



**UNIVERSITY OF
CAMBRIDGE**
Department of Chemistry



**National Centre for
Atmospheric Science**
NATURAL ENVIRONMENT RESEARCH COUNCIL

Emissions in UKCA: What goes in must... do something?!

Introduction to UKCA, 4-8 Jan 2016, Cambridge

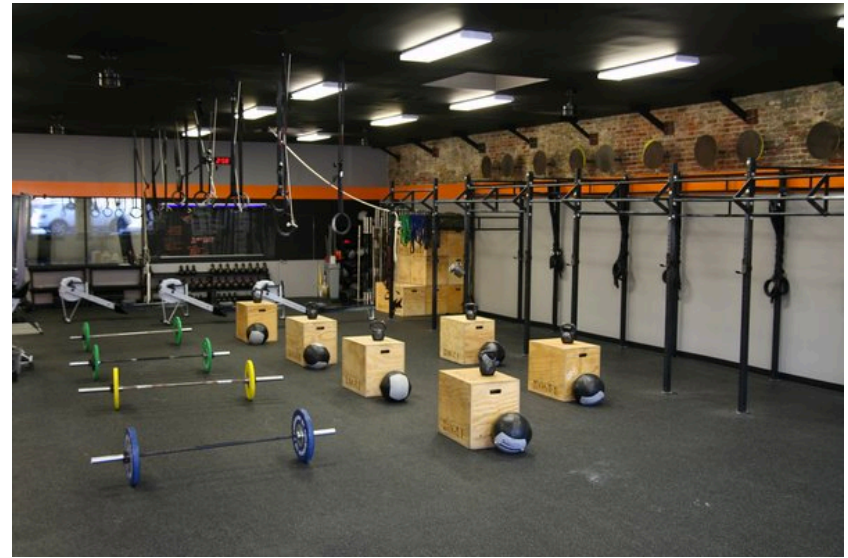
Carlos Ordóñez carlos.ordonez@metoffice.gov.uk
Alex Archibald ata27@cam.ac.uk

Garbage in garbage out!

- It's as simple as that really. Numerical models take inputs, operate on these and produce outputs.
- If you want high quality, meaningful outputs then you need to think about BOTH what your model is doing and what you are putting into your model.



+

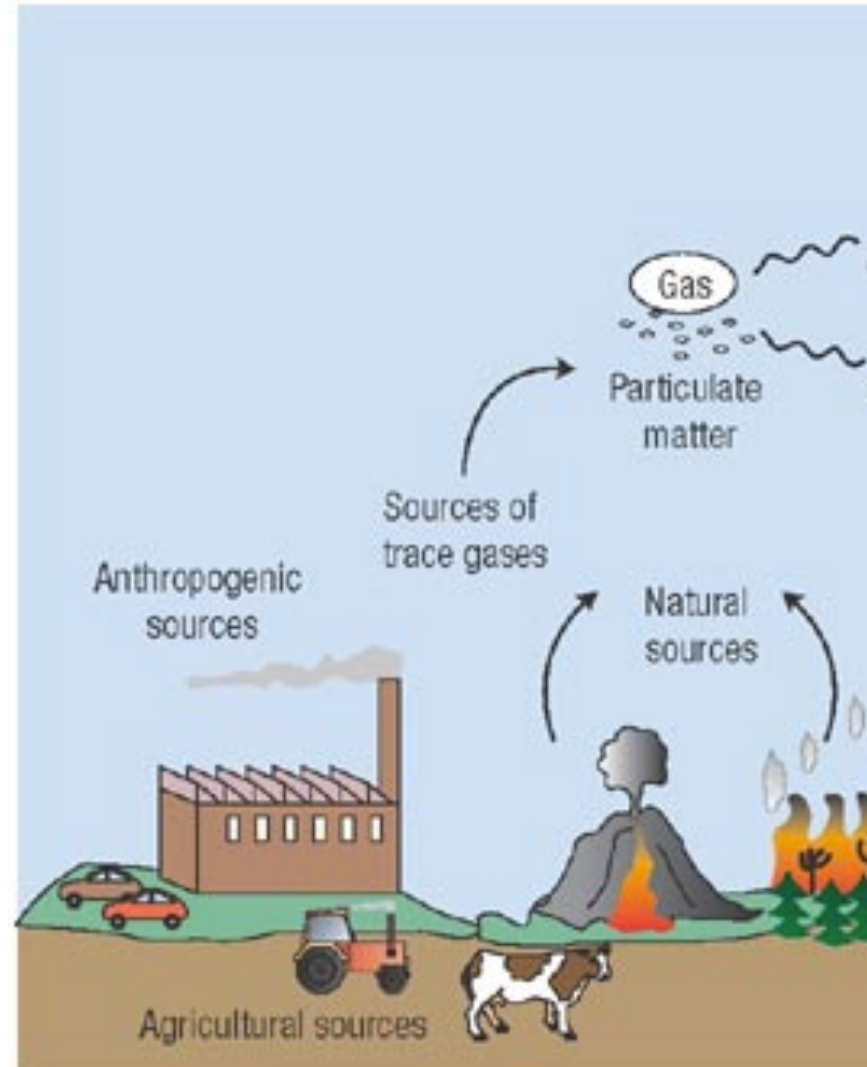


~~||~~



Sources of emissions:

- There are mainly two sources of emissions:
 - Natural
 - Anthropogenic
- We can also divide emissions by their phase (gas vs aerosol).

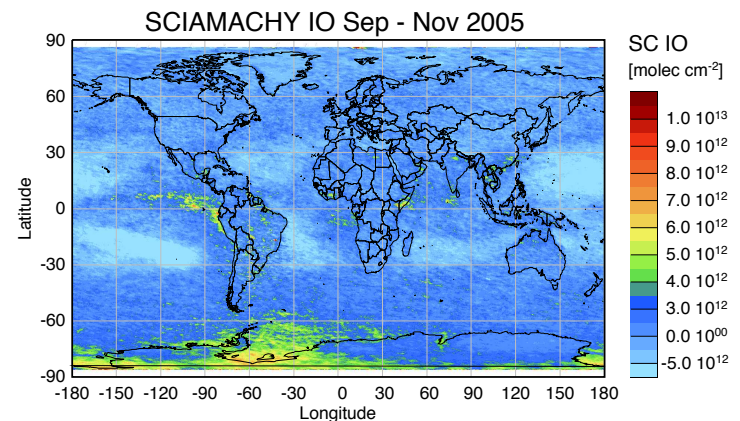
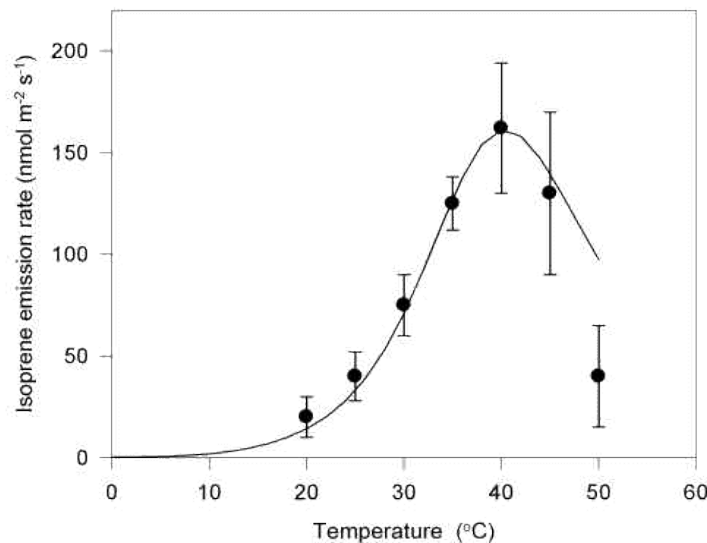




Met Office

Natural emissions

- Produced through natural processes e.g. photosynthesis, respiration, wild fires etc. So often have a dependence on environmental factors
- Some of the most interesting “new” research topics deal with the feedbacks between climate and natural emissions.



$$Flux_{HOI} = [O_{3(g)}] * \sqrt{[I^-(aq)]} * \left(\frac{3.56 \times 10^5}{ws} - 2.16 \times 10^4 \right)$$



Met Office

Anthropogenic emissions

- Produced by man-made processes e.g. pasture burning, agriculture, industry etc. Can have “environmental dependence” but generally not.
- For some compounds anthropogenic emissions are the dominate source.
- Predicted to change due to socio-economic factors.



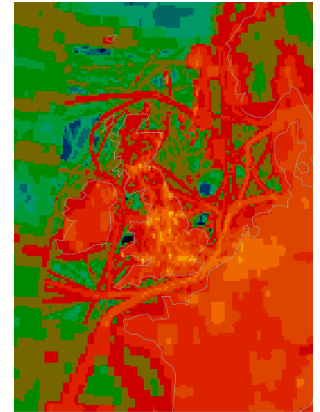


Treatment of emissions in UKCA

1. Introduction
2. Ancillary emission system
3. NetCDF emission system
4. Final notes on UKCA
5. Getting hold of emissions data

Emission inputs to UKCA

- **Offline:** Pre-computed fluxes
 - Read in from ancillary or NetCDF files
 - 12-monthly varying 2-D or 3-D fields
- **Online:** Computed in real-time using fields from the UM. Examples:
 - Lightning NO_x
 - Sea-salt (in GLOMAP-mode)
 - Wetland CH_4 (from JULES)
 - Interactive BVOC (from JULES; for isoprene, terpenes, methanol, acetone; since UM vn9.2).



