**COUPLING CMEM WITH MESH**

**Kurt C. Kornelsen**

**McMaster University, 2014**

**Document Version 1.0**

**Amendments: NA**

This documentation will describe how the coupling of the Community Microwave Emission Model (CMEM) ver 4.1 with SA-MESH.

For a description of CMEM please visit <https://software.ecmwf.int/wiki/display/LDAS/CMEM> . The author of this MESH update was not involved in CMEM development and accepts no responsibility for errors that relate to CMEM. **If you are using CMEM+MESH please cite the original CMEM work as referenced at the ECMWF CMEM site** and {to be added when publication ready}.

For information on CMEM inputs please see the CMEM documentation.

**NOTE: At the current time not all options available in CMEM are available in MESH+CMEM. These differences include the LIEBE option, and model output (see OUTPUT below).**

If you have questions, need to report a bug or wish for help using MESH+CMEM please contact:

Kurt C. Kornelsen

[kornelkc@mcmaster.ca](mailto:kornelkc@mcmaster.ca)

1. **Coupling**

The CMEM functionality used for the coupling takes advantage of the CMEM ascii input options. CMEM operates independently for each grid cell and accepts all inputs as a 1D array. Therefore at each time step of MESH, all inputs from the MESH GAT arrays are re-written to CMEM input arrays for processing. There is therefore duplication of data in the arrays, however this has allowed the structure of CMEM to be unaltered. The MESH GAT arrays are of length ILG = NoGrids x NoGRU/CLASS Cells. However, during the CLASSG call only those grid cells/GRU’s that are relevant are written to the GAT array. The result is elements 1:NML are useful data and NML+1:ILG contain meaningless information. Therefore, the CMEM arrays are of length 1:NML. The MESH update has been written to be modular with minimal changes to the MESH driver. Most additional variables are stored in the MESH\_CMEM\_MODULE and as much as possible calls are made through the use of SUBROUTINEs, resulting in minimal driver and MESH source file changes.

The following minor alterations were made to the CMEM and MESH codes.

1. CMEM is designed as a modular code, and contains many modules and source files. However, some variables in these modules are used both in MESH and CMEM. To differentiate these variables/parameters in all CMEM source files N, PI, and K have been changed to N\_CMEM, PI\_CMEM and K\_CMEM respectively. If using the CMEM code from the MESH repository this change has already been made.

WARNING: In some CMEM files N is used as an internal counter as well. Therefore, use caution when changing N to N\_CMEM to ensure it is only the global parameter used by the module that is being changed.

1. In the file CMEM\_setup.F90 around Line 177, below the CASE(‘netcdf’) and before END SELECT insert

! > \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ADD FOR COUPLING WITH MESH\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CASE ('MESH')

CALL ABOR1('Wrong LSM Layer choice. When using MESH use default layers.')

!>=========================================================================

In the same file replace the final line of ‘3.1 check namelist option exist’ with

IF(CFINOUT/= 'gribapi'.AND.CFINOUT/= 'gribex'.AND.CFINOUT/= 'netcdf'.AND.CFINOUT/= 'ascii' .AND.CFINOUT/= 'MESH')& !KCK-> ADD MESH

& CALL ABOR1('Wrong CFINOUT choice. Choose gribapi, gribex, netcdf, ascii, or MESH') !> KCK -> ADD FOR MESH

Again the above substitutions are not required if using the CMEM 4.1 code from the MESH repository.

1. The following changes are required for the MESH driver. This will need to be done with each new release of SA\_MESH in the repository TRUNK.

At the beginning of the MESH driver include the following snippet at the end of the list of modules

!> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FOR CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USE PARKIND1, ONLY : JPIM, JPRM

USE YOMLUN, ONLY : NULOUT, NULTMP

USE YOMCMEMPAR

USE YOMCMEMFIELDS

USE YOMCMEMSOIL

USE YOMCMEMVEG, ONLY : wc\_veg, tb\_veg, w\_eff, bj, tauN, &

& tb\_veg, tau\_veg, t\_veg, a\_geo, a\_geoL, a\_geoH, tth, ttv

USE YOMCMEMATM, ONLY : tau\_atm,tb\_au,tb\_ad,tb\_toa,tb\_tov,fZ,tair &

&, fs\_tatm, fs\_spres, fs\_R, t\_atm, r\_atm, p\_atm, z\_atm, rh\_atm, ah

USE MESH\_CMEM\_MODULE

!>===============================================================================

Most of the MESH related variables that are related to CMEM are all stored in MESH\_CMEM\_MODULE.

