\frac{\rho_0}{\rho}\right)^{\mathcal{G}_x} n_{ax} \Gamma\left(\alpha_x + b_x + d_x + 1\right)}{F_x} \right)^{\frac{1}{\alpha_x + b_x + d_x - n_{bx} + 1}}. \tag{8}$$

The effective radius is then given by:

$$R_x = \frac{\mu_x^3}{2\mu_x^2} = \frac{\Gamma(\alpha_x + 4)}{2\Gamma(\alpha_x + 3)} \lambda_x^{-1} \tag{9}$$

3.2 Mixing ratios from precipitation fluxes

The radar reflectivities are computed from the hydormeteor mixing ratios. However, as large scale models typically diagnose precipitation fluxes, there exists the possibility of passing precipitation fluxes and let COSP convert them into mixing ratios before calling the radar simulator. The variable `use_precipitation_fluxes` in the `COSP_INPUT` namelist controls whether the COSP should do this conversion or use the input mixing ratios instead.

Expanding and integrating Eq. (3.1), the expression for the precipitation flux as a function of the mixing ratio and the parameters that define the PSD is given by:

$$F_x = \rho q_x \left(\frac{\rho_0}{\rho}\right)^{\mathcal{G}_x} c_x \frac{\Gamma(\alpha_x + b_x + d_x + 1)}{\Gamma(\alpha_x + b_x + 1)} \left(\frac{\rho q_x}{n_{ax} a_x \Gamma(\alpha_x + b_x + 1)} \right)^{\frac{d_x}{\alpha_x + b_x - n_{bx} + 1}}. \tag{10}$$

Solving for the mixing ratio gives:

$$q_x = \rho^{-1} \left[F_x \left(\frac{\rho}{\rho_0}\right)^{\mathcal{G}_x} \sigma \right]^{\frac{1}{\xi + 1}}, \tag{11}$$

where
$\xi = \frac{d_x}{\alpha_x + b_x - n_{bx} + 1}$,
$\Gamma_1 = \Gamma(\alpha_x + b_x + d_x + 1)$,
$\Gamma_2 = \Gamma(\alpha_x + b_x + 1)$,
$\sigma = \frac{\Gamma_2}{c_x \Gamma_1} (n_{ax} a_x \Gamma_2)^{\xi}$.

3.3 Setting the microphysical constants

The formulation presented here is available since COSP v1.3. The conversion is done by the subroutine `cosp_precip_mxratio`, which generalises the previous subroutine `pf_to_mr` that was only compatible with the method from *Khairoutdinov and Randall* [2003]. The microphysical constants needed for the precipitation are stored in `cosp_constants.f90`, along with the HCLASS table used for the reflectivity computations (see below). These two sets of constants have to be filled carefully with consistent constants. Table 4 lists the correspondence between FORTRAN names stored in `cosp_constants.f90` and the constants in used in this document.

If the formulation presented here is not compatible with your model's formulation, then you will have to set `use_precipitation_fluxes=.false.`, do the conversion off-line following your

FORTRAN name	COSP manual
`N_ax`	n_{ax}
`N_bx`	n_{bx}
`alpha_x`	α_x
`c_x`	c_x
`d_x`	d_x
`g_x`	g_x
`a_x`	a_x
`b_x`	b_x
`gamma_1`	Γ_1
`gamma_2`	Γ_2

Table 4. Correspondence between the FORTRAN names used in COSP and the formulation in used in this document.

model's fomulation, and fill in the arrays `gbx%mr_hydro(:,:,i)` with the precipitation mixing ratios in `cosp_test` (`i` is the index of each precipitation class: I_LSRAIN, I_LSSNOW, I_CVRAIN, I_CVSNOW, I_LSGRPL). The standard list of hydrometeors is defined in `cosp_constants.f90`:

```
integer,parameter :: I_LSCLIQ = 1
integer,parameter :: I_LSCICE = 2
integer,parameter :: I_LSRAIN = 3
integer,parameter :: I_LSSNOW = 4
integer,parameter :: I_CVCLIQ = 5
integer,parameter :: I_CVCICE = 6
integer,parameter :: I_CVRAIN = 7
integer,parameter :: I_CVSNOW = 8
integer,parameter :: I_LSGRPL = 9
```

3.4 Setting the HCLASS table

The microphysical assumptions for the radar simulation in COSP are stored in the HCLASS table, in `cosp_constants.f90`. The meaning of the HCLASS constants are given in the Quickbeam User's guide [*Haynes*, 2007]. For the sake of completeness, here we also give an overview and the settings. The HCLASS table consists of several lines, each one stored in a different variable. These variables are vectors with as many elements as number of hydrometeors so that the settings for each hydrometeor can be set up independently. These variables are:

- HCLASS_TYPE: Set to 1 for modified gamma distribution, 2 for exponential distribution, 3 for power law distribution, 4 for monodisperse distribution, 5 for lognormal distribution. Set to a negative number to ignore the hydrometeor class defined in that position.
- HCLASS_COL: Reserved for future use, value is ignored.

