Supplemental information:
How non-equilibrium aerosol chemistry impacts particle acidity: the GMXe AERosol CHEMistry (GMXe–AERCHEM, v1.0) sub-submodel of MESSy

Correspondence: Simon Rosanka (s.rosanka@fz-juelich.de, srosanka@uci.edu)

Contents

1 Representation of secondary organic aerosols (SOA) – the GMXe–SOA sub-submodel **4**

2 GMXe–AERCHEM and GMXe–SOA manual **7**

2.1 Selecting the in-cloud aqueous-phase mechanism and related adjustments in SCAV 7

2.2 Selecting the aqueous-phase mechanism for deliquescent aerosols in GMXe-AERCHEM 8

2.3 Necessary changes to the MECCA submodel and selecting the gas-phase mechanism 8

2.4 GMXe setup for GMXe–AERCHEM and GMXe–SOA 9

2.4.1 GMXe–AERCHEM sub-submodel options 9

2.4.2 GMXe–SOA sub-submodel options 10

References **20**

List of Figures

S1 Mean yearly aerosol pH for the soluble accumulation mode (as) simulated by ISORROPIA-II and by AERCHEM. 2

S2 Mean yearly aerosol pH for the soluble coarse mode (cs) simulated by ISORROPIA-II and by AERCHEM. . . 3

List of Tables

S1 Temperature dependent rate constants for particular precursors with OH, O_3, and NO_3. 6

S2 SOA parameters used for the implemented two-product model. 6

List of Namelists

S1 Example GMXe setup when using GMXe-AERCHEM 12

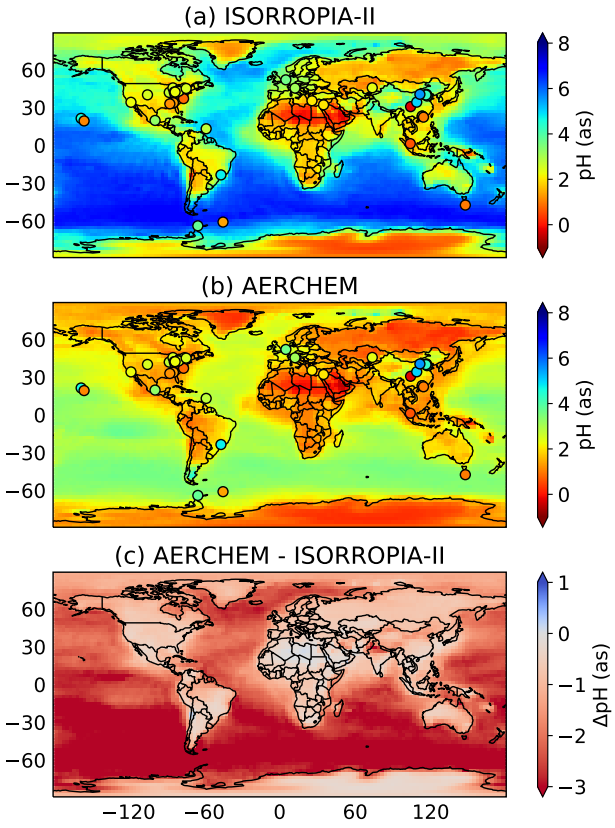

Figure S1. Mean yearly aerosol pH for the soluble accumulation mode (as) simulated by (a) ISORROPIA-II and by (b) AERCHEM. Subfigure (c) represents the absolute difference of the yearly means. Please note that for the figure showing the absolute pH differences, an increase in acidity (decrease in pH) is indicated by red shading, whereas an increase in pH is indicated in blue. For comparison, observed fine particle acidity, based on the dataset published by Pye (2020), is indicated by circles in subfigures (a) and (b).

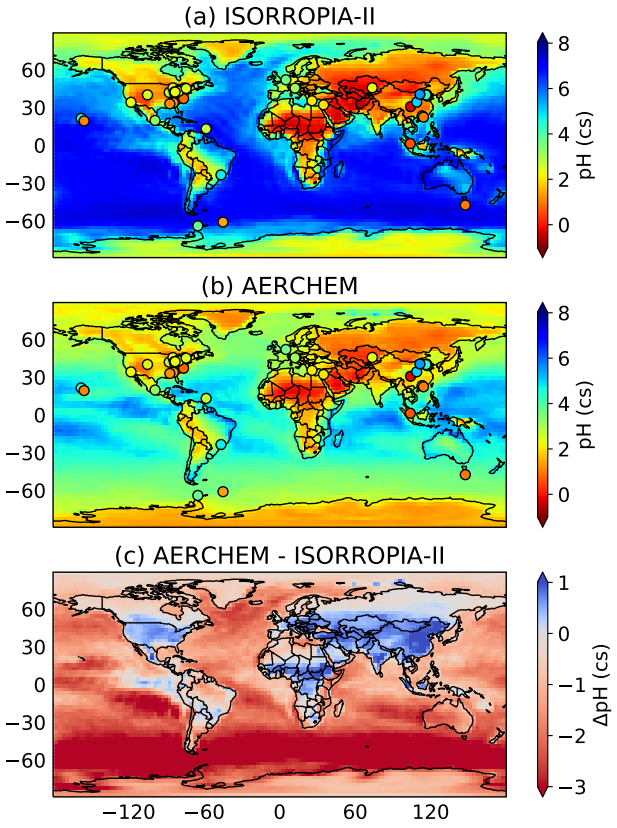

Figure S2. Mean yearly aerosol pH for the soluble coarse mode (cs) simulated by (a) ISORROPIA-II and by (b) AERCHEM. Subfigure (c) represents the absolute difference of the yearly means. Please note that for the figure showing the absolute pH differences, an increase in acidity (decrease in pH) is indicated by red shading, whereas an increase in pH is indicated in blue. For comparison, observed fine particle acidity, based on the dataset published by Pye (2020), is indicated by circles in subfigures (a) and (b).