Gas phase emissions

Emission Species	Units	Std Trop	Trop-Isop	RAQ Chem	Strat Chem	Strat+ Trop
NOx Surface Emissions	kg(NO ₂)/m ² /s	Y	Y	Y	Y	Y
CH ₄ Surface Emissions	kg(CH ₄)/m ² /s	Y	Y	Y	Y	Y
CO Surface Emissions	kg/m ² /s	Y	Y	Y	Y	Y
HCHO Surface Emissions	kg/m ² /s	Y	Y	Y	Y	Y
C ₂ H ₆ Surface Emissions	kg/m ² /s	Y	Y	Y		Y
C ₃ H ₈ Surface Emissions	kg/m ² /s	Y	Y	Y		Y
CH ₃ COCH ₃ Surface Emissions	kg/m ² /s	Y	Y	Y		Y
CH ₃ CHO Surface Emissions	kg/m ² /s	Y	Y	Y		Y
C ₅ H ₈ (Isop) Surface Emiss	kg(C)/m ² /s		Y	Y		Y
H ₂ Surface Emissions	kg/m ² /s			Y		
C ₄ H ₁₀ Surface Emissions	kg/m ² /s			Y		
C ₂ H ₄ Surface Emissions	kg/m ² /s			Y		
C ₃ H ₆ Surface Emissions	kg/m ² /s			Y		
Toluene Surface Emissions	kg/m ² /s			Y		
o-xylene Surface Emissions	kg/m ² /s			Y		
CH ₃ OH Surface Emissions	kg(C)/m ² /s			Y		
NOx Aircraft Emissions (3D)	kg(NO ₂)/m ² /s	Y	Y	Y	Y	Y

Aerosol & online emissions

Emission Species	Units	Std Trop	Trop-Isop	RAQ Chem	Strat Chem	Strat+ Trop
<i>If using Aerosol Chem</i>						
Monoterpene Surface Emiss	kg(C)/m ² /s	Y	Y			Y
NVOC Surface Emiss	kg(C)/m ² /s		Y			Y
SO2 Surface Emissions	kg(S)/m ² /s	Y	Y		Y	Y
DMS Surf Emiss (Land,ocean)	kg(S)/m ² /s	Y	Y		Y	Y
NH3 Surface Emissions	kg/m ² /s	Y	Y			Y
SO2 High (Ind,forest,ship)	kg(S)/m ² /s	Y	Y		Y	Y
SO2 Volcanic Emissions (3D)	kg(S)/m ² /s	Y	Y		Y	Y
<i>for GLOMAP-mode (if using 5-mode setup)</i>						
BC Fossil Fuel Emiss	kg(C)/m ² /s	Y	Y			Y
OC Fossil Fuel Emiss	kg(C)/m ² /s	Y	Y			Y
BC Biofuel Emissions	kg(C)/m ² /s	Y	Y			Y
OC Biofuel Emissions	kg(C)/m ² /s	Y	Y			Y
BC Biomass burn Emiss (3D)	kg(C)/m ² /s	Y	Y			Y
OC Biomass burn Emiss (3D)	kg(C)/m ² /s	Y	Y			Y
<i>Online Emissions</i>						
CH4 Wetland Emiss (in LSH - Optional)	kg(C)/m ² /s	Y	Y	Y	Y	Y
Lightning NOx (in UKCA)	kg(NO2)/kg(air)/cell/s	Y	Y	Y	Y	Y
Sea-Salt Emiss (in GLOMAP-mode)	number/m ² /s	Y	Y		Y	Y

2. Ancillary emission system

Emission files & STASH

UKCA Makes use of **user ancillary files**:

- **single-level**: items 301-320 (STASH section 0)
- **multi-level**: items 321-340 (STASH section 0)

STASH code	Emission Species	Units	Std Trop	Trop- Isop	RAQ Chem	Strat Chem	Strat+ Trop
301	NOx Surface Emissions	kg(NO ₂)/m ² /s	Y	Y	Y	Y	Y
302	CH ₄ Surface Emissions	kg(CH ₄)/m ² /s	Y	Y	Y	Y	Y
303	CO Surface Emissions	kg/m ² /s	Y	Y	Y	Y	Y
304	HCHO Surface Emissions	kg/m ² /s	Y	Y	Y	Y	Y
305	C ₂ H ₆ Surface Emissions	kg/m ² /s	Y	Y	Y		Y
306	C ₃ H ₈ Surface Emissions	kg/m ² /s	Y	Y	Y		Y
307	CH ₃ COCH ₃ Surface Emissions	kg/m ² /s	Y	Y	Y		Y
308	CH ₃ CHO Surface Emissions	kg/m ² /s	Y	Y	Y		Y
309	C ₅ H ₈ (Isop) Surface Emiss	kg(C)/m ² /s		Y	Y		Y
...							
340	NOx Aircraft Emissions (3D)	kg(NO ₂)/m ² /s	Y	Y	Y	Y	Y

Rest of items in section 0 reserved for UM prognostics

Array “em_chem_spec” (1)

- Array with list of emitted species/fields**

Scheme specific. Declared in **ukca_d1_defs.F90**:

```
CHARACTER(LEN=10), DIMENSION(:), ALLOCATABLE, SAVE :: em_chem_spec
```

- Allocated & filled in ukca_setd1defs.F90**

(IF block for different chemistry schemes)

```
...
ELSE IF (L_UKCA_RAQ) THEN      ! Regional air quality chemistry (RAQ)
  n_chem_emissions = 16
  n_3d_emissions   = 1         ! aircraft NOx
  ALLOCATE(em_chem_spec(n_chem_emissions+n_3d_emissions))
  em_chem_spec =
    (/ 'NO', 'CH4', 'CO', 'HCHO',
      'C2H6', 'C3H8', 'Me2CO', 'MeCHO',
      'C5H8', 'C4H10', 'C2H4', 'C3H6',
      'TOLUENE', 'oXYLENE', 'CH3OH', 'H2',
      'NO_aircraft' /)
    &
    &
    &
    &
```

NO_x emissions: Expressed as kg(NO₂) m⁻² s⁻¹ but assigned to the NO tracer!

Array “em_chem_spec” (2)

...

```
ELSE IF (L_ukca_strattrop .AND. L_ukca_achem) THEN
  n_chem_emissions = 21
  n_3d_emissions   = 2           ! volc SO2 & aircraft NOX
  ALLOCATE(em_chem_spec(n_chem_emissions+n_3d_emissions))
  em_chem_spec =
    (/ 'NO          ', 'CH4          ', 'CO          ', 'HCHO          ', &
      'C2H6          ', 'C3H8          ', 'Me2CO         ', 'MeCHO         ', &
      'C5H8          ', 'BC_fossil ', 'BC_biofuel', 'OC_fossil ', &
      'OC_biofuel', 'Monoterp ', 'NVOC          ', 'SO2_low      ', &
      'SO2_high      ', 'NH3          ', 'DMS          ', 'SO2_nat      ', &
      'BC_biomass', 'OC_biomass', 'NO_aircraft' /)
```

NO_x emissions: Expressed as kg(NO₂) m⁻² s⁻¹ but assigned to the NO tracer!

If adding new emission field:

- Look for your chemistry scheme in the IF block of UKCA_SETD1DEFS. Update em_chem_spec.
- Need new STASH code?
- Need to update long block towards the end of UKCA_SETD1DEFS?

(See tutorial in the afternoon)

Code in UKCA_MAIN (ukca_main1-ukca_main1.F90)

```
IF ( L_ukca_new_emiss ) THEN  
    CALL ukca_new_emiss_ctl ( ... )  
ELSE  
    CALL UKCA_EMISSION_CTL ( ... )  
END IF
```

← NetCDF (since UM vn8.6)

← ancillaries

Both cases accept online emissions

ukca_emission_ctl.F90 (1)

Beginning of the routine → Some initialisations & IF blocks affecting different UKCA configurations/options. Examples:

```
! Call routine for primary emissions for UKCA-MODE
IF (L_ukca_mode) THEN                                ! Number & mass emission fluxes
    CALL ukca_mode_ems_um                            ! are assembled for each tracer
END IF                                                ! in array em_field_mode

! Regrid interactive BVOC emissions from landpoints (1D) to 2D-grid
IF (L_bvoc_emis) THEN
    We get isoprene_2D, terpene_2D, methanol_2D, acetone_2D
END IF
```

Then very **long loop through all gas phase tracers** to (i) **add surface emissions**, including unit conversions if needed, and (ii) do **boundary layer mixing of tracers**. See next slide ...



ukca_emission_ctl.F90 (2)

! Check if tracer has **surface emissions** and set emission. Otherwise emission field is zero from initialisation.

```
DO k = 1, jpctr      ! loop over tracers
  DO l = 1, n_chem_emissions
```

```
!    Rest of emissions
    ELSE IF (adv_t(k) == em_chem_spec(l) ) THEN
      em_field(:, :, k) = all_emissions(:, :, l)
    ENDIF      ! end adv_t(k)
  END DO      ! l=1,n_use_emissions
```

```
END DO      ! end of loop over tracers
```



ukca_emission_ctl.F90 (2)

! Check if tracer has **surface emissions** and set emission. Otherwise emission field is zero from initialisation.

DO k = 1, jpctr **! loop over tracers**

DO l = 1, n_chem_emissions

IF (adv_t(k) == em_chem_spec(l) .AND. em_chem_spec(l) == 'NO ') **THEN**

! Convert from kg NO₂/m²/s to kg NO/m²/s

 em_field(:, :, k) = all_emissions(:, :, l) * m_no / m_no2

! **Rest of emissions**

ELSE IF (adv_t(k) == em_chem_spec(l)) **THEN**

 em_field(:, :, k) = all_emissions(:, :, l)

ENDIF **! end adv_t(k)**

END DO **! l=1, n_use_emissions**

END DO **! end of loop over tracers**



ukca_emission_ctl.F90 (2)

! Check if tracer has **surface emissions** and set emission. Otherwise emission field is zero from initialisation.

DO k = 1, jpctr **! loop over tracers**

DO l = 1, n_chem_emissions

IF (adv_t(k) == em_chem_spec(l) .AND. em_chem_spec(l) == 'NO') THEN

! Convert from kg NO₂/m²/s to kg NO/m²/s

 em_field(:, :, k) = all_emissions(:, :, l) * m_no / m_no2

ELSE IF ... **! Many other checks (for SO₂_low, DMS, Monoterp, C₅H₈, MeOH, Me₂CO, ...).**

! **Note that BVOC emissions might be interactive**

!

! If you update the code check if there are specific unit conversions for your

! species and make sure that is consistent with your emission fields.

