

# Updates to the CLM fire module for coupling mode

Maria Val Martin (Sheffield), Colette Heald (MIT) and Jeff Pierce (CSU)

Francis Vitt and Christine Wiedinmyer (NCAR)



\*Part of the project *Estimating the Effects of Changing Climate on Fires and Consequences for U.S. Air Quality*

# UPDATED AND EXPANDED FIRE EMISSION FACTORS

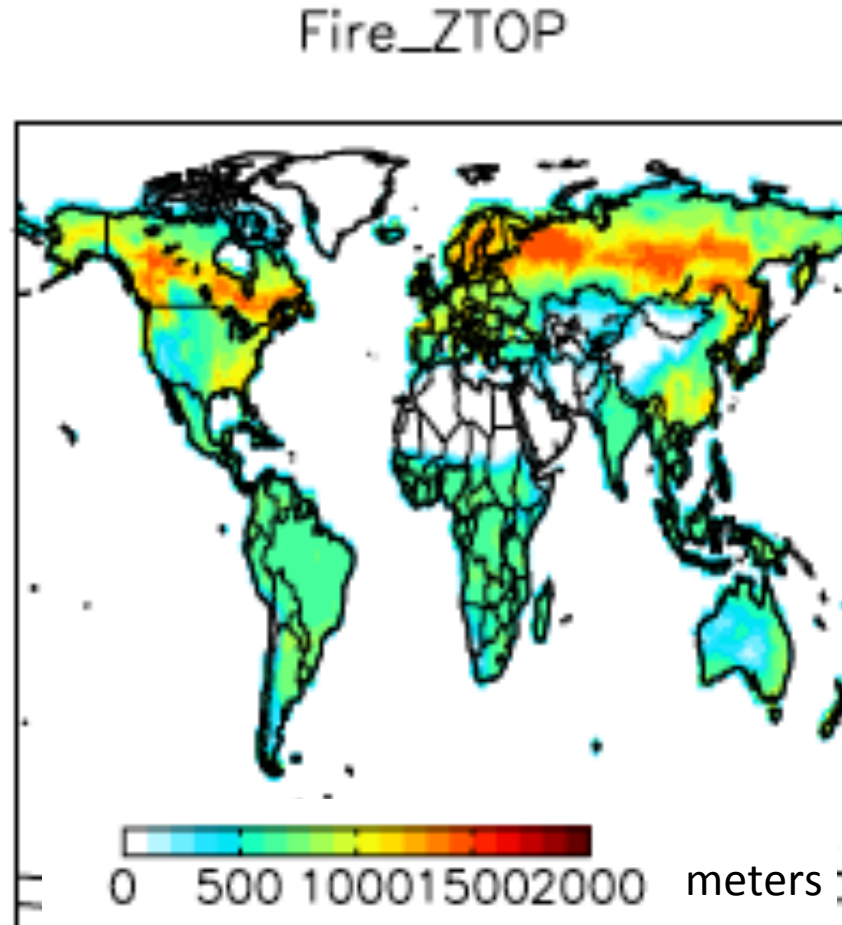
**CO2 CO CH4 NMHC H2 NOx NO NO2 NH3 N2O PM25**  
**TPM TC OC BC SO2 APIN BCARY BENZENE BIGALK**  
**BIGENE BPIN BZALD C2H2 C2H4 C2H5OH C2H6 C3H6**  
**C3H8 CH2O CH3CHO CH3CN CH3COCH3 CH3COCHO**  
**CH3COOH CH3OH GLYALD GLYOXAL HCN HCOOH HONO**  
**HYAC ISOP LIMON MACR MEK MVK MYRC ONIT**  
**PHENOL TOLUENE XYLENE**

Original Species

New Species

- Updated original fire EFs [Andrea and Merlet, (2001)] with the most up-to-date information [Akagi et al., 2011, 2013; Yokelson et al., 2013; McMeeking et al JGR 2009]
- Most VOC EFs are based on Wiedinmyer et al., (2011), with some updates
- VOCs and HONO species will provide a better representation of fire impacts on surface ozone and secondary organic aerosols.