1 Representation of secondary organic aerosols (SOA) – the GMXe–SOA sub-submodel

The representation of the formation and evolution of secondary organic aerosols (SOA) is computationally demanding. A detailed representation is thus outside the scope of this study. In order to be able to calculate the water uptake due to organic aerosols, we implement a simple computationally efficient sub-submodel in GMXe (GMXe-SOA) to calculate the formation of SOA from isoprene, α-pinene, β-pinene, toluene, and xylens. This model follows the description presented by Tsigaridis and Kanakidou (2003), Zhang et al. (2007), and O'Donnell et al. (2011).

In short, we implement the two-product model for SOA formation based on the work of Odum et al. (1996). They demonstrated that for a reaction yielding multiple semi-volatile products, the aerosol yield can be represented by assuming that only two condensable species are produced ($\mathrm{SP_1}$ and $\mathrm{SP_2}$, respectively). For known mass-based stoichiometric coefficients for each product (α_1 and α_2, respectively), the gas-phase reaction of a given precursor (PRE, e.g., isoprene) with a given oxidant (OX, e.g., OH, $\mathrm{O_3}$, $\mathrm{NO_3}$) is given by:

$$PRE + OX \rightarrow \alpha_1 \times SP_1 + \alpha_2 \times SP_2 \tag{1}$$

Tables S1 and S2 provide a summary of all rate constants and SOA parameters, respectively.

For each condensable species (i) produced from VOC oxidation, the equilibrium between the gas and aerosol phase is governed by:

$$A_i = K_i M_0 G_i \tag{2}$$

where $\mathrm{K_i}$ is the partitioning coefficient of the condensable species, $\mathrm{A_i}$ and $\mathrm{G_i}$ are the aerosol-phase and gas-phase concentrations of the condensable species, respectively. $\mathrm{M_0}$ is the total particulate phase concentration, which is the sum of the primary particles (PP) that can serve as absorptive medium and the sum of the aerosol phase concentrations ($\mathrm{A_i}$) of all condensable species:

$$M_0 = PP + \sum_i A_i \tag{3}$$

Here, we represent the bulk partitioning coefficient ($\mathrm{K_i(T)}$) at a specific temperature (T) as:

$$K_i(T) = K_{i,ref} \times \frac{T}{T_{ref}} \times \exp\left[\frac{\Delta H_i}{R}\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right] \tag{4}$$

where $\mathrm{K_{i,ref}}$ is the reference partitioning coefficient at 298 K, $\mathrm{T_{ref}}$ the reference temperature of 298 K, R the universal gas constant, and $\mathrm{\Delta H_i}$ the enthalpy of vaporization of the specific compound. In this study, we assume $\mathrm{\Delta H_i}$ to be 156 $\mathrm{kJ\,mol^{-1}}$, the upper estimate discussed in Zhang et al. (2007). Table S2 lists all reference partitioning coefficients at 298 K ($\mathrm{K_{i,ref}}$).

By assuming that the total amount of semi-volatile products repartitions in each time step and all SOA has the potential to fully evaporate from the particles at each time step and defining $\mathrm{S_i}$ as the total mass of semi-volatile SOA of the given species

($\mathrm{S_i} = \mathrm{A_i} + \mathrm{G_i}$), we can rearrange Eq. (2) to solve for the gas and aerosol mass to:

$$G_i = \frac{S_i}{1 + K_i(T)M_0}, \text{and} \tag{5}$$

$$A_i = \frac{K_i(T)M_0 S_i}{1 + K_i(T)M_0} \tag{6}$$

Substituting equation (6) into equation (3), we can solve $\mathrm{M_0}$ iteratively, since PP and $\mathrm{S_i}$ are known.

O'Donnell et al. (2011) demonstrate that an equilibrium SOA partitioning between different size classes is consistent, when assuming unity for the activity coefficient for all SOA species, with the fraction of non-volatile absorbing aerosol mass in each mode:

$$\frac{A_{i_j}}{A_i} = \frac{M_{NV_j}}{M_{NV}} \tag{7}$$

where $\mathrm{M_{NV}}$ is the total non-volatile absorbing mass, $\mathrm{A_i}$ the total aerosol phase mass of SOA species i, $\mathrm{M_{NV_j}}$ the non-volatile absorbing mass in mode j, and $\mathrm{A_{i_j}}$ the aerosol phase mass of SOA species i in mode j. This allows us to calculate the partitioning for each SOA species for each aerosol mode in which SOA partitioning is permitted.

In a last step, the water uptake due to each SOA species for each mode is calculated. Here, we assume a variable hygroscopicity parameter for each SOA species. Table S2 provides an overview on the hygroscopicity parameters used for each SOA species.

Table S1. Temperature dependent rate constants for particular precursors with OH, O_3, and NO_3. Unless noted, reaction rates are presented by the Arrhenius equation, given as: $\mathrm{K} = \mathrm{A} \times \exp\left(\frac{-\mathrm{E_a}}{\mathrm{RT}}\right)$, where T is temperature in Kelvin (K), A is given in s^{-1}, and $-\mathrm{E_a}/\mathrm{R}$ is given in K.

	OH		O_3		NO_3		
Precursor	A	$-\mathrm{E_a}/\mathrm{R}$	A	$-\mathrm{E_a}/\mathrm{R}$	A	$-\mathrm{E_a}/\mathrm{R}$	Reference
Isoprene	2.7×10^{-11}	390	1.03×10^{-14}	-1995	3.15×10^{-12}	-450	O'Donnell et al. (2011)
α-pinene	1.21×10^{-11}	444	1.01×10^{-15}	-732	3.15×10^{-13}	841	Tsigaridis and Kanakidou (2003)
β-pinene	2.38×10^{-11}	357	1.5×10^{-17}		1.6×10^{-10}	-1248	Tsigaridis and Kanakidou (2003)
Toluene	5.96×10^{-11}		2.34×10^{-12}	-6694	6.8×10^{-17}		Tsigaridis and Kanakidou (2003)
Xylene	1.72×10^{-11}		a	a	3.54×10^{-16}		Tsigaridis and Kanakidou (2003)