- HCLASS_PHASE: Set to 0 for liquid, 1 for ice.
- HCLASS_CP: Not used in COSP.
- HCLASS_DMIN: The minimum drop size for this class (μm), ignored for monodisperse.
- HCLASS_DMAX: The maximum drop size for this class (μm), ignored for monodisperse.
- HCLASS_APM: The a_x coefficient in in the mass-diameter relationship. If used, then set HCLASS_RHO to -1.
- HCLASS_BPM: The b_x coefficient in in the mass-diameter relationship. If used, then set HCLASS_RHO to -1.
- HCLASS_RHO: hydrometeor density [kgm^3]. If used, then set HCLASS_APM and HCLASS_BPM to -1.
- HCLASS_P1, HCLASS_P2, HCLASS_P3: these parameters depend on the type of distribution. For the modified gamma distribution used in the UM, P1 is the total particle number concentration, P2 is the particle mean diameter $[\mu m]$, and P3 is the distribution width, $\alpha_x + 1$. One of the parameters (P1,P2) must be specified, and the other one should be set to -1. P3 must be specified.

The settings for DMIN and DMAX are ignored in the current version for all distributions except for power law. Except when the power law distribution is used, particle size is fixed to vary from zero to infinity.

Since COSP v0.2, a capability of Quickbeam to pass the effective radius as input parameter is used. In that case, the settings in HCLASS_P[1-3] are defaults. If the input R_{eff} is zero at any spatial or hydrometeor coordinate at which there is condensate, then the HCALSS default is used. Hence, if the effective radius is not zero when there is hydrometeor present, the values in HCLASS_P2 are not used.

The default values in the COSP HCLASS table reflect those used by Roj Marchand to run the simulator for the MMF [*Marchand et al.*, 2009].

4 Configuration for CFMIP-2 experiments

The directory `./cfmip2` contains the namelists with the configuration for the CFMIP-2 experiments. These files are also available on the CFMIP web site. There are two different configurations:

- Long time series (*long_inline.txt). This is the configuration for the 30 yr monthly and daily means from ISCCP and CALIPSO/PARASOL. These are global gridded data computed from model gridded inputs, with the simulators run inline. The production version for these experiments is COSP v1.3. Experiments already run with v1.2.2 should be fine as long as the outputs look ok. COSP v1.3 includes a bug fix in the ISCCP simulator that may cause problem in some circumstances.

- Short time series (*short_offline.txt). This is the configuration for the 1 yr time series, both for the curtain outputs and global gridded monthly means from curtain outputs. Outputs from CloudSat and CALIPSO/PARASOL are requested. It is hoped that v1.3 will be the production version for these experiments. It will contain the final version of the MIP tables released by PCMDI.

5 Using your own cloud generator

Currently, COSP only includes treatment for cloud/precipitation overlap, but not subgrid variability. Please see Section 6.5 of the README.txt file if you require this extra capability.

Acknowledgements

COSP is a collaborative effort, and many people have been involved in the development of the software. Thanks to: M. J. Webb, S. Bony, H. Chepfer, J.-L. Dufresne, S. A. Klein, Y. Zhang, R. Marchand, J. M. Haynes, R. Pincus, and V. O. John.

Appendix A. Structure of the NetCDF input data files.

The structure of the input data NetCDF files are listed below. Examples of these files are distributed with COSP, namely, `cosp_input_um.nc` for 1D mode, and `cosp_input_um_2d.nc` for 2D mode. The 1D mode represents data along a trajectory, like the orbit track. The 2D mode is a gridded lat-lon input, suitable for model outputs.