....

! **Rest of emissions**

ELSE IF (adv_t(k) == em_chem_spec(l)) THEN

 em_field(:, :, k) = all_emissions(:, :, l)

ENDIF **! end adv_t(k)**

END DO **! l=1, n_use_emissions**

END DO **! end of loop over tracers**



ukca_emission_ctl.F90 (2)

! Check if tracer has **surface emissions** and set emission. Otherwise emission field is zero from initialisation.

DO k = 1, jpctr **! loop over tracers**

DO l = 1, n_chem_emissions

IF (adv_t(k) == em_chem_spec(l) .AND. em_chem_spec(l) == 'NO') THEN

! Convert from kg NO₂/m²/s to kg NO/m²/s

 em_field(:, :, k) = all_emissions(:, :, l) * m_no / m_no2

ELSE IF ... **! Many other checks (for SO₂_low, DMS, Monoterp, C₅H₈, MeOH, Me₂CO, ...).**

! **Note that BVOC emissions might be interactive**

!

! If you update the code check if there are specific unit conversions for your
! species and make sure that is consistent with your emission fields.

....

! **Rest of emissions**

ELSE IF (adv_t(k) == em_chem_spec(l)) THEN

 em_field(:, :, k) = all_emissions(:, :, l)

ENDIF **! end adv_t(k)**

END DO **! l=1, n_use_emissions**

IF (adv_t(k) == 'CH₄' ...) THEN

! Add wetland CH₄ emissions or prescribe the surface mixing ratio

IF (L_ukca_strat .OR. L_ukca_stratcfc .OR. L_ukca_strattrop) THEN

! Treatment of long-lived species with lower boundary condition

END DO **! end of loop over tracers**



ukca_emission_ctl.F90 (2)

! Check if tracer has **surface emissions** and set emission. Otherwise emission field is zero from initialisation.

DO k = 1, jpctr **! loop over tracers**

DO l = 1, n_chem_emissions

IF (adv_t(k) == em_chem_spec(l) .AND. em_chem_spec(l) == 'NO ') THEN

! Convert from kg NO₂/m²/s to kg NO/m²/s

em_field(:, :, k) = all_emissions(:, :, l) * m_no / m_no2

ELSE IF ... ! Many other checks (for SO₂_low, DMS, Monoterp, C₅H₈, MeOH, Me₂CO, ...).

! **Note that BVOC emissions might be interactive**

!

! If you update the code check if there are specific unit conversions for your
! species and make sure that is consistent with your emission fields.

....

! **Rest of emissions**

ELSE IF (adv_t(k) == em_chem_spec(l)) THEN

em_field(:, :, k) = all_emissions(:, :, l)

ENDIF ! end adv_t(k)

END DO ! l=1,n_use_emissions

IF (adv_t(k) == 'CH₄ ' ...) THEN

! Add wetland CH₄ emissions or prescribe the surface mixing ratio

IF (L_ukca_strat .OR. L_ukca_stratcfc .OR. L_ukca_strattrop) THEN

! Treatment of long-lived species with lower boundary condition

! **Call boundary layer mixing and add surface emissions.**

CALL TR_MIX (..., em_field(:, :, k), ..., tracers(:, :, 1:bl_levels, k),)

END DO ! end of loop over tracers

ukca_emission_ctl.F90 (3a)

Similar stuff for GLOMAP-mode

! Beginning of ukca_emission_ctl. Call routine for primary emissions for UKCA-MODE

```
IF (L_ukca_mode) THEN  
  CALL ukca_mode_ems_um → em_field_mode is filled with nr & mass  
                           emission fluxes for each tracer  
END IF
```

1. Initial emission arrays for aerosols are created from the **input data** (depending on the model set-up)
2. **CALL ukca_mode_ems**: returns **mass** & **number emission arrays** for sulphate, sea-salt, OC, BC and dust (as required)
3. The number and mass fluxes are then assembled for each tracer in the **array em_field_mode**
4. **Diagnostics** for emitted component mass are stored in STASH

```
| 38 | 201 | PRIMARY SO4 TO AITKEN (SOL)  
| 38 | 202 | PRIMARY SO4 TO ACCUMULATION (SOL)  
| 38 | 203 | PRIMARY SO4 TO COARSE (SOL)  
| 38 | 204 | PRIMARY SEA-SALT TO ACCUMULTN (SOL)  
| 38 | 205 | PRIMARY SEA-SALT TO COARSE (SOL)  
| 38 | 206 | PRIMARY BLACK CARBON TO AITKEN (SOL)  
| 38 | 207 | PRIMARY BLACK CARBON TO AITKEN (INS)  
| 38 | 208 | PRIMARY ORG. CARBON TO AITKEN (SOL)  
| 38 | 209 | PRIMARY ORG. CARBON TO AITKEN (INS)  
| 38 | 210 | PRIMARY DUST TO ACCUMULATION (SOL)  
| 38 | 211 | PRIMARY DUST TO ACCUMULATION (INS)  
| 38 | 212 | PRIMARY DUST TO COARSE (SOLUBLE)
```

ukca_emission_ctl.F90 (3b)

Similar stuff for GLOMAP-mode

! Beginning of ukca_emission_ctl. Call routine for primary emissions for UKCA-MODE

```
IF (L_ukca_mode) THEN
  CALL ukca_mode_ems_um    → em_field_mode is filled with nr & mass
                           emission fluxes for each tracer
END IF
```

! After doing injection & mixing of gas tracers. Do same for MODE aerosols

```
IF (L_ukca_mode) THEN

  ! Call boundary layer mixing and add surface emissions
  ! for GLOMAP-mode aerosol tracers

  DO kaer = 1,n_mode_tracers                ! loop over tracers
    CALL TR_MIX (... , em_field_mode (:, :, 1, kaer), ...,
                  mode_tracers (:, :, 1:bl_levels, kaer), ...)

  END DO                                     ! end of loop over tracers (kaer)

END IF ! if L_UKCA_MODE
```

ukca_emission_ctl.F90 (3b)

Similar stuff for GLOMAP-mode

! Beginning of ukca_emission_ctl. Call routine for primary emissions for UKCA-MODE

```
IF (L_ukca_mode) THEN
  CALL ukca_mode_ems_um    → em_field_mode is filled with nr & mass
                           emission fluxes for each tracer
END IF
```

! After doing injection & mixing of gas tracers. Do same for MODE aerosols

```
IF (L_ukca_mode) THEN

  ! Call boundary layer mixing and add surface emissions
  ! for GLOMAP-mode aerosol tracers

  DO kaer = 1,n_mode_tracers                ! loop over tracers

    CALL TR_MIX (... , em_field_mode (:, :, 1, kaer), ...,
                   mode_tracers (:, :, 1:bl_levels, kaer), ...)

    ! Add in emission fluxes not at surface level
    DO k = 2,model_levels
      CALL TRSRCE (... , mode_tracers (:, :, k, kaer), ...,
                  em_field_mode(:, :, k, kaer), ...)
    END DO ! loop over model levels 2 to top (k)

  END DO                                     ! end of loop over tracers (kaer)

END IF ! if L_UKCA_MODE
```

ukca_emission_ctl.F90 (4)

! Diagnose NO2 lightning emissions

```
lightningem_n_gridbox = 0.0
```

```
lightningem_no2_to_air = 0.0
```

```
CALL UKCA_LIGHT_CTL( ...                                &
    lightningem_n_gridbox (1:row_length,1:rows,1:model_levels), &
    lightningem_no2_to_air (1:row_length,1:rows,1:model_levels))
```

! Convert aircraft emissions from kg NO2/gridbox/s to kg NO/m2/s

Within some loops:

```
conv_aircraftems(j,k,l) = aircraftems(j,k,l)           &
    * m_no/(surf_area(j,k)*m_no2)
```

! Add aircraft emissions to NO or NOx tracer

```
DO k = 1,model_levels
```

```
CALL TRSRCE (                                           &
    ..., tracers(:, :, k, inox), conv_aircraftems(:, :, k), k, ...)
```

```
END DO
```

! Update tracer fields with NO/NOx lightning emissions

```
tracers(1:row_length,1:rows,:,inox) =                  &
    tracers(1:row_length,1:rows,:,inox) +                &
    timestep*lightningem_no2_to_air*m_no/m_no2
```

ukca_emission_ctl.F90 (5)

Finally, **treatment of SO₂ emissions** (if aerosol chemistry):

- Add (3-D volcanic + high-level anthropogenic) emissions to SO₂ tracer.
NOTE: The code removes direct sulphate fraction of emissions and converts from kg(S) to kg (SO₂)
- Add 3-D emissions from explosive volcanic eruptions (e.g. Pinatubo 1991 and 5 others):

```
CALL UKCA_VOLCANIC_SO2
```

Last note (only for ASAD framework). Some calls to output emission diagnostics:

```
CALL asad_emissions_diagnostics
```

```
CALL asad_3D_emissions_diagnostics
```



Well tested for all UKCA configurations:

- ## Limitations

- ```
em_chem_spec =
 ('NO ',....
 'SO2_low ', 'SO2_high ', 'SO2_nat ',
 ..., 'NO aircraft'/)
```

- © Crown copyright Met Office



## 3. NetCDF emission system



# Introduction

Available since UM vn8.6

## Advantages

- Allow the use of different emission fields to account for independent source sectors for any given tracer
- Emissions can be injected at different altitudes and with different temporal variability
- Avoid inconsistencies in the units of the emission fields (to comply with CF conventions units should be “ $\text{kg m}^{-2} \text{s}^{-1}$ ”)

## Limitations

- Currently tested only for RAQ & Strat-Trop chemistry
- Needs to be extended for:
  - aerosol emissions (`l_ukca_mode .OR. l_ukca_aerchem .OR. l_ukca_nr_aqchem`)
  - iBVOC emissions (`l_ukca_ibvoc .AND. l_bvoc_emis`)

# UKCA code for NetCDF emissions

- As before “**em\_chem\_spec**” is allocated & filled within **ukca\_setd1defs.F90**  
Each element of the array could correspond to 1 or more emission fields
- From **ukca\_main1-ukca\_main1.F90**:

```
IF (L_ukca_new_emiss) THEN
 CALL ukca_new_emiss_ctl (...)
ELSE
 CALL UKCA_EMISSION_CTL (...)
END IF
```

← NetCDF (since UM vn8.6)

← ancillaries

- **ukca\_new\_emiss\_ctl.F90** includes calls to many routines in order to:
  - Go through NetCDF emission files to automatically identify emission fields present in them
  - Initialise an “emissions” structure which will hold all emission fields
  - Update the “emissions” structure when needed
  - Output emission diagnostics



# NAMELIST input through UMUI (only available at UM vn8.6)

Section 34: UKCA Chemistry and Aerosols : Job dlna.f: "cp ansyq (AQU std job from UMSys)"

Choose the relevant section release

- ☐ <0A> UKCA not included.
- ☒ <1A> UKCA included.

