Supplementary Information for Modelling the Fate of Mercury Emissions from Artisanal and Small Scale Gold Mining

Ian M. Hedegcock, Francesco De Simone, Francesco Carbone, and Nicola Pirrone CNR-Institute of Atmospheric Pollution Research, 87036 Rende, Italy **Correspondence:** Ian M. Hedgecock (ian.hedgecock@iia.cnr.it)

S1 Emission domain relief



Figure S1. Emissions and relief of the South-East Asian Domains, emissions in $\mathrm{mol}\,\mathrm{km}^{-2}\,\mathrm{hr}^{-1}$

Figure S2. Emissions and relief of the African Domains, emissions in $\rm mol\,km^{-2}\,hr^{-1}$ $\rm Emissions$

Burkina Faso

0°W

Sudan

32°E

34°E

2°E

4°E

18°N

16°N

14°N

12°N

18°N

16°N

14°N

30°E

2°W



30°E 32°E 34°E

Figure S3. Normalised Hg Deposition vs. distance from emission source for the South American domains. The left colour bar represents deposition to land, the right to seas and oceans



(c) Brazil

<page-header>



(d) Guyana



(b) Columbia

Figure S4. Normalised Hg Deposition vs. distance from emission source for the South-East Asian domains. The left colour bar represents

10⁻³

10-4

10-1

deposition to land, the right to seas and oceans

(a) Mekong



(c) Kalimantan



(d) Papua New Guinea

Papua New Guinea, slow oxidation and slow foliar uptake



(b) Sumatra

Figure S5. Normalised Hg Deposition vs. distance from emission source for theAfrican domains. The left colour bar represents deposition to land, the right to seas and oceans

(a) Ghana



(c) Senegal







(b) Burkina Faso

(d) Sudan





(e) Lake Victoria

Lake Victoria, slow oxidation and slow foliar uptake



S3 Deposition Distance Barcharts

Note on deposition velocity



Figure S6. Deposition vs. distance, South-East Asian emission domains. Percentages of the total within domain deposition Deposition vs. Distance ($v_d = 2 cms^{-1}$)



Figure S7. Deposition vs. distance, South-East Asian emission domains. Percentages of the total within domain deposition Deposition vs. Distance ($v_d = 2 cms^{-1}$)



Figure S8. Deposition vs. distance, African emission domains. Percentages of the total within domain deposition

5 S4 Deposition Maps

These maps show the deposition from the simulations using the most rapid oxidation rate and the fastest foliar uptake, illustrating the differences in deposition fields for the different domains and different deposition pathways. The normalisation is to the maximum deposition for each pathway for each domain.





Figure S10. Normalised deposition from the domain in Sumatra

Sumatra Domain, fast oxidation, rapid foliar uptake



Figure S11. Normalised deposition from the domain in Kalimantan Kalimantan Domain, fast oxidation, rapid foliar uptake



Figure S12. Normalised deposition from the domain in Papua New Guinea

Papua New Guinea Domain, fast oxidation, rapid foliar uptake



Figure S13. Normalised deposition from the domain in Colombia Colombia Domain, fast oxidation, rapid foliar uptake



Figure S14. Normalised deposition from the domain in Bolivia

Bolivia Domain, fast oxidation, rapid foliar uptake



Figure S15. Normalised deposition from the domain in Brazil Brazil Domain, fast oxidation, rapid foliar uptake



Figure S16. Normalised deposition from the domain in Guyana

Guyana Domain, fast oxidation, rapid foliar uptake



Figure S17. Normalised deposition from the domain in Ghana Ghana Domain, fast oxidation, rapid foliar uptake



Figure S18. Normalised deposition from the domain in Burkina Faso Burkina Faso Domain, fast oxidation, rapid foliar uptake



Figure S19. Normalised deposition from the domain in Senegal Senegal Domain, fast oxidation, rapid foliar uptake



Figure S20. Normalised deposition from the domain in Sudan

Sudan Domain, fast oxidation, rapid foliar uptake



Figure S21. Normalised deposition from the domain near Lake Victoria Lake Victoria Domain, fast oxidation, rapid foliar uptake