[a] Average of temperature dependence for ortho-, meta- and para-isomers:
$(2.4 \times 10^{-13} \times \exp(\frac{-5586}{\mathrm{T}}) + 5.37 \times 10^{-13} \times \exp(\frac{-6039}{\mathrm{T}}) + 1.91 \times 10^{-13} \times \exp(\frac{-5586}{\mathrm{T}}))/3$

Table S2. SOA parameters used for the implemented two-product model. α_1 and α_2 represent the stoichiometric coefficients and $\mathrm{K}_{1,\mathrm{ref}}$ and $\mathrm{K}_{2,\mathrm{ref}}$ represent the reference equilibrium partitioning coefficient, respectively. Values for α_1, α_2, $\mathrm{K}_{1,\mathrm{ref}}$, and $\mathrm{K}_{2,\mathrm{ref}}$ are taken from Zhang et al. (2007) and sources therein. κ_1 and κ_2 represent the hygroscopicity parameter for each SOA product. Hygroscopicity parameter taken from Pye et al. (2017) and sources terein.

Precursor	α_1	α_2	$\mathrm{K}_{1,\mathrm{ref}}[\mathrm{m}^3\mu\mathrm{g}^{-1}]$	$\mathrm{K}_{2,\mathrm{ref}}[\mathrm{m}^3\mu\mathrm{g}^{-1}]$	κ_1	κ_2
Isoprene	0.232	0.0288	0.00459	0.8628	0.14	0.15
α-pinene	0.0864	0.3857	1.1561	0.0847	0.10	0.10
β-pinene	0.0864	0.3857	1.1561	0.0847	0.10	0.10
Toluene	0.071	0.138	0.5829	0.0209	0.15	0.10
Xylene	0.038	0.167	0.4619	0.0154	0.14[a]	0.14[a]

[a] Average for all SOA products presented by Pye et al. (2017).

2 GMXe–AERCHEM and GMXe–SOA manual

MESSy includes a very comprehensive set of chemical reactions in both, the gas-phase and the aqueous-phase. To simplify the usage and enhance the consistency between all mechanisms used for the different phases (gas – MECCA, aqueous phase – SCAV , aerosol phase – GMXe–AERCHEM) the full mechanism is hosted within the MECCA submodel. Before compiling the MESSy code, the user is able to choose the required mechanisms. The selection of the aqueous-phase mechanism used in clouds and deliquescent aerosols needs to be performed separately. In order to make use of both GMXe–AERCHEM and GMXe–SOA the user needs to perform the following steps in the given order before compiling the MESSy code:

1. Select the in-cloud aqueous-phase mechanism and adjust the creation of cloud and aerosol tracer in SCAV
2. Select the aqueous-phase mechanism for deliquescent aerosols in GMXe-AERCHEM
3. Modify and select the gas-phase mechanism in MECCA
4. Adjust the GMXe setup for the usage of GMXe-AERCHEM and GMXe-SOA

2.1 Selecting the in-cloud aqueous-phase mechanism and related adjustments in SCAV

The choice of the in-cloud aqueous-phase mechanism employed is determined by the `xscav` script:

```
cd messy/mbm/scav/mechanism
./xscav
```

The script will prompt various queries with the intention of creating the mechanism. A general description of the creation script is provided in `messy/mbm/caaba/manual/caaba_mecca_manual.pdf`. It is strongly recommended to utilize the precompiled batch files which are outlined within the `batch` directory. For the simulation performed in this manuscript, the `jamoc.bat` was used in order to employ the JAMOC mechanism.

In order to utilize GMXe–AERCHEM, the SCAV submodel needs to make use of cloud tracer and evaporate all cloud species to aerosol tracers upon cloud evaporation. The creation of aerosol tracer in SCAV is enabled in the submodel interface layer. Before compiling the MESSy code, the user needs to ensure that the following logical switches are set to `.TRUE.` in the `messy/smil/messy_scav_si.f90` file:

```
! l_trac_aer: make aerosol tracers for kpp species;  using this switch,
!             but not l_trac_aer_all will create tracers only for charged
!             compounds
l_trac_aer     = .TRUE.
l_trac_aer_all = .TRUE. ! make tracers for all kpp species
```

In order to enable the creation of cloud tracer in SCAV, the user needs to enable the appropriate switch in the SCAV namelist (`scav.nml`):

```
&CPL
[...]
! Create cloud tracer for all KPP species
l_trac_cloud   = T,
[...]
/
```

2.2 Selecting the aqueous-phase mechanism for deliquescent aerosols in GMXe-AERCHEM

The choice of the aqueous-phase mechanism employed for deliquescent aerosols is determined by the `xaerchem` script:

```
cd messy/mbm/gmxe/aerchem/mechanism
./xaerchem
```

The script will prompt various queries with the intention of creating the mechanism. For an explanation of all options, the user may consult `messy/mbm/caaba/manual/caaba_mecca_manual.pdf`. It is recommended to consult the same general description manual It is strongly recommended to utilize the precompiled batch files which are outlined within the `batch` directory. For the simulation performed in this manuscript, the `jamoc.bat` was used in order to employ the JAMOC mechanism.

2.3 Necessary changes to the MECCA submodel and selecting the gas-phase mechanism

In its standard setup, MECCA represents the reactive aerosol uptake of N_2O_5 across all aerosol modes. At the same time, the newly developed GMXe–AERCHEM sub-submodel also represents this uptake for each individual mode for which GMXe–AERCHEM is used. In order to avoid a double accounting of this process, the user needs to disable the uptake of N_2O_5 across all aerosol modes before selecting the gas-phase mechanism by excluding in `messy/mbm/caaba/mecca/gas.eqn`:

```
<HET301>  N2O5  = 2 NO3m_cs + 2 Hp_cs : {%TrHet} khet_Tr(iht_N2O5){$};     {&SGN}
```

and enable the mode specific uptake by including the following in the same file:

```
<HET301a> N2O5  = 2 NO3m_ns + 2 Hp_ns : {%TrHet} khet_Tr(iht_N2O5_ns){$}; {&SGN}
<HET301b> N2O5  = 2 NO3m_ks + 2 Hp_ks : {%TrHet} khet_Tr(iht_N2O5_ks){$}; {&SGN}
<HET301c> N2O5  = 2 NO3m_as + 2 Hp_as : {%TrHet} khet_Tr(iht_N2O5_as){$}; {&SGN}
<HET301d> N2O5  = 2 NO3m_cs + 2 Hp_cs : {%TrHet} khet_Tr(iht_N2O5_cs){$}; {&SGN}
```

