**WRF-Chem v3.4, WRF-GHG, and EDGAR CO2 Source Data**

1. **Overview**

The WRF-GHG code has been modified to fulfill the requirements for a distribution within the official WRF-Chem code from version 3.4 onwards. The general intention was to shape the WRF-GHG code in WRF-Chem structure, using the existing WRF-Chem capabilities, and to provide user-friendly code.

According to the WRF-Chem conventions, the assigned array for anthropogenic flux variable in the WRF-Chem v3.4 code is **emis\_ant** and the flux variable is **e\_co2.** A test flux for the anthropogenic fluxes has been included and the dimension of the anthropogenic fluxes is chosen in the **namelist.input** file by the **kemit (**number of vertical levels in the emissions input data file**)** option. With this option the injection level of the anthropogenic emissions can be explicitly specified.

The anthropogenic CO2 tracers (now in units of ppm) are added up to the total contribution of the atmospheric mixing ratio in a post-processing step. The chemistry option previously defined as **chem\_opt = 98** in the **namelist.input** file for WRF-GHG has now been moved to the package **co2\_tracer** with **chem\_opt=16** including anthropogenic CO2 and the package **ghg\_tracer** with **chem\_opt=17** including all tracers of chem\_opt=16 (and biomass burning tracers for CO2, etc.) The anthropogenic CO2 tracer variable(**chem)** corresponding to the two chemistry options 16 and 17 for the transport of greenhouse gases is called **co2\_ant**.

A new routine called **module\_ghg\_fluxes.F** has been added to the WRF-Chem v3.4 code. This routine was previously split into two different modules in the stand-alone WRF-GHG code called **module\_greenhouse\_gases.F** and **module\_add\_emissions.F**. The variables previously defined in **registry.ghg** in stand-alone WRF-GHG are now included in **registry.chem**.

A general practice within the WRF-Chem community is the initialization of tracers with standardized profiles. For simplicity, anthropogenic CO2 tracers are initialized with a constant background value of 380 ppm that has to be subtracted in a post-processing step. These values are also used for the forcing at the lateral boundaries. The initialization with background values further prevents the tracers from mass conservation problems that were observed for cases where strong gradients occur.

1. **Description of some of the WRF-Chem 3.4 code structure relevant to anthropogenic CO2 tracer emissions, fluxes, and concentrations.**

The following diagram gives a rough overview of the WRF-Chem v3.4 code structure:

**dyn\_em/solve\_em.F---> share/solve\_interface.F---> chem/chem\_driver.F**

**---> chem/module\_ghg\_fluxes.F**

**----> add\_ghg\_fluxes subroutine**

All tracer and flux variables used in WRF-Chem v 3.4 are defined in the **registry.chem** file in the WRF-Chem v3.4 registry. In the routine **module\_ghg\_fluxes.F** there is a subroutine called **add\_ghg\_fluxes**

for converting fluxes into atmospheric concentrations and adding them to the corresponding tracer variable (e.g., anthropogenic CO2). The subroutine for the greenhouse gas module is, in turn, called by the **chem\_driver.F** routine if **chem\_opt = 16** is set in the **namelist.input** file.

**registry.chem**

All variables for tracers and fluxes used by WRF-Chem v3.4, variables needed for the online calculation of fluxes, and all the WRF-Chem v3.4 –specific namelist options are defined in the **registry.chem** file. All variables defined in **registry.chem** are global variables and the empty arrays are created in the **wrfinput\* and wrfbdy\*** files generated by **real.exe** if it is specified in the **namelist.input** file that lateral boundary conditions are requested. If it was defined in **registry.chem**, the variables are also written to the **wrfoutput\*** file. Additionally, all flux variables representing fluxes (units in kg/m2s) of different processes emitting or consuming CO2 are defined in **registry.chem**.

