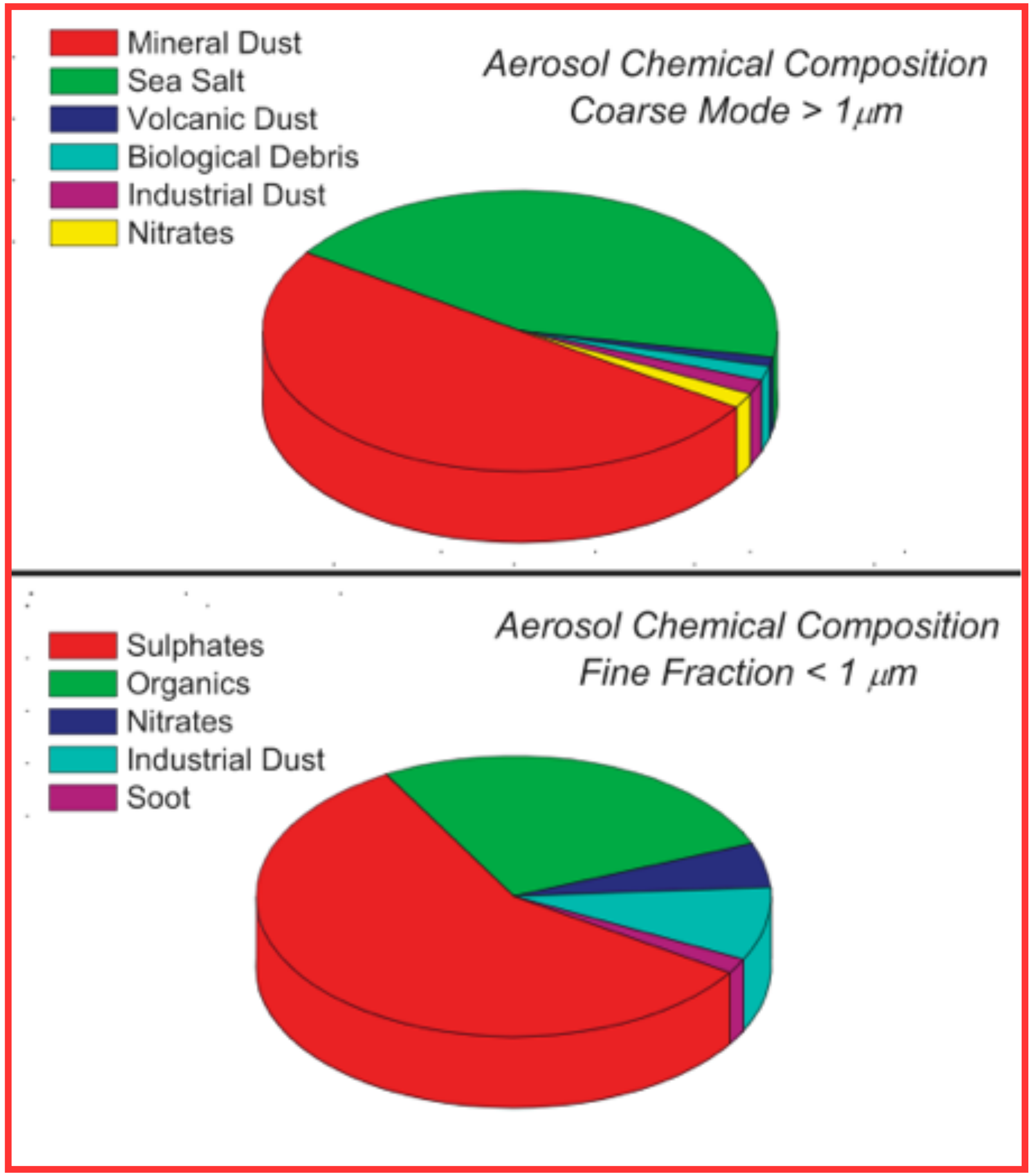
Heterogeneous chemistry - how to get from the literature to UKCA model treatment

Paul Griffiths

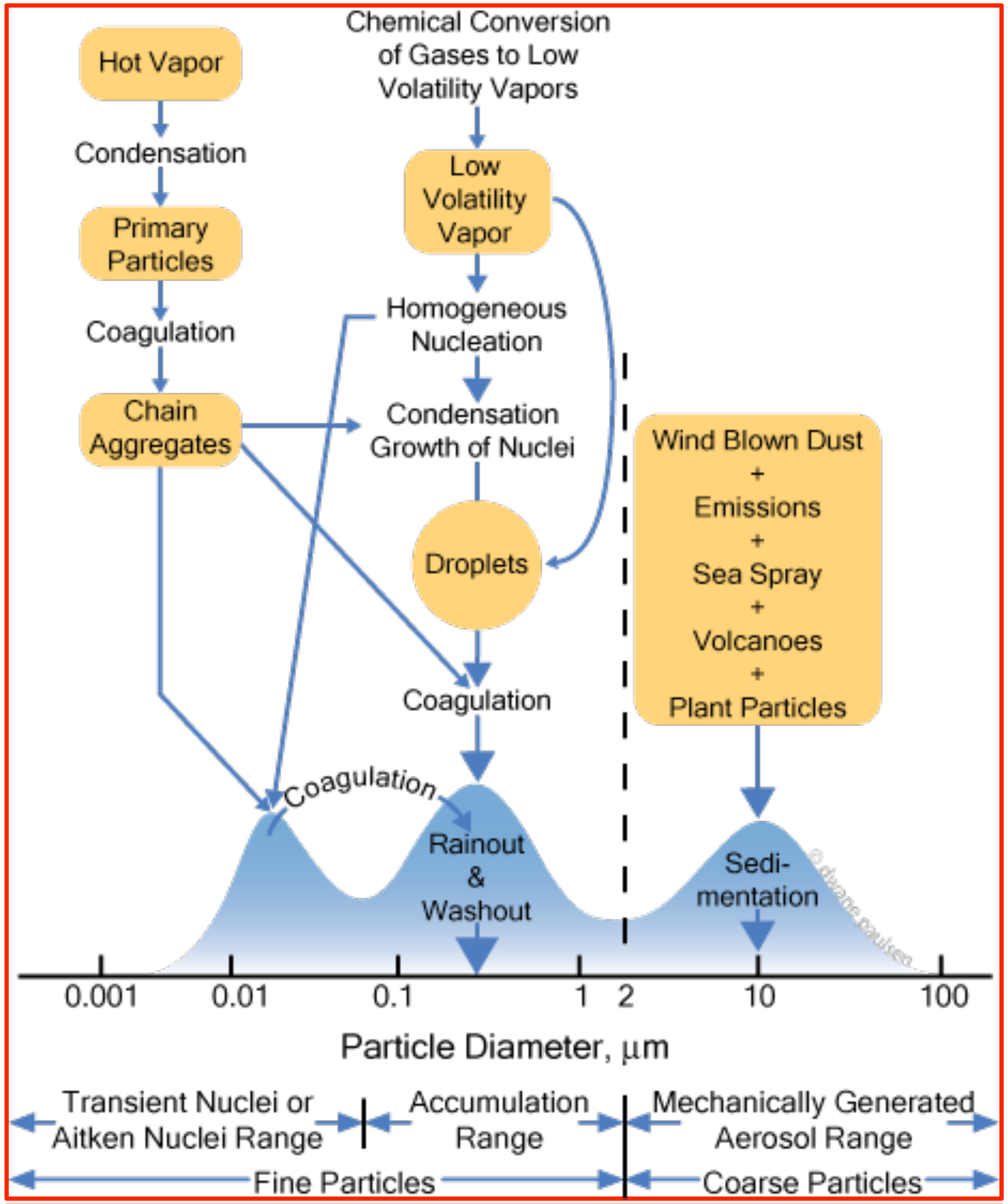
Chemical composition of tropospheric aerosol

- The particulate matter in the atmosphere is very diverse.
- Wide variety of primary sources - dust and sea salt are the strongest by mass.
- Chemical transformation of pollutants occurs within the atmosphere, and can lead to aerosol formation.
- $NO_x \rightarrow HNO_3 \rightarrow$ aerosol nitrate
- $SO_2 \rightarrow H_2SO_4 \rightarrow$ aerosol sulfate
- VOC → organics (carbon-rich)

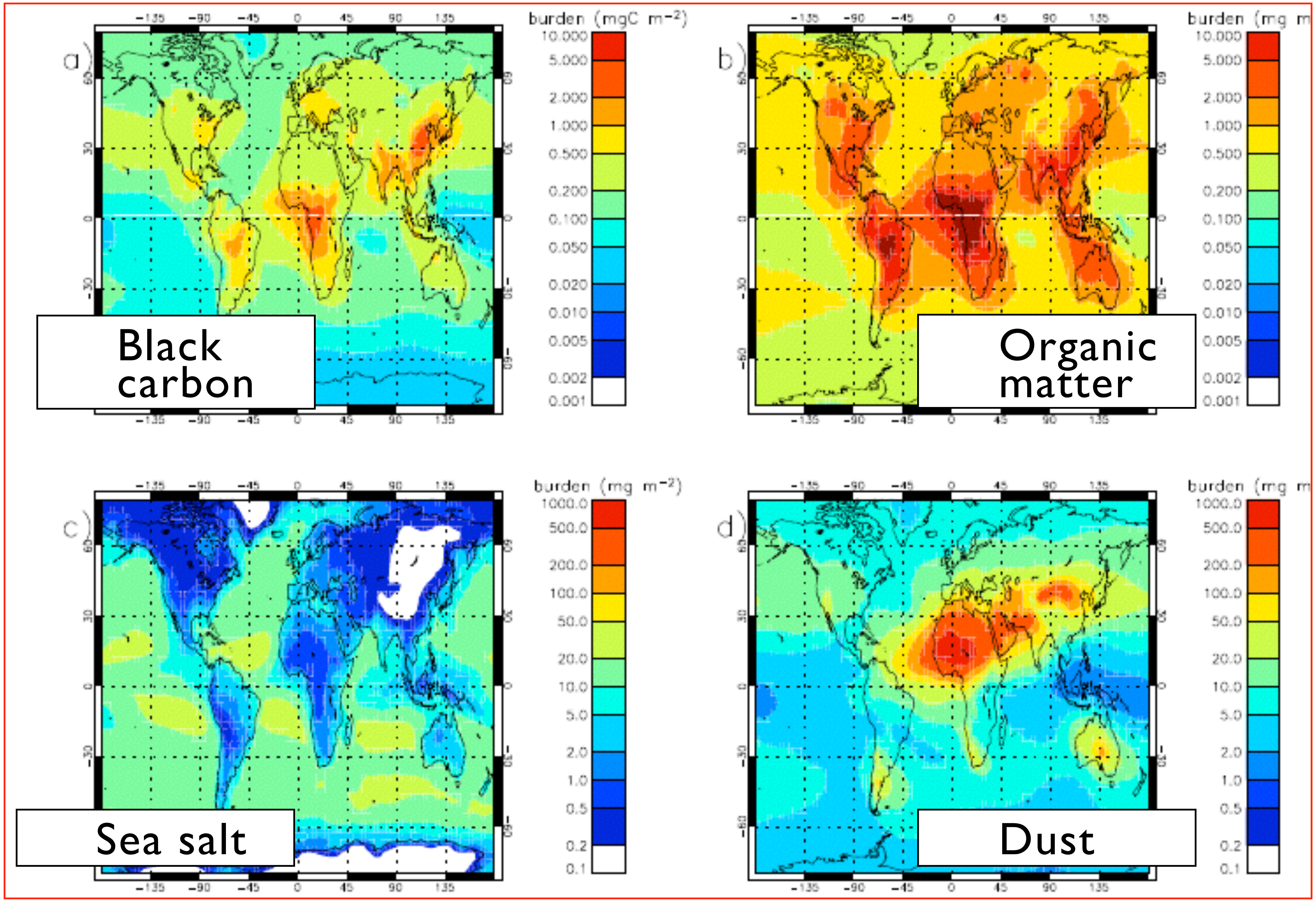


Size range spanned by atmospheric aerosol

- Sizes range from few nm up to hundreds of microns.
- The lifetime of small and large particles is short, but between 100nm and 2 μm, the lifetime can be up to a couple of weeks.
- Dust can be transported across oceans, volcanic aerosol can survive in the stratosphere for years.
- The integrated impact can be large.