S5 Vegetation uptake module

```
10
     1: MODULE module_veg_uptake
     2:
     3: USE module_hg_lai_data
     4: USE module state description
15
     5:
     6: CONTAINS
     7:
     8: SUBROUTINE Hg tracer ASGM(id, dtstep, ktau, pbl h, chem, t phy, p phy, rho phy, dz8w, chem opt, num chem, &
     9:
                                         z, ht, ids, ide, jds, jde, kds, kde,
                                                                                                 &
20
    10:
                                         ims, ime, jms, jme, kms, kme,
                                                                                                 &
    11:
                                         its, ite, its, ite, kts, kte,
                                                                                                 &
    12:
                                         lai, raincv_hg, rainncv_hg, Hg_dep, moist, area2d)
    13:
              IMPLICIT NONE
    14:
25
    15:
              INTEGER,
                                                :: id
    16:
                             INTENT(IN
                                          )
                                                :: ktau, chem_opt, num_chem
    17:
              INTEGER,
                             INTENT(IN
                                          )
                                                :: ids, ide, jds, jde, kds, kde
    18:
              INTEGER.
                             INTENT(IN
                                          )
                                                :: ims, ime, jms, jme, kms, kme
    19:
              INTEGER,
                             INTENT(IN
                                          )
              INTEGER.
                             INTENT (IN
                                          )
                                                :: its, ite, jts, jte, kts, kte
30
    20:
                     DIMENSION(ims:ime,kms:kme,jms:jme,num chem), INTENT(INOUT) :: chem
    21:
              REAL,
    22:
              REAL.
                     DIMENSION(ims:ime,1,jms:jme,num_Hg_dep), INTENT(INOUT) :: Hg_dep
    23:
              REAL,
                     DIMENSION(ims:ime,kms:kme,jms:jme), INTENT(IN) :: t_phy,p_phy,rho_phy,dz8w,z
              REAL, DIMENSION( ims:ime, kms:kme, jms:jme, num_moist ),
                                                                                     &
    24:
                 INTENT(INOUT) ::
35
    25:
                                                                          moist
                     DIMENSION(ims:ime,jms:jme), INTENT(IN) :: PBL_H,HT, area2d
    26:
              REAL,
    27:
              REAL.
                     DIMENSION(ims:ime, jms:jme), INTENT(IN) :: lai, raincv_hg, rainncv_hg
              REAL,
                     INTENT(IN) :: dtstep
    28:
              !local variables
    29:
40
    30:
              REAL,
                     DIMENSION(ims:ime,jms:jme) :: fraction_on_ground
             REAL
                                  :: colsum_hgii_180_slo_asgm
    31:
    32:
             REAL
                                  :: scav_hgii_180_slo_asgm
             REAL
                                  :: fraction_this_cell, frac, mass_deposited_this_lev, lost_fraction
    33:
             REAL
                                  :: Hg_em_sum, hg_mass, hg_scav_fac, wet_hg_mass, ppm2mgm3
    34:
45
    35:
             REAL
                                  :: LWC, ppm2ngm3, ng_per_L, ngL_sum, ngL_average, dz8wsum, qr_sum, qr_average
              INTEGER
                                  :: count_pbl, count_trop, qr_count
    36:
```

```
37:
                                                                        INTEGER
                                                                                                                                                                               :: i,j,k
                       38:
                        39:
50
                      40:
                                              !! Uptake by vegatation
                       41:
                       42:
                       43:
                       44:
                                                                                         do i = its, ite
55
                                                                                         do j = jts, jte
                      45:
                                                                                                           Leaf_{lo} = lai(i, j) + 9.218e - 5 + dtstep / 1.4 + dtstep is in seconds (unlike the chemistry time
                        46:
                       47:
                                                                                                          Leaf_mid = lai(i,j)*0.0001777*dtstep/1.4
                       48:
                                                                                                           Leaf_hi = lai(i, j) * 0.0002632 * dtstep / 1.4
                        49:
60
                                                                                                           if (Leaf_lo.gt.1.) Leaf_lo = 1.
                      50:
                                                                                                           if (Leaf_mid.gt.1.) Leaf_mid = 1.
                       51:
                       52:
                                                                                                           if (Leaf_hi.gt.1.) Leaf_hi = 1.
                       53:
                       54:
                                                                                         do k = 1, 1
                                                                                                                hg_mass = ((200.59 * p_phy(i,1,j)) / (t_phy(i,1,j) * 8.314472)) * 1.e-6 !ppm to mug/m3
65
                      55:
                        56:
                                                                                                                Hg_dep(i, 1, j, p_hg_{180\_slo\_asgm}) = Hg_dep(i, 1, j, p_hg_{180\_slo\_asgm}) + hg_mass*chem(i, k, j, p_hg_{180
                       57:
                       58:
                                                                                                          chem(i,k,j,p_{10}) = chem(i,k,j,p_{10}) = chem(i,k,j,p_{10}) + (1. - (Leaf_{10} / dz8w(i,1)) + (1. - (Leaf_{10} / dz8w(i,1))) + (1
                        59:
                                                                                         enddo
70
                      60:
                                                                                         enddo
                      61:
                                                                                         enddo
                       62:
                       63:
                      64: !! Rain
                     65: !! convective
75
                       66:
                       67:
                       68: do i = its, ite
                       69: do j = jts, jte
80
                      70: if (raincv_hg(i,j).gt.0.1) then
                                                                                                    dz8wsum=0.
                      71:
                       72:
                                                                                                    ppm2ngm3=0.
                       73:
                                                                                                     colsum_hgii_180_slo_asgm=0.
                       74:
                                                                                                     colsum_hgii_270_slo_asgm=0.
```