In addition, the user needs to disable the calculation of the reactive aerosol uptake for each mode considered in GMXe–AERCHEM by adjusting `mecca.nml`. By default, GMXe–AERCHEM represents non-equilibrium aqueous-phase chemistry for the soluble accumulation (as, mode 3 in GMXe) and coarse (cs, mode 4 in GMXe) mode. Accordingly, `mecca.nml` needs to be adjusted to:

```
&CPL_KHET
[...]
! aerosol submodel and modes:
asm(1) = 'gmxe',  '1,2,5,6,7'
[...]
/
```

If other heterogeneous uptake processes are considered in MECCA, the user is advised to keep `mecca.nml` unchanged and exclude the uptake of N_2O_5 for each GMXe-AERCHEM mode in `messy/mbm/caaba/mecca/gas.eqn`. For a standard GMXe-AERCHEM setup the uptake of N_2O_5 in MECCA would only include:

```
<HET301a> N2O5  = 2 NO3m_ns + 2 Hp_ns : {%TrHet} khet_Tr(iht_N2O5_ns){§}; {&SGN}
<HET301b> N2O5  = 2 NO3m_ks + 2 Hp_ks : {%TrHet} khet_Tr(iht_N2O5_ks){§}; {&SGN}
```

The choice of the gas-phase mechanism employed is determined by the `xmecca` script:

```
cd messy/mbm/caaba/mecca
./xmecca
```

The script will prompt various queries with the intention of creating the mechanism. For an explanation of all options, the user may consult `messy/mbm/caaba/manual/caaba_mecca_manual.pdf`. It is strongly recommended to utilize the precompiled batch files which are outlined within the `ini` directory. For the simulation performed in this manuscript, the `mecca_mom3d.ini` was used in order to employ the MOM mechanism.

2.4 GMXe setup for GMXe–AERCHEM and GMXe–SOA

Adhering to the general methodology employed by MESSy for managing submodels (Jöckel et al., 2006, 2010) all necessary controls to utilize both newly implemented sub-submodels (i.e., GMXe–AERCHEM and GMXe–SOA) are established within the GMXe namelist (`gmxe.nml`). Namelist S1 provides an overview of the recommended GMXe configuration when employing GMXe–AERCHEM and GMXe–SOA. In the following, each setting is described in detail. The supplemental material of Pringle et al. (2010) provides a detailed description of all GMXe options not discussed in this section.

2.4.1 GMXe–AERCHEM sub-submodel options

The GMXe–AERCHEM sub-submodel is enabled by using the appropriate logical switch in the control setup:

```
&CTRL
[...]
l_aerchem  = T, ! calculate aerosol chemistry using a kpp based reaction mechanism
[...]
/
```

In its default setup, all GMXe calculations are performed in the `radiation` routine of MESSy. When using GMXe–AERCHEM, all GMXe calculations need to be performed in the `physc` routine of MESSy to represent the intended operator splitting described in the main manuscript (see Figure 1).

```
&CPL
[...]
driver_call = 'physc'    ! Swich indicated where GMXe_driver is called from
                         !  (radiation or physc)
[...]
/
```

When using GMXe-AERCHEM, all aerosols are assumed to be in a metastable state. This is ensured by using the following setting:

```
&CTRL_GMXE_TD
[...]
LHYSTER    = F, ! Inclusion of hysteresis effect (includes solids/metastable)
[...]
/
```

All additional GMXe–AERCHEM settings are defined in its control instant (`CTRL_GMXE_AERCHEM`):

- `lmode`: Defines the lower mode for which the KPP routine will be executed. The default values is 3 representing the soluble accumulation mode (as).
- `umode`: Defines the upper mode for which the KPP routine will be executed. The default values is 4 representing the soluble coarse mode (cs).

2.4.2 GMXe–SOA sub-submodel options

The GMXe–SOA sub-submodel is enabled by using the appropriate logical switch in the control setup:

```
&CTRL
[...]
l_soa = T,      ! calculate SOA with an explicit SOA model
[...]
/
```

All additional GMXe–SOA settings are defined in its control instant (`CTRL_GMXE_SOA`):

- `lmode_soa`: Defines the lower mode for which the SOA calculations will be executed. The default values is 2 representing the soluble aitken mode (ks).

– umode_soa: Defines the upper mode for which the SOA calculations will be executed. The default values is 5 representing the insoluble aitken mode (ki).

– N_GASPREC(:): Defines the tracer name of each SOA precursor considered in the SOA calculations. GMXe–SOA will create the tracer, if it is not part of the gas-phase mechanism. If the tracer is not part of the gas-phase mechanism, ensure that l_GASPREC(:,2) is set to true (T) for the precursor. In addition, appropriate emissions need to be implemented via the respective submodel.

– l_GASPREC(:,1): Logical switch defining if the SOA calculations will be performed for the specific precursor.

– l_GASPREC(:,2): Logical switch to decide whether total chemical degradation or only degradation due to SOA formation should be considered. If set to T, the total loss is considered.

– l_OXI(:): Logical switch for oxidants (i.e., O_3, OH, and NO_3) to consider in the SOA calculation.

– EXCL_STR_SOA(:): Tracer name of species to be excluded for bulk SOA uptake. By default, only mineral dust (DU) is excluded.