**module\_ghg\_fluxes.F**

**module\_ghg\_fluxes.F** contains all subroutines for adding the emissions of CO2 calculated per time step to the corresponding atmospheric concentrations. Emissions are usually added to the atmospheric concentrations at the first model layer. The **add\_ghg\_fluxes** subroutine in **module\_ghg\_fluxes.F** multiplies the three-dimensional flux variable **emis\_ant(i,k,j)** by a conversion factor **conv\_rho(i,k,j)** and adds this to the first layer of the three-dimensional tracer variable **chem(i,k,j)**. The addition of fluxes to the concentrations is illustrtated in the following formulas using the variable names of the WRF-Chem v3.4 source code:

**chem(I,k,j) = chem(I,k,j) + emis\_ant(I,k,j)\*conv\_rho(i,k,j)**

with

**conv\_rho(i,k,j) = 1/rho\_phy(i,k,j)[kg/m3] \* dstep[s]/dz8w(i,k,j)[m]**

Here **rho\_phy(i,k,j)** denotes the air density in the model layer **k**, **dstep** the time step of the model, and **dz8w(i,k,j)** the thickness of the model layer in meters. For three-dimensional addition of anthropogenic CO2 emissions, a loop over all model levels **k** with their corresponding air density **rho\_phy(i,k,j)** and layer thickenss **dz8w(i,k,j)** is conducted.

The main subroutine **add\_ghg\_fluxes** is called from **enissions\_driver.F**, which is, in turn, ultimately called from the **chem\_dirver.F** routine (e.g., see the **chem\_prep** subroutine in **chem\_driver.F**).

**Other relevant WRF-Chem v3.4 routines**

The **chem\_init** subroutine in **chemics\_init.F** initializes anthropogenic CO2 for the 16th and 17th chemistry options.:

* CO2 mixing ratios for background GHG tracers are set at a constant value (e.g., 380 ppmv).
* Some spin up is necessary to obtain the correct spatial variability.
* Initial values for CO2 components need to be subtracted from the output. The purpose of this is to avoid negative values due to biospheric uptake.
* This subroutine is called at every initialization, however it is carried out only if **restart = 0** and **chem\_in\_opt = 0** in the **namelist.input**  file.

The green-house gas emissions are specified “outside” of the WPS or WRF system via an external input data file (see Section 3 below), and this data file is opened and read via the subroutine **med\_read\_wrf\_chem\_emissions** in the **../share/mediation\_integrate.F** routine.

1. **The prep\_chem\_sources pre-processor and EDGAR Anthropogenic emissions for CO2**

For anthropogenic CO2 emissions, WRF-Chem v3.4 can be used with **EDGAR** emission inventories. The **EDGAR V4.1** emission inventory (available online via [**http://edgar.jrc.ec.europa.eu**](http://edgar.jrc.ec.europa.eu)) is on a 0.1o x 0.1o resolution and globally available for greenhouse gases such as CO2 on a yearly basis up to 2005. All IPCC categories can be downloaded from the webpage and have to be summed up to get the final emissions. I believe that the **prep\_chem\_sources v 1.2** software includes ascii files containing summed up EDGAR CO2 anthropogenic final emissions data. These emissions are gridded on the WRF grid using the **prep\_chem\_sources** pre-processor along with an R script and/or NCL tools. The **prep\_chem\_sources** pre-processor can be used to obtain **EDGAR** re-gridded anthropogenic CO2 emissions output fields containing a diurnal cycle. The pre-processor has been augmented for NetCDF output and the routines dealing with the EDGAR emission database (i.e., **edgar\_emissions.f90**) have also been upgraded to be able to handle the **EDGAR V4.1** emission inventory. In general, the units of the output fields are [kg/m2 day]. In case of using diurnal cycle routines, the units are [kg/m2 s].

**APPENDIX: Code fragments associated with anthropogenic CO2 emissions in relevant WRF-Chem 3.4 routines.**

**chem\_driver.F**

CALL wrf\_debug ( 15 , ' call chem\_prep' )

CALL chem\_prep ( config\_flags, &

.