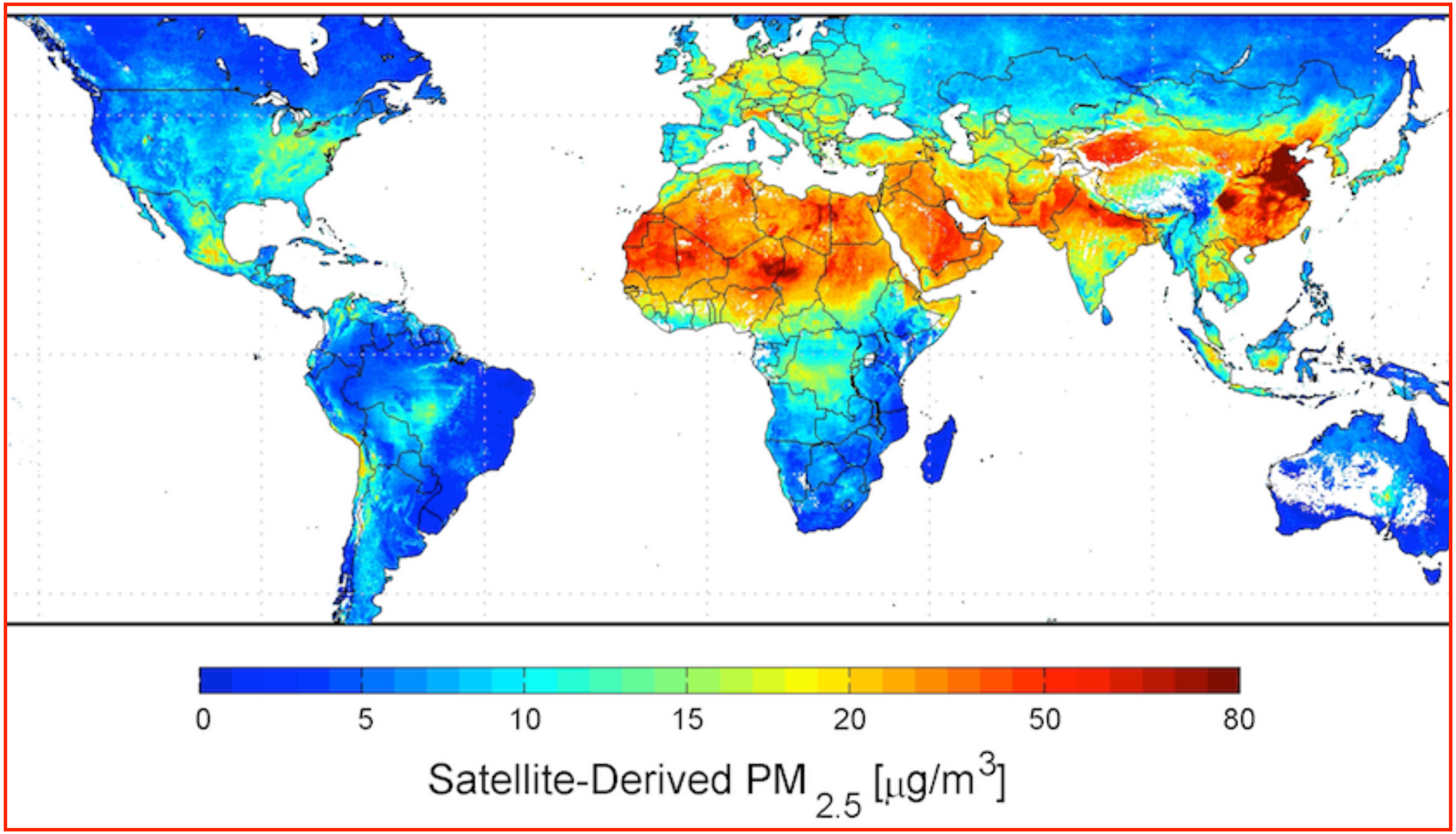


Current work - aerosol chemistry on a global scale



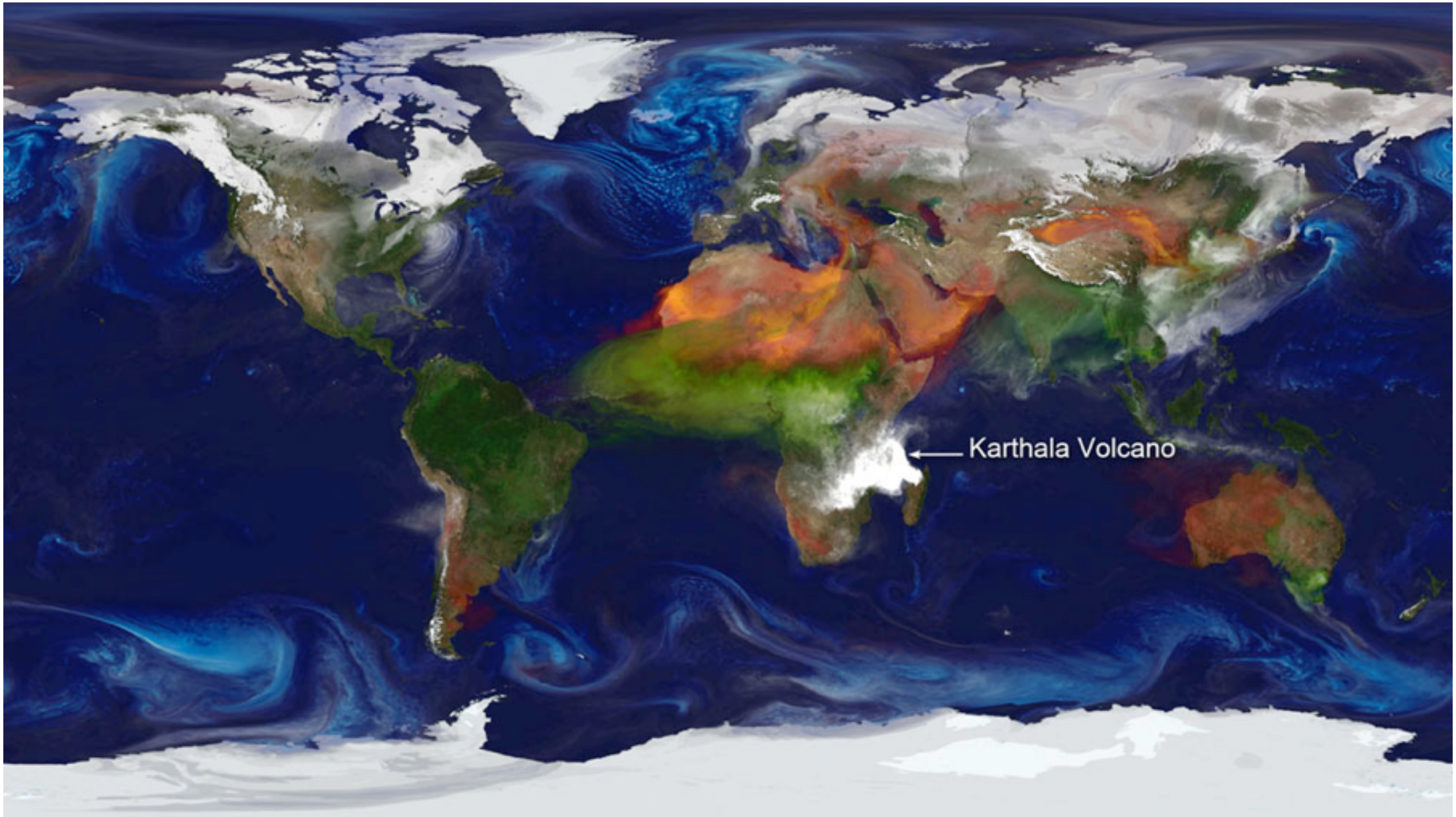
This aerosol model runs within the UK Met Office ‘Unified’ Climate Model.

Global near-surface distribution of aerosol



Total (mass) particulate matter is shown (NB no measurements over oceans)

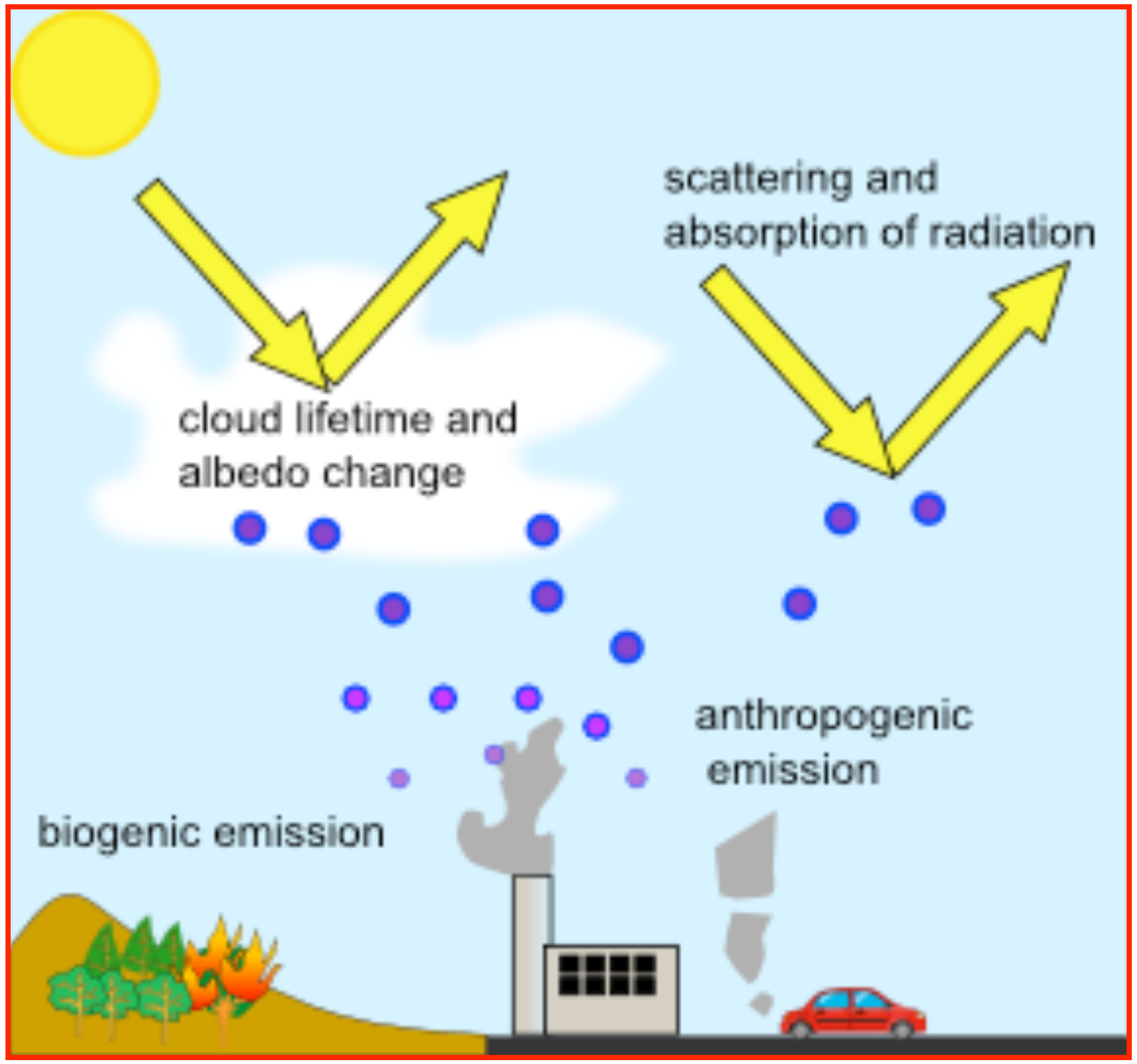
Global distribution of aerosol



White: sulfate blue: sea salt red: dust green: soot

Aerosols and the chemistry of the troposphere

- Anthropogenic pollutants are degraded by the action of sunlight in the atmosphere to form aerosol.
- Absorption and scattering by the aerosol then change the amount of incident sunlight - local climate different in the neighbourhood of the emissions.
- Also changes the rate of degradation - a feedback loop is established.


scattering and absorption of radiation
cloud lifetime and albedo change
anthropogenic emission
biogenic emission

Coupling between aerosol and oxidation

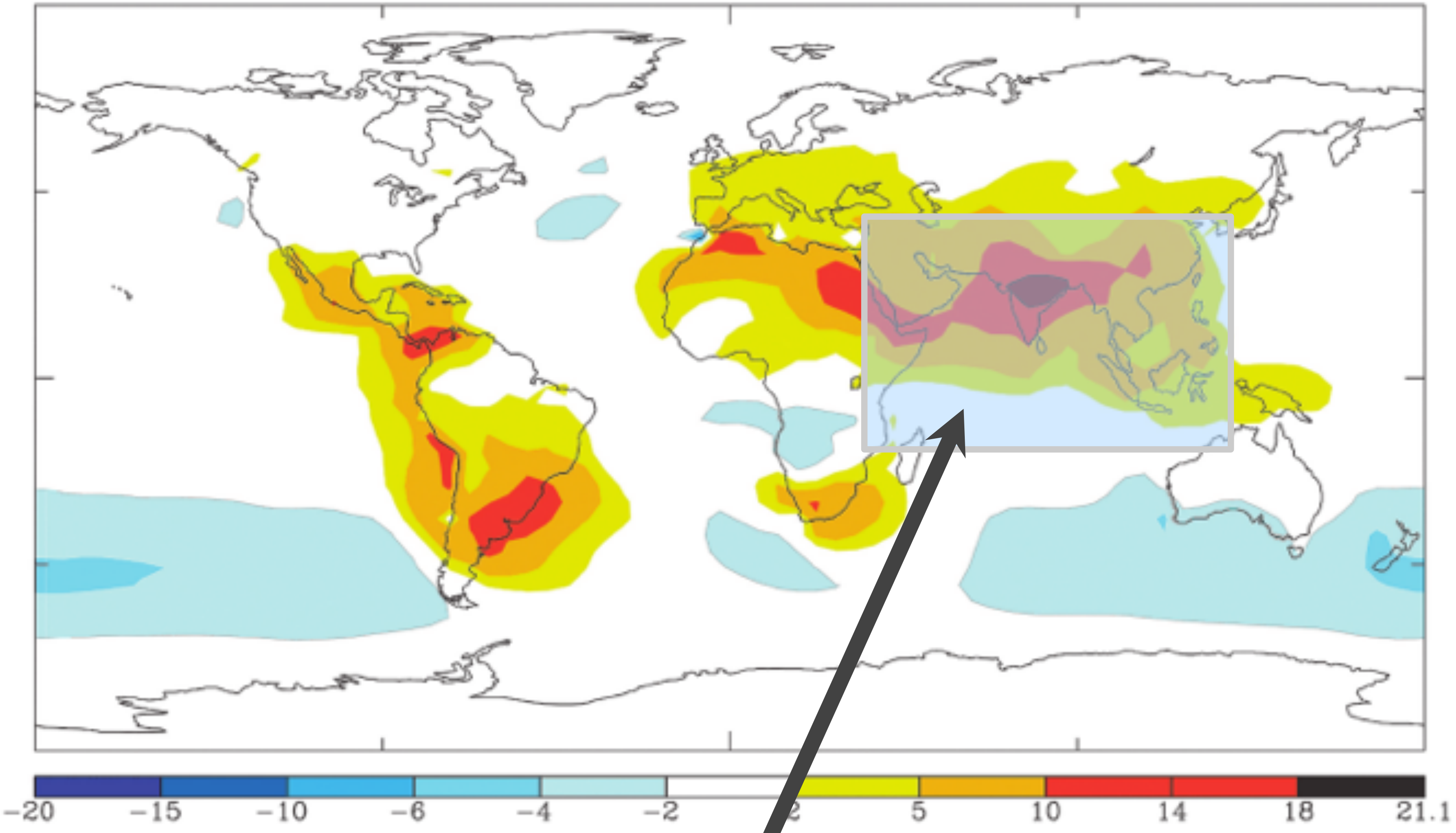


Fig. 1. Percentage difference in surface sulfate mixing ratio between future control simulation (2030C) and sensitivity simulation with future SO_2 emissions but present-day O_3 precursor emissions ($2030SO_4$) [i.e., 100 × (2030C − $2030SO_4$)/2030C].

Regional variations in impact: higher future NOx emissions lead to large increase in [OH], more OH+SO_2, more sulfate. More aerosol means a decrease in solar radiation, moderating climate change.

Aerosols and their effect on ozone

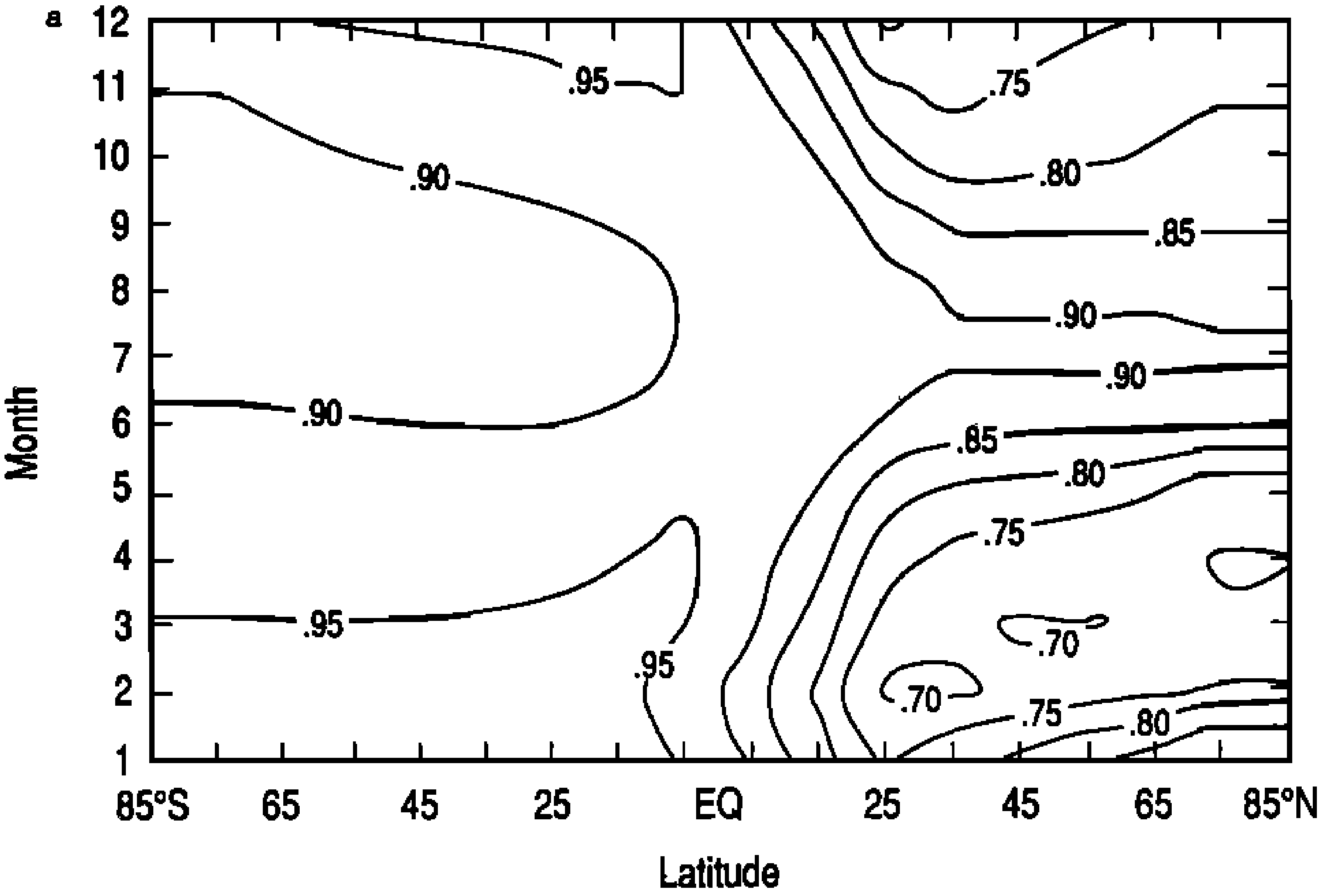


Fig. 9 from Dentener and Crutzen, 1993. The zonal mean monthly average ratio of O3 concentrations with aerosol reactions to those without for all months at 1000 hPa

Particulate matter in the atmosphere

- Serve as the germ or nucleus for cloud formation.
- Scatter incoming solar radiation (reduce heating at the surface).
- Absorb outgoing radiation (increase strength of greenhouse effect).
- Remove (and transform) trace gases and pollutants.
- Health effects

Chemical effects of particles in the atmosphere

- Partitioning to the aqueous phase - solubility controlled
- Adsorption onto solid surfaces
- Reactive uptake of trace gases
- Release of trace gases following reactive uptake

- **Aerosols may enhance concentration and so open up new pathways of reactivity.**