Namelist S1. Example GMXe setup when using GMXe-AERCHEM

```
! -*- f90 -*-
!
&CTRL
                    ! <--- GMXe general options
                    !
  LGMXe      = T, ! Aerosol composition scheme GMXe (interface)
  LOUTPUT    = T, ! Write GMXe channel to diagnose output
  LMASS_DIAG = F, ! Mass balance check in gmXe_physc
  LSTRAT     = F, ! Exclude stratosphere (requires TROPOP channel)
  LPSC       = T, ! Exclude PSC region   (requires PSC channel)

  LNUCL      = T, ! Calculate nucleation of aerosol particles

  LCOND      = T, ! Calculate amount of condensable species
                  ! due to kinetic limitations otherwise distribute equally across modes
  LCOAG      = T, ! Reshape the size distribution due to particle
                  ! coagulation (if LSIZE=.F.)

  LCOAT      = T, ! Calculate coating/ageing of primary particles
                  ! (requires THERMO interface (NEQM > -1))

  LAH2O      = T, ! Calculate aerosol water concentration limited
                  ! by specific humidity (otherwise equilibrium is assumed)
  LGH2O      = F, ! Update water vapor concentration (H2O tracer only
                  ! if defined in H2O tracer coupling) !currently not used!
  LSHUM      = F, ! Update specific humidity (ECHAM5)

  LSIZE      = F, ! Use only mass to determine numbers (originates from EQSAM3)

! switches that deal with aerosol cloud coupling
! (to be (re)moved to the cloud submodel)
  LACC       = F, ! Aerosol-cloud-coupling (diag only, if not LCLC,LCLWC,LCIWC,LCDNC,LICNC)
  LCLC       = F, ! Update cloud cover
  LADYN      = F, ! Calculate aerosol dynamics (limits EQSAM3 gas/liquid/solid equilibration time by
      transport time scales)
  LCLWC      = F, ! Update cloud liquid water concentration (sets LCDNC = T)
  LCIWC      = F, ! Update cloud ice water concentration    (sets LICNC = T)
  LCDNC      = F, ! Update cloud droplet number concentration (CLWC calc. by ECHAM5)
  LICNC      = F, ! Update ice crystal number concentration   (CIWC calc. by ECHAM5)
```

```
42: LGAS      = T, ! Update gas phase chemistry     - Gases  : H3PO4, H2SO4, HNO3, HCl, HBr, HI, H2CO3
      , H2O, CH2O2, C2H4O2,
43:                !                                            C2H2O4, C6H8O7, Hg, NH3
44: LAEROSOL  = T, ! Update aerosol phase chemistry - Cations: H+, NH4+, Na+, K+, Mg++, Ca++, Fe++/Fe
      +++
45:                !                                - Anions : PO4---, SO4--, HSO4-, NO3, Cl-, Br-, I
      -, CO3--, HCO3-, OH-, CHO2-,
46:                !                                            C2H3O2-, C2O4--, C6H5O7---
47:                !                                - Solutes: NH3, C3H6O, CH3OH, C2H6O, C6H12O6,
      C6H13O5, C12H22O11,
48:                !                                            Hg, Pb, C6H10O5, C4H6O4, C33H32O19
49:                !                                - Gases  : H3PO4, H2SO4, HNO3, HCl, HBr, HI, H2CO3
      , H2O, CH2O2, C2H4O2,
50:                !                                            C2H2O4, C6H8O7, Hg, NH3
51:                !                              Note - Within EQSAM3 all major cation/anion
      combinations as well as gases
52:                !                                   and (neutral) solutes are considered for
      the aerosol water uptake;
53:                !                                   All these compounds, but especially the
      salts are important for the
54:                !                                   aerosol water mass and hyrgoscopic growth
      due their different RH regimes,
55:                !                                   where they are able to absorb or desorb
      water vapor from the atmosphere
56:                !                                   (deliquescence/efflorescence relative
      humidity, respectively).
57:                !                                - Resolving the hyrgoscopic growth is
      important, as this determines the
58:                !                                   ambient particle size, life-time of
      aerosols and trace gasses, as well as
59:                !                                   the interaction of the aerosols with
      sunlight (direct forcing) and clouds
60:                !                                   (indirect forcing).
61:                ! <--- GMXe channel
62:                !
63: LNUMBER   = F, ! T = Calculate aerosol numbers from aerosol mass; the GMXe channel contains a
      channel element 'anumber'
64:                ! F = Transport aerosol numbers as tracers (initial aerosol numbers are assigned
      from emissions, etc)
65: LWETRAD   = T, ! T = Calculate wet aerosol radius  from EQSAM3 thermodynamics; F from bulk
      thermodynamics
66: LDRYRAD   = T, ! T = Calculate dry aerosol radius  from EQSAM3 thermodynamics; F from bulk
      thermodynamics
```

```
67:   LDRYDENS    = T, ! T = Calculate dry aerosol density from EQSAM3 thermodynamics; F from bulk
          thermodynamics
68:   LEVAP       = T, ! Calculate H2SO4 evaporation (default T)
69:   LCALCPM     = T, ! Calculate Particulate Matter PM1/2.5/10 (default F)
70:   LCALCFPM    = T, ! Calculate fraction of each mode in Particulate Matter PM1/2.5/10 (default F,
          requires LCALCPM = T)
71:                    !
72:                    ! Calculate aerosol composition and hygroscopic growth for selected modes:
73:                    !
74:   NLOWERMODE = 1   ! Lowest mode (min. mode number = 1; nucleation mode,    soluble)
75:   NUPPERMODE = 7   ! Uppest mode (max. mode number = 7; coarse      mode, insoluble)
76:                    !
77:                    ! <--- Choice of aerosol composition module
78:                    !
79:   NEQM       = 2     ! Thermodynamic module: -1, 0 = None (bulk hygroscopic growth only)
80:                      !                        1= EQSAM4CLIM
81:                      !                        2= ISOROPIA2
82:                      !
83:                      ! <--- Choice of nucleation scheme
84:                      !
85:   NNUCL      = 1     !  0 = test, 1 = Vehkamaeki (2002), 2 = Kulmala (1998)
86:                      ! (0 requires EQSAM3 interface, neqm >= 0)
87:                      !
88:
89:   l_aerchem  = T, ! calculate aerosol chemistry using a kpp based reaction mechanism
90:   l_oracle   = F, ! calculate organic aerosols using oracle
