

Adding passive (CO-like) tracers to GEOS-Chem

Jenny Fisher, U. Wollongong & Jingqiu Mao, GFDL

1 Basic Info

These instructions are based on v9-01-03 and mostly follow instructions on these pages:

http://wiki.seas.harvard.edu/geos-chem/index.php/Adding_tracers_to_GEOS-Chem

http://acmg.seas.harvard.edu/forum_files2/tmf_addchemistry.pdf

The first parts of this problem are relatively straightforward. The second part, namely deciding where to set the emissions for each passive tracer, is subjective. I have chosen to set them independently for fossil fuel, biomass burning and biofuels. This allows these to be specified as separate tracers for each region/source, or as one tracer for each region (all sources). Alternatively, if no source dependence is required, this could probably be done in setemis.F by just setting the REMIS array (but I haven't thought too much about that option).

Note that these instructions only work for running with SMVGear. To run with KPP, you will also need to follow the instructions on this page:

http://wiki.seas.harvard.edu/geos-chem/index.php/Interfacing_GEOS-Chem_with_KPP#Generating_KPP_input_files_from_GEOS-Chem_globchem.dat

However, I have not yet been successful at running KPP on my system, so I can't comment on these.

After implementing the passive tracers, a good sanity check is to archive the fossil fuel, biofuel, and biomass burning emissions for both CO and the new passive tracers. For biofuel and biomass burning, no new code is required - just add the new tracer number to the output for the diagnostic in *input.geos*. For fossil fuel, the required change is included below.

Please direct any questions to Jenny Fisher: jennyf@uow.edu.au

2 Acknowledgment

These passive tracers have not been submitted to the standard code because implementation is application-dependent, but required effort commensurate with a normal GEOS-Chem development. We encourage you to follow the standard GEOS-Chem credits and references policy (http://www.geos-chem.org/geos_credit_and_refs.html). This work has not yet been published, but citation information will be included once available.

3 Run Directory

3.1 *input.geos*

- a. Increase number of tracers by one for each passive tracer

```
Number of Tracers      : 57
```

- b. Add each passive tracer to tracer list

```
Tracer #57             : 57  PCOAS  28.0  (PCOAS)
```

3.2 *globchem.dat*

- a. Add each passive tracer to species list

```
A PCOAS                1.00 1.000E-07 1.000E-07 1.000E-07 1.000E-07
                        0 0 0 0 0 0 0 0 0
```

- b. Add emission reaction for each passive tracer

```
A 573 0.00E+00 0.0E+00 0 0 0.00 0. 0.
    EMISSION      +
=1.000PCOAS      +          +          +
+                +          +          +
+                +          +          +
+                +          +          +
+                +          +          +
```

The 573 here is the reaction number, which I just copied from CO. This number doesn't really matter as the model doesn't use it for calculation (and is set to 0 for several of the reactions in *globchem.dat*).

- c. Add decay reaction for each passive tracer

```
A 813 4.63E-07 0.0E+00 0 0 0.00 0. 0.
    PCOAS        +
=                +          +          +
+                +          +          +
+                +          +          +
+                +          +          +
+                +          +          +
```

The lifetime of the passive tracer is specified by the third column in the first line. This is the reaction rate for a first-order reaction, given in units of s^{-1} . In this example, the CO passive tracer has a lifetime of 25 days. So the specified reaction rate is $1/(25d)$, which is equal to $4.63e-07 s^{-1}$.

As above, the 813 is just a generic reaction number and doesn't matter for computation. I added this reaction to the very end of the kinetic reactions list (but before photolysis).