In the section of the driver where the run options are printed for future use include:

!> \*\*\*\*\*\*\*\*\* ADD IF USING CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WRITE (58,\*) 'CMEMFLAG = ', CMEMFLAG !KCK ADDED FOR CMEM

DO N = 1,CMEMFLAG

WRITE (58,\*) 'CMEMTHETA = ', CMEMTHETA(N) !KCK ADDED FOR CMEM

ENDDO

WRITE (58,\*) 'CMEMR2COUTPUTFLAG = ',CMEMR2COUTPUTFLAG

!>=======================================================================

Following the call to the subroutine GATPREP we initialize CMEM.

!>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ADD FOR USING CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

!>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INITIALIZE CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

!> ADDED BY KCK

IF (CMEMFLAG > 0) THEN

CALL INIT\_MESH\_CMEM(ILG,NML,NA,NTYPE,NMTEST,IGND,ICAN,ILMOS,JLMOS, sl%ZBOT,ELEV)

END IF ! IF CMEM

!>=====================================================================

!> WRITE R2C FILE HEADERS

IF (CMEMR2COUTPUTFLAG == 1) THEN

CALL WRITE\_CMEM\_R2C\_HEADER(NMTEST,coordsys1,datum1,zone1,XORIGIN,YORIGIN, &

XDELTA,YDELTA,XCOUNT,YCOUNT)

ENDIF

!>======================================================================

Following this is a good place to move the section of the driver that writes ===SA\_MESH IS RUNNING ….STARTING MESH (PRECIP, EVAP, RUNOFF)====. This is not strictly required but makes the printed output look nicer.

Following the 430 CONTINUE in the driver insert

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ADD FOR RUNNING CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF (CMEMFLAG > 0) THEN

!> KCK - CALL CMEM SUBROUTINES

!>

!> CONVERT MESH VARIABLES TO CMEM FOR READING

CALL RDCMEMMESHSTATIC(NA,NTYPE,ILG,NMTEST,IGND,ICP1, &

ZSNOW, RHOSGAT, THLQGAT, TBARGAT, TAGAT, TSFSGAT, &

FCANGAT, SANDGAT, CLAYGAT)

! Loop for each incidence angle

DO M = 1,CMEMFLAG

CALL RUN\_MESHCMEM(M)

!> NATIVE CMEM OUTPUTS ARE:

! LEVEL 1OUTPUTS OF CMEM WILL BE TBH(K), TBV(K), TEFF\_H(K)

! ftb\_toa(JJ,1), ttb\_toa(JJ,2), fteffC(JJ,2)

! LEVEL 2 OUTPUTS OF CMEM WILL BE Tau\_Veg(H,V), Total bare soil fraction, Total VWC, Atmospheric optical depth &

! TB Atmosphere Upward

! ftau\_veg(JJ,1), ftau\_veg(JJ,2), ftfrac(JJ,1)+ftfrac(JJ,2), &

! fwc\_veg(JJ,1)\*(ftfrac(JJ,3)+ftfrac(JJ,4)) + fwc\_veg(JJ,2)\*(ftfrac(JJ,5)+ftfrac(JJ,6)), &

! ftau\_atm(JJ), ftb\_au(JJ)

! LEVEL 3 OUTPUTS OF CMEM WILL BE Pure (no snow) bare soil fraction, Pure(no snow) Low veg fraction, &

! Low Veg WC, High Veg WC, Low veg b param, High veg b param, Roughness param h, Wilting point SM, &

! Horizontal emissivity, vertical emissivity, C parameter

! ftfrac(JJ,1), ftfrac(JJ,3), fwc\_veg(JJ,1), fwc\_veg(JJ,2), fb(JJ,1),fb(JJ,2), &

! fh(JJ), fWP(JJ), fsurf\_emis(JJ,1), fsurf\_emis(JJ,2), fteffC(JJ,1)

!> FOR MESH THESE CANNOT ALL BE IMPLEMENTED. THEREFORE ONLY TBh, TBv, Tau\_h, and Tau\_v will be available.