Prognostic equation for reaction of aerosol with gas X

$$\frac{d[X]}{dt} = -k_{het}[X]$$

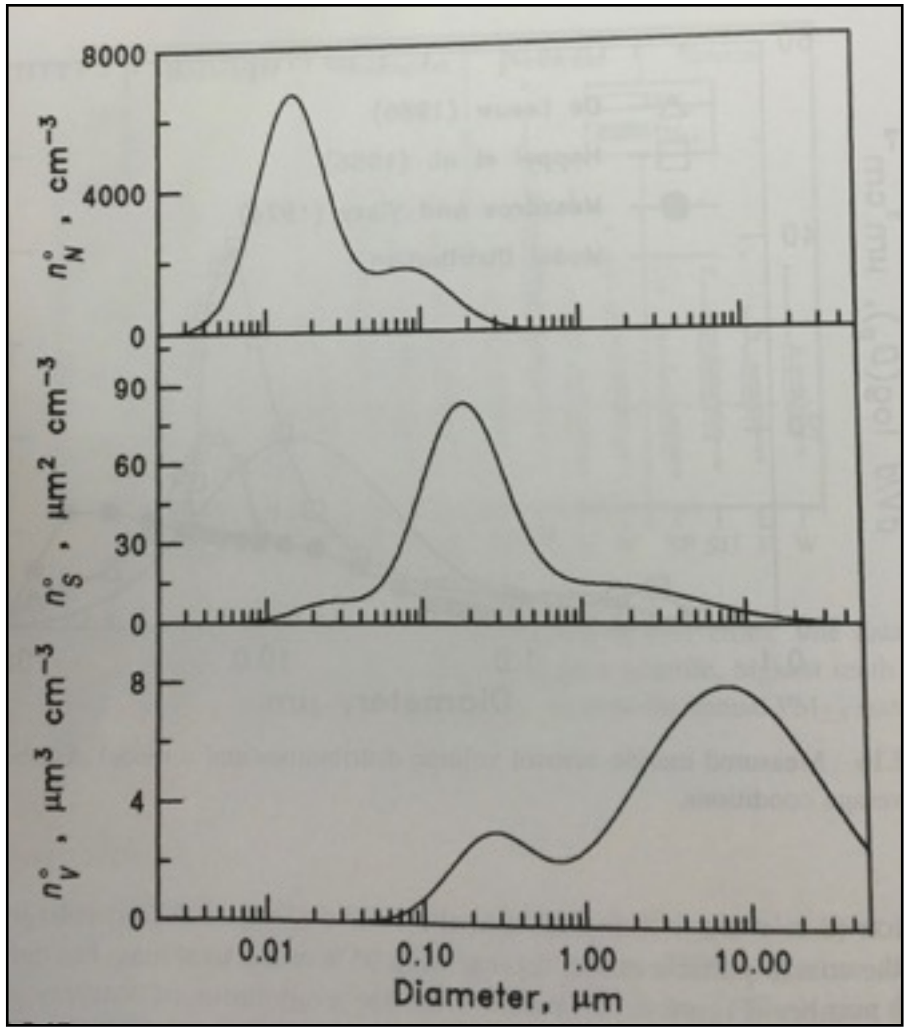
$$k_{het} = \gamma S_a c/4$$

$$c = \sqrt{\frac{8kT}{\pi m}}$$

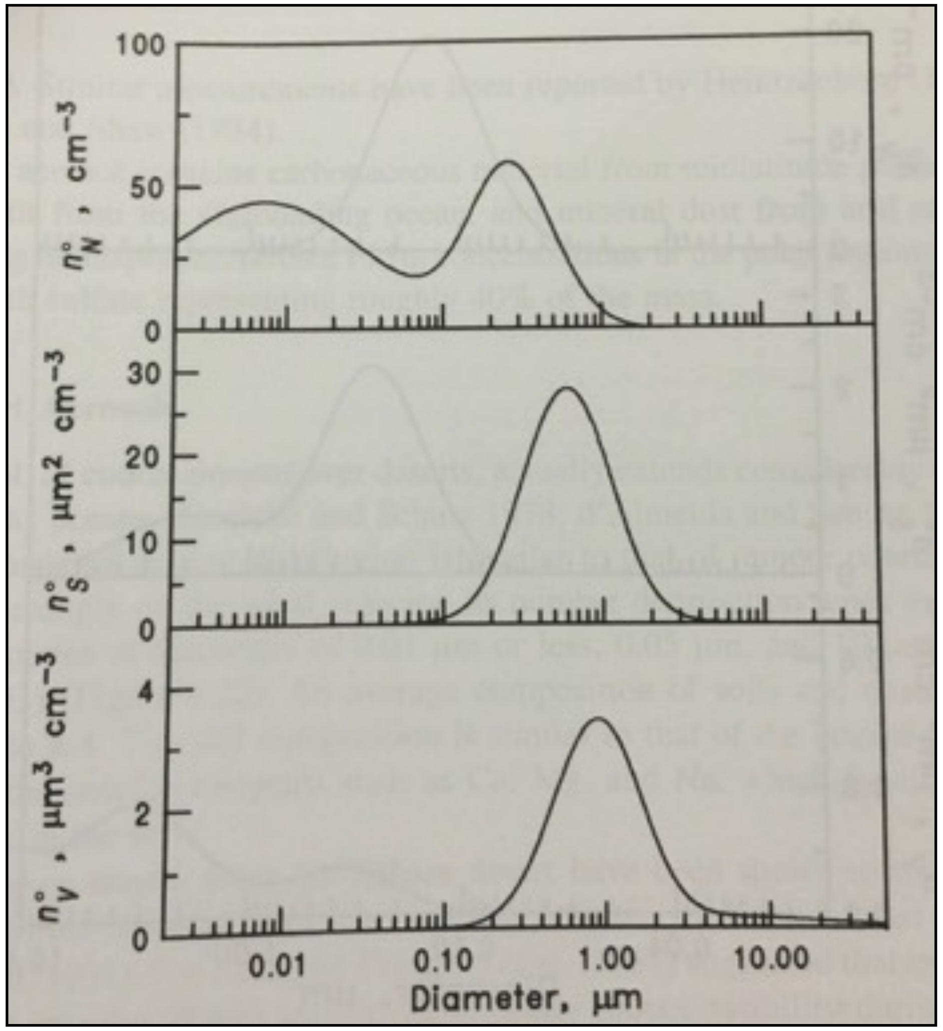
- Parameterise reaction/uptake via uptake coefficient, γ, (0 <γ≤1)
- Represents the probability that the gas phase species X is lost per collision with the aerosol surface.
- Make sure units of surface area density (m^2 m^{-3}) and c (mean molecular speed of gas phase X, ms^{-1}) match.
- E.g. typical units of SA density - μm^2 m^{-3}

Representative aerosol number, SA and mass distributions

Rural continental

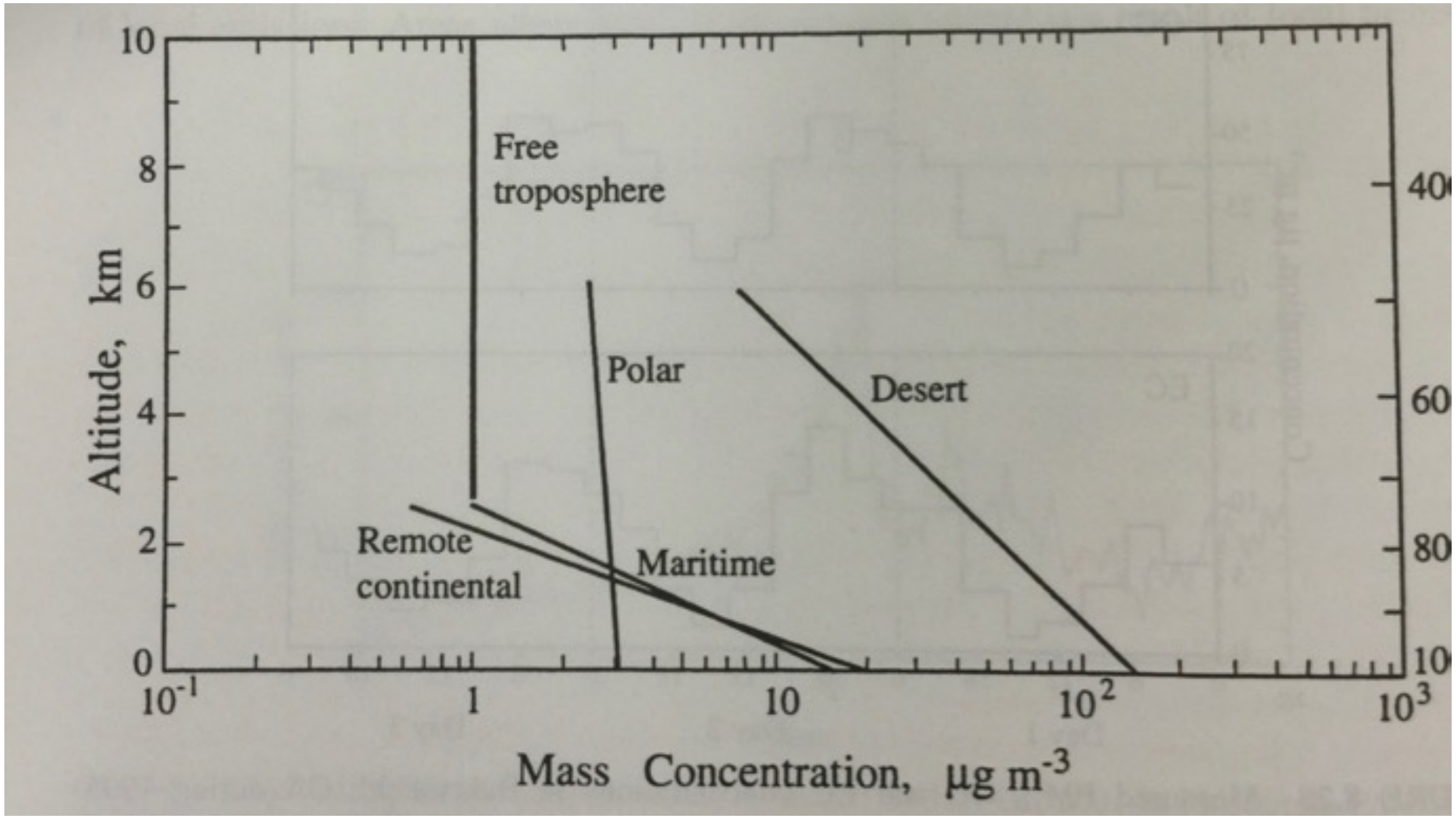


Free troposphere



Maximum of aerosol surface area density is in the accumulation mode

Tropospheric profile of aerosol mass



Figures taken from Seinfeld and Pandis, p. 375

Kinetics of removal of trace gases

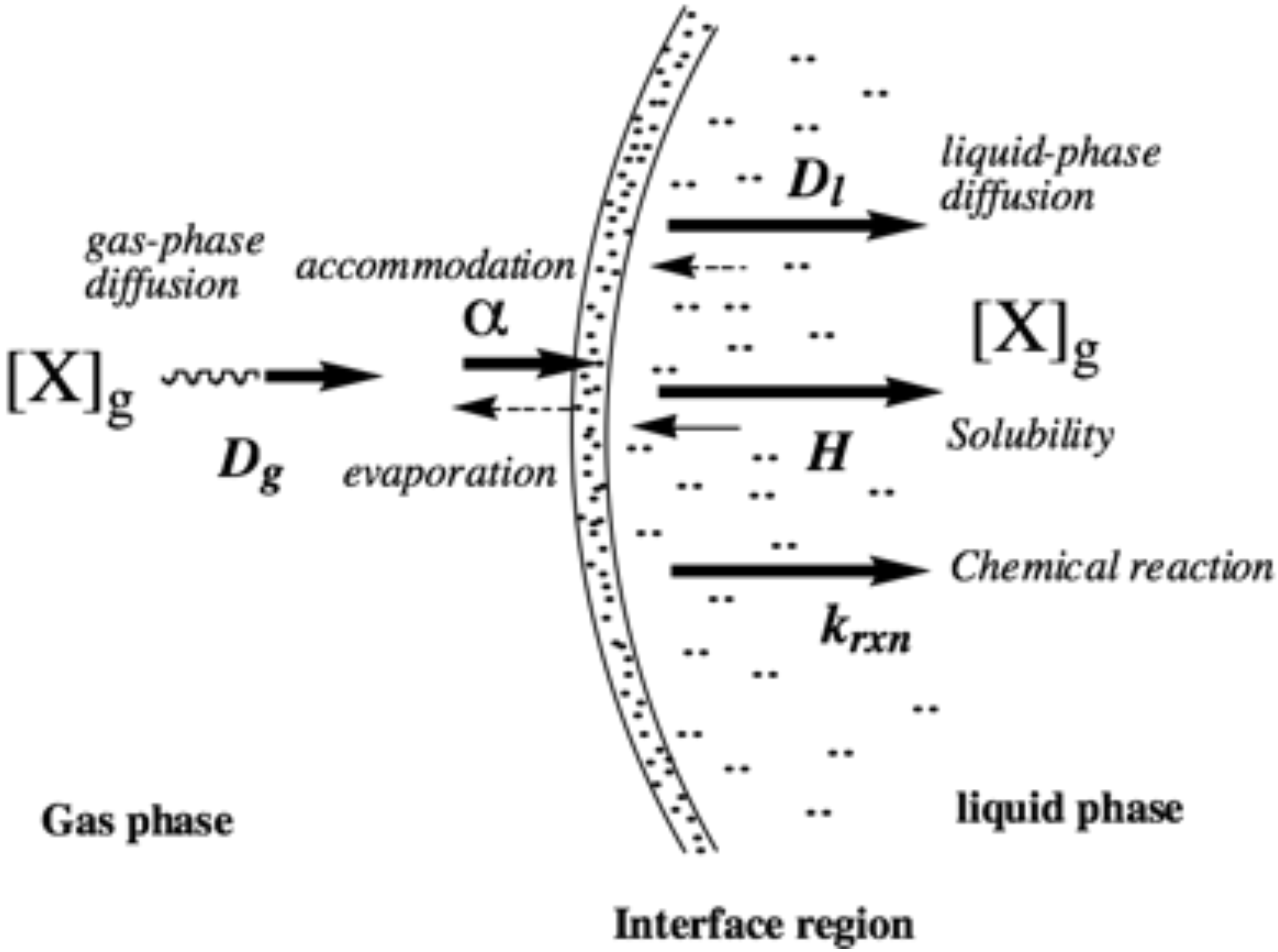
$$\frac{d[X]}{dt} = -k_{het}[X]$$

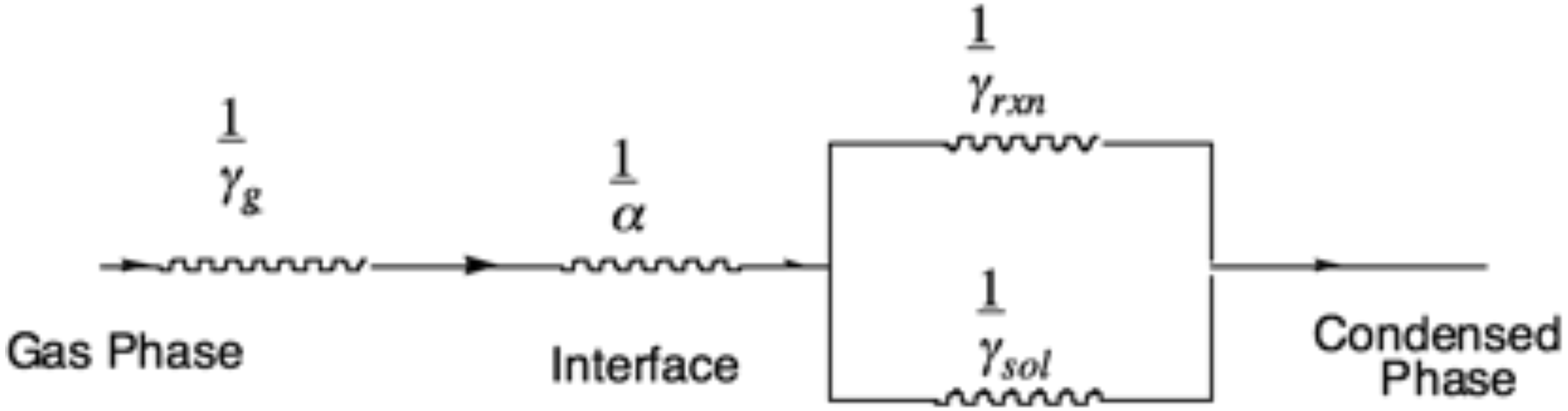
$$k_{het} = \gamma S_a c/4$$

$$c = \sqrt{\frac{8kT}{\pi m}}$$

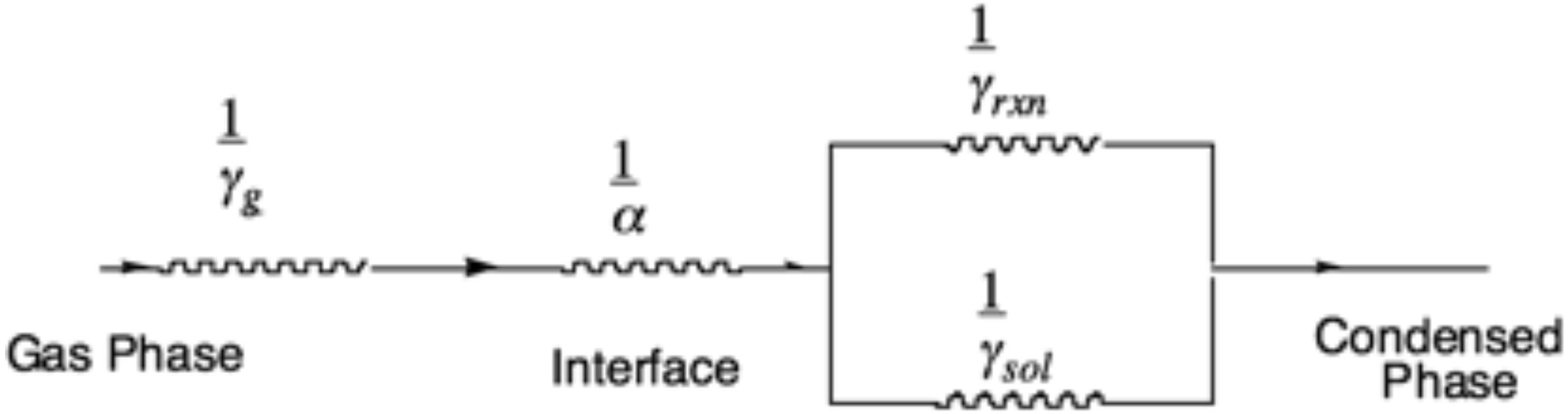
- The uptake coefficient, γ, contains the physical chemistry that you need to think about.
- This is generally what the experimentalists report.
- Tempting just to take the number from the paper and put it into the model, but there's usually scope to add a bit of value.
- The resistor framework provides the best way currently to do this, although there are other modelling frameworks available - e.g. Shiraiwa (PRA framework).