3.3 *restart file*

- a. Add a line to *tracerinfo.dat* for each passive tracer

PCOAS Asian CO passive tracer 2.800E-02 1 57 1.000E+09 ppbv

- b. Use `make_restart.pro` to create a fake restart containing just the passive tracers, with the same model, resolution, and `tau0` as your original restart.

```
MAKE_RESTART, OUTMODELNAME='GEOS5_47L', OUTRESOLUTION=2, $
              OUTFILENAME='restart.pcoas.2x25.geos5', TAU0=NYMD2TAU( 20050101L ), $
              DATAVALUE=1e-8, TRACERLIST=57, UNIT='v/v', DIAGN='IJ-AVG-$'
```

- c. Use `bpch_link.pro` to combine the two restarts.

```
bpch_link, ['restart.2x25.geos5', 'restart.pcoas.2x25.geos5'], 'restart.new.2x25.geos5'
```

4 Code Directory

4.1 *Headers/CMN_SIZE_mod.F*

This section can be very confusing because `CMN_SIZE_mod.F` contains the entire module twice. The first version is delineated with the `#if defined(DEVEL)` tag. Since generally we won't be using the `DEVEL` tag, it's important to make sure these quantities are updated in the *second* version (although preferably, update them in both places to be consistent).

- a. Increase `NNPAR` by one for every new passive tracer

```
INTEGER, PARAMETER :: NNPAR = 86
```

- b. Increase `NEMPARA` by one for every new passive tracer (anthropogenic emissions)

```
INTEGER, PARAMETER :: NEMPARA = 30
```

- c. Increase `NBIOMAX` by one for every new passive tracer (biomass burning emissions)

```
INTEGER, PARAMETER :: NBIOMAX = 26
```

Note that there are no direct biogenic emissions of CO, so we don't need to edit `NEMPARB`

4.2 *Headers/comode_loop_mod.F*

- a. Increase `IGAS` by one for every new passive tracer

```
PARAMETER ( IGAS = 311, IAERTY = 1 )
```

- b. Increase `NMRATE` by two for every new passive tracer (one for emission, one for destruction)

```
PARAMETER ( NMRATE = 728, IPHOT = 92 )
```

4.3 *GeosCore/tracerid_mod.F*

- a. Define tracer IDs (`ID`, `IDT`, `IDE`, `IDBF`, `IDB`) for each passive tracer

```
INTEGER      :: IDPCOAS
INTEGER      :: IDTPCOAS
INTEGER      :: IDEPCOAS
```

```

INTEGER          :: IDBFPCOAS
INTEGER          :: IDBPCOAS

```

- b. Add case statements for each passive tracer to initialize IDT, IDBF, and IDB (subroutine TRACERID)

```

CASE ( 'PCOAS' )
  COUNT      = COUNT + 1
  IDTPCOAS = N
  IDBFPCOAS = COUNT

  COUNT_BB = COUNT_BB + 1
  IDBPCOAS = COUNT_BB

```

- c. Add if statements for each passive tracer to initialize IDE (subroutine TRACERID)

```

IF ( IDTPCOAS > 0 ) THEN
  NEMANTHRO = NEMANTHRO + 1
  IDEPCOAS  = NEMANTHRO
ENDIF

```

Make sure this is done *before* the non-anthropogenic tracers are initialised. Note that if you are implementing separate biomass burning tracers (which therefore have no anthropogenic emissions), they *still* need to be included in NEMANTHRO for the species to be treated by the chemistry solver.

- d. Add IDs for each passive tracer to IDEMS (subroutine TRACERID)

```

IF ( IDEPCOAS /= 0 ) IDEMS(IDEPCOAS) = IDTPCOAS

```

As for (c), even biomass burning (non-anthropogenic) tracers need to be added to IDEMS.

- e. Update writing statements (subroutine TRACERID)

```

WRITE( 6, 115 ) '          + 1 for CO passive tracers'
WRITE( 6, 115 ) 'NEMANTHRO: 18 max'
WRITE( 6, 120 ) 'IDEPCOAS ', IDEPCOAS

```

- f. Add if statements for each passive tracer to initialise ID (subroutine SETTRACE)

```

IF ( NAMEGAS(I) == 'PCOAS' ) IDPCOAS = I

```

- g. Zero ID, IDT, IDE, IDBF, and IDB for each passive tracer (subroutine INIT_TRACERID)

```

IDPCOAS      = 0
IDTPCOAS     = 0
IDEPCOAS     = 0
IDBFPCOAS   = 0
IDBPCOAS     = 0