Select Chemical Scheme

- ☐ Age of air only
- ☐ Standard Tropospheric(BE)
- ☒ RAQ(BE)
- ☐ Tropospheric+Isoprene
- ☐ Standard Stratospheric
- ☐ Stratospheric + Tropospheric Chemistry

☒ Set Backward Euler Solver Settings to non-default values?

Backward Euler Timestep:

Number of iterations for BE solver:

☐ Include aerosol chemistry

☐ UKCA\_MODE Aerosol Scheme

☐ Interactive wetland CH4 emissions

Specify Tropospheric Options to be included

☐ Use 2D top boundary data?

Directory pathname for the 2D top boundary data:

☐ Switch on Tropospheric Heterogenous Chemistry

Select Stratospheric options to be included:

☐ Switch on water feedback from chemistry

☐ Switch on Heterogenous / PSC chemistry

☐ Use climatological Aerosol for Surface Area

Directory containing climatological aerosol file:

File containing climatological aerosol data:

☐ Use a cyclic, monthly-varying 'background' aerosol field instead of timeseries

Push PHOTO button for photolysis parameters

Push LOWBC button to specify Trace gases and Lower Boundary Conditions

Push COUPL button for Coupling between UKCA and Atmosphere

Push UKCA\_TRA to initialise tracers available

Push MODE to setup aerosol model parameters

Push NEW\_EMISS to set up the new NetCDF emission system

Buttons: Help, Abandon changes, Close, PHOTO, LOWBC, COUPL, UKCA\_TRA, MODE, NEW\_EMISS

Window Name : atmos\_Science\_Section\_UKCA. Job dlna.f.

UKCA NetCDF Emission System : Job dlna.f: "cp ansyq (AQU ...)"

☒ Use new UKCA emissions

Directory pathname for NetCDF emission files:

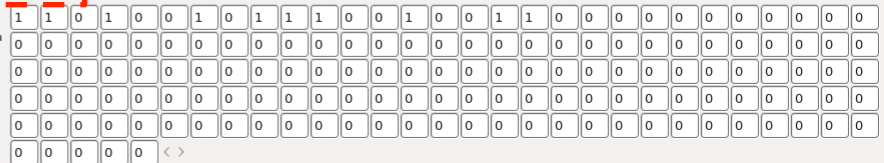
| NetCDF emission files |           |             |
|-----------------------|-----------|-------------|
| Number                | File name | Include Y/N |
| 31                    |           |             |
| 32                    |           |             |
| 33                    |           |             |
| 34                    |           |             |
| 35                    |           |             |
| 36                    |           |             |
| 37                    |           |             |
| 38                    |           |             |
| 39                    |           |             |
| 40                    |           |             |
| Inert                 | Edit      | Edit        |

Remove Blank Lines

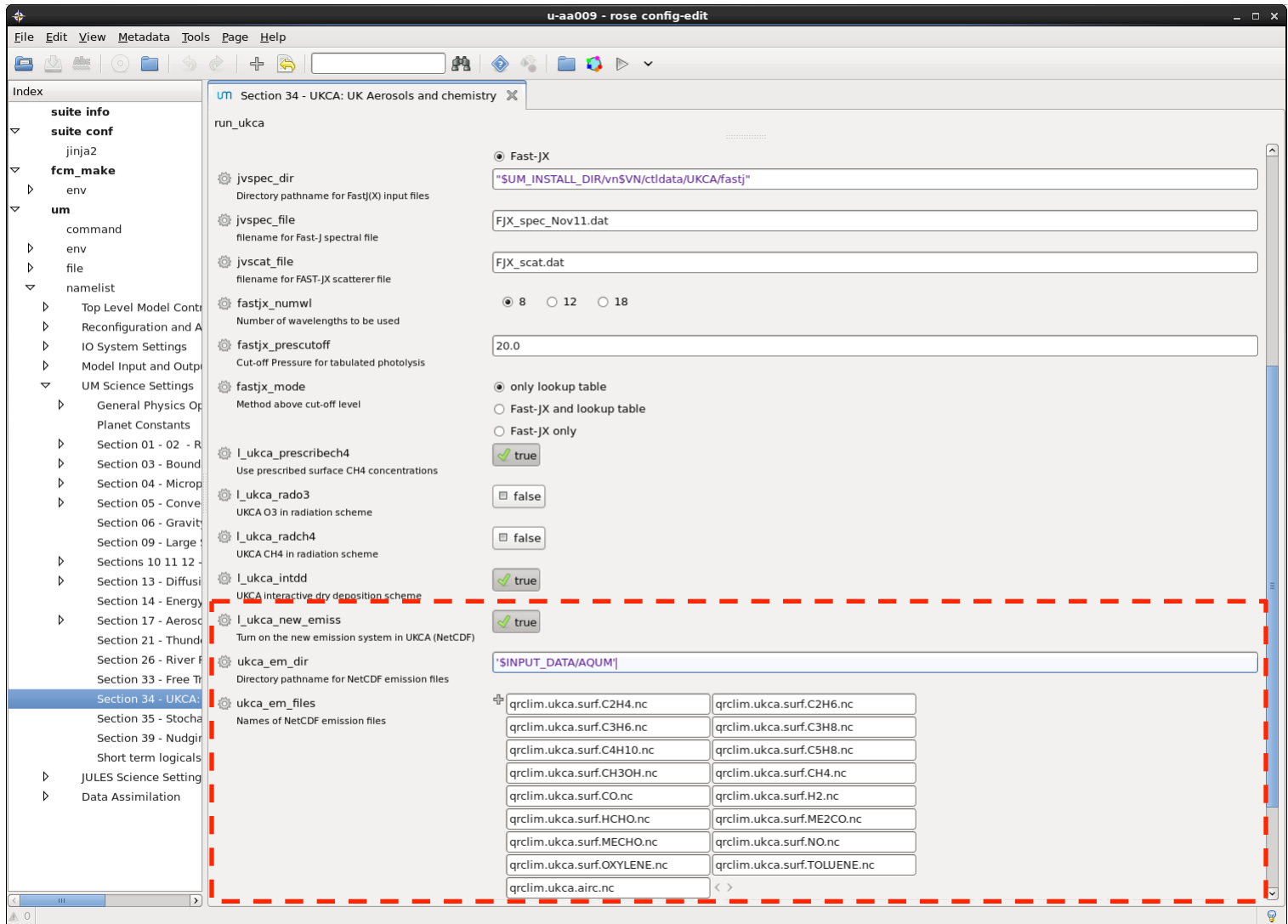
Push UKCA to go to the parent window

Buttons: Help, Abandon changes, Close, UKCA

Window Name : atmos\_Science\_Section\_UKCA\_Emiss. Job dlna.f.



# NAMELIST input through Rose (II)



The screenshot shows the Rose configuration editor for the 'u-aa009 - rose config-edit' window. The left sidebar displays the configuration tree, with 'Section 34 - UKCA: UK Aerosols and chemistry' selected. The main panel shows the configuration for 'run\_ukca' under 'Section 34 - UKCA: UK Aerosols and chemistry'.

The configuration includes the following settings:

- run\_ukca**
  - Fast-JX**
    - `SUM_INSTALL_DIR/vnSVN/ctl/data/UKCA/fastj*`
    - `FJX_spec_Nov11.dat`
    - `FJX_scatter.dat`
    - fastjx\_numwl**: 8 (selected), 12, 18
    - fastjx\_prescutoff**: 20.0
    - fastjx\_mode**
      - only lookup table** (selected)
      - Fast-JX and lookup table
      - Fast-JX only
    - I\_ukca\_prescribech4**: true
    - I\_ukca\_rado3**: false
    - I\_ukca\_radch4**: false
    - I\_ukca\_intdd**: true
    - I\_ukca\_new\_emiss**: true
    - ukca\_em\_dir**: `'$INPUT_DATA/AQUUM'`
    - ukca\_em\_files**
      - Names of NetCDF emission files
      - Table of emission files:

|                             |                             |
|-----------------------------|-----------------------------|
| qrclim.ukca.surf.C2H4.nc    | qrclim.ukca.surf.C2H6.nc    |
| qrclim.ukca.surf.C3H6.nc    | qrclim.ukca.surf.C3H8.nc    |
| qrclim.ukca.surf.C4H10.nc   | qrclim.ukca.surf.C5H8.nc    |
| qrclim.ukca.surf.CH3OH.nc   | qrclim.ukca.surf.CH4.nc     |
| qrclim.ukca.surf.CO.nc      | qrclim.ukca.surf.H2.nc      |
| qrclim.ukca.surf.HCHO.nc    | qrclim.ukca.surf.ME2CO.nc   |
| qrclim.ukca.surf.MECHO.nc   | qrclim.ukca.surf.NO.nc      |
| qrclim.ukca.surf.OXYLENE.nc | qrclim.ukca.surf.TOLUENE.nc |
| qrclim.ukca.airc.nc         |                             |



# Requirements of NetCDF files (1)

- **Maximum nr of emission files: 40**

ukca\_option\_mod.F90:

```
INTEGER, PARAMETER :: nr_cdf_files = 40
CHARACTER (LEN=120) :: ukca_em_dir
CHARACTER (LEN=50) :: ukca_em_files(nr_cdf_files)
```

Each NetCDF file may include one or several emission fields

- **Should comply with:**
  - **CF** conventions (<http://cfconventions.org/>)
  - **Met Office NetCDF** conventions

However UM code could still work if files don't comply.