Trace gas uptake by liquids – a series of coupled steps





Trace gas uptake as a series of coupled steps



$$\frac{1}{\gamma} = \frac{1}{\gamma_g} + \frac{1}{\alpha} + \frac{1}{\gamma_{sol} + \gamma_{rxn}}$$

- The value of gamma can be set by any one of these processes.
- smallest γ ⇒ 1/γ dominates sum - pays to identify the slowest step!
- If no reaction(γ_{rxn} = 0) , then expect the aerosol to become saturated in the trace gas of interest. Over time, the net uptake will reduce to zero.

Trace gas uptake as a series of coupled steps

$$\frac{1}{\gamma} = \frac{1}{\gamma_g} + \frac{1}{\alpha} + \frac{1}{\gamma_{sol} + \gamma_{rxn}}$$

- γ_g represents the modification to the collision frequency (and uptake rate) if the particle is large or diffusion is slow to the particle surface. Depends of particle radius,r, and D_g, gas phase diffusion coefficient

$$\frac{1}{\gamma_g} = \frac{cr}{4D_g}$$

- α is the probability that having collided with the particle surface, the molecule X enters or sticks to the aerosol (NB one way). This can be the smallest probability for solid particles for which this probability is low and so this may control gamma.

- Conversely, when reaction is very fast (γ_{rxn} is large) the accommodation step may again be the slowest step and control γ (so-called 'accommodation-controlled uptake')

Trace gas uptake as a series of coupled steps

$$\frac{1}{\gamma} = \frac{1}{\gamma_g} + \frac{1}{\alpha} + \frac{1}{\gamma_{sol} + \gamma_{rxn}}$$

- γ_{sol} represents the uptake due to the equilibration of the condensed and gas phases - establishing a Henry's law equilibrium at which uptake and evaporation are equal - net uptake at this point is zero.

- Key points: that solubility is composition dependent - e.g. solubility of SO_2 increases in acidic aerosol; aerosol has probably equilibrated in a UKCA timestep.

$$\frac{1}{\gamma_{sol}} = \frac{\sqrt{\pi} c}{4HRT} \left(\frac{t}{D_l}\right)^{1/2}$$

- Fraction of gas X in aqueous phase can be calculated as

$$\frac{[X_{aq}]}{[X]} = 10^{-6} HRTL$$

- *L* is liquid water content in g cm^{-3}. Can be significant for soluble gases in clouds!

Trace gas uptake as a series of coupled steps

$$\frac{1}{\gamma} = \frac{1}{\gamma_g} + \frac{1}{\alpha} + \frac{1}{\gamma_{sol} + \gamma_{rxn}}$$

- γ_{rxn} represents the uptake due to reaction within the aerosol volume.
- Key points: is composition dependent - N_2O_5 onto sulphate (fast) vs nitrate (slow); HO_2 onto organic aerosol; O_3 uptake by SOA. Most general form (rarely used)

$$\frac{1}{\gamma_{rxn}} = \frac{c}{4HRT\sqrt{D_l k_{rxn}}}\left(\coth q - \frac{1}{q}\right)$$

- krxn is the first-order (s^{-1}) rate constant for loss of X in solution; q is the ratio of particle radius to reacto-diffusive length, l.

$$q = \frac{r}{l} \qquad\qquad l = \sqrt{\frac{D_l}{k_{rxn}}}$$

- In general, the above equation can be simplified.

Uptake onto solid surfaces

$$\frac{1}{\gamma} = \frac{1}{\alpha} + \frac{1}{\gamma_s}$$

- Here reaction is between a surface site or surface bound molecule
- Two limiting cases - Langmuir-Hinshelwood (most common by far) or Eley-Rideal
- For Langmuir-Hinshelwood, the reaction is between gas-phase X and a surface-bound molecule Y (here Y_s). Need info on concentration of Y and on partitioning at equilibrium between X and surface-bound X, given by K and N.

$$\frac{1}{\gamma} = \frac{1}{\alpha} + \frac{c(1+K[X])}{4k_{rxn}[Y_s]KN}$$

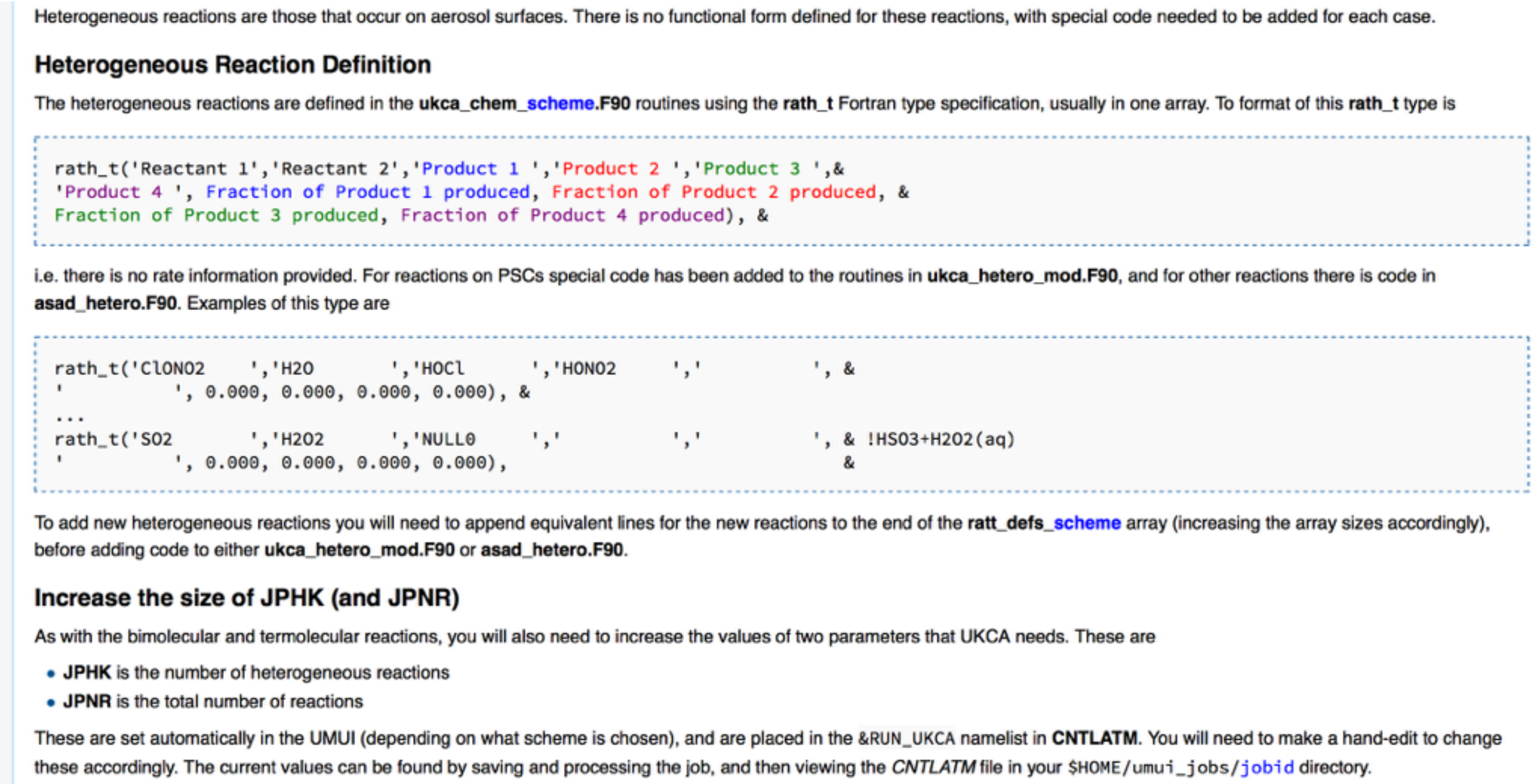
- For more information see the IUPAC website, http://iupac.pole-ether.fr/htdocs/supp_info/NewHetIntroOct2009.pdf

Back of the envelope

- Need a representative surface area density (or range)
- And a gamma (or range of gammas)
- Then calculate k_{het} for the trace gas of interest
- Compare with other relevant rates e.g. reaction with O_3, dry deposition, wet removal, photolysis, …
- Can also put into a box model and do an aerosol on/off experiment
  - E.g. KPP - http://people.cs.vt.edu/~asandu/Software/Kpp/
  - MECCA - http://www.rolf-sander.net/messy/mecca/mecca_box.html
  - DSMACC - https://github.com/barronh/DSMACC

Putting heterogeneous chemistry into UKCA

- Follow the tutorial for putting in a new chemical reaction.
- http://www.ukca.ac.uk/wiki/index.php/UKCA_Chemistry_and_Aerosol_Tutorial_6#Heterogeneous_Reactions



Heterogeneous reactions are those that occur on aerosol surfaces. There is no functional form defined for these reactions, with special code needed to be added for each case.

Heterogeneous Reaction Definition

The heterogeneous reactions are defined in the **ukca_chem_scheme.F90** routines using the **rath_t** Fortran type specification, usually in one array. To format of this **rath_t** type is

```
rath_t('Reactant 1','Reactant 2','Product 1 ','Product 2 ','Product 3 ',&
'Product 4 ', Fraction of Product 1 produced, Fraction of Product 2 produced, &
Fraction of Product 3 produced, Fraction of Product 4 produced), &
```

i.e. there is no rate information provided. For reactions on PSCs special code has been added to the routines in **ukca_hetero_mod.F90**, and for other reactions there is code in **asad_hetero.F90**. Examples of this type are

```
rath_t('ClONO2    ','H2O       ','HOCl      ','HONO2     ','          ', &
'          ', 0.000, 0.000, 0.000, 0.000), &
...
rath_t('SO2       ','H2O2      ','NULL0     ','          ','          ', & !HSO3+H2O2(aq)
'          ', 0.000, 0.000, 0.000, 0.000),                                &
```

To add new heterogeneous reactions you will need to append equivalent lines for the new reactions to the end of the **ratt_defs_scheme** array (increasing the array sizes accordingly), before adding code to either **ukca_hetero_mod.F90** or **asad_hetero.F90**.

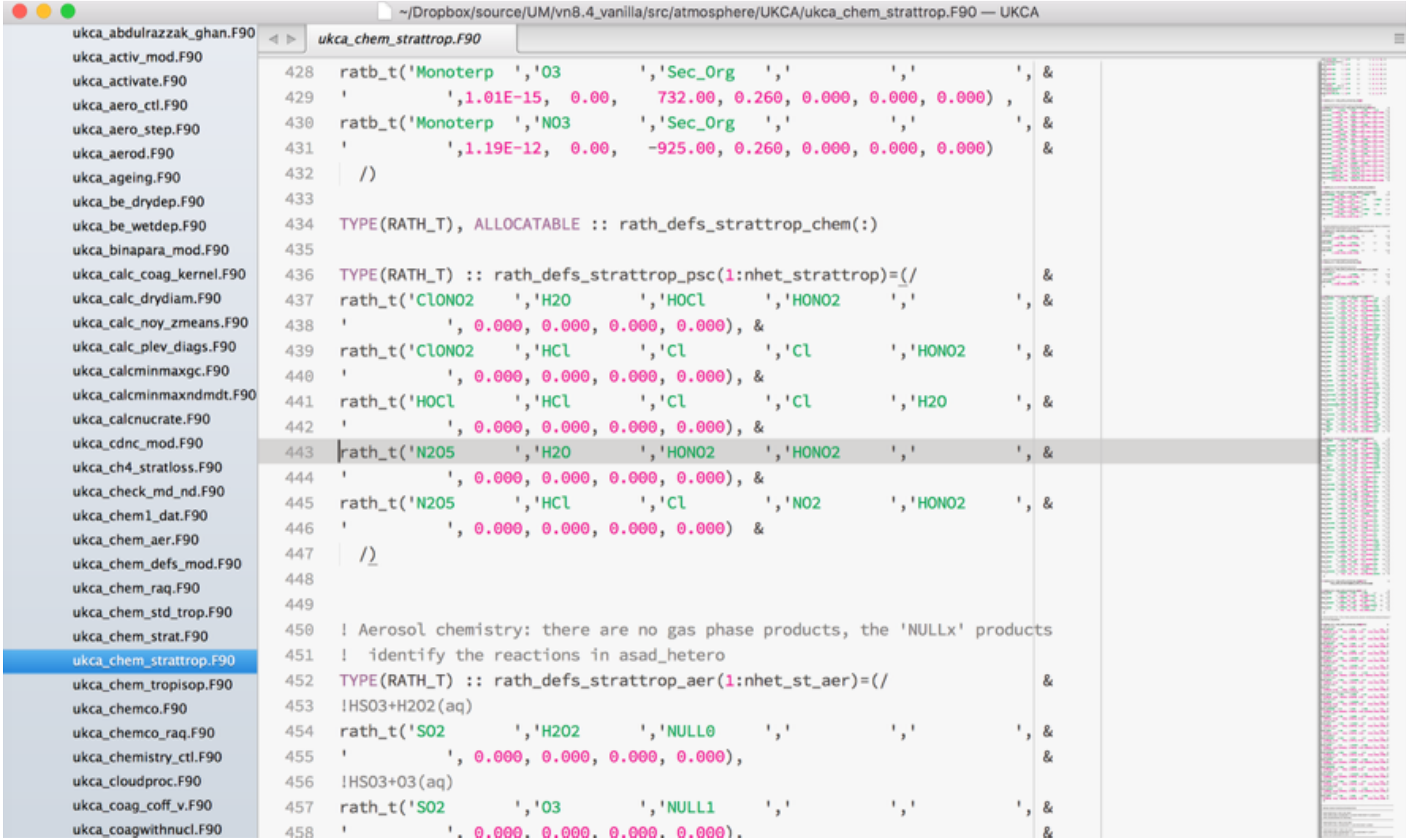
Increase the size of JPHK (and JPNR)

As with the bimolecular and termolecular reactions, you will also need to increase the values of two parameters that UKCA needs. These are

- **JPHK** is the number of heterogeneous reactions
- **JPNR** is the total number of reactions