This is the Common Data Language (CDL) structure of the COSP input NetCDF file in 1D mode:

```
netcdf cosp_input_um {
dimensions:
        point = 1236 ;
        level = 50 ;
        hydro = 9 ;
variables:
        short year(point) ;
                year:long_name = "year" ;
                year:_FillValue = -32767s ;
                year:units = "yr" ;
        byte month(point) ;
                month:long_name = "month" ;
                month:_FillValue = -127b ;
        byte day(point) ;
                day:long_name = "day" ;
```

```
        day:_FillValue = -127b ;
        day:units = "day" ;
byte hour(point) ;
        hour:long_name = "hour" ;
        hour:_FillValue = -127b ;
        hour:units = "hr" ;
byte minute(point) ;
        minute:long_name = "minute" ;
        minute:_FillValue = -127b ;
        minute:units = "min" ;
float second(point) ;
        second:long_name = "second" ;
        second:_FillValue = -1.e+30f ;
        second:units = "s" ;
float t(point) ;
        t:long_name = "t" ;
        t:_FillValue = -1.e+30f ;
        t:units = "min" ;
float tUM(point) ;
        tUM:long_name = "tUM" ;
        tUM:_FillValue = -1.e+30f ;
        tUM:units = "min" ;
float lst(point) ;
        lst:long_name = "lst" ;
        lst:_FillValue = -1.e+30f ;
        lst:units = "h" ;
float lon(point) ;
        lon:long_name = "longitude" ;
        lon:_FillValue = -1.e+30f ;
        lon:units = "degree_east" ;
float lat(point) ;
        lat:long_name = "latitude" ;
        lat:_FillValue = -1.e+30f ;
        lat:units = "degree_north" ;
float landmask(point) ;
        landmask:long_name = "landmask" ;
        landmask:_FillValue = -1.e+30f ;
        landmask:units = "1" ;
float orography(point) ;
        orography:long_name = "orography" ;
        orography:_FillValue = -1.e+30f ;
        orography:units = "m" ;
```

```
float psfc(point) ;
        psfc:long_name = "surface_pressure" ;
        psfc:_FillValue = -1.e+30f ;
        psfc:units = "Pa" ;
float height(level, point) ;
        height:long_name = "height_in_full_levels" ;
        height:_FillValue = -1.e+30f ;
        height:units = "m" ;
float height_half(level, point) ;
        height_half:long_name = "height_in_half_levels" ;
        height_half:_FillValue = -1.e+30f ;
        height_half:units = "m" ;
float T_abs(level, point) ;
        T_abs:long_name = "air_temperature" ;
        T_abs:_FillValue = -1.e+30f ;
        T_abs:units = "K" ;
float qv(level, point) ;
        qv:long_name = "specific_humidity" ;
        qv:_FillValue = -1.e+30f ;
        qv:units = "%" ;
float rh(level, point) ;
        rh:long_name = "relative_humidity_liquid_water" ;
        rh:_FillValue = -1.e+30f ;
        rh:units = "%" ;
float pfull(level, point) ;
        pfull:long_name = "p_in_full_levels" ;
        pfull:_FillValue = -1.e+30f ;
        pfull:units = "Pa" ;
float phalf(level, point) ;
        phalf:long_name = "p_in_half_levels" ;
        phalf:_FillValue = -1.e+30f ;
        phalf:units = "Pa" ;
float mr_lsliq(level, point) ;
        mr_lsliq:long_name = "mixing_ratio_large_scale_cloud_liquid" ;
        mr_lsliq:_FillValue = -1.e+30f ;
        mr_lsliq:units = "kg/kg" ;
float mr_lsice(level, point) ;
        mr_lsice:long_name = "mixing_ratio_large_scale_cloud_ice" ;
        mr_lsice:_FillValue = -1.e+30f ;
        mr_lsice:units = "kg/kg" ;
float mr_ccliq(level, point) ;
        mr_ccliq:long_name = "mixing_ratio_convective_cloud_liquid" ;
```

```
                mr_ccliq:_FillValue = -1.e+30f ;
                mr_ccliq:units = "kg/kg" ;
        float mr_ccice(level, point) ;
                mr_ccice:long_name = "mixing_ratio_convective_cloud_ice" ;
                mr_ccice:_FillValue = -1.e+30f ;