- Next slides: **Metadata attributes** needed in the files so that the UM code works!

# Requirements of NetCDF files (2)

## Global attributes needed in each emission file:

- **update\_freq\_in\_hours**: frequency in hours at which all emission fields present in that file should be read.

If you want 5 days (as done for ancillaries in many UMUI jobs):

**update\_freq\_in\_hours = 120**

- **emission type** (follows same conventions as for ancillary files):
  - 0: Single time (not allowed yet)
  - 1: Time series
  - 2: Periodic time series



# Requirements of NetCDF files (3)

Some **metadata attributes** required for **each emission field**:

- **name**: Name of the emission field (80 characters, only for debugging)
- **tracer\_name**: This has to be equal to one of the names in the list of emissions for the given chemical scheme, i.e. **em\_chem\_spec**
- **standard\_name** : Compulsory if available. Example for NO:  
    “tendency\_of\_atmosphere\_mass\_content\_of\_nitrogen\_monoxide\_due\_to\_emission”  
    See <http://cfconventions.org/Data/cf-standard-names/26/build/cf-standard-name-table.html>
- **long\_name**: Compulsory if there is no ‘standard\_name’ available. Example:  
    “tendency of atmosphere mass content of nitrogen monoxide due to emission”
- **units** = “kg m<sup>-2</sup> s<sup>-1</sup>”
- **hourly\_scaling, daily\_scaling & vertical\_scaling**: Characters read by the UM to apply corresponding numeric scaling factors



# What UKCA code does with NetCDF attributes

```
! Emission Data structure
TYPE ukca_em_struct
 CHARACTER (LEN=256) :: file_name
 CHARACTER (LEN=80) :: var_name
 CHARACTER (LEN=10) :: tracer_name
 CHARACTER (LEN=256) :: std_name
 CHARACTER (LEN=256) :: lng_name
 CHARACTER (LEN=30) :: units
 INTEGER :: update_freq
 INTEGER :: update_type
 LOGICAL :: l_update
 LOGICAL :: three_dim
 REAL :: base_fact
 REAL, POINTER :: vert_scaling_3d (:,:,)
 CHARACTER (LEN=20) :: hourly_fact
 CHARACTER (LEN=20) :: daily_fact
 CHARACTER (LEN=30) :: vert_fact
 INTEGER :: lowest_lev
 INTEGER :: highest_lev
 REAL, POINTER :: values (:,:,)
 REAL, POINTER :: diags (:,:,)
END TYPE ukca_em_struct

! Super array of emissions
TYPE (ukca_em_struct), ALLOCATABLE :: emissions (:)

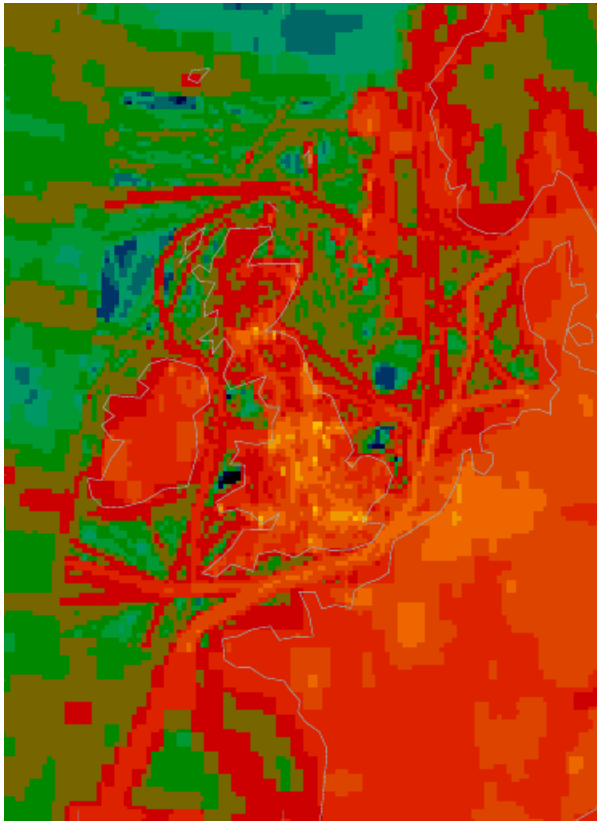
ALLOCATE (emissions(num_em_flds))
```

1. The UKCA code creates an “emissions” structure (derived type variable) which contains objects representing emissions. Each object can be distinguished by some components: emission name, values, units, vertical/temporal profiles, ...
2. Then some of the NetCDF attributes are assigned to those components

| Attribute in NetCDF files     | Corresponding name in emissions structure within the UKCA code |
|-------------------------------|----------------------------------------------------------------|
| name                          | emissions(:)%var_name                                          |
| standard_name                 | emissions(:)%std_name                                          |
| long_name                     | emissions(:)%lng_name                                          |
| tracer_name                   | emissions(:)%tracer_name                                       |
| units                         | emissions(:)%units                                             |
| hourly_scaling                | emissions(:)%hourly_fact                                       |
| daily_scaling                 | emissions(:)%daily_fact                                        |
| vertical_scaling              | emissions(:)%vert_fact                                         |
| lowest_level                  | emissions(:)%lowest_lev                                        |
| highest_level                 | emissions(:)%highest_lev                                       |
| update_freq.in.hours (global) | emissions(:)%update_freq                                       |
| emission_type (global)        | emissions(:)%update_type                                       |

# Example of time / vertical profiles for RAQ (1)

Typical 12-monthly emission field for NO<sub>x</sub> (all source sectors combined)

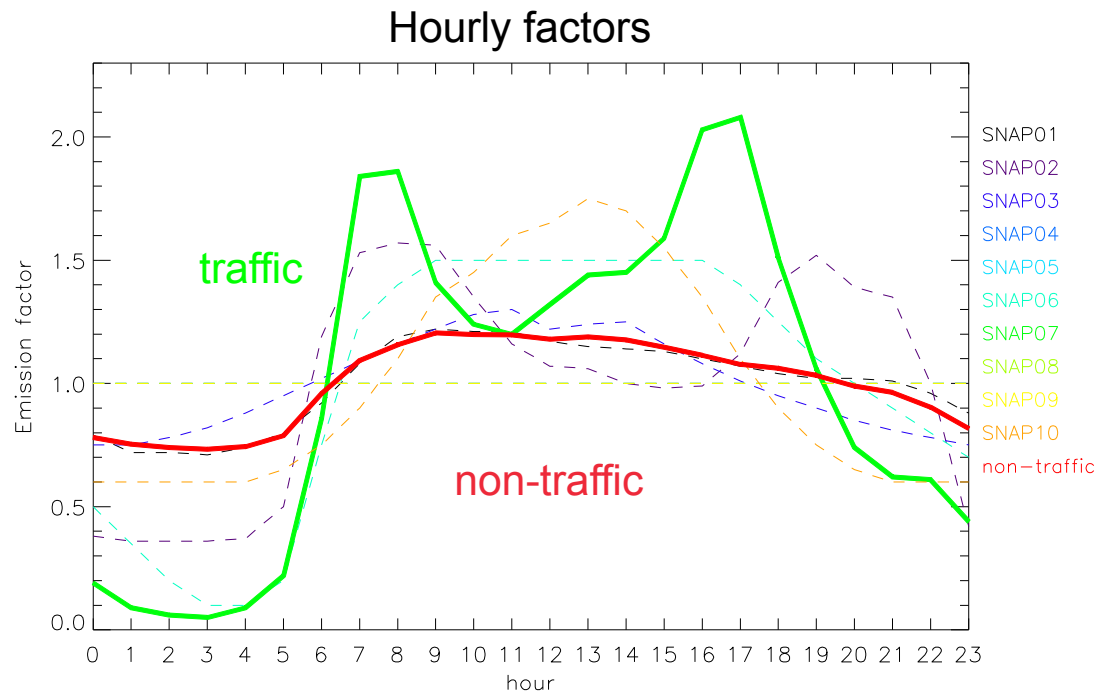


Better approach: Use separate SNAP source sectors

- 1 Combustion in energy and transformation industries
- 2 Non-industrial combustion plants
- 3 Combustion in manufacturing industry
- 4 Production processes
- 5 Extraction and distribution of fossil fuels
- 6 Solvents
- 7 Road transport ( “traffic” )
- 8 Other mobile sources (e.g. shipping)
- 9 Waste treatment and disposal
- 10 Agriculture
- 11 Natural and biogenic sources

# Example of time / vertical profiles for RAQ (1a)

- **Hourly & daily factors** to account for daily and weekly variability in emissions
- Based on data provided by TNO for the MACC project



# Example of time / vertical profiles for RAQ (1b)

## Module UKCA\_EMISS\_FACTORS - Subroutine HOURLY\_EMISS\_FACTORS

**SELECT CASE** (TRIM (hourly\_fact))      ← NetCDF attribute hourly\_scaling

! No hourly factors applied unless specified

CASE ('none', '')

```
 hourly_scaling =
 (/1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00,
 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00,
 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00/)
 &
 &
 &
```

! -----  
! Hourly factors of emissions for Europe. Calculated by TNO  
! for the MACC project.

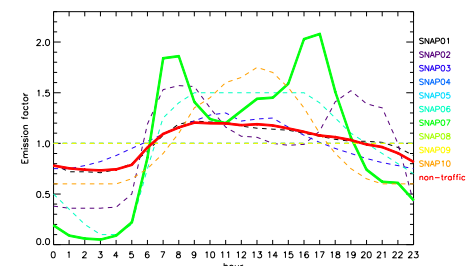
...