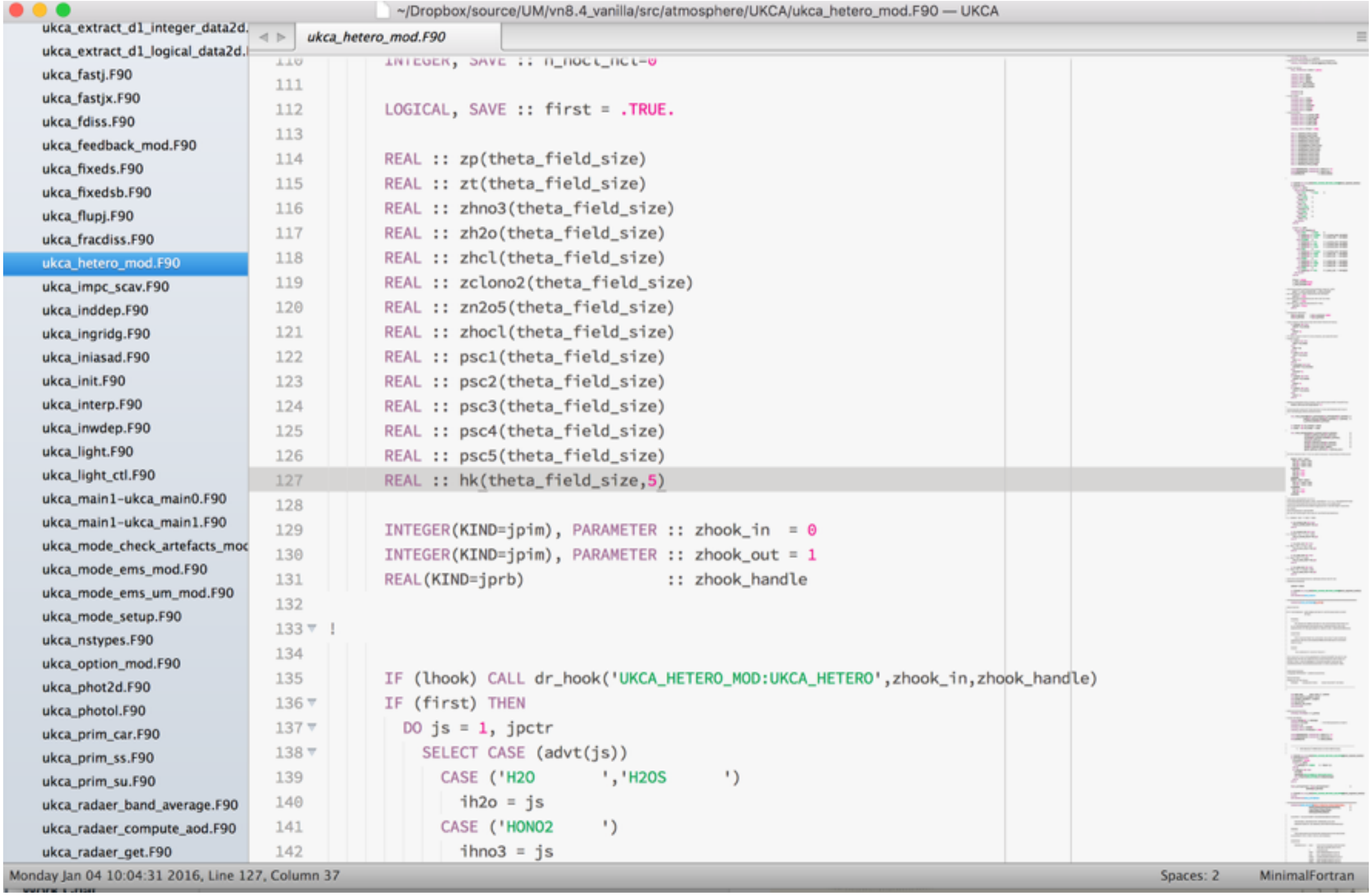
These are set automatically in the UMUI (depending on what scheme is chosen), and are placed in the `&RUN_UKCA` namelist in **CNTLATM**. You will need to make a hand-edit to change these accordingly. The current values can be found by saving and processing the job, and then viewing the *CNTLATM* file in your `$HOME/umui_jobs/jobid` directory.

UKCA/ukca_chem_strattrop.F90



```fortran
ratb_t('Monoterp  ','O3        ','Sec_Org   ','          ','          ', &
'          ',1.01E-15,  0.00,    732.00, 0.260, 0.000, 0.000, 0.000) ,   &
ratb_t('Monoterp  ','NO3       ','Sec_Org   ','          ','          ', &
'          ',1.19E-12,  0.00,   -925.00, 0.260, 0.000, 0.000, 0.000)     &
  /)

TYPE(RATH_T), ALLOCATABLE :: rath_defs_strattrop_chem(:)

TYPE(RATH_T) :: rath_defs_strattrop_psc(1:nhet_strattrop)=(/            &
rath_t('ClONO2    ','H2O       ','HOCl      ','HONO2     ','          ', &
'          ', 0.000, 0.000, 0.000, 0.000), &
rath_t('ClONO2    ','HCl       ','Cl        ','Cl        ','HONO2     ', &
'          ', 0.000, 0.000, 0.000, 0.000), &
rath_t('HOCl      ','HCl       ','Cl        ','Cl        ','H2O       ', &
'          ', 0.000, 0.000, 0.000, 0.000), &
rath_t('N2O5      ','H2O       ','HONO2     ','HONO2     ','          ', &
'          ', 0.000, 0.000, 0.000, 0.000), &
rath_t('N2O5      ','HCl       ','Cl        ','NO2       ','HONO2     ', &
'          ', 0.000, 0.000, 0.000, 0.000)  &
  /)


! Aerosol chemistry: there are no gas phase products, the 'NULLx' products
!  identify the reactions in asad_hetero
TYPE(RATH_T) :: rath_defs_strattrop_aer(1:nhet_st_aer)=(/                &
!HSO3+H2O2(aq)
rath_t('SO2       ','H2O2      ','NULL0     ','          ','          ', &
'          ', 0.000, 0.000, 0.000, 0.000),                               &
!HSO3+O3(aq)
rath_t('SO2       ','O3        ','NULL1     ','          ','          ', &
'          ', 0.000, 0.000, 0.000, 0.000),                               &
```

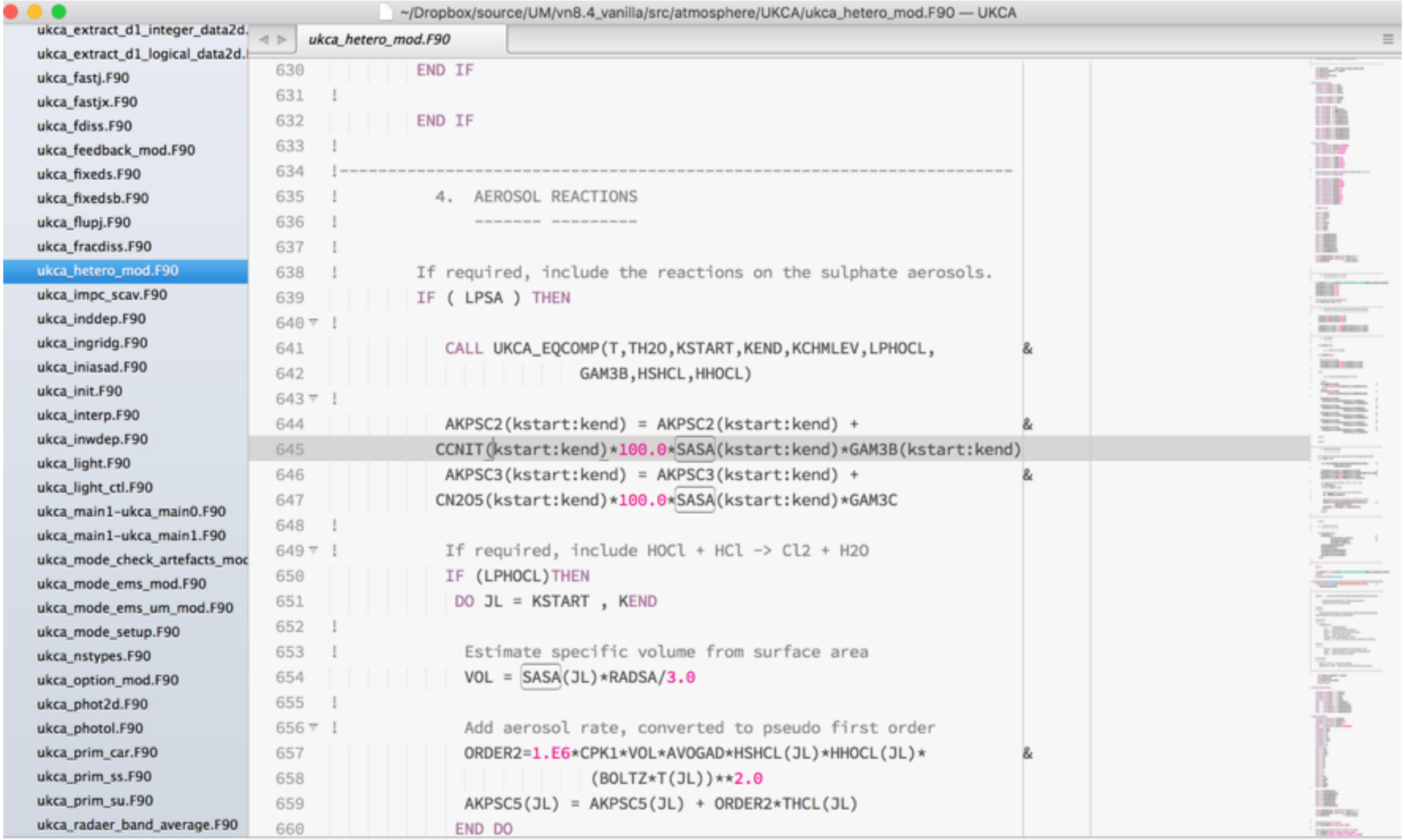
UKCA/ukca_hetero_mod.F90



```
      INTEGER, SAVE :: n_hoct_hct=0

      LOGICAL, SAVE :: first = .TRUE.

      REAL :: zp(theta_field_size)
      REAL :: zt(theta_field_size)
      REAL :: zhno3(theta_field_size)
      REAL :: zh2o(theta_field_size)
      REAL :: zhcl(theta_field_size)
      REAL :: zclono2(theta_field_size)
      REAL :: zn2o5(theta_field_size)
      REAL :: zhocl(theta_field_size)
      REAL :: psc1(theta_field_size)
      REAL :: psc2(theta_field_size)
      REAL :: psc3(theta_field_size)
      REAL :: psc4(theta_field_size)
      REAL :: psc5(theta_field_size)
      REAL :: hk(theta_field_size,5)

      INTEGER(KIND=jpim), PARAMETER :: zhook_in  = 0
      INTEGER(KIND=jpim), PARAMETER :: zhook_out = 1
      REAL(KIND=jprb)               :: zhook_handle

!

      IF (lhook) CALL dr_hook('UKCA_HETERO_MOD:UKCA_HETERO',zhook_in,zhook_handle)
      IF (first) THEN
        DO js = 1, jpctr
          SELECT CASE (advt(js))
            CASE ('H2O       ','H2OS      ')
              ih2o = js
            CASE ('HONO2     ')
              ihno3 = js
```

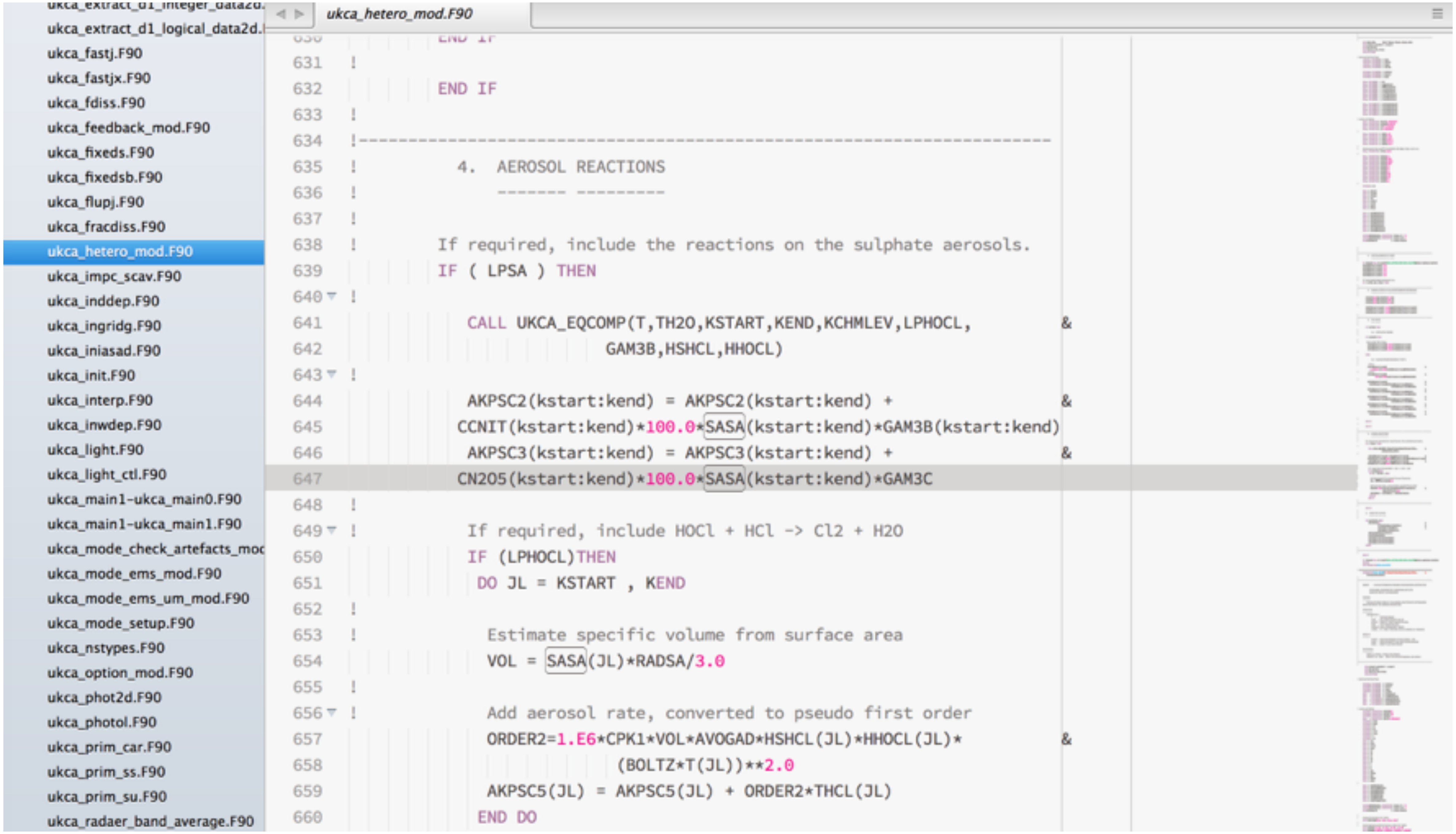
Monday Jan 04 10:04:31 2016, Line 127, Column 37

UKCA/ukca_chem_hetero_mod.F90



```fortran
        END IF
!
        END IF
!
!---------------------------------------------------------------------
!         4.  AEROSOL REACTIONS
!             ------- ---------
!
!       If required, include the reactions on the sulphate aerosols.
        IF ( LPSA ) THEN
!
           CALL UKCA_EQCOMP(T,TH2O,KSTART,KEND,KCHMLEV,LPHOCL,         &
                         GAM3B,HSHCL,HHOCL)
!
           AKPSC2(kstart:kend) = AKPSC2(kstart:kend) +                 &
          CCNIT(kstart:kend)*100.0*SASA(kstart:kend)*GAM3B(kstart:kend)
           AKPSC3(kstart:kend) = AKPSC3(kstart:kend) +                 &
          CN2O5(kstart:kend)*100.0*SASA(kstart:kend)*GAM3C
!
!          If required, include HOCl + HCl -> Cl2 + H2O
           IF (LPHOCL)THEN
            DO JL = KSTART , KEND
!
!            Estimate specific volume from surface area
             VOL = SASA(JL)*RADSA/3.0
!
!            Add aerosol rate, converted to pseudo first order
             ORDER2=1.E6*CPK1*VOL*AVOGAD*HSHCL(JL)*HHOCL(JL)*          &
                          (BOLTZ*T(JL))**2.0
             AKPSC5(JL) = AKPSC5(JL) + ORDER2*THCL(JL)
            END DO
```