                mr_ccice:units = "kg/kg" ;
        float fl_lsrain(level, point) ;
                fl_lsrain:long_name = "flux_large_scale_cloud_rain" ;
                fl_lsrain:_FillValue = -1.e+30f ;
                fl_lsrain:units = "kg m^-2 s^-1" ;
        float fl_lssnow(level, point) ;
                fl_lssnow:long_name = "flux_large_scale_cloud_snow" ;
                fl_lssnow:_FillValue = -1.e+30f ;
                fl_lssnow:units = "kg m^-2 s^-1" ;
        float fl_lsgrpl(level, point) ;
                fl_lsgrpl:long_name = "flux_large_scale_cloud_graupel" ;
                fl_lsgrpl:_FillValue = -1.e+30f ;
                fl_lsgrpl:units = "kg m^-2 s^-1" ;
        float fl_ccrain(level, point) ;
                fl_ccrain:long_name = "flux_convective_cloud_rain" ;
                fl_ccrain:_FillValue = -1.e+30f ;
                fl_ccrain:units = "kg m^-2 s^-1" ;
        float fl_ccsnow(level, point) ;
                fl_ccsnow:long_name = "flux_convective_cloud_snow" ;
                fl_ccsnow:_FillValue = -1.e+30f ;
                fl_ccsnow:units = "kg m^-2 s^-1" ;
        float tca(level, point) ;
                tca:long_name = "total_cloud_amount" ;
                tca:_FillValue = -1.e+30f ;
                tca:units = "0-1" ;
        float cca(level, point) ;
                cca:long_name = "convective_cloud_amount" ;
                cca:_FillValue = -1.e+30f ;
                cca:units = "0-1" ;
        float Reff(hydro, level, point) ;
                Reff:long_name = "hydrometeor_effective_radius" ;
                Reff:_FillValue = -1.e+30f ;
                Reff:units = "m" ;
        float dtau_s(level, point) ;
                dtau_s:long_name = "Optical depth of stratiform cloud at 0.67 micron" ;
                dtau_s:_FillValue = -1.e+30f ;
                dtau_s:units = "1" ;
```

```
        float dtau_c(level, point) ;
                dtau_c:long_name = "Optical depth of convective cloud at 0.67 micron" ;
                dtau_c:_FillValue = -1.e+30f ;
                dtau_c:units = "1" ;
        float dem_s(level, point) ;
                dem_s:long_name = "Longwave emissivity of stratiform cloud at 10.5 micron" ;
                dem_s:_FillValue = -1.e+30f ;
                dem_s:units = "1" ;
        float dem_c(level, point) ;
                dem_c:long_name = "Longwave emissivity of convective cloud at 10.5 micron" ;
                dem_c:_FillValue = -1.e+30f ;
                dem_c:units = "1" ;
        float skt(point) ;
                skt:long_name = "Skin temperature" ;
                skt:_FillValue = -1.e+30f ;
                skt:units = "K" ;
        float sunlit(point) ;
                sunlit:long_name = "Day points" ;
                sunlit:_FillValue = -1.e+30f ;
                sunlit:units = "1" ;
        float u_wind(point) ;
                u_wind:long_name = "eastward_wind" ;
                u_wind:_FillValue = -1.e+30f ;
                u_wind:units = "m s-1" ;
        float v_wind(point) ;
                v_wind:long_name = "northward_wind" ;
                v_wind:_FillValue = -1.e+30f ;
                v_wind:units = "m s-1" ;
        float mr_ozone(level, point) ;
                mr_ozone:long_name = "mass_fraction_of_ozone_in_air" ;
                mr_ozone:_FillValue = -1.e+30f ;
                mr_ozone:units = "kg/kg" ;
        float emsfc_lw ;
                emsfc_lw:long_name = "Surface emissivity at 10.5 micron (fraction)" ;
                emsfc_lw:_FillValue = -1.e+30f ;
                emsfc_lw:units = "1" ;

// global attributes:
                :title = "COSP inputs UKMO N320L50" ;
                :Conventions = "CF-1.0" ;
                :description = "" ;
}
```