.

.

chem\_minval = epsilc !chem\_minval can be case dependant and set below...

chem\_select: SELECT CASE(config\_flags%chem\_opt)

.

.

.

CASE (CO2\_TRACER,GHG\_TRACER)

CALL wrf\_debug(15,'Greenhouse gas mode: fluxes and transport of GHG')

.

.

.

!

! For the chemistry tracer mode, only emissions and vertical mixing are done.

! So, finish any remaining tiles and then skip to the end of chem\_driver.

!

if( do\_chemstep .and. &

.

.

.

config\_flags%chem\_opt /= CO2\_TRACER .and. &

config\_flags%chem\_opt /= GHG\_TRACER ) then

!

! chemical mechanisms

!

! save old concentrations for chemistry tendencies

#if (EM\_CORE == 1)

DO nv=PARAM\_FIRST\_SCALAR,num\_chem\_ct

chem\_old(its:ite,kts:kte,jts:jte,nv) = chem(its:ite,kts:kte,jts:jte,chem\_ct\_indices(nv))

ENDDO

**chemics\_init.F**

subroutine chem\_init (id,chem,emis\_ant,scalar,dt,bioemdt,photdt,chemdt,stepbioe, &

.

.

.

chem\_select: SELECT CASE(config\_flags%chem\_opt)

.

.

.

CASE (CO2\_TRACER, GHG\_TRACER )

call wrf\_message("WARNING: Users interested in the GHG options should check the comments/references in header of module\_ghg\_fluxes")

.

.

.

CALL nl\_get\_mminlu( 1, mminlu\_loc )

IF ( (mminlu\_loc .NE. 'USGS') .AND. (config\_flags%chem\_opt .NE. CO2\_TRACER .OR. config\_flags%chem\_opt .NE. GHG\_TRACER) ) THEN

call wrf\_error\_fatal(" ERROR: CHEM\_INIT: Chemistry routines function with USGS data. Need to change land use option. ")

ENDIF

.

.

.

if(config\_flags%chem\_in\_opt == 0 )then

do j=jts,jte

do k=kts,kte

do i=its,ite

chem(i,k,j,p\_co2)=370

.

.

.

**emissions\_driver.F**

MODULE module\_emissions\_driver

IMPLICIT NONE

CONTAINS

subroutine emissions\_driver(id,ktau,dtstep,DX,

.

.

.

USE module\_ghg\_fluxes

.

.

.

call wrf\_debug(15,'emissions\_driver calling add\_anthropogenics')

call add\_anthropogenics(id,dtstep,dz8w,config\_flags,rho\_phy,alt, &

chem, emis\_ant,emis\_aircraft, &

ids,ide, jds,jde, kds,kde, &

ims,ime, jms,jme, kms,kme, &

its,ite, jts,jte, kts,kte )

.

.

.

! Add all the GHG fluxes to chem species, this step is for both

! anthropogenic and biospheric fluxes

IF (config\_flags%emiss\_inpt\_opt==16) THEN

CALL add\_ghg\_fluxes( ids,ide, jds,jde, kds,kde, &

ims,ime, jms,jme, kms,kme, &

its,ite, jts,jte, kts,kte, &

dtstep,dz8w,config\_flags,rho\_phy, &

chem,emis\_ant,eghg\_bio,ebio\_co2oce )

END IF

.

.

.

**module\_input\_chem\_data.F**

**.**

**.**

**.**

MODULE module\_input\_chem\_data

.

.

.

! Interpolate the chemistry data to the SI grid. These values should typically

! be set to match the values in bdy\_chem\_value\_tracer so that the boundaries

! and interior match each other.

.

.

.

IF ( si\_grid%chem\_opt == CHEM\_TRACER ) THEN

.

.

.