DO JJ = 1,N\_CMEM !WRITE TO MESH GAT VARIABLES

TBH(JJ,M) = ftb\_toa(JJ,1)

TBV(JJ,M) = ftb\_toa(JJ,2)

TAUH(JJ,M) = ftau\_veg(JJ,1)

TAUV(JJ,M) = ftau\_veg(JJ,2)

END DO

END DO ! END OF CMEMFLAG LOOP

! WRITE CMEM VALUES TO R2CDATA

IF (CMEMR2COUTPUTFLAG == 1) THEN

IF(NR2CFILES\_CMEM > 0 .AND. MOD(NCOUNT\*30,DELTR2C\_CMEM) == 0)THEN

CALL FIND\_MONTH (IDAY, IYEAR, ensim\_month)

CALL FIND\_DAY (IDAY, IYEAR, ensim\_day)

CALL WRITE\_CMEM\_R2C\_DATA(NML,NLTEST,NMTEST,NCOUNT,IMIN,ACLASS, &

NA,XXX,YYY,XCOUNT,YCOUNT,ILMOS,JLMOS,ILG,&

NR2C\_CMEM,NR2CFILES\_CMEM,R2CFILEUNITSTART\_CMEM,GRD\_CMEM,GAT\_CMEM, &

GRDGAT\_CMEM,NR2CSTATES\_CMEM,R2C\_ATTRIBUTES\_CMEM,FRAME\_NO\_CMEM,IYEAR,&

ensim\_MONTH,ensim\_DAY,IHOUR,IMIN,ICAN, &

ICAN+1,CMEMFLAG,TBH,TBV,TAUH,TAUV)

FRAME\_NO\_CMEM = FRAME\_NO\_CMEM + 1 !UPDATE COUNTERS

ENDIF

ENDIF

!

ENDIF !IF CMEMFLAG > 0

!>==================END OF CMEM SECTION=======================================================

1. CMEM requires the leaf area index which is included in MESH as an internal variable of the subroutine CLASSA named AIL. To make it available outside of CLASSA, AIL is included in MESH\_CMEM\_MODULE. In the CLASSA code at the beginning of the code prior to IMPLICIT NONE insert

!>\*\*\*\*\*\*\*\*\*REQUIRED FOR CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USE MESH\_CMEM\_MODULE, ONLY: AIL

!>=============================================================

If CMEMFLAG > 0 AIL is allocated in INIT\_MESH\_CMEM. Otherwise it should be allocated here by inserting the following immediately after the declarations in CLASSA.

!> \*\*\*\*TO BE ADDED FOR CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IF(.NOT. ALLOCATED(AIL)) ALLOCATE(AIL(ILG,IC))

!>=============================================

This also requires that you comment out AIL from the declarations in CLASSA.

1. The final changes are to be made in READ\_RUN\_OPTIONS.F90. First we again add the modules.

Following this we need to initialize these values by adding the following section where the FLAGS are initialized.

!>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ADDED FOR CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

USE MESH\_CMEM\_MODULE, ONLY : CMEMCOUNT, CMEMFLAG,

+ CMEMR2COUTPUTFLAG, CMEMTHETA

!>=========================================================

!> FLAGS FOR CMEM

!\* If CMEMFLAG is 0 CMEM does not run

!\* If CMEMFLAG > 0 the number of angles simulated by CMEM will be equal to CMEMFLAG

!\* CMEMTHETA is the angles to be used and should have the same number as CMEMFLAG

CMEMFLAG = 0

CMEMCOUNT = 1

CMEMR2COUTPUTFLAG = 0

At the end of the ELSE IF blocks where IRONAME AND IROVAL are prior to the ELSE insert the following code block:

!> \*\*\*\*\* ADD FOR CMEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ELSE IF (IRONAME == "CMEMFLAG") THEN

CMEMFLAG = IROVAL

ALLOCATE(CMEMTHETA(CMEMFLAG))

ELSE IF (IRONAME == "CMEMTHETA") THEN

IF(CMEMFLAG == 0) THEN

WRITE(6,\*) "CMEMTHETA has been found in the run options",

1 " file. CMEMFLAG is set to 0. To use CMEM set",

2 " CMEMFLAG to greater than 0 prior to use of ",

3 " CMEMTHETA."