85	75:	colsum_hgii_360_slo_asgm=0.
	76:	colsum_hgii_180_l10cfpp=0.
	77:	colsum_hgii_270_110cfpp=0.
	78:	colsum_hgii_360_l10cfpp=0.
	79:	colsum_hgii_180_l15cfpp=0.
90	80:	colsum_hgii_270_115cfpp=0.
	81:	colsum_hgii_360_115cfpp=0.
	82:	do k=kts, kte
	83:	if (p_phy(i,k,j).gt.18000.) dz8wsum=dz8wsum + dz8w(i,k,j)
	84:	enddo
95	85:	do k=kts, kte
	86:	if (p_phy(i,k,j).gt.18000and.dz8wsum.gt.100.) then
	87:	$ppm2ngm3 = ((200.59 * p_phy(i,k,j)) / (t_phy(i,k,j) * 8.314472)) * 1000. *1.e-6 !this$
	88:	colsum_hgii_180_slo_asgm=colsum_hgii_180_slo_asgm+ppm2ngm3*chem(i,k,j,p_hgii_180_slo_
	89:	else
100	90:	continue
	91:	endif
	92:	enddo
	93:	
	94:	
105	95:	if (colsum_hgii_180_slo_asgm.gt.1.e-13) then
	96:	$Hg_dep(i, 1, j, p_hg_{180_l05cdep_cfpp}) = Hg_dep(i, 1, j, p_hg_{180_l05cdep_cfpp}) + 750.*(colsum_hg_dep(i, 1, j, j,$
	97:	scav_hgii_180_slo_asgm=(colsum_hgii_180_slo_asgm-(750.*(colsum_hgii_180_slo_asgm/dz8wsu
	98:	else
	99:	$scav_hgii_180_slo_asgm = 1.0$
110	100:	endif
	101:	
	102:	do k=kts, kte
	103:	if (p_phy(i,k,j).gt.18000and.dz8wsum.gt.100.) then
	104:	$chem(i,k,j,p_hgii_180_slo_asgm) = scav_hgii_180_slo_asgm*chem(i,k,j,p_hgii_180_slo_asgm*chem(i,k,j,k,j,k,k,k,k,k,k,k,k,k,k,k,k,k,k,k,$
115	105:	else
	106:	continue
	107:	endif
	108:	enddo
	109:	
120	110: else	
	111: continu	
	112: endif	

```
113: enddo
            114: enddo
125 115:
            116: ! non-convective
            117: do i = its, ite
            118: do j = jts, jte
            119: if (rainnev_hg(i,j).gt.0.1) then
130 120:
                                                      dz8wsum=0.
            121:
                                                      ppm2ngm3=0.
                                                      colsum_hgii_180_slo_asgm=0.
            122:
            123:
                                                      colsum_hgii_270_slo_asgm=0.
            124:
                                                      colsum_hgii_360_slo_asgm=0.
135 125:
                                                      colsum_hgii_180_l10cfpp=0.
                                                      colsum_hgii_270_110cfpp=0.
            126:
            127:
                                                      colsum_hgii_360_110cfpp=0.
            128:
                                                      colsum_hgii_180_l15cfpp=0.
            129:
                                                      colsum_hgii_270_l15cfpp=0.
140 130:
                                                      colsum_hgii_360_115cfpp=0.
            131:
                                                      do k=kts, kte
            132:
                                                      if (p_phy(i,k,j), gt.60000) dz8wsum=dz8wsum + dz8w(i,k,j)
            133:
                                                      enddo
            134:
145 135:
                                                      do k=kts, kte
            136:
                                                               if (p_phy(i,k,j).gt.60000..and.dz8wsum.gt.100.) then
            137:
                                                               ppm2ngm3 = ((200.59 * p_phy(i,k,j)) / (t_phy(i,k,j) * 8.314472)) * 1000. *1.e-6 !this
                                                               colsum_hgii_180_slo_asgm=colsum_hgii_180_slo_asgm+ppm2ngm3*chem(i,k,j,p_hgii_180_slo_asgm+ppm2ngm3*chem)
            138:
            139:
                                                               else
150 140:
                                                               continue
                                                               endif
            141:
            142:
                                                      enddo
            143:
                                                   if (colsum_hgii_180_slo_asgm.gt.1.e-13) then
            144:
155 145:
                                                        Hg_dep(i, 1, j, p_hg_{180_105ncdep_cfpp}) = Hg_dep(i, 1, j, p_hg_{180_105ncdep_cfpp}) + 750.*(colsum_b) + 750.*(colsum
            146:
                                                         scav_hgii_180_slo_asgm=(colsum_hgii_180_slo_asgm-(750.*(colsum_hgii_180_slo_asgm/dz8wsu
            147:
                                                   else
                                                         scav_hgii_180_slo_asgm=1.0
            148:
            149:
                                                   endif
                                                   do k=kts, kte
160 150:
```