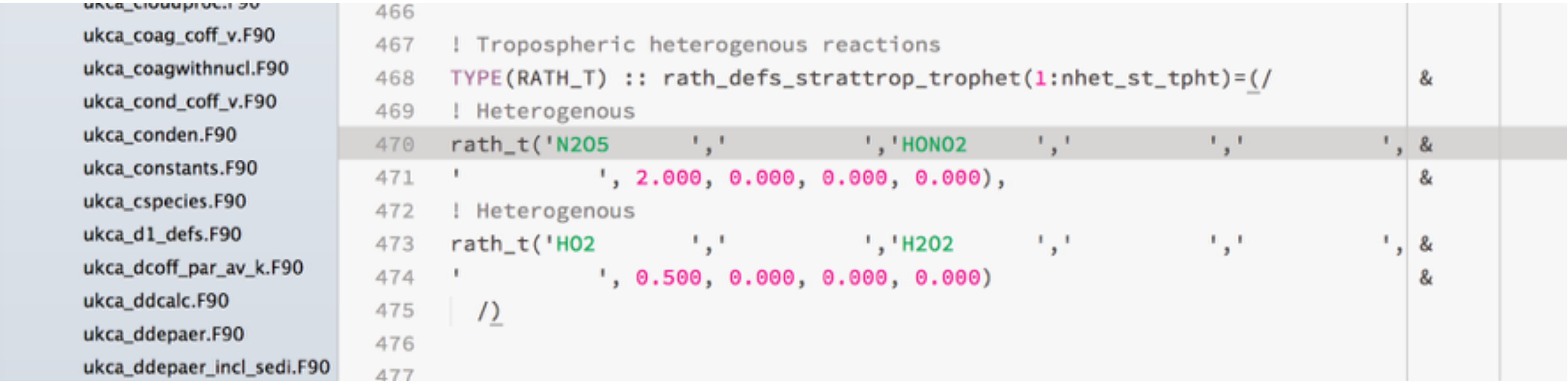
UKCA/ukca_chem_hetero_mod.F90



```
          END IF
!
          END IF
!
!-----------------------------------------------------------------------
!           4.  AEROSOL REACTIONS
!               ------- ---------
!
!         If required, include the reactions on the sulphate aerosols.
          IF ( LPSA ) THEN
!
             CALL UKCA_EQCOMP(T,TH2O,KSTART,KEND,KCHMLEV,LPHOCL,               &
                                   GAM3B,HSHCL,HHOCL)
!
             AKPSC2(kstart:kend) = AKPSC2(kstart:kend) +                    &
            CCNIT(kstart:kend)*100.0*SASA(kstart:kend)*GAM3B(kstart:kend)
             AKPSC3(kstart:kend) = AKPSC3(kstart:kend) +                    &
            CN2O5(kstart:kend)*100.0*SASA(kstart:kend)*GAM3C
!
             If required, include HOCl + HCl -> Cl2 + H2O
             IF (LPHOCL)THEN
              DO JL = KSTART , KEND
!
               Estimate specific volume from surface area
               VOL = SASA(JL)*RADSA/3.0
!
               Add aerosol rate, converted to pseudo first order
               ORDER2=1.E6*CPK1*VOL*AVOGAD*HSHCL(JL)*HHOCL(JL)*             &
                            (BOLTZ*T(JL))**2.0
               AKPSC5(JL) = AKPSC5(JL) + ORDER2*THCL(JL)
              END DO
```

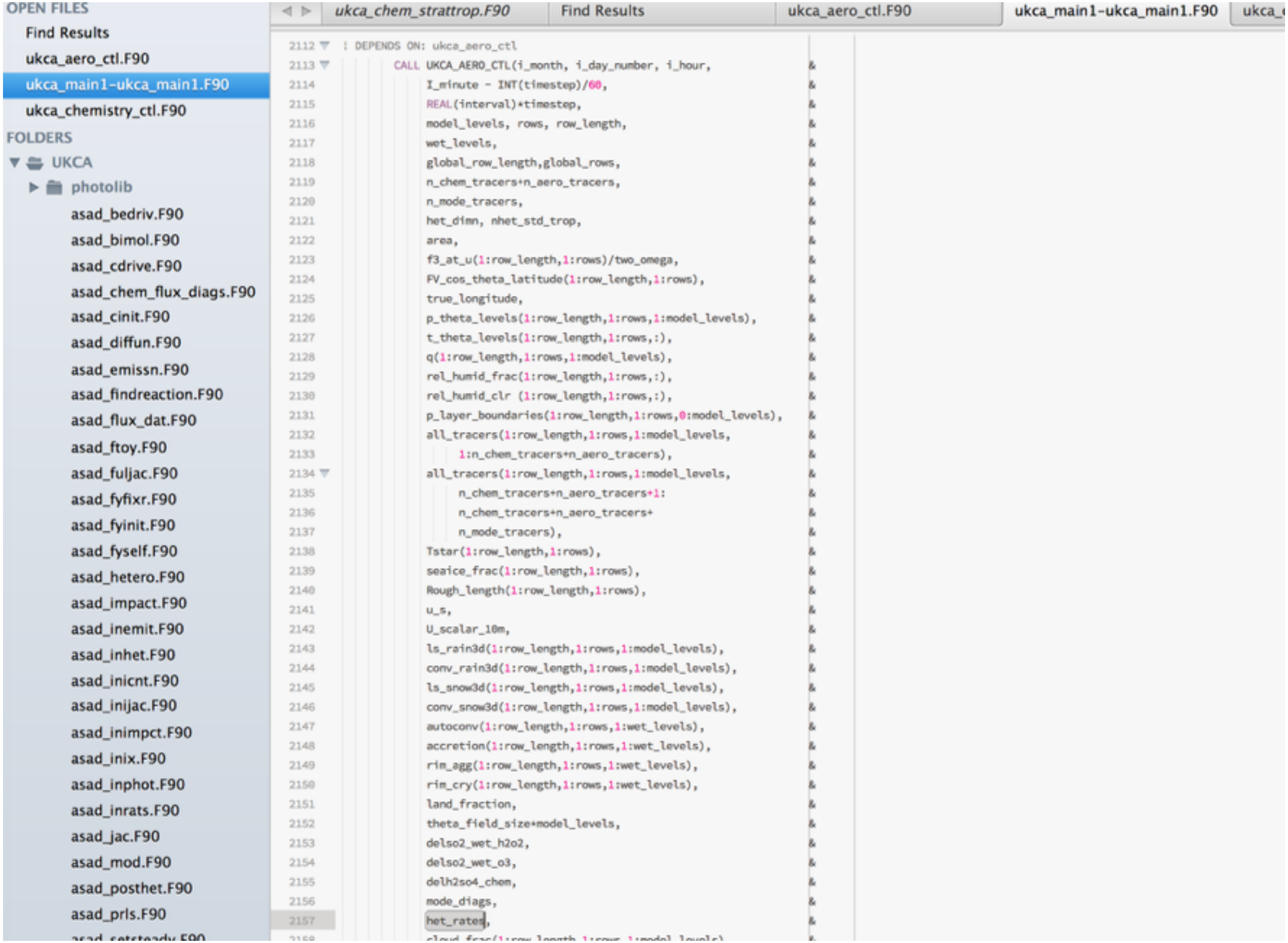
The MODE aerosol chemistry - slightly different

UKCA/ukca_chem_strattrop.F90



```

! Tropospheric heterogenous reactions
TYPE(RATH_T) :: rath_defs_strattrop_trophet(1:nhet_st_tpht)=(/           &
! Heterogenous
rath_t('N2O5      ','          ','HONO2     ','          ','          ', &
'          ', 2.000, 0.000, 0.000, 0.000),                               &
! Heterogenous
rath_t('HO2       ','          ','H2O2      ','          ','          ', &
'          ', 0.500, 0.000, 0.000, 0.000)                                &
  /)

```

UKCA/ukca_main1-ukca_main1.F90



OPEN FILES

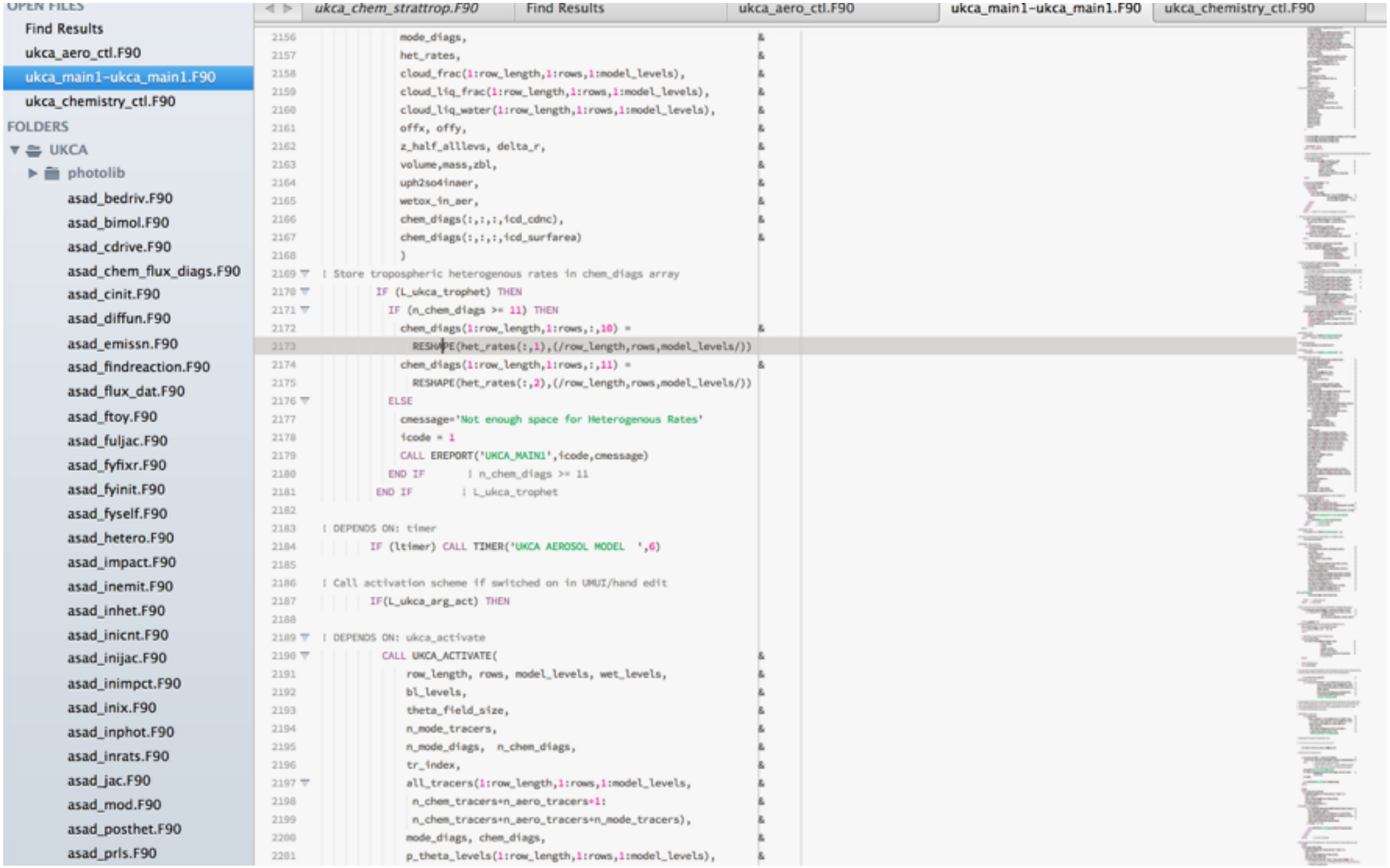
Find Results
ukca_aero_ctl.F90
ukca_main1-ukca_main1.F90
ukca_chemistry_ctl.F90

FOLDERS

▼ UKCA
▶ photolib
asad_bedriv.F90
asad_bimol.F90
asad_cdrive.F90
asad_chem_flux_diags.F90
asad_cinit.F90
asad_diffun.F90
asad_emissn.F90
asad_findreaction.F90
asad_flux_dat.F90
asad_ftoy.F90
asad_fuljac.F90
asad_fyfixr.F90
asad_fyinit.F90
asad_fyself.F90
asad_hetero.F90
asad_impact.F90
asad_inemit.F90
asad_inhet.F90
asad_inicnt.F90
asad_inijac.F90
asad_inimpct.F90
asad_inix.F90
asad_inphot.F90
asad_inrats.F90
asad_jac.F90
asad_mod.F90
asad_posthet.F90
asad_prls.F90
asad_setsteady.F90

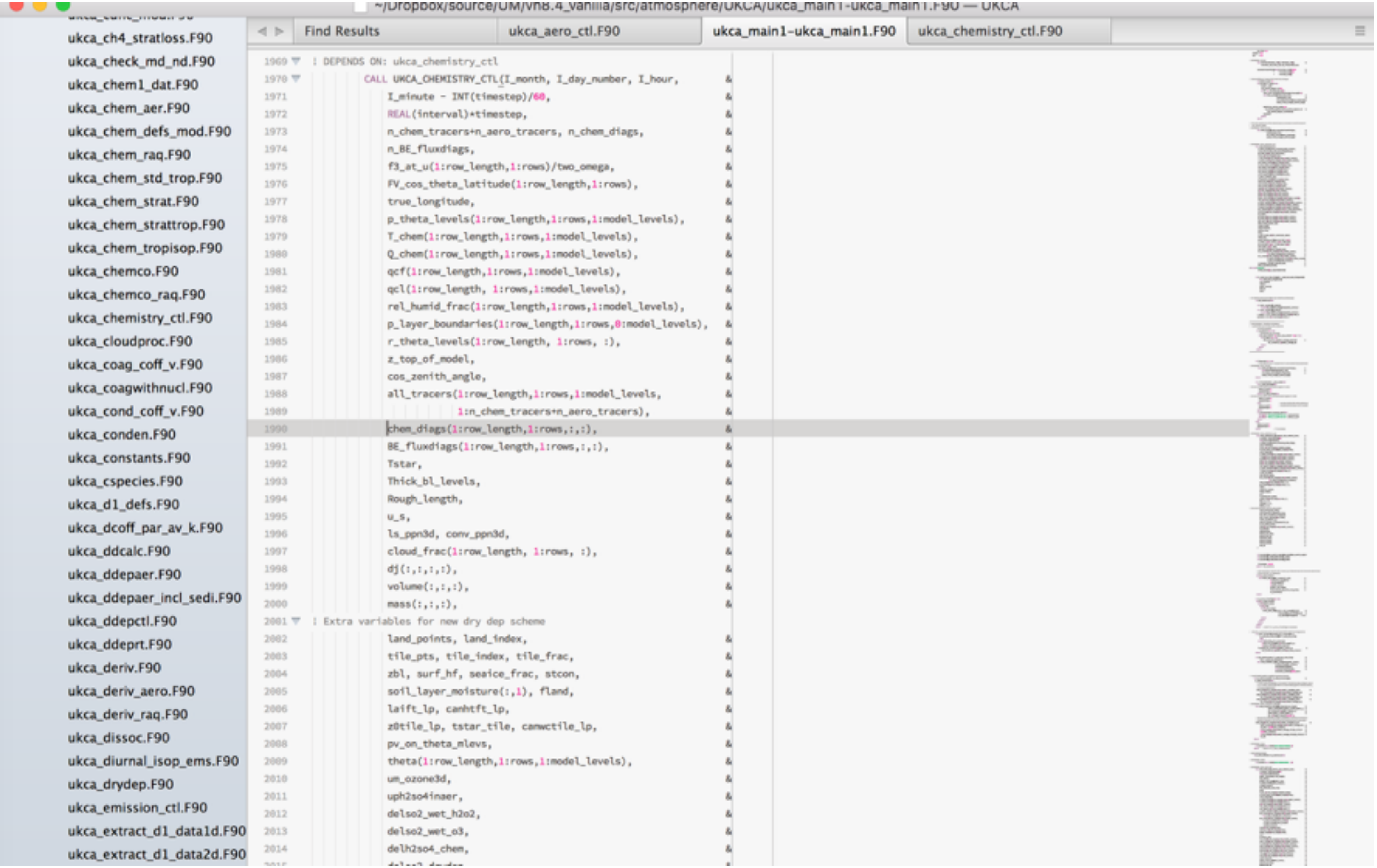
```
! DEPENDS ON: ukca_aero_ctl
      CALL UKCA_AERO_CTL(i_month, i_day_number, i_hour,                &
              I_minute - INT(timestep)/60,                             &
              REAL(interval)*timestep,                                 &
              model_levels, rows, row_length,                          &
              wet_levels,                                              &
              global_row_length,global_rows,                           &
              n_chem_tracers+n_aero_tracers,                           &
              n_mode_tracers,                                          &
              het_dimn, nhet_std_trop,                                 &
              area,                                                    &
              f3_at_u(1:row_length,1:rows)/two_omega,                  &
              FV_cos_theta_latitude(1:row_length,1:rows),              &
              true_longitude,                                          &
              p_theta_levels(1:row_length,1:rows,1:model_levels),      &
              t_theta_levels(1:row_length,1:rows,:),                   &
              q(1:row_length,1:rows,1:model_levels),                   &
              rel_humid_frac(1:row_length,1:rows,:),                   &
              rel_humid_clr (1:row_length,1:rows,:),                   &
              p_layer_boundaries(1:row_length,1:rows,0:model_levels),  &
              all_tracers(1:row_length,1:rows,1:model_levels,          &
                  1:n_chem_tracers+n_aero_tracers),                    &
              all_tracers(1:row_length,1:rows,1:model_levels,          &
                  n_chem_tracers+n_aero_tracers+1:                     &
                  n_chem_tracers+n_aero_tracers+                       &
                  n_mode_tracers),                                     &
              Tstar(1:row_length,1:rows),                              &
              seaice_frac(1:row_length,1:rows),                        &
              Rough_length(1:row_length,1:rows),                       &
              u_s,                                                     &
              U_scalar_10m,                                            &
              ls_rain3d(1:row_length,1:rows,1:model_levels),           &
              conv_rain3d(1:row_length,1:rows,1:model_levels),         &
              ls_snow3d(1:row_length,1:rows,1:model_levels),           &
              conv_snow3d(1:row_length,1:rows,1:model_levels),         &
              autoconv(1:row_length,1:rows,1:wet_levels),              &
              accretion(1:row_length,1:rows,1:wet_levels),             &
              rim_agg(1:row_length,1:rows,1:wet_levels),               &
              rim_cry(1:row_length,1:rows,1:wet_levels),               &
              land_fraction,                                           &
              theta_field_size*model_levels,                           &
              delso2_wet_h2o2,                                         &
              delso2_wet_o3,                                           &
              delh2so4_chem,                                           &
              mode_diags,                                              &
              het_rates,                                               &
              cloud_frac(1:row_length,1:rows,1:model_levels)           &
```