CASE ('TNO\_MACC\_EU\_SNAP07')

```
 hourly_scaling =
 (/0.19, 0.09, 0.06, 0.05, 0.09, 0.22, 0.86, 1.84,
 1.86, 1.41, 1.24, 1.20, 1.32, 1.44, 1.45, 1.59,
 2.03, 2.08, 1.51, 1.06, 0.74, 0.62, 0.61, 0.44/)
 &
 &
 &
```

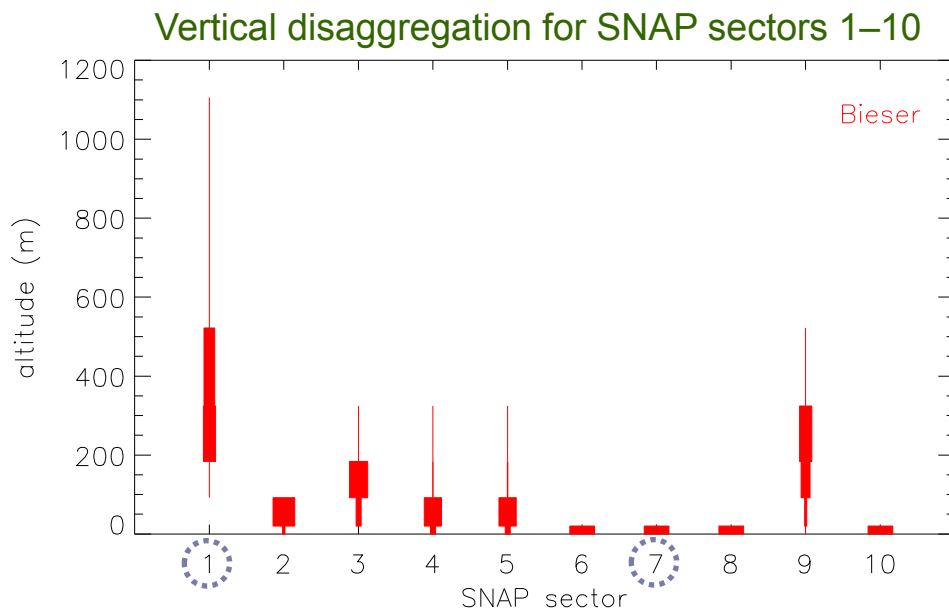
...

**END SELECT**



## Example of time / vertical profiles for RAQ (2a)

**Vertical profiles** tested for anthropogenic emissions in RAQ chemistry scheme



# Example of time / vertical profiles for RAQ (2b)

## Module UKCA\_EMISS\_FACTORS - Subroutine VERTICAL\_EMISS\_FACTORS

```
! Altitude of the interfaces in metres
interf_ref = (/0.0, 20.0, 92.0, 184.0, 324.0, 522.0, 781.0, 1106.0/)

! Get a 3D vertical scaling factor for the given profile
SELECT CASE (TRIM (vert_fact)) ← NetCDF attribute vertical_scaling

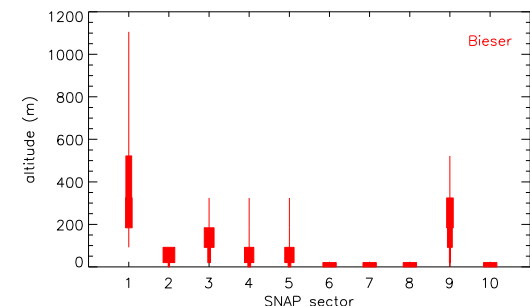
! Average vertical profiles for various SNAP source sectors.
CASE ('Bieser_modified_SNAP01')
 vert_scaling_ref (:) = (/ 0.0, 0.0, 0.25, 51.0, 45.3, 3.25, 0.2 /)
 ...

CASE ('Bieser_modified_SNAP07')
 vert_scaling_ref (:) = (/ 100.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 /)
 ...

! Surface emiss in lowest model lev unless something different specified
CASE ('surface', '')
 vert_scaling_3d (:, :, 1) = 1.0
 vert_scaling_3d (:, :, 2:) = 0.0

! 3D emissions over all model levels
CASE ('all_levels', '3D')
 vert_scaling_3d (:, :, :) = 1.0
 ...

END SELECT
```





Met Office

# How to deal with special units

- **Attribute units in all NetCDF emission fields:**

units = "kg m<sup>-2</sup> s<sup>-1</sup>" or units = "kg/m<sup>2</sup>/s"

- **If you want to express in kg(N), kg(C) or kg(S):**

```
standard_name = "tendency_of_atmosphere_mass_content_of_nox
 expressed_as_nitrogen_due_to_emission"
```

Check <http://cfconventions.org/Data/cf-standard-names/27/build/cf-standard-name-table.html>

```
long_name = "tendency of atmosphere mass content of isoprene
 expressed as carbon due to emission"
```

- See code in 2 routines within the module **ukca\_emiss\_factors**:
  - Strings automatically detected by **base\_emiss\_factors**
  - Conversions done in **get\_base\_scaling**

You might need to adapt them



# New emission diagnostics (1)

## **ukca\_add\_emiss\_mod.F90**

emissions(l)%diags (:, :, 1) filled

Column integrated & with time profiles applied

## **ukca\_emdiags\_struct\_mod.F90:**

Declares a structure for emission diagnostics:

```
TYPE emdiags_struct
 ! Flags indicating if diagnostics selected
 LOGICAL :: l_em_no
 LOGICAL :: l_em_ch4
 ...

 ! Pointers to hold emission diagnostics
 REAL, POINTER :: em_no (:, :)
 REAL, POINTER :: em_ch4 (:, :)
 ...

 REAL, POINTER :: em_no_air (:, :, :)
END TYPE emdiags_struct
```

## **ukca\_update\_emdiagstruct\_mod.F90**

Update emission diagnostic values and flags in the "emdiags\_struct"

## **ukca\_emiss\_diags\_mod.F90**

Output emission diagnostics

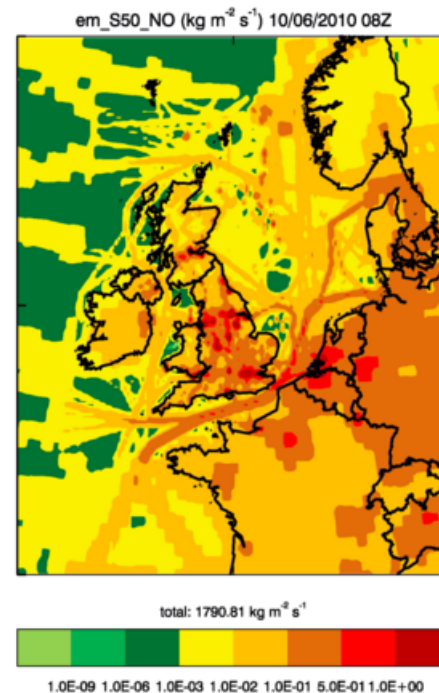
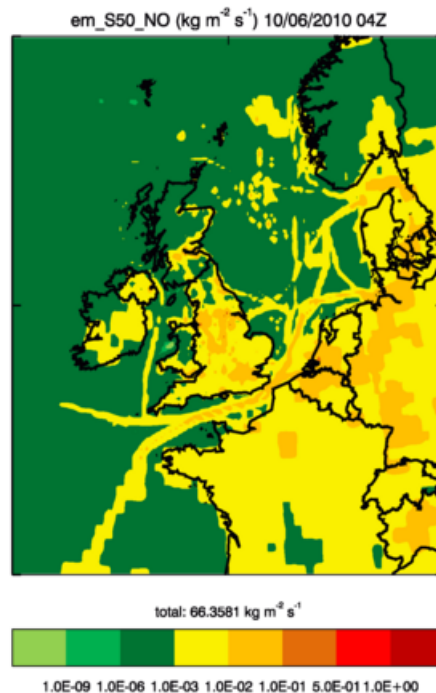
| Emission field<br>(in em_chem_spec) | Item number<br>(in Sect. 50) |
|-------------------------------------|------------------------------|
| 'NO '                               | 156                          |
| 'CH4 '                              | 157                          |
| 'CO '                               | 158                          |
| 'HCHO '                             | 159                          |
| 'C2H6 '                             | 160                          |
| 'C3H8 '                             | 161                          |
| 'Me2CO '                            | 162                          |
| 'MeCHO '                            | 163                          |
| 'C5H8 '                             | 164                          |
| 'C4H10 '                            | 165                          |
| 'C2H4 '                             | 166                          |
| 'C3H6 '                             | 167                          |
| 'TOLUENE '                          | 168                          |
| 'oXYLENE '                          | 169                          |
| 'CH3OH '                            | 170                          |
| 'H2 '                               | 171                          |
| 'NO <sub>aircraft</sub> '           | 172                          |

- Available for RAQ & other schemes
- New items can be added in S50

# New emission diagnostics (2)

## Emission diagnostics for NO:

1. With temporal factors applied
2. Column integrated (always for 2-D emissions)
3. In  $\text{kg (NO) m}^{-2} \text{ s}^{-1}$