This is the CDL structure of the COSP input NetCDF file in 2D mode:

```
netcdf cosp_input_um_2d {
dimensions:
lon = 17 ;
lat = 9 ;
level = 38 ;
bnds = 2 ;
hydro = 9 ;
variables:
float lon(lon) ;
lon:axis = "X" ;
lon:units = "degrees_east" ;
lon:long_name = "longitude" ;
lon:bounds = "lon_bnds" ;
float lat(lat) ;
lat:axis = "Y" ;
lat:units = "degrees_north" ;
lat:long_name = "latitude" ;
lat:bounds = "lat_bnds" ;
float lon_bnds(lon, bnds) ;
float lat_bnds(lat, bnds) ;
float height(level, lat, lon) ;
height:units = "m" ;
height:long_name = "height_in_full_levels" ;
height:FillValue = -1.e+30f ;
float pfull(level, lat, lon) ;
pfull:units = "Pa" ;
pfull:long_name = "p_in_full_levels" ;
pfull:FillValue = -1.e+30f ;
float phalf(level, lat, lon) ;
phalf:units = "Pa" ;
phalf:long_name = "p_in_half_levels" ;
phalf:FillValue = -1.e+30f ;
float T_abs(level, lat, lon) ;
T_abs:units = "K" ;
T_abs:long_name = "air_temperature" ;
T_abs:FillValue = -1.e+30f ;
float qv(level, lat, lon) ;
qv:units = "kg/kg" ;
qv:long_name = "specific_humidity" ;
qv:FillValue = -1.e+30f ;
```

```
float rh(level, lat, lon) ;
rh:units = "%" ;
rh:long_name = "relative_humidity" ;
rh:FillValue = -1.e+30f ;
float tca(level, lat, lon) ;
tca:units = "1" ;
tca:long_name = "total_cloud_amount" ;
tca:FillValue = -1.e+30f ;
float cca(level, lat, lon) ;
cca:units = "1" ;
cca:long_name = "convective_cloud_amount" ;
cca:FillValue = -1.e+30f ;
float mr_lsliq(level, lat, lon) ;
mr_lsliq:units = "kg/kg" ;
mr_lsliq:long_name = "mixing_ratio_large_scale_cloud_liquid" ;
mr_lsliq:FillValue = -1.e+30f ;
float mr_lsice(level, lat, lon) ;
mr_lsice:units = "kg/kg" ;
mr_lsice:long_name = "mixing_ratio_large_scale_cloud_ice" ;
mr_lsice:FillValue = -1.e+30f ;
float mr_ccliq(level, lat, lon) ;
mr_ccliq:units = "kg/kg" ;
mr_ccliq:long_name = "mixing_ratio_convective_cloud_liquid" ;
mr_ccliq:FillValue = -1.e+30f ;
float mr_ccice(level, lat, lon) ;
mr_ccice:units = "kg/kg" ;
mr_ccice:long_name = "mixing_ratio_convective_cloud_ice" ;
mr_ccice:FillValue = -1.e+30f ;
float fl_lsrain(level, lat, lon) ;
fl_lsrain:units = "kg m^-2 s^-1" ;
fl_lsrain:long_name = "flux_large_scale_cloud_rain" ;
fl_lsrain:FillValue = -1.e+30f ;
float fl_lssnow(level, lat, lon) ;
fl_lssnow:units = "kg m^-2 s^-1" ;
fl_lssnow:long_name = "flux_large_scale_cloud_snow" ;
fl_lssnow:FillValue = -1.e+30f ;
float fl_lsgrpl(level, lat, lon) ;
fl_lsgrpl:units = "kg m^-2 s^-1" ;
fl_lsgrpl:long_name = "flux_large_scale_cloud_graupel" ;
fl_lsgrpl:FillValue = -1.e+30f ;
float fl_ccrain(level, lat, lon) ;
fl_ccrain:units = "kg m^-2 s^-1" ;
```

```
fl_ccrain:long_name = "flux_convective_cloud_rain" ;
fl_ccrain:FillValue = -1.e+30f ;
float fl_ccsnow(level, lat, lon) ;
fl_ccsnow:units = "kg m^-2 s^-1" ;
fl_ccsnow:long_name = "flux_convective_cloud_snow" ;
fl_ccsnow:FillValue = -1.e+30f ;
float orography(lat, lon) ;
orography:units = "m" ;
orography:long_name = "orography" ;
orography:FillValue = -1.e+30f ;
float landmask(lat, lon) ;
landmask:units = "1" ;
landmask:long_name = "land_mask" ;
landmask:FillValue = -1.e+30f ;
float height_half(level, lat, lon) ;
height_half:units = "m" ;
height_half:long_name = "height_in_half_levels" ;
height_half:FillValue = -1.e+30f ;
float psfc(lat, lon) ;
psfc:units = "Pa" ;
psfc:long_name = "surface_pressure" ;
psfc:FillValue = -1.e+30f ;
float Reff(hydro, level, lat, lon) ;
Reff:units = "m" ;
Reff:long_name = "hydrometeor_effective_radius" ;
Reff:FillValue = -1.e+30f ;
float dtau_s(level, lat, lon) ;
dtau_s:units = "1" ;
dtau_s:long_name = "Optical depth of stratiform cloud at 0.67 micron" ;
dtau_s:FillValue = -1.e+30f ;
float dtau_c(level, lat, lon) ;
dtau_c:units = "1" ;
dtau_c:long_name = "Optical depth of convective cloud at 0.67 micro" ;
dtau_c:FillValue = -1.e+30f ;
float dem_s(level, lat, lon) ;
dem_s:units = "1" ;
dem_s:long_name = "Longwave emissivity of stratiform cloud at 10.5 micron" ;
dem_s:FillValue = -1.e+30f ;
float dem_c(level, lat, lon) ;
dem_c:units = "1" ;
dem_c:long_name = "Longwave emissivity of convective cloud at 10.5 micron" ;
dem_c:FillValue = -1.e+30f ;
```

```
float skt(lat, lon) ;
skt:units = "K" ;
skt:long_name = "Skin temperature" ;
skt:FillValue = -1.e+30f ;
float sunlit(lat, lon) ;
sunlit:units = "1" ;
sunlit:long_name = "Day points" ;
sunlit:FillValue = -1.e+30f ;
float emsfc_lw ;
emsfc_lw:units = "1" ;
emsfc_lw:long_name = "Surface emissivity at 10.5 micron (fraction)" ;
emsfc_lw:FillValue = -1.e+30f ;
float mr_ozone(level, lat, lon) ;
mr_ozone:units = "kg/kg" ;
mr_ozone:long_name = "mass_fraction_of_ozone_in_air" ;
mr_ozone:FillValue = -1.e+30f ;
float u_wind(lat, lon) ;
u_wind:units = "m s-1" ;
u_wind:long_name = "eastward_wind" ;
u_wind:FillValue = -1.e+30f ;
float v_wind(lat, lon) ;
v_wind:units = "m s-1" ;
v_wind:long_name = "northward_wind" ;
v_wind:FillValue = -1.e+30f ;
}
```