UKCA/ukca_main1-ukca_main1.F90



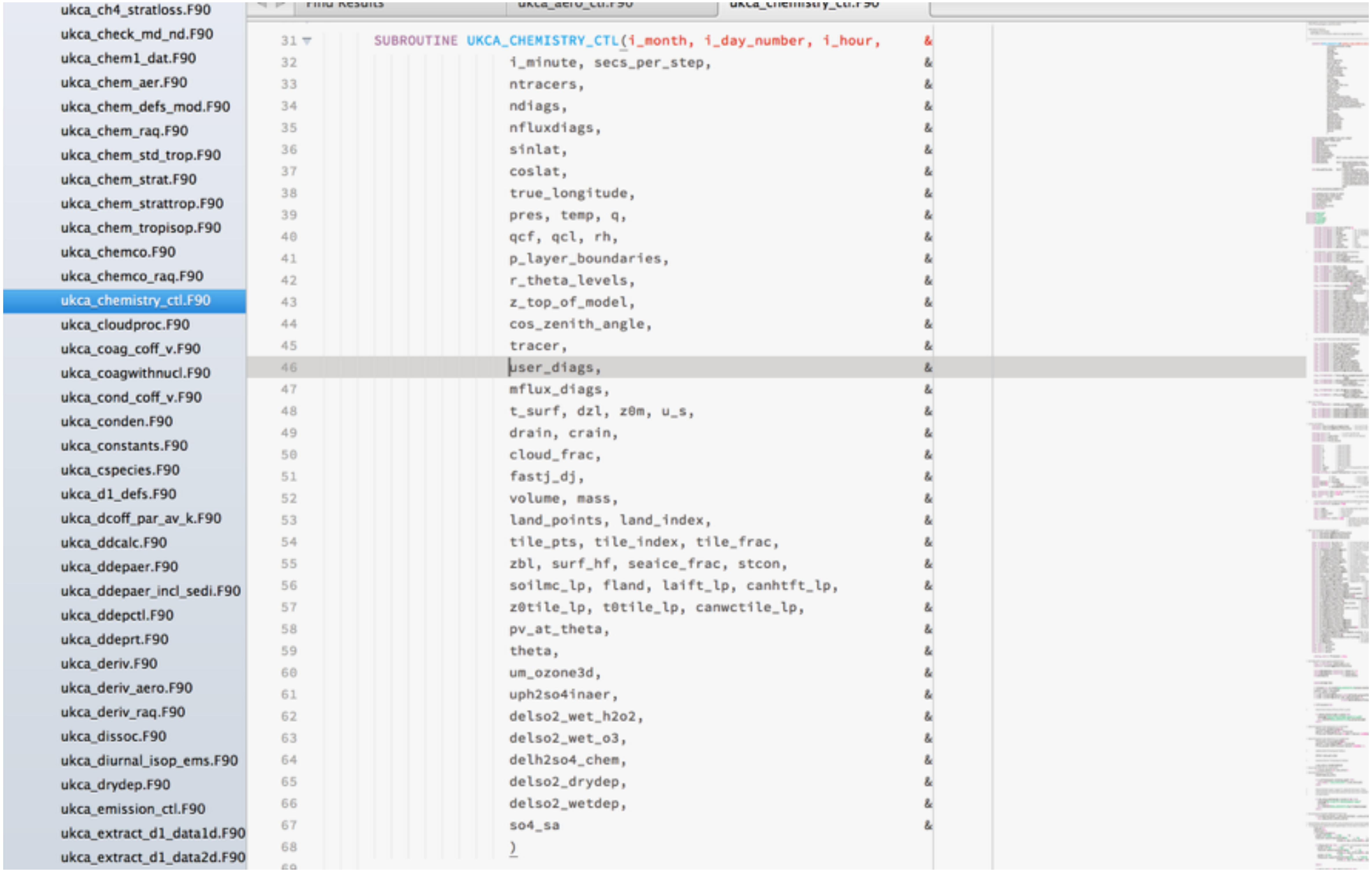
```
             mode_diags,                                                &
             het_rates,                                                 &
             cloud_frac(1:row_length,1:rows,1:model_levels),            &
             cloud_liq_frac(1:row_length,1:rows,1:model_levels),        &
             cloud_liq_water(1:row_length,1:rows,1:model_levels),       &
             offx, offy,                                                &
             z_half_alllevs, delta_r,                                   &
             volume,mass,zbl,                                           &
             uph2so4inaer,                                              &
             wetox_in_aer,                                              &
             chem_diags(:,:,:,icd_cdnc),                                &
             chem_diags(:,:,:,icd_surfarea)                             &
             )
! Store tropospheric heterogenous rates in chem_diags array
         IF (L_ukca_trophet) THEN
           IF (n_chem_diags >= 11) THEN
             chem_diags(1:row_length,1:rows,:,10) =                     &
               RESHAPE(het_rates(:,1),(/row_length,rows,model_levels/))
             chem_diags(1:row_length,1:rows,:,11) =                     &
               RESHAPE(het_rates(:,2),(/row_length,rows,model_levels/))
           ELSE
             cmessage='Not enough space for Heterogenous Rates'
             icode = 1
             CALL EREPORT('UKCA_MAIN1',icode,cmessage)
           END IF       ! n_chem_diags >= 11
         END IF       ! L_ukca_trophet

! DEPENDS ON: timer
        IF (ltimer) CALL TIMER('UKCA AEROSOL MODEL  ',6)

! Call activation scheme if switched on in UMUI/hand edit
        IF(L_ukca_arg_act) THEN

! DEPENDS ON: ukca_activate
          CALL UKCA_ACTIVATE(                                           &
              row_length, rows, model_levels, wet_levels,               &
              bl_levels,                                                &
              theta_field_size,                                         &
              n_mode_tracers,                                           &
              n_mode_diags,  n_chem_diags,                              &
              tr_index,                                                 &
              all_tracers(1:row_length,1:rows,1:model_levels,           &
               n_chem_tracers+n_aero_tracers+1:                         &
               n_chem_tracers+n_aero_tracers+n_mode_tracers),           &
              mode_diags, chem_diags,                                   &
              p_theta_levels(1:row_length,1:rows,1:model_levels),       &
```

UKCA/ukca_main1-ukca_main1.F90



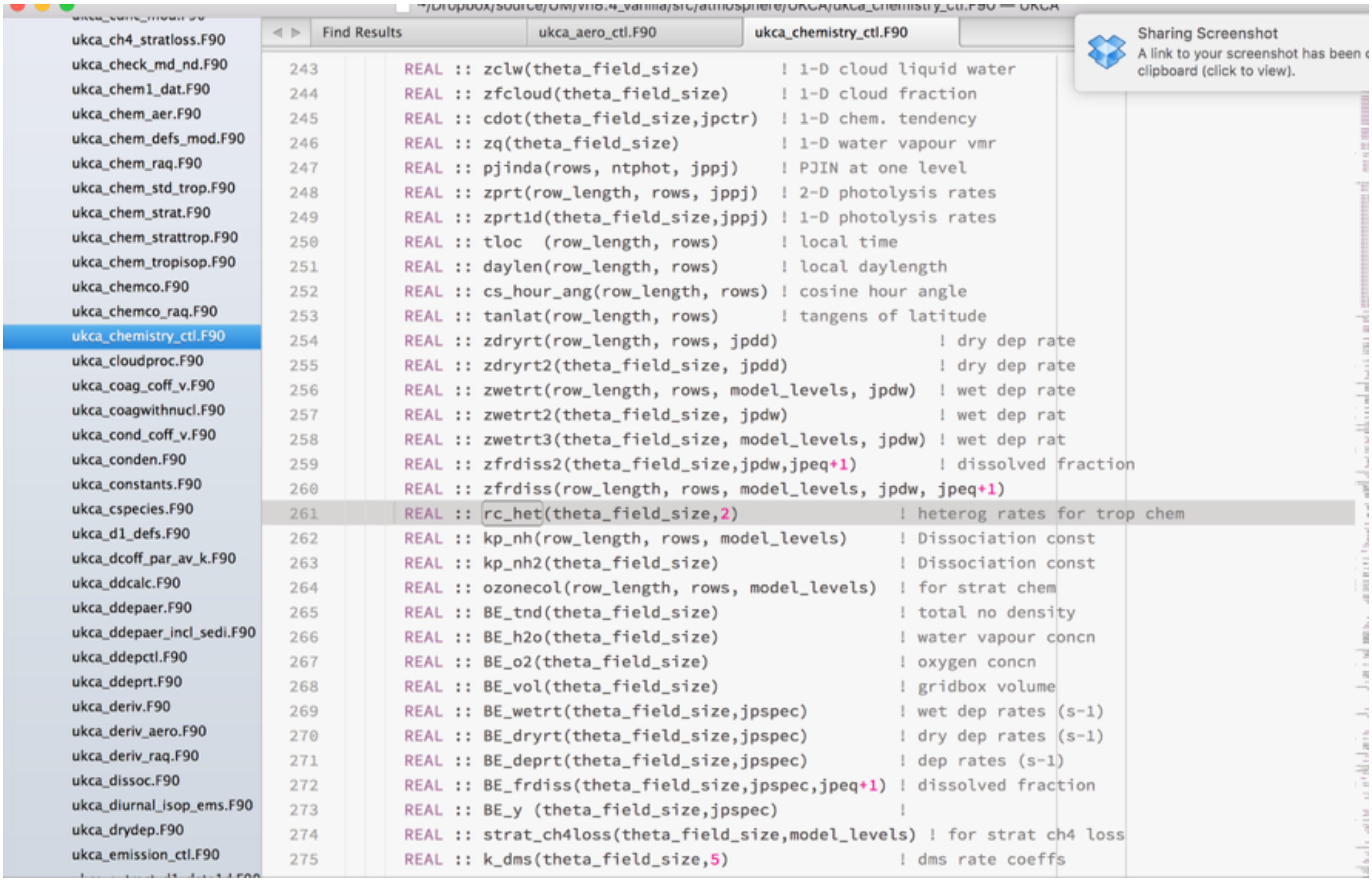
```
! DEPENDS ON: ukca_chemistry_ctl
       CALL UKCA_CHEMISTRY_CTL(I_month, I_day_number, I_hour,          &
          I_minute - INT(timestep)/60,                                 &
          REAL(interval)*timestep,                                     &
          n_chem_tracers+n_aero_tracers, n_chem_diags,                 &
          n_BE_fluxdiags,                                              &
          f3_at_u(1:row_length,1:rows)/two_omega,                      &
          FV_cos_theta_latitude(1:row_length,1:rows),                  &
          true_longitude,                                              &
          p_theta_levels(1:row_length,1:rows,1:model_levels),          &
          T_chem(1:row_length,1:rows,1:model_levels),                  &
          Q_chem(1:row_length,1:rows,1:model_levels),                  &
          qcf(1:row_length,1:rows,1:model_levels),                     &
          qcl(1:row_length, 1:rows,1:model_levels),                    &
          rel_humid_frac(1:row_length,1:rows,1:model_levels),          &
          p_layer_boundaries(1:row_length,1:rows,0:model_levels),      &
          r_theta_levels(1:row_length, 1:rows, :),                     &
          z_top_of_model,                                              &
          cos_zenith_angle,                                            &
          all_tracers(1:row_length,1:rows,1:model_levels,              &
                      1:n_chem_tracers+n_aero_tracers),                &
          chem_diags(1:row_length,1:rows,:,:),                         &
          BE_fluxdiags(1:row_length,1:rows,:,:),                       &
          Tstar,                                                       &
          Thick_bl_levels,                                             &
          Rough_length,                                                &
          u_s,                                                         &
          ls_ppn3d, conv_ppn3d,                                        &
          cloud_frac(1:row_length, 1:rows, :),                         &
          dj(:,:,:,:),                                                 &
          volume(:,:,:),                                               &
          mass(:,:,:),                                                 &
! Extra variables for new dry dep scheme
          land_points, land_index,                                     &
          tile_pts, tile_index, tile_frac,                             &
          zbl, surf_hf, seaice_frac, stcon,                            &
          soil_layer_moisture(:,1), fland,                             &
          laift_lp, canhtft_lp,                                        &
          z0tile_lp, tstar_tile, canwctile_lp,                         &
          pv_on_theta_mlevs,                                           &
          theta(1:row_length,1:rows,1:model_levels),                   &
          um_ozone3d,                                                  &
          uph2so4inaer,                                                &
          delso2_wet_h2o2,                                             &
          delso2_wet_o3,                                               &
          delh2so4_chem,                                               &
```

UKCA/ukca_chemistry_ctl.F90



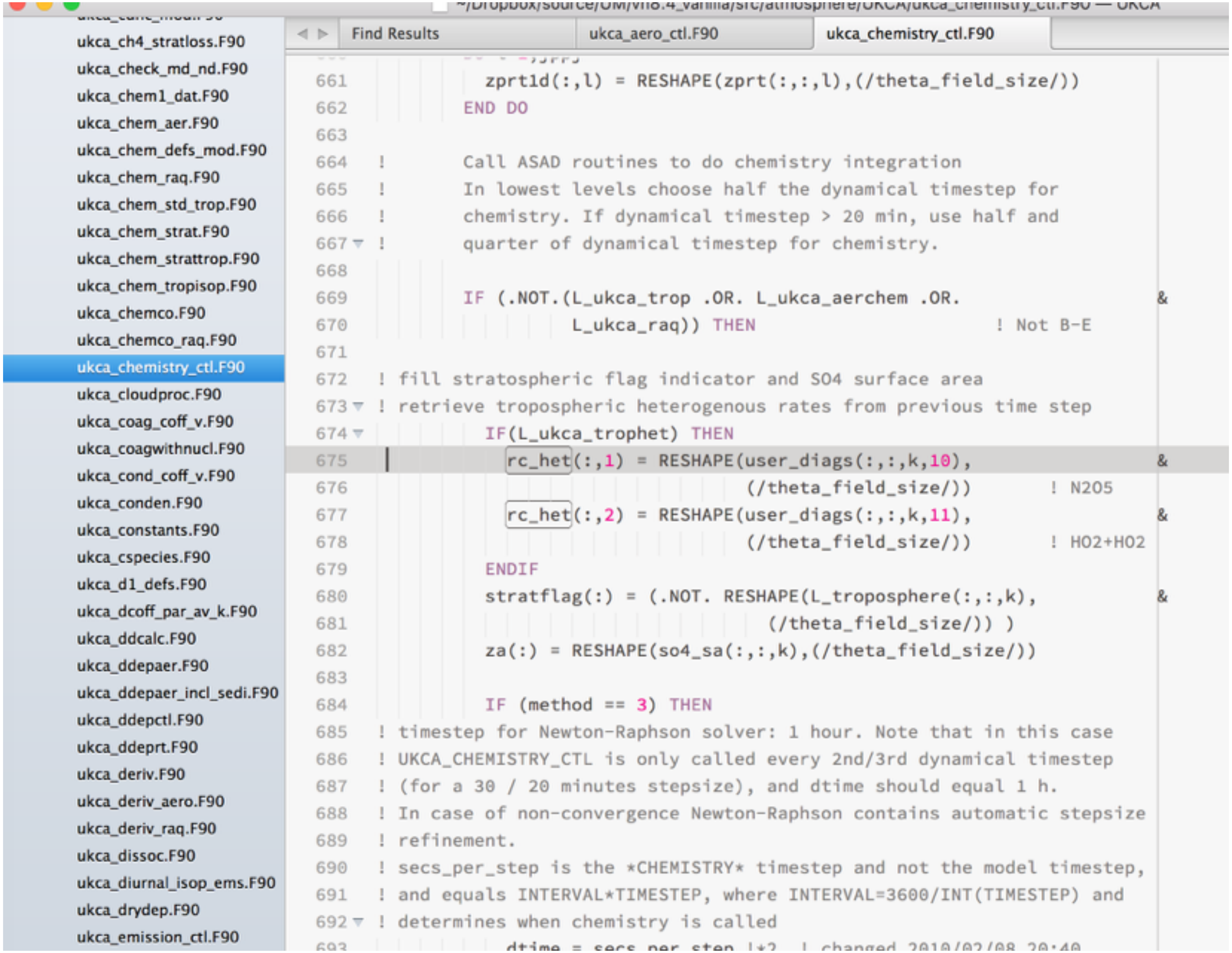
```
SUBROUTINE UKCA_CHEMISTRY_CTL(i_month, i_day_number, i_hour,     &
                i_minute, secs_per_step,                         &
                ntracers,                                        &
                ndiags,                                          &
                nfluxdiags,                                      &
                sinlat,                                          &
                coslat,                                          &
                true_longitude,                                  &
                pres, temp, q,                                   &
                qcf, qcl, rh,                                    &
                p_layer_boundaries,                              &
                r_theta_levels,                                  &
                z_top_of_model,                                  &
                cos_zenith_angle,                                &
                tracer,                                          &
                user_diags,                                      &
                mflux_diags,                                     &
                t_surf, dzl, z0m, u_s,                           &
                drain, crain,                                    &
                cloud_frac,                                      &
                fastj_dj,                                        &
                volume, mass,                                    &
                land_points, land_index,                         &
                tile_pts, tile_index, tile_frac,                 &
                zbl, surf_hf, seaice_frac, stcon,                &
                soilmc_lp, fland, laift_lp, canhtft_lp,          &
                z0tile_lp, t0tile_lp, canwctile_lp,              &
                pv_at_theta,                                     &
                theta,                                           &
                um_ozone3d,                                      &
                uph2so4inaer,                                    &
                delso2_wet_h2o2,                                 &
                delso2_wet_o3,                                   &
                delh2so4_chem,                                   &
                delso2_drydep,                                   &
                delso2_wetdep,                                   &
                so4_sa                                           &
                )
```