STOP

END IF

IF(CMEMCOUNT > CMEMFLAG) THEN

WRITE(6,\*) "There are more CMEMTHETA values to read",

1 " then there are values in CMEMFLAG. Ignoring",

2 " all CMEMTHETA values greater than CMEMFLAG."

ELSE

CMEMTHETA(CMEMCOUNT) = REAL(IROVAL)

CMEMCOUNT = CMEMCOUNT + 1

END IF

ELSE IF (IRONAME == "CMEMR2COUTPUTFLAG") THEN

IF(CMEMFLAG == 0) THEN

WRITE(6,\*) "ERROR: CMEMR2COUTPUT is not available without",

1 "CMEMFLAG > 0. Specify CMEMFLAG > 0 first"

STOP

ELSE

CMEMR2COUTPUTFLAG = IROVAL

END IF

!>===========================================================

Once these changes are made the coupling should be complete.

**Most of these changes are already implemented and can be found in the MESH repository. Simply add the folder ‘cmem\_4.1\_mesh’, and replace MESH\_driver.F90, read\_run\_options.F90, CLASSA.F90, makefile and makefile.def. You can then compile the SA\_MESH\_CMEM.**

1. **Input**
2. **MESH INPUT RUN OPTIONS**

To run MESH+CMEM, additional flags need to be added to the MESH\_input\_run\_options.ini file. Don’t forget to change the number of Control Flags as well. MESH+CMEM can simulate any number of valid incidence angles that can be represented by an integer value (See CMEM documentation). This is done by setting the CMEMFLAG.

If CMEMFLAG = 0 , CMEM does not run. Otherwise set CMEMFLAG to the number of incidence angles that you would like to simulate.

CMEMTHETA should occur in the control flags CMEMFLAG times and should be set to the incidence angle to simulate. This will replace the THETA value set in the CMEM input file. So to set CMEM to run at 2 incidence angles (40° & 50°):

CMEMFLAG 2

CMEMTHETA 40

CMEMTHETA 50

If you require R2C output of the CMEM values then:

CMEMR2COUTPUTFLAG = 1. The default is 0.

1. **Vegetation Types**

CMEM naturally considers vegetation types based on the ECOCLIMAP system of land surface classification. Since CMEM was written by ECMWF the options are also available for TESSEL and HTESSEL classification types, which are then converted in CMEM to ECOCLIMAP types. Therefore only ECOCLIMAP types are available with MESH in the CIDVEG option in the input file must be set to ‘Ecoclimap’.

A complication of MESH is the two levels of heterogeneity that are available, the first being the CLASS mosaic and the second the MESH GRU. Implementing both in CMEM is impractical for this coupling. Therefore each GRU is limited to a single High Vegetation (Tree) and a single Low vegetation (Grass/Crop) type based on ECOCLIMAP types. It is recommended to pick the dominant land cover type for each GRU. The ECOCLIMAP land cover types for each GRU are assigned in the file ‘MESH\_cmem\_ecoclimap.ini’ where the number of rows in the file is equal to the number of GRU’s.

1. **Soil Depth**

CMEM is designed to simulate the microwave emissions at the surface. Currently, if CMEMFLAG > 0 MESH will give a warning if the soil depths are not consistent with the CMEM defaults of 0.07, 0.28 and 1.00m for the first three layers. Therefore, the MESH\_input\_soil\_levels.ini need to be set to 0.07, 0.21 and 0.72 for the first three layers. If the first depth is greater than 0.1m MESH will give a warning and terminate as this is not consistent with CMEM. The ‘Liebe’ option of CMEM may provide better results for microwave emission but is disables in the coupling with MESH as it requires more soil layers and atmospheric layers than are available.

1. **Output**
2. **CMEM R2C OUTPUT**

The CMEMR2COUTPUTFLAG will print out the outputs requested in r2c\_output\_cmem.txt for each incidence angle. Be aware that this can turn into a lot of R2C files for each variable. The format is the same as r2c\_output.txt and can print the GRD, GAT, or GRDGAT variables subject to the vegetation types limitations (below).

CMEM natively has 3 levels of output which each provide progressively more information. The implementation of all of these for R2C output is left as a future upgrade. Currently only the temperature brightness at the top of the atmosphere at both H and V polarization and the vegetation opacity (Tau) at H and V polarization are available.