References

Chepfer, H., S. Bony, D. Winker, M. Chiriaco, J.-L. Dufresne, and G. Sèze, Use of CALIPSO lidar observations to evaluate the cloudiness simulated by a climate model., *Geophys. Res. Lett.*, *35*, L15,704, doi:10.1029/2008GL034207, 2008.

Haynes, J. M., *QuickBeam radar simulation software*, Colorado State University, Fort Collins, CO, USA, v1.1 ed., 2007.

Haynes, J. M., R. T. Marchand, Z. Luo, A. Bodas-Salcedo, and G. L. Stephens, A multi-purpose radar simulation package: Quickbeam, *Bull. Am. Meteorol. Soc.*, *88*(11), 1723–1727, doi: 10.1175/BAMS-88-11-1723, 2007.

Khairoutdinov, M. F., and D. A. Randall, Cloud resolving modeling of the ARM summer 1997 IOP: Model formulation, results, uncertainties, and sensitivities, *J. Atmos. Sci.*, *60*(4), 607–625, doi:10.1175/1520-0469(2003)060¡0607:CRMOTA¿2.0.CO;2, 2003.

Klein, S. A., and C. Jakob, Validation and sensitivities of frontal clouds simulated by the ECMWF model., *Mon. Weather Rev.*, *127*(10), 2514–2531, 1999.

Marchand, R., J. Haynes, G. G. Mace, T. Ackerman, and G. Stephens, A comparison of simulated cloud radar output from the multiscale modeling framework global climate model with CloudSat cloud radar observations, *J. Geophys. Res.*, *114*, D00A20, doi: 10.1029/2008JD009790, 2009.

Pincus, R., C. Hannay, S. A. Klein, K.-M. Xu, and R. Hemler, Overlap assumptions for assumed probability distribution function cloud schemes in large-scale models, *J. Geophys. Res.*, *110*, D15S09, doi:doi:10.1029/2004JD005100, 2005.

Saunders, R., M. Matricardi, and P. Brunel, An improved fast radiative transfer model for assimilation of satellite radiance observations, *Q. J. R. Meteorol. Soc.*, *125*, 1407–1425, 1999.

Webb, M., C. Senior, S. Bony, and J. J. Morcrette, Combining ERBE and ISCCP data to assess clouds in the Hadley Centre, ECMWF and LMD atmospheric climate models, *Clim. Dyn.*, *17*, 905–922, 2001.

Zhang, Y., S. A. Klein, J. Boyle, and G. G. Mace, Evaluation of tropical cloud and precipitation statistics of CAM3 using CloudSat and CALIPSO data, *J. Geophys. Res.*, doi: 10.1029/2009JD012006, 2010.