UKCA/ukca_chemistry_ctl.F90



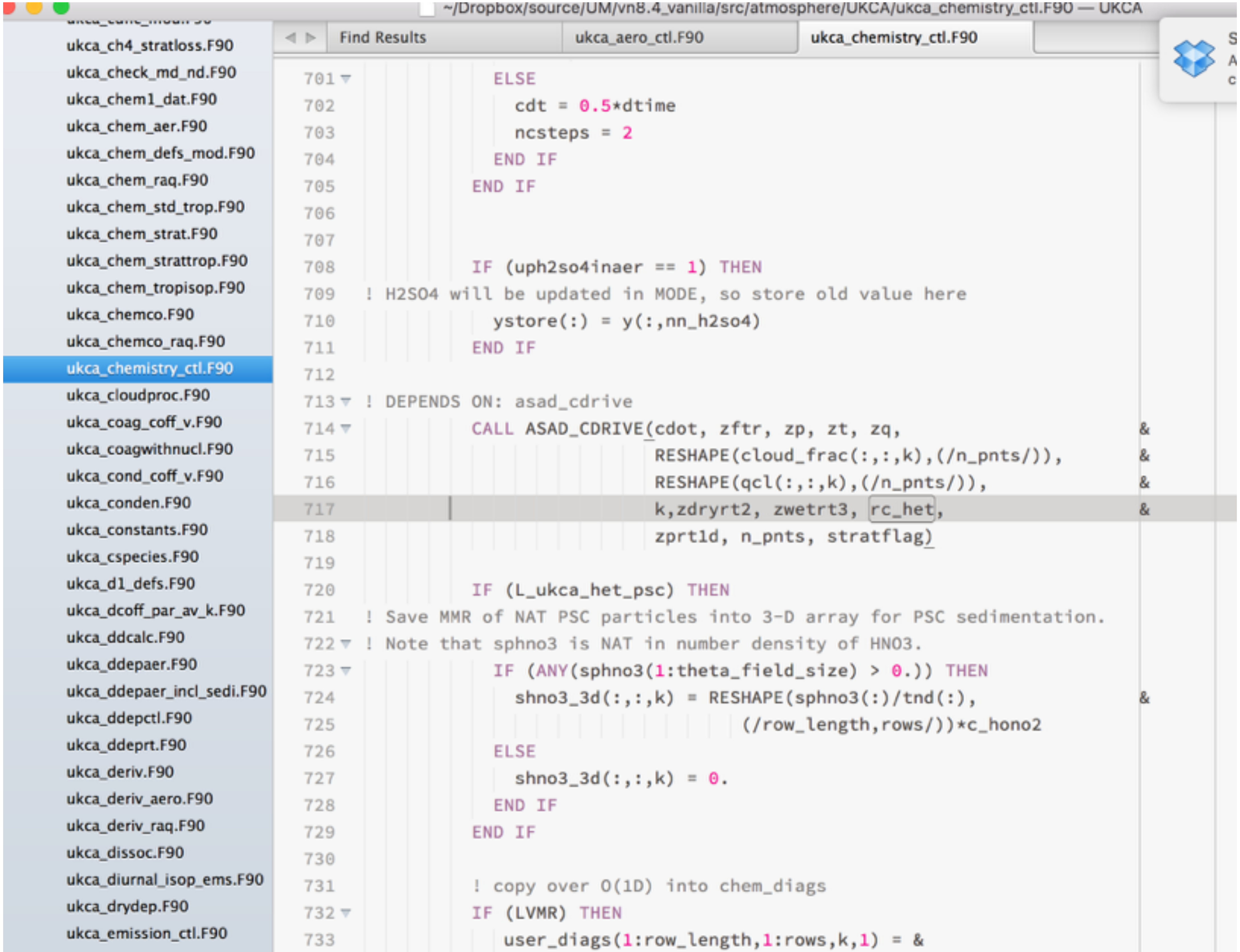
```
REAL :: zclw(theta_field_size)        ! 1-D cloud liquid water
REAL :: zfcloud(theta_field_size)     ! 1-D cloud fraction
REAL :: cdot(theta_field_size,jpctr)  ! 1-D chem. tendency
REAL :: zq(theta_field_size)          ! 1-D water vapour vmr
REAL :: pjinda(rows, ntphot, jppj)    ! PJIN at one level
REAL :: zprt(row_length, rows, jppj)  ! 2-D photolysis rates
REAL :: zprt1d(theta_field_size,jppj) ! 1-D photolysis rates
REAL :: tloc  (row_length, rows)      ! local time
REAL :: daylen(row_length, rows)      ! local daylength
REAL :: cs_hour_ang(row_length, rows) ! cosine hour angle
REAL :: tanlat(row_length, rows)      ! tangens of latitude
REAL :: zdryrt(row_length, rows, jpdd)                ! dry dep rate
REAL :: zdryrt2(theta_field_size, jpdd)               ! dry dep rate
REAL :: zwetrt(row_length, rows, model_levels, jpdw)  ! wet dep rate
REAL :: zwetrt2(theta_field_size, jpdw)               ! wet dep rat
REAL :: zwetrt3(theta_field_size, model_levels, jpdw) ! wet dep rat
REAL :: zfrdiss2(theta_field_size,jpdw,jpeq+1)        ! dissolved fraction
REAL :: zfrdiss(row_length, rows, model_levels, jpdw, jpeq+1)
REAL :: rc_het(theta_field_size,2)                ! heterog rates for trop chem
REAL :: kp_nh(row_length, rows, model_levels)     ! Dissociation const
REAL :: kp_nh2(theta_field_size)                  ! Dissociation const
REAL :: ozonecol(row_length, rows, model_levels)  ! for strat chem
REAL :: BE_tnd(theta_field_size)                  ! total no density
REAL :: BE_h2o(theta_field_size)                  ! water vapour concn
REAL :: BE_o2(theta_field_size)                   ! oxygen concn
REAL :: BE_vol(theta_field_size)                  ! gridbox volume
REAL :: BE_wetrt(theta_field_size,jpspec)         ! wet dep rates (s-1)
REAL :: BE_dryrt(theta_field_size,jpspec)         ! dry dep rates (s-1)
REAL :: BE_deprt(theta_field_size,jpspec)         ! dep rates (s-1)
REAL :: BE_frdiss(theta_field_size,jpspec,jpeq+1) ! dissolved fraction
REAL :: BE_y (theta_field_size,jpspec)            !
REAL :: strat_ch4loss(theta_field_size,model_levels) ! for strat ch4 loss
REAL :: k_dms(theta_field_size,5)                 ! dms rate coeffs
```

UKCA/ukca_chemistry_ctl.F90



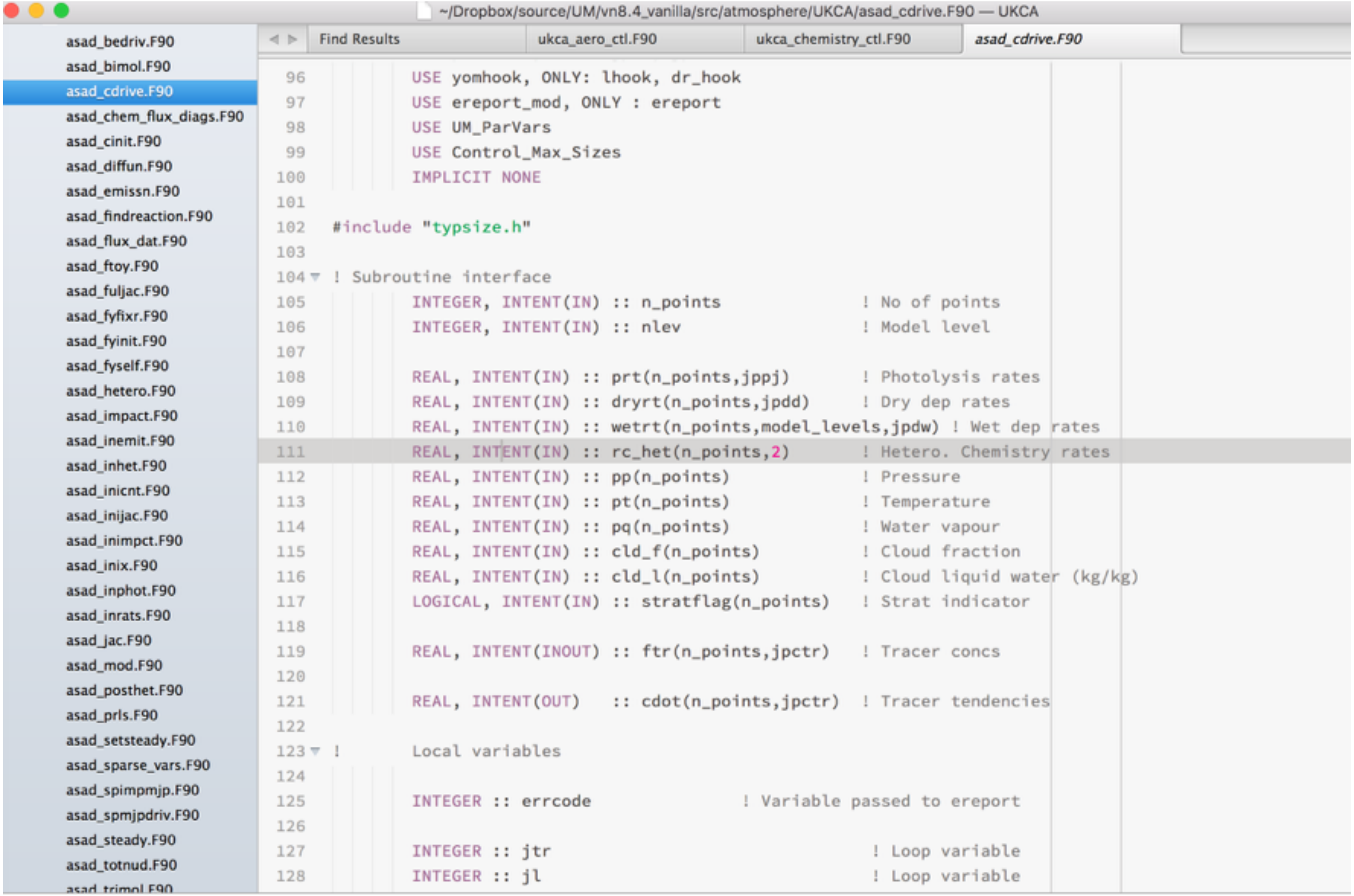
```
     zprt1d(:,l) = RESHAPE(zprt(:,:,l),(/theta_field_size/))
    END DO

!   Call ASAD routines to do chemistry integration
!   In lowest levels choose half the dynamical timestep for
!   chemistry. If dynamical timestep > 20 min, use half and
!   quarter of dynamical timestep for chemistry.

    IF (.NOT.(L_ukca_trop .OR. L_ukca_aerchem .OR.                  &
         L_ukca_raq)) THEN                     ! Not B-E

! fill stratospheric flag indicator and SO4 surface area
! retrieve tropospheric heterogenous rates from previous time step
     IF(L_ukca_trophet) THEN
      rc_het(:,1) = RESHAPE(user_diags(:,:,k,10),                   &
                 (/theta_field_size/))              ! N2O5
      rc_het(:,2) = RESHAPE(user_diags(:,:,k,11),                   &
                 (/theta_field_size/))              ! HO2+HO2
     ENDIF
     stratflag(:) = (.NOT. RESHAPE(L_troposphere(:,:,k),            &
                  (/theta_field_size/)) )
     za(:) = RESHAPE(so4_sa(:,:,k),(/theta_field_size/))

     IF (method == 3) THEN
! timestep for Newton-Raphson solver: 1 hour. Note that in this case
! UKCA_CHEMISTRY_CTL is only called every 2nd/3rd dynamical timestep
! (for a 30 / 20 minutes stepsize), and dtime should equal 1 h.
! In case of non-convergence Newton-Raphson contains automatic stepsize
! refinement.
! secs_per_step is the *CHEMISTRY* timestep and not the model timestep,
! and equals INTERVAL*TIMESTEP, where INTERVAL=3600/INT(TIMESTEP) and
! determines when chemistry is called
      dtime = secs_per_step !*2   ! changed 2010/02/08 20:40
```

UKCA/ukca_chemistry_ctl.F90



```
            ELSE
              cdt = 0.5*dtime
              ncsteps = 2
            END IF
          END IF


          IF (uph2so4inaer == 1) THEN
! H2SO4 will be updated in MODE, so store old value here
            ystore(:) = y(:,nn_h2so4)
          END IF

! DEPENDS ON: asad_cdrive
          CALL ASAD_CDRIVE(cdot, zftr, zp, zt, zq,                      &
                           RESHAPE(cloud_frac(:,:,k),(/n_pnts/)),       &
                           RESHAPE(qcl(:,:,k),(/n_pnts/)),              &
                           k,zdryrt2, zwetrt3, rc_het,                  &
                           zprt1d, n_pnts, stratflag)

          IF (L_ukca_het_psc) THEN
! Save MMR of NAT PSC particles into 3-D array for PSC sedimentation.
! Note that sphno3 is NAT in number density of HNO3.
            IF (ANY(sphno3(1:theta_field_size) > 0.)) THEN
              shno3_3d(:,:,k) = RESHAPE(sphno3(:)/tnd(:),               &
                                   (/row_length,rows/))*c_hono2
            ELSE
              shno3_3d(:,:,k) = 0.
            END IF
          END IF

          ! copy over O(1D) into chem_diags
          IF (LVMR) THEN
             user_diags(1:row_length,1:rows,k,1) = &
```

UKCA/asad_cdrive.F90



```
        USE yomhook, ONLY: lhook, dr_hook
        USE ereport_mod, ONLY : ereport
        USE UM_ParVars
        USE Control_Max_Sizes
        IMPLICIT NONE

#include "typsize.h"

! Subroutine interface
        INTEGER, INTENT(IN) :: n_points              ! No of points
        INTEGER, INTENT(IN) :: nlev                  ! Model level

        REAL, INTENT(IN) :: prt(n_points,jppj)       ! Photolysis rates
        REAL, INTENT(IN) :: dryrt(n_points,jpdd)     ! Dry dep rates
        REAL, INTENT(IN) :: wetrt(n_points,model_levels,jpdw) ! Wet dep rates
        REAL, INTENT(IN) :: rc_het(n_points,2)       ! Hetero. Chemistry rates
        REAL, INTENT(IN) :: pp(n_points)             ! Pressure
        REAL, INTENT(IN) :: pt(n_points)             ! Temperature
        REAL, INTENT(IN) :: pq(n_points)             ! Water vapour
        REAL, INTENT(IN) :: cld_f(n_points)          ! Cloud fraction
        REAL, INTENT(IN) :: cld_l(n_points)          ! Cloud liquid water (kg/kg)
        LOGICAL, INTENT(IN) :: stratflag(n_points)   ! Strat indicator

        REAL, INTENT(INOUT) :: ftr(n_points,jpctr)   ! Tracer concs

        REAL, INTENT(OUT)   :: cdot(n_points,jpctr)  ! Tracer tendencies

!       Local variables

        INTEGER :: errcode               ! Variable passed to ereport

        INTEGER :: jtr                                ! Loop variable
        INTEGER :: jl                                 ! Loop variable
```