91:   l_oc_aging = F, ! calculate aging of organic carbon
92:
93:   l_soa = T,      ! calculate SOA with an explicit SOA model
94:
95:   l_passive_aer = F ! use passive aerosol tracers
96: /
97:
98: &CTRL_GMXE_DISC
99:   nmod        = 7  ! number of modes
100:
101:                  ! string for sigma must contain the width of each mode, separated by ";"
102:                  ! If the number of entries does not match nmod, the simulation will be terminated
103:   sigma_nml   = '1.59;1.59;1.59;2.0;1.59;1.59;2.0'
104:                  ! string for crdiv must contain the lower boundary of each mode in [cm],
105:                  ! separated by ";"
106:                  ! If the number of entries does not match nmod, the simulation will be terminated
107:   crdiv_nml = '0.0005e-4;0.006e-4;0.06e-4;0.7e-4;0.006e-4;0.06e-4;0.7e-4'
108:                  ! string for cmodes must contain the names for each mode, separated by ";"
```

```
109:                     ! If the number of entries does not match nmod, the simulation will be terminated
110:                     ! Note, that many submodels coupling to aerosol species use small letters...
111:   cmodes_nml = 'ns;ks;as;cs;ki;ai;ci'
112: /
113:
114: &CTRL_GMXE_SPECIES
115:                     ! enter all gas phase species which should be considered in the thermodynamics
116:                     ! the first string is for the name - CASE SENSITIVE
117:                     ! the second string is for the modes, in which this species should occur
118:                     ! the second string should contain the numbers of the modes, separated by ";"
119:   CASK_GASES(1,:) = "HNO3", "0;1;2;3;4;5;6;7"
120:   CASK_GASES(2,:) = "H2SO4","0;1;2;3;4;5;6;7"
121:   CASK_GASES(3,:) = "HCl",  "0;1;2;3;4;5;6;7"
122:   CASK_GASES(4,:) = "NH3",  "0;1;2;3;4;5;6;7"
123:                     ! enter all cation species which should be considered in the thermodynamics
124:                     ! the first string is for the name - CASE SENSITIVE
125:                     ! the second string is for the modes, in which this species should occur
126:                     ! the second string should contain the numbers of the modes, separated by ";"
127:   CASK_CATIONS(1,:) = "Hp",  "1;2;3;4;5;6;7"
128:   CASK_CATIONS(2,:) = "NH4p","1;2;3;4;5;6;7"
129:   CASK_CATIONS(3,:) = "Nap", "1;2;3;4;5;6;7"
130:   CASK_CATIONS(4,:) = "Kp",  "1;2;3;4;5;6;7"
131:   CASK_CATIONS(5,:) = "Capp","1;2;3;4;5;6;7"
132:   CASK_CATIONS(6,:) = "Mgpp","1;2;3;4;5;6;7"
133:                     ! enter all anion pecies which should be considered in the thermodynamics
134:                     ! the first string is for the name - CASE SENSITIVE
135:                     ! the second string is for the modes, in which this species should occur
136:                     ! the second string should contain the numbers of the modes, separated by ";"
137:   CASK_ANIONS(1,:) = "SO4mm","1;2;3;4;5;6;7"
138:   CASK_ANIONS(2,:) = "HSO4m","1;2;3;4;5;6;7"
139:   CASK_ANIONS(3,:) = "OHm",  "1;2;3;4;5;6;7"
140:   CASK_ANIONS(4,:) = "Clm",  "1;2;3;4;5;6;7"
141:   CASK_ANIONS(5,:) = "NO3m", "1;2;3;4;5;6;7"
142:                     ! enter all solute species which should be considered in the thermodynamics
143:                     ! the first string is for the name - CASE SENSITIVE
144:                     ! the second string is for the modes, in which this species should occur
145:                     ! the second string should contain the numbers of the modes, separated by ";"
146:   CASK_SOLUTES(1,:) = "H2O","0;1;2;3;4;5;6;7"
147:                     ! enter all bulk species which should be considered in the thermodynamics
148:                     ! the first string is for the name - CASE SENSITIVE
149:                     ! the second string is for the modes, in which this species should occur
150:                     ! the second string should contain the numbers of the modes, separated by ";"
151:   CASK_BULK(1,:) = "SS", "2;3;4"
152:   CASK_BULK(2,:) = "DU", "3;4;6;7"
```

```
153:   CASK_BULK(3,:) = "OC", "2;3;4;5"
154:   CASK_BULK(4,:) = "BC", "2;3;4;5"
155: /
156:
157: &CTRL_GMXE_TD
158:                    ! <--- Thermodynamic specific options
159:                    !
160:   LDRY       = F, ! Force aerosol particles to be dry (no aerosol water)
161:   LHYSTER    = F, ! Inclusion of hysteresis effect (includes solids/metastable)
162: /
163:
164: &CTRL_GMXE_AERCHEM
165:                    ! namelist for aerchem subsubmodel
166:   lmode = 3        ! lower mode boundary for aerchem
167:   umode = 4        ! upper mode boundary for aerchem
168: /
169:
170: &CTRL_GMXE_SOA
171:                    ! namelist for SOA subsubmodel
172:   lmode_soa = 2    ! lower mode boundary for SOA
173:   umode_soa = 5    ! upper mode boundary for SOA
174:
175: ! tracer name for each SOA precursor
176: !                  Isoprene, Toluene, Xylene, Cresol, Alkenes, total terpenes, a-Pinene, b-Pinene,
177:   N_GASPREC(:)   = 'Isop',  'Tol',  'Xyl',  'Cre',  'Alk',  'Terp',          'aPin',  'bPin'
178:
179: ! logical switch for species to consider
180: !                  Isoprene, Toluene, Xylene, Cresol, Alkenes, total terpenes, a-Pinene, b-Pinene
181: !                     total terpenes are true if a-Pinene or b-Pinene are true
182:   l_GASPREC(:,1) =    T ,     T,      T,      T,      T,      T,              T,       T
183:
184: ! logical switch to decide whether total chemical degradation or
185: ! only degradation due to SOA formation should be considered
186: ! T means that total loss is considered
187: !                  Isoprene, Toluene, Xylene, Cresol, Alkenes, total terpenes, a-Pinene, b-Pinene,
188:   l_GASPREC(:,2) =    T,      T,      T,      T,      T,      T,              T,       T
189:
190: ! logical switch for oxidants to consider
191: !                  O3, OH, NO3
192:   l_OXI(:)       =  T , T,  T
193:
194: ! string with potential names of species excluded for bulk SOA uptake
195: !   maximum number of species at the moment is 10
196:   EXCL_STR_SOA(:)= 'DU','','','','','','','','',''
```

```
/

&CTRL_GMXE_PASSIVE
! namelist for passive aerosol tracers
! NOTES: If passive aerosols are turned on here, then consider the following:
!        - SCAV: if it should be possible to scavenge them, then the
!          full name has to be included in the out_string_aer in scav.nml;