```

4.4 *GeosCore/biomass_mod.F*

- a. Add IDs for each passive tracer to BIOTRCE array and NBIOTRCE count (subroutine SET_BIOTRCE)

```

USE TRACERID_MOD, ONLY : IDBPCOAS
USE TRACERID_MOD, ONLY : IDTPCOAS

IF ( IDBPCOAS /= 0 ) NBIOTRCE = NBIOTRCE + 1

```

```
IF ( IDBPCOAS /= 0 ) BIOTRCE(IDBPCOAS) = IDTPCOAS
```

- b. Set BIOBGAS to TRUE for each passive CO tracer as emission will be handled as for CO (subroutine SET_BIOTRCE)

```
IF ( IDBPCOAS /= 0 ) BIOBGAS(IDBPCOAS) = .TRUE.
```

- c. Add biomass emissions to BIOMASS for each passive tracer (subroutine COMPUTE_BIOMASS_EMISSIONS)

```
USE TRACERID_MOD,      ONLY : IDBPCOAS
USE GRID_MOD,         ONLY : GET_XMID, GET_YMID
INTEGER               :: X,      Y

      IF ( IDBPCOAS > 0 ) THEN
        DO J = 1, JJPAR
          DO I = 1, IIPAR

            X = GET_XMID(I)
            Y = GET_YMID(J)

            IF ( (X >= 90.0) .AND. (X < 142.0) .AND.
&              (Y >= 6.0) .AND. (Y < 40.0) )
&              BIOMASS(I,J,IDBPCOAS) = BIOMASS(I,J,IDBCO)

          ENDDO
        ENDDO
      ENDIF
```

Note that this should be done *towards the end* of the subroutine, after the BIOMASS(:, :, IDBCO) array has been set. This prevents having to rewrite the BIOMASS array for the passive tracers each time CO is scaled or a different inventory is used. I chose to put this just before the diagnostics are written. This section is outside the I,J loops so we need to loop here.

The IF statement in the middle of this section should specify the bounds of the region appropriate for each passive tracer. The outer IF statement should include IDB for each passive tracer.

4.5 *GeosCore/biofuel_mod.F*

- a. Add IDs for each passive tracer to BFTRCE array and NBFTRCE count (subroutine SET_BFTRCE)

```
USE TRACERID_MOD, ONLY : IDBFPCOAS
USE TRACERID_MOD, ONLY : IDTPCOAS

IF ( IDBFPCOAS /= 0 ) NBFTRACE = NBFTRACE + 1
IF ( IDBFPCOAS /= 0 ) BFTRACE(IDBFPCOAS) = IDTPCOAS
```

- b. DON'T write to BIOFUEL array based on BIOFUEL_KG for passive tracers (subroutine BIOFUEL_BURN)

```
USE TRACERID_MOD,      ONLY : IDTPCOAS

      IF ( NN /= IDTPCOAS ) THEN
```

```

      ! Biofuel emissions in [molec/cm3/s]
      BIOFUEL(N,I,J) = BIOFUEL_KG(N,I,J)
&
&      ( 6.023d23 / MOLWT(N) ) /
&      ( 365d0 * 86400d0 * BOXVL(I,J,1) )
      *
    ENDIF

```

Note that BIOFUEL code is already present; only the surrounding IF statement is new.

- c. Add biofuel emissions to BIOFUEL for each passive tracer (subroutine BIOFUEL_BURN)

```

      USE TRACERID_MOD,      ONLY : IDBFPCOAS
      USE GRID_MOD,        ONLY : GET_XMID, GET_YMID
      INTEGER              :: X, Y

!$OMP+PRIVATE( I, J, BXHEIGHT_CM, N, NN, EPA_NEI, X, Y )

      IF ( IDBFPCOAS > 0 ) THEN

          X = GET_XMID(I)
          Y = GET_YMID(J)

          IF ( (X >= 90.0) .AND. (X < 142.0) .AND.
&
&            (Y >= 6.0) .AND. (Y < 40.0) )
&            BIOFUEL(IDBFPCOAS,I,J) = BIOFUEL(IDBFCO,I,J)

      ENDIF

```

Note that this should be done *towards the end* of the subroutine, after the BIOFUEL(IDBFCO,,:) array has been set. This prevents having to rewrite the BIOFUEL array for the passive tracers each time CO is scaled, overwritten, etc. I chose to put this just before ND29 is written. Note that if done this way, we are still in the loops over I and J, so X and Y must be added to the OMP PRIVATE declaration.