Met Office

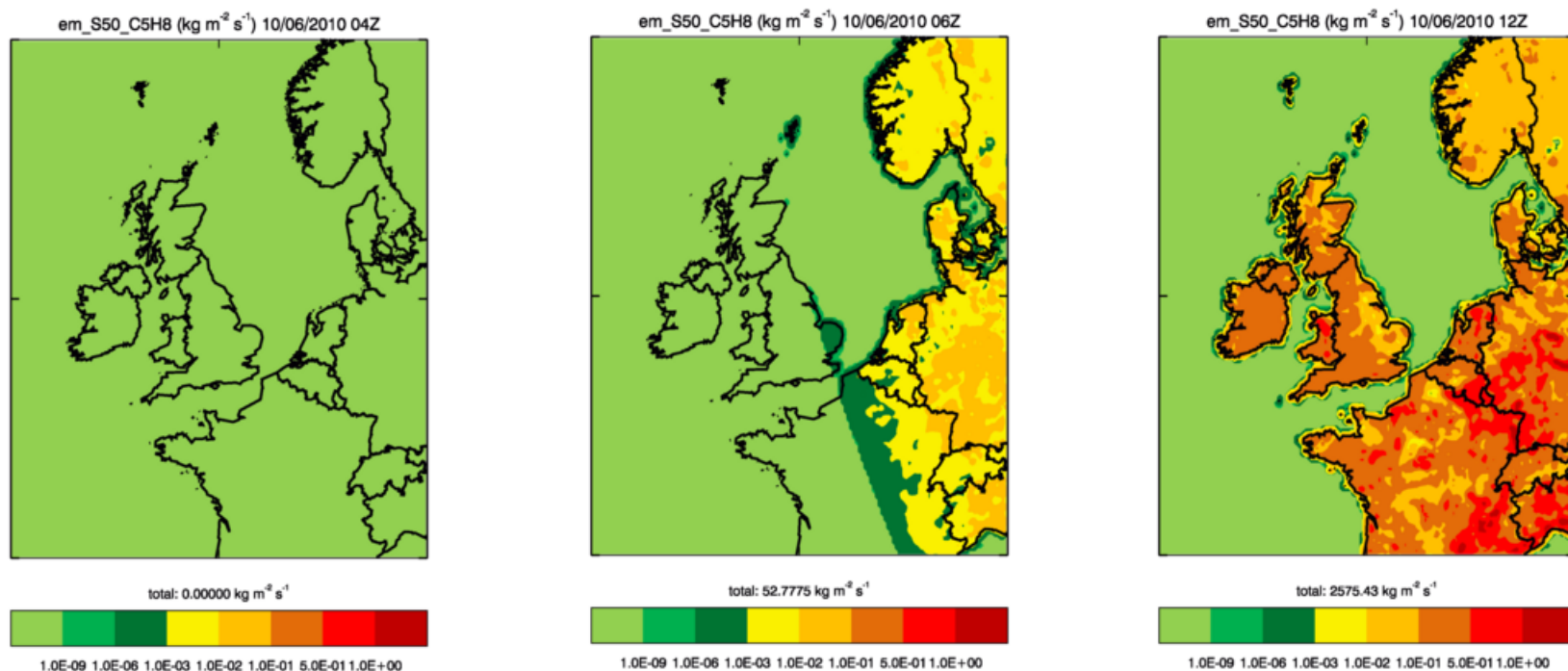
# New emission diagnostics (3)

Emission diagnostics for (non-interactive) isoprene emissions, with netCDF attributes:

tracer\_name = 'C5H8'

hourly\_scaling = 'diurnal\_isopems'

→ diurnal cycle applied via call to **ukca\_diurnal\_isop\_ems.F90**



NetCDF emission system will be extended for iBVOC emissions of  $\text{C}_5\text{H}_8$  and other species

# ukca\_new\_emiss\_ctl.F90 (1)

```
IF (I_first) THEN
```

```
! Read emission NetCDF files and look for the the emission fields
! in them to allocate all variables in the emissions structure.
```

```
CALL ukca_emiss_init (...)
```

```
! Get scaling factors indicating how to to spread emissions
! over different vertical levels. Done it only once here and
! stored in the emissions structure, then valid for all time steps.
```

```
DO I = 1, num_em_flds
```

```
CALL vertical_emiss_factors (... , vert_fact_3d)
```

```
emissions(I)%vert_scaling_3d (:,:,) = vert_fact_3d (:,:,)
```

```
END DO
```

```
END IF
```

```
! -----
! Check if it is time to update the emiss fields that
! we read from NetCDF files (depending on time step
! and update frequency). If needed then update the files.
```

```
CALL ukca_emiss_update (...)
```

```
! -----
! Deal with online emissions. Always updated at each time step
! For the moment only NOx from lightning and CH4 from wetlands
```

# ukca\_new\_emiss\_ctl.F90 (2)

```
! -----
! Update all fields in the emissions super array:
! * Do conversions so that emissions are given as 'kg(tracer) m-2 s-1'
! * Update isoprene emissions if they are diurnally varying
```

```
DO I = 1, num_em_flds
```

```
 IF (emissions(I)%I_update) THEN
```

```
 CALL base_emiss_factors
```

```
 update "emissions(I)%values"
```

```
 END IF
```

```
 For isoprene: CALL ukca_diurnal_isop_ems
 to apply diurnal cycle
```

```
END DO
```

```
! -----
! Inject emissions and do tracer mixing
CALL ukca_add_emiss (...)
```

```
! -----
! Call the emission diagnostics code if any of the diagnostics present
! in the routine GET_EMDIAG_STASH has been selected via stash.
```

```
CALL ukca_emiss_diags (...)
```

## 4. Final notes



# Final notes

- This talk is based on UM vn9.2. Some functionalities not available for older UM versions.
- I have not covered some topics in detail (aerosol emissions, BVOC emissions, ...).

## Useful reading material

- **Online UKCA tutorial**

[http://www.ukca.ac.uk/wiki/index.php/UKCA\\_Chemistry\\_and\\_Aerosol\\_Tutorial\\_5](http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorial_5)

- **UKCA - UMDP 84**

- Chapter 10: Emissions (ancillary system)
- Chapter 11: New NetCDF emission system

<https://code.metoffice.gov.uk/doc/um/vn9.2/umdp.html>

<https://code.metoffice.gov.uk/doc/um/vn10.0/umdp.html>

# Code in ukca\_setd1defs.F90

Complicated block of code further down in this routine

```

J = n_use_tracers
IF (n_chem_emissions+n_3d_emissions+n_mode_emissions > 0) THEN
 DO i=1,n_chem_emissions + n_3d_emissions
 UkcaDlCodes(J+i)%section = 0
 UkcaDlCodes(J+i)%item = n_emiss_first+i-1
 UkcaDlCodes(J+i)%len_dim1 = row_length
 UkcaDlCodes(J+i)%len_dim2 = rows
 IF (.NOT. (L_ukca_new_emiss)) THEN
 UkcaDlCodes(J+i)%required = .TRUE.
 END IF
 UkcaDlCodes(J+i)%prognostic = .TRUE.
 Special cases, emissions already available in UM
 IF (em_chem_spec(i)(1:7) == 'SO2_low') THEN
 UkcaDlCodes(J+i)%item = 58
 ELSEIF (em_chem_spec(i)(1:7) == 'SO2_nat') THEN
 UkcaDlCodes(J+i)%item = 121
 UkcaDlCodes(J+i)%len_dim3 = tr_levels
 ELSEIF (em_chem_spec(i)(1:8) == 'SO2_high') THEN
 UkcaDlCodes(J+i)%item = 126
 ELSEIF (em_chem_spec(i)(1:3) == 'NH3') THEN
 UkcaDlCodes(J+i)%item = 127
 ELSEIF (em_chem_spec(i) == 'BC_fossil ') THEN
 UkcaDlCodes(J+i)%item = 310
 ELSEIF (em_chem_spec(i) == 'BC_biofuel') THEN
 UkcaDlCodes(J+i)%item = 311
 ELSEIF (em_chem_spec(i) == 'OC_fossil ') THEN
 UkcaDlCodes(J+i)%item = 312
 ELSEIF (em_chem_spec(i) == 'OC_biofuel') THEN
 UkcaDlCodes(J+i)%item = 313
 ELSEIF (em_chem_spec(i) == 'Monoterp ') THEN
 UkcaDlCodes(J+i)%item = 314
 ELSEIF (em_chem_spec(i) == 'NVOC ') THEN
 UkcaDlCodes(J+i)%item = 315
 ELSEIF (em_chem_spec(i) == 'BC_biomass') THEN
 UkcaDlCodes(J+i)%item = 322
 UkcaDlCodes(J+i)%len_dim3 = tr_levels
 ELSEIF (em_chem_spec(i) == 'OC_biomass') THEN
 UkcaDlCodes(J+i)%item = 323
 UkcaDlCodes(J+i)%len_dim3 = tr_levels
 ELSEIF (em_chem_spec(i) == 'SO2_biomass') THEN
 UkcaDlCodes(J+i)%item = 324
 UkcaDlCodes(J+i)%len_dim3 = tr_levels
 ELSEIF (em_chem_spec(i)(1:3) == 'DMS') THEN
 UkcaDlCodes(J+i)%section = 17
 UkcaDlCodes(J+i)%item = 205
 UkcaDlCodes(J+i)%prognostic = .FALSE.
 IF (.NOT. L_sulpc_dms) UkcaDlCodes(J+i)%required = .FALSE.
 ! DMS internally calculated if CLASSIC is OFF
 ELSEIF (em_chem_spec(i)(1:7) == 'NO_airc') THEN
 UkcaDlCodes(J+i)%item = 340
 UkcaDlCodes(J+i)%len_dim3 = tr_levels
 ENDIF
 ENDDO
ENDIF

```

## Meaning

- STASH 301–309 in same order as **em\_chem\_spec**
- STASH items defined for some emissions (e.g. items 322, 323, 324 reserved for 3D emissions)

Main thing to remember:

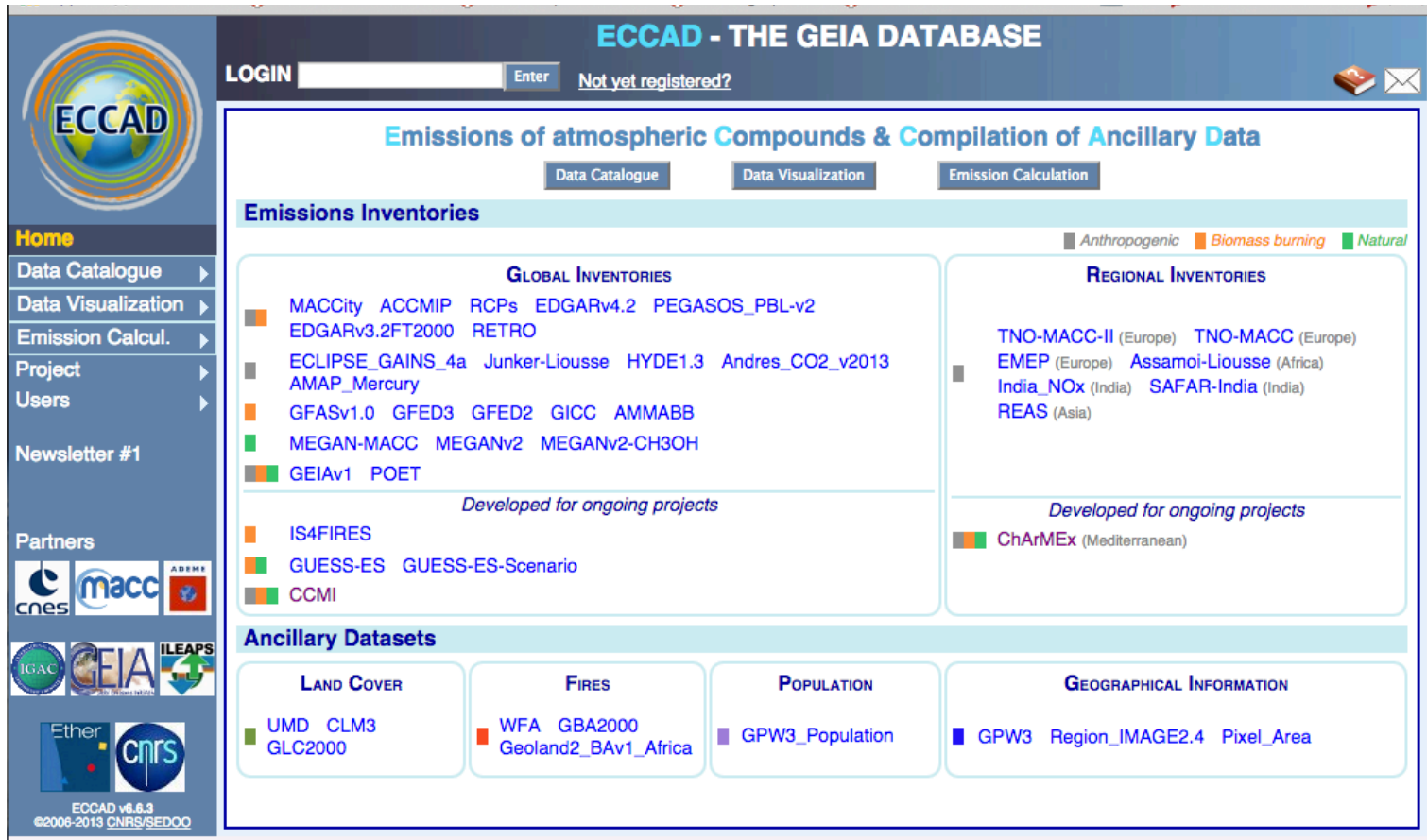
- single-level: items 301-320
- multi-level: items 321-340



## 5. Getting hold of some data

# So where/how do we get information on emissions?

- By far the best source of information is GEIA:

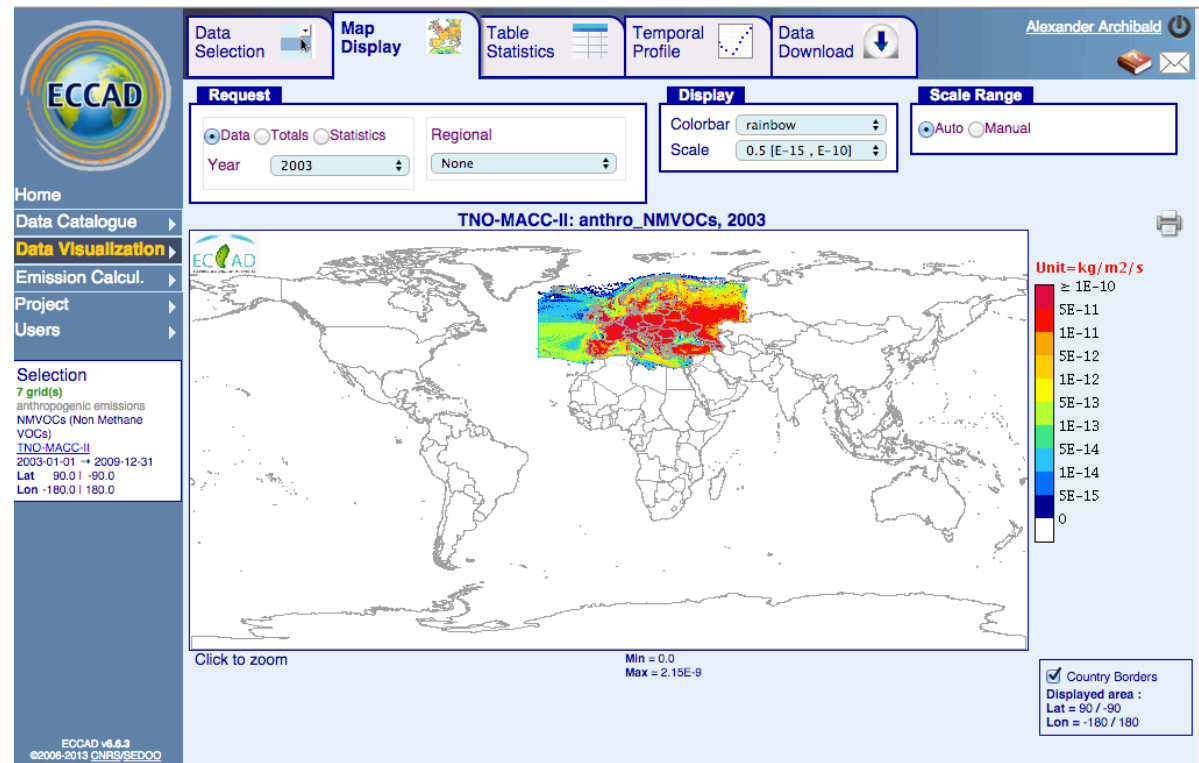


The screenshot shows the ECCAD - THE GEIA DATABASE website. The header includes a login field and a link for 'Not yet registered?'. The main title is 'Emissions of atmospheric Compounds & Compilation of Ancillary Data'. Below this are three tabs: 'Data Catalogue', 'Data Visualization', and 'Emission Calculation'. The 'Emissions Inventories' section is divided into 'GLOBAL INVENTORIES' and 'REGIONAL INVENTORIES'. The 'GLOBAL INVENTORIES' list includes MACCity, ACCMIP, RCPs, EDGARv4.2, PEGASOS\_PBL-v2, EDGARv3.2FT2000, RETRO, ECLIPSE\_GAINS\_4a, Junker-Lioussé, HYDE1.3, Andres\_CO2\_v2013, AMAP\_Mercury, GFASv1.0, GFED3, GFED2, GICC, AMMABB, MEGAN-MACC, MEGANv2, MEGANv2-CH3OH, GEIAv1, and POET. The 'REGIONAL INVENTORIES' list includes TNO-MACC-II (Europe), TNO-MACC (Europe), EMEP (Europe), Assamoi-Lioussé (Africa), India\_NOx (India), SAFAR-India (India), and REAS (Asia). Both sections note they are 'Developed for ongoing projects'. The 'Ancillary Datasets' section is divided into four categories: 'LAND COVER' (UMD, CLM3, GLC2000), 'FIRES' (WFA, GBA2000, Geoland2\_BAv1\_Africa), 'POPULATION' (GPW3\_Population), and 'GEOGRAPHICAL INFORMATION' (GPW3, Region\_IMAGE2.4, Pixel\_Area). The footer includes the ECCAD v6.6.3 logo and copyright information for 2006-2013 CNRS/SED00.

# So where/how do we get information on emissions?

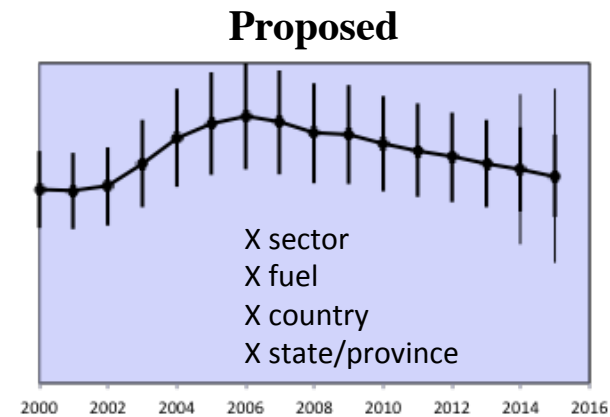
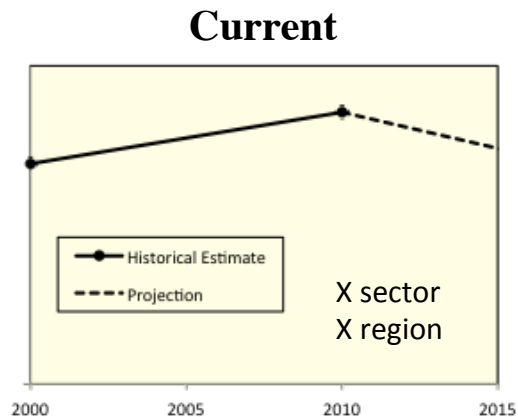
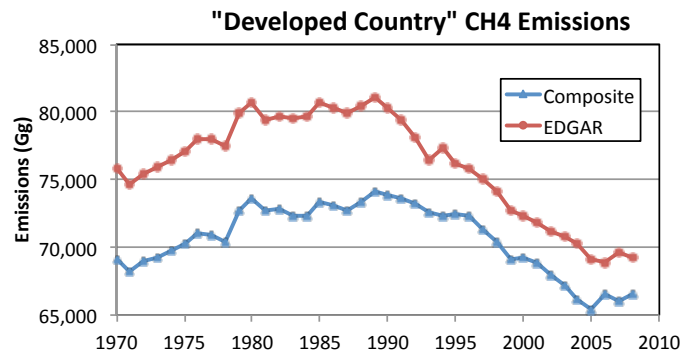
- By far the best source of information is GEIA:

Easily select data for different compounds, from different datasets for different time periods and plot, download, visualize etc.



# What about the future of emissions?

- Uncertainty quantification will be more important.



- Community Emissions Data System (CEDS) will be key for providing info for CMIP6 – you can get involved!