!        - AEROPT: if it is turned on and the passive tracers should
!          NOT contribute to the AOT, then exclude them in aeropt.nml;
!        - emissions: so far only offlem emissions, see offlem46.nml
!          and emis_PA46.nml
!        - tracers are initialized in messy_gmxe_e5.nml

   num_pa = 1       ! number of passive aerosol tracers
                    ! modes for tracer initialization:
                    ! (1-4: hydrophilic, 5-7: hydrophobic)
   pamode1 = 2      ! lower mode boundary for passive aerosol
   pamode2 = 4      ! upper mode boundary for passive aerosol
/

&CPL
   l_calc_emis = T ! calculate direct emission of compounds within gmxe
! 1.1   add emission to tendency (TRUE) or in xtems (FALSE)
!       note that 3D or X-NLEV2D are always added in the tendency
   l_tendency = T
!
! <--- 1. Choice of tropopause channel
!
   Tropopchannel = 'tropop' ! channel containing tropopause index
   TropopIndex   = 'tp_i'   ! tropopause index channel element
!
! <--- 2. Choice of PSC channel
!
   Pscchannel = 'msbm'       ! channel containing PSC mask
   Pscreg     = 'PSC_region' ! mask for psc region
   phase      = 'phase'      ! phase of psc (to be used as a mask)

   driver_call = 'physc'    ! Swich indicated where GMXe_driver is called from
                            !  (radiation or physc)

! emission setup (new way)
! for each emission flux one EMIS_CASK should be filled
! all possible fluxes are listed in the messy_gmxe_e5.f90
! in the subroutine gmxe_emis_e5_init
```

```
241: ! certain characteristics of the fluxes are defined there as well,
242: ! e.g. if only a fraction of the total flux should be used for
243: ! another flux (e.g. SO2 emissions, 2.5% are used for particulate sulfate emissions)
244: ! emis_casks - array:
245: !                     1st entry: name (used for identification in the
246: !                                list of the e5 file section emis_init)
247: !                     2nd entry: total scaling factor for incoming flux
248: !                     3rd entry: channel name of the emission flux
249: !                     4th entry: name of the mass emission flux object
250: !                     5th entry: name of the corresponding number
251: !                                emission flux object (If it does not exist a
252: !                                number is calculated from the mass)
253: !                     6th entry: ";" separated list of tracers which should
254: !                                receive emissions from this flux
255: !                     7th entry: ";" separated list of fractions of this emission
256: !                                flux for all the tracers defined in entry 6
257: ! Theoretically, one can define an emission flux several time for several tracers.
258: ! BUT, BE AWARE, that is means that the number flux is also added several times, which
259: !                is most likely not desired
260: ! WARNING: In case the 1st entry from a cask is not matching any of the fluxes in the list
261: !          it is ignored
262:
263: !------------------
264: ! SEASALT
265: !------------------
266: ! AEROCOM
267: !------------------
268: EMIS_CASK(1,:) = "seasalt_mass_cs","1.46",     "onemis","mss_cs_lsce" ,"","Nap_cs;Mgpp_cs;Capp_cs;
        Kp_cs;Clm_cs;SS_cs;SO4mm_cs","0.306;0.037;0.012;0.011;0.550;0.007;0.077"
269: EMIS_CASK(2,:) = "seasalt_mass_as","1.458905", "onemis","mss_as_lsce" ,"","Nap_as;Mgpp_as;Capp_as;
        Kp_as;Clm_as;SS_as;SO4mm_as","0.306;0.037;0.012;0.011;0.550;0.007;0.077"
270: EMIS_CASK(3,:) = "seasalt_mass_ks","0.001095", "onemis","mss_as_lsce" ,"","Nap_ks;Mgpp_ks;Capp_ks;
        Kp_ks;Clm_ks;SS_ks;SO4mm_ks","0.306;0.037;0.012;0.011;0.550;0.007;0.077"
271:
272: !------------------
273: ! SULPHATE
274: !------------------
275: ! 0.25% iof emissions divided equally to KS and AS mode
276: EMIS_CASK(4,:) = "so2_mlc_ks",   "0.0125", "offemis",    "EDGAR_ANTH_SO2_flux","", "SO4mm_ks;Hp_ks","
        1.0;2.0"
277: EMIS_CASK(5,:) = "so2_mlc_as",   "0.0125", "offemis",    "EDGAR_ANTH_SO2_flux","", "SO4mm_ks;Hp_ks","
        1.0;2.0"
278: EMIS_CASK(6,:) = "so2_mlc_ks",   "0.0125", "offemis",    "SO2_flux",           "", "SO4mm_ks;Hp_ks","
        1.0;2.0"
```

```
279: EMIS_CASK(7,:) = "so2_mlc_as",    "0.0125", "offemis",     "SO2_flux",            "", "SO4mm_ks;Hp_ks","
        1.0;2.0"
280: EMIS_CASK(8,:) = "so2_mlc_ks",    "0.0125", "import_grid","VOLCANIC_SO2_SO2",    "", "SO4mm_ks;Hp_ks","
        1.0;2.0"
281: EMIS_CASK(9,:) = "so2_mlc_as",    "0.0125", "import_grid","VOLCANIC_SO2_SO2",    "", "SO4mm_as;Hp_as","
        1.0;2.0"
282:
283: !------------------
284: ! ORGANIC CARBON
285: !------------------
286: ! Scaling factor contains water soluble fraction and conversion om2oc
287: ! ANTHROPOGENIC
288: EMIS_CASK(22,:)= "oc_mass_ff_ks",  "0.91",     "offemis",      "EDGAR_ANTH_OC_flux",  ""  ,"OC_ks","
        1.0"
289: EMIS_CASK(23,:)= "oc_mass_ff_ki",  "0.49",     "offemis",      "EDGAR_ANTH_OC_flux",  ""  ,"OC_ki","
        1.0"
290: EMIS_CASK(24,:)= "oc_mass_ff_ks",  "1.0868",   "import_grid",  "EDGAR_AIR_OC_flux",   "",  "OC_ks","
        1.0"
291: EMIS_CASK(25,:)= "oc_mass_ff_ki",  "0.2032",   "import_grid",  "EDGAR_AIR_OC_flux",   "",  "OC_ki","
        1.0"
292: ! BIOMASS BURNING
293: EMIS_CASK(26,:)= "oc_mass_bb_ks",  "0.59728",  "offemis",      "OC_flux",             "",  "OC_ks","
        1.0"
294: EMIS_CASK(27,:)= "oc_mass_bb_ki",  "1.00272",  "offemis",      "OC_flux",             "",  "OC_ki","
        1.0"
295:
296: !------------------
297: ! BLACK CARBON
298: !-----------------
299: EMIS_CASK(28,:)= "bc_mlc_3d_ki",   "1.0", "offemis",      "EDGAR_ANTH_BC_flux" ,""   ,"BC_ki","1.0"
300: EMIS_CASK(29,:)= "bc_mlc_3d_ki",   "1.0", "import_grid", "EDGAR_AIR_BC_flux"  ,""   ,"BC_ki","1.0"
301: EMIS_CASK(30,:)= "bc_mass_3d_ki",  "1.0", "offemis",      "BC_flux"            ,""   ,"BC_ki","1.0"
302:
303: !------------------
304: ! DUST
305: !------------------
306: !KKDU-Asthita resolution dependent
307: ! T106 = 1.0
308: ! T63  = 0.8
309: ! T42  = 0.6
310: EMIS_CASK(40,:)= "dust_mass_ci",    "1.00", "onemis", "kkdu_misc_emflux_ci", "", "DU_ci",   "0.60"
311: EMIS_CASK(41,:)= "dust_mass_ai",    "1.00", "onemis", "kkdu_misc_emflux_ai", "", "DU_ai",   "0.60"
312: /
```