The IF statement in the middle of this section should specify the bounds of the region appropriate for each passive tracer. The outer IF statement should include IDBF for each passive tracer.

4.6 *GeosCore/emfossil.F*

- a. Add fossil fuel emissions to EMISRR for each passive tracer (subroutine EMFOSSIL)

```

      USE TRACERID_MOD,      ONLY : IDECO, IDTPCOAS
      USE GRID_MOD,        ONLY : GET_XMID, GET_YMID

      INTEGER :: X, Y

      IF ( NN == IDTPCOAS ) THEN

          X = GET_XMID(I)
          Y = GET_YMID(J)

          IF ( (X >= 90.0) .AND. (X < 142.0) .AND.
&
&            (Y >= 6.0) .AND. (Y < 40.0) ) THEN

```

```

EMISRR(I,J,N) = EMISRR(I,J,IDECO)
EMX(1) = EMISRR(I,J,N) * DTSRCE / XNUMOL(NN)

```

```

ENDIF

```

```

ENDIF

```

Again, note that this should be done *towards the end* of the subroutine, after the EMISRR(I,J,IDECO) value has been set. This prevents having to rewrite the EMISRR array for the passive tracers each time CO is overwritten (e.g. by a regional inventory). I chose to put this just before ND29 is written. Note that EMFOSSIL is called for a specific (I,J) so there is no need to loop here.

The IF statement in the middle of this section should specify the bounds of the region appropriate for each passive tracer. The outer IF statement should include IDT for each passive tracer. Note that we use IDT not IDE here because this is the NN value (the GEOS-Chem tracer) although we use IDE for indexing the CO already stored in EMISRR (the emissions tracer).

Note also that additional terms are added to EMISRR(:, :, IDECO) in emissdr.F to account for monoterpene oxidation. While this is treated in GEOS-Chem as an emission, it is actually a parameterisation of chemistry. Because passive tracers should only reflect *direct emissions*, we do not include this overwrite for the passive tracers.

Finally, EMX(1) is redefined here for the passive tracer because EMX(1) is the quantity used to archive the fossil fuel emissions diagnostic (ND36). Here, we simply invert EMISRR to recover EMX(1). This provides an additional sanity check: if for some reason something besides direct emissions were added to EMISRR for CO, we would see a discrepancy between ND36 for CO and ND36 for the passive tracer. The way the code is currently structured there are no differences, but this is something to be aware of in future implementations.

- b. Zero any non-anthropogenic passive tracers (subroutine EMFOSSIL)

Because of legacy code, we have to include non-anthropogenic passive tracers (e.g. biomass burning tracers) in NEMANTHRO, which means EMFOSSIL will be called for these tracers as well. It is important that EMISRR and EMX(1) get set to 0d0 for any such tracers.

4.7 *GeosCore/gamap_mod.F*

- a. Increase NTRAC(28) by 1 for each passive tracer

```

NTRAC(28) = 15

```

- b. Add each passive tracer to tracerinfo labelling for ND28 (subroutine INIT_TRACERINFO)

```

USE TRACERID_MOD, ONLY : IDBPCOAS, IDTPCOAS

```

```

ELSE IF ( T == IDBPCOAS ) THEN
  NAME (T,28) = 'PCOAS'
  INDEX(T,28) = IDTPCOAS + ( SPACING * 45 )
  MWT (T,28) = 28e-3
  MOLC (T,28) = 1

```

UNIT (T,28) = 'molec/cm2/s'

This isn't essential, but if it is skipped then the GAMAP routines won't recognise the name/unit of the new tracers for biomass burning emissions. This isn't a problem for the other emissions.