ELSE IF ( si\_grid%chem\_opt==CO2\_TRACER .OR. si\_grid%chem\_opt==GHG\_TRACER ) THEN

! Do nothing, since for GHGs initialization is done in chemics\_init.F

.

.

.

SUBROUTINE bdy\_chem\_value\_racm ( chem, z, nch, numgas,p\_co2 )

IMPLICIT NONE

REAL, intent(OUT) :: chem

REAL, intent(IN) :: z ! 3D height array

INTEGER, intent(IN) :: nch,p\_co2 ! index number of chemical species

.

.

.

! Check the number of species

! if((nch-1).gt.logg)return

if (nch.eq.p\_co2)then

chem=370.

return

.

.

.

! Check the number of species

! if((nch-1).gt.logg)return

if (nch.eq.p\_co2)then

chem=370.

return

.

.

.

SUBROUTINE bdy\_chem\_value\_ghg ( chem, nch )

! This subroutine is called to set the boundary constan values for

! greenhouse gases when gas\_bc\_opt=16

IMPLICIT NONE

REAL, intent(OUT) :: chem

INTEGER, intent(IN) :: nch ! index number of chemical species

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

if( nch==p\_co2\_bck .OR. nch==p\_co2\_bio .OR. nch==p\_co2\_oce .OR. nch==p\_co2\_ant &

.OR. nch==p\_co2\_bbu .OR. nch==p\_co2\_tst ) then

chem = 380.

.

.

.

SUBROUTINE flow\_dep\_bdy\_chem ( chem,

.

.

.

! This subroutine sets zero gradient conditions for outflow and a set profile value

! for inflow in the boundary specified region. Note that field must be unstaggered.

! The velocities, u and v, will only be used to check their sign (coupled vels OK)

! spec\_zone is the width of the outer specified b.c.s that are set here.

! (JD August 2000)

.

.

.

else if (config\_flags%chem\_opt==CO2\_TRACER .OR. config\_flags%chem\_opt==GHG\_TRACER) then

i\_bdy\_method = 16

! CALL wrf\_message( "For GHGs strongly recommended to include LBCs from a global model!" )

,

,

,

**mediation\_integrate.F**

WRITE(message,'(A,A)')'mediation\_integrate: med\_read\_wrf\_chem\_emissions: Read emissions for time ',TRIM(current\_date\_char)

CALL wrf\_message( TRIM(message) )

.

.

.

!---

! io\_style\_emissions option 2: use dated emission files whose length is

! set via frames\_per\_auxinput5...

!---

else if( config\_flags%io\_style\_emissions == 2 ) then

WRITE(message,\*)'mediation\_integrate: med\_read\_wrf\_chem\_emissions: Read emissions for time ',TRIM(current\_date\_char)

CALL wrf\_message( TRIM(message) )

.

.

.

#include <wrf\_io\_flags.h>

! IF ( grid%id .EQ. 1 ) THEN

CALL domain\_clock\_get( grid, current\_timestr=current\_date\_char )

CALL construct\_filename1 ( inpname , config\_flags%auxinput12\_inname , grid%id , 2 )

WRITE(message,\*)'mediation\_integrate: med\_read\_wrf\_chem\_input: Open file ',TRIM(inpname)

CALL wrf\_message( TRIM(message) )

if( grid%auxinput12\_oid .NE. 0 ) then

CALL close\_dataset ( grid%auxinput12\_oid , config\_flags , "DATASET=AUXINPUT12" )

endif

.

.

.

!

! Code to read hourly emission files...

!

if( grid%auxinput5\_oid == 0 ) then

CALL construct\_filename2a(inpname , grid%emi\_inname, grid%id , 2, current\_date\_char)

WRITE(message,\*)'mediation\_integrate: med\_read\_wrf\_chem\_emissions: Open file ',TRIM(inpname)

CALL wrf\_message( TRIM(message) )

CALL open\_r\_dataset ( grid%auxinput5\_oid, TRIM(inpname) , grid , config\_flags, &

.

.

.