	151:	if (p_phy(i,k,j).gt.60000and.dz8wsum.gt.100.) then
	152:	chem(i,k,j,p_hgii_180_slo_asgm) = scav_hgii_180_slo_asgm*chem(i,k,j,p_hgii_180_slo_asgr
	153:	
	154:	else
165	155:	continue
	156:	endif
	157:	enddo
	158:	
	159: else	
170	160: continu	ue
	161: endif	
	162: enddo	
	163: enddo	
	164:	
175	165: ! dry_de	p
	166:	
	167: do i = i	ts, ite
	168: do j = j	ts,jte
	169:	do k=1,1
180	170:	$ppm2mgm3 = ((200.59 * p_phy(i,k,j)) / (t_phy(i,k,j) * 8.314472)) * 0.001$
	171:	$Hg_dep(i, 1, j, p_hg_{180_{105_{dd_cfpp}}) = Hg_dep(i, 1, j, p_hg_{180_{105_{dd_cfpp}}) + ppm_2mg_{m3} * chem(i, 1, j, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$
	172:	chem(i,k,j,p_hgii_180_slo_asgm) = chem(i,k,j,p_hgii_180_slo_asgm) * (1 (0.01 * dtst
	173:	enddo
	174: enddo	
185	175: enddo	
	176:	
	177:	end subroutine Hg_tracer_ASGM