# IMPLEMENTED A SIMPLE SMOKE INJECTION HEIGHT PARAMETERIZATION



Based on satellite-based plume height observations [Val Martin et al., 2010, Tosca et al., 2011, Jian et al., 2013]

# IMPLEMENTED A SMOKE INJECTION HEIGHT PARAMETERIZATION

## FireEmissionsMod.F90

```
function vert_dist_top( veg_type ) result(ztop)
  integer, intent(in) :: veg_type
  real(r8) :: ztop

!ztop = 3.e3_r8 ! m -- this is the value of ztop hard wired in vert_fire_emis.F90
!ztop compiled from Val Martin et al ACP 2010, Tosca et al. JGR 2011 and Jian et al., ACP 2013

!ndllf_evr_tmp_tree (1)
!ndllf_evr_brl_tree (2)
!ndllf_dcd_brl_tree (3)
!nbrdlf_dcd_brl_tree (8)
  if (veg_type == 1 .or. veg_type == 2 .or. veg_type == 3 .or. veg_type == 8) then
    ztop = 1.5e3_r8 ! m
  end if

  ! nbrdlf_evr_tmp_tree (5)
!nbrdlf_dcd_tmp_tree (7)
!nbrdlf_evr_shrub (9)
!nbrdlf_dcd_tmp_shrub (10)
!nbrdlf_dcd_brl_shrub (11)
  if (veg_type == 5 .or. veg_type == 7 .or. veg_type == 9 .or. veg_type == 10 .or. veg_type == 11) then
    ztop = 1.0e3_r8 ! m
  end if

!nbrdlf_evr_trp_tree (4)
!nbrdlf_dcd_trp_tree (6)
!nc3_arctic_grass (12)
!nc3_nonarctic_grass (13)
!nc4_grass (14)
!nc3corn (15)
!nc3wheat (16)
  if (veg_type == 4 .or. veg_type == 6 .or. veg_type == 12 .or. veg_type == 13 .or. veg_type == 14 .or. veg_type == 15 .or.
veg_type == 16) then
    ztop = 0.7e3_r8 ! m
  end if

end function vert_dist_top
```

# IMPLEMENTED A SIMPLE SMOKE INJECTION HEIGHT PARAMETERIZATION

vert\_fire\_emis.F90

```
=====
! vertically distributes surface fire fluxes
=====
subroutine vert_fire_emis_distrib( ncol, lchnk, zint, fire_sflx, fire_ztop, frcing )

  use shr_const_mod,    only : avogad => SHR_CONST_AVOGAD
  use chem_mods,        only : adv_mass ! g/mole

  ! args
  integer,          intent(in) :: ncol,lchnk
  real(r8),         intent(in) :: zint(:,:) ! interface geopot above surface (km)
  real(r8),pointer, intent(in) :: fire_sflx(:,:) ! fire surface emissions (kg/m2/sec)
  real(r8),pointer, intent(in) :: fire_ztop(:) ! top of vert distribution of fire surface emissions (m)
  real(r8),         intent(inout) :: frcing(:,,:) ! insitu forcings (molecules/cm3/sec)

  ! local vars
  real(r8) :: vertical_fire(pcols,pver), ztop
  integer :: n, i,k
  real(r8) :: fire_frc(ncol,pver)
  real(r8) :: sflx(ncol)

  if (.not.shr_fire_emis_elevated) return
  if (shr_fire_emis_mechcomps_n<1) return

  ! define vertical_fire from Dentener units /m
  do k=1,pver
    do i=1,ncol
      ztop = fire_ztop(i)*1.e-3_r8 ! convert m to km
      if(zint(i,k)<ztop)then
        vertical_fire(i,k)=1.e-3_r8/(ztop-zint(i,pverp))
      elseif(zint(i,k)>ztop.and.zint(i,k+1)<ztop)then
        vertical_fire(i,k)=1.e-3_r8*(ztop-zint(i,k+1))/(zint(i,k)-zint(i,k+1))/(ztop-zint(i,pverp))
      else
        vertical_fire(i,k)=0._r8
      endif
    enddo
  enddo
enddo
```