References

Jöckel, P., Tost, H., Pozzer, A., Brühl, C., Buchholz, J., Ganzeveld, L., Hoor, P., Kerkweg, A., Lawrence, M. G., Sander, R., Steil, B., Stiller, G., Tanarhte, M., Taraborrelli, D., van Aardenne, J., and Lelieveld, J.: The atmospheric chemistry general circulation model ECHAM5/MESSy1: consistent simulation of ozone from the surface to the mesosphere, Atmospheric Chemistry and Physics, 6, 5067–5104, https://doi.org/10.5194/acp-6-5067-2006, publisher: Copernicus GmbH, 2006.

Jöckel, P., Kerkweg, A., Pozzer, A., Sander, R., Tost, H., Riede, H., Baumgaertner, A., Gromov, S., and Kern, B.: Development cycle 2 of the Modular Earth Submodel System (MESSy2), Geoscientific Model Development, 3, 717–752, https://doi.org/10.5194/gmd-3-717-2010, publisher: Copernicus GmbH, 2010.

O'Donnell, D., Tsigaridis, K., and Feichter, J.: Estimating the direct and indirect effects of secondary organic aerosols using ECHAM5-HAM, Atmospheric Chemistry and Physics, 11, 8635–8659, https://doi.org/10.5194/acp-11-8635-2011, publisher: Copernicus GmbH, 2011.

Odum, J. R., Hoffmann, T., Bowman, F., Collins, D., Flagan, R. C., and Seinfeld, J. H.: Gas/Particle Partitioning and Secondary Organic Aerosol Yields, Environmental Science & Technology, 30, 2580–2585, https://doi.org/10.1021/es950943+, publisher: American Chemical Society, 1996.

Pringle, K. J., Tost, H., Message, S., Steil, B., Giannadaki, D., Nenes, A., Fountoukis, C., Stier, P., Vignati, E., and Lelieveld, J.: Description and evaluation of GMXe: a new aerosol submodel for global simulations (v1), Geoscientific Model Development, 3, 391–412, https://doi.org/10.5194/gmd-3-391-2010, publisher: Copernicus GmbH, 2010.

Pye, H. O. T.: The Acidity of Atmospheric Particles and Clouds, https://catalog.data.gov/dataset/the-acidity-of-atmospheric-particles-and-clouds, 2020.

Pye, H. O. T., Murphy, B. N., Xu, L., Ng, N. L., Carlton, A. G., Guo, H., Weber, R., Vasilakos, P., Appel, K. W., Budisulistiorini, S. H., Surratt, J. D., Nenes, A., Hu, W., Jimenez, J. L., Isaacman-VanWertz, G., Misztal, P. K., and Goldstein, A. H.: On the implications of aerosol liquid water and phase separation for organic aerosol mass, Atmospheric Chemistry and Physics, 17, 343–369, https://doi.org/10.5194/acp-17-343-2017, publisher: Copernicus GmbH, 2017.

Tsigaridis, K. and Kanakidou, M.: Global modelling of secondary organic aerosol in the troposphere: a sensitivity analysis, Atmospheric Chemistry and Physics, 3, 1849–1869, https://doi.org/10.5194/acp-3-1849-2003, publisher: Copernicus GmbH, 2003.

Zhang, Y., Huang, J.-P., Henze, D. K., and Seinfeld, J. H.: Role of isoprene in secondary organic aerosol formation on a regional scale, Journal of Geophysical Research: Atmospheres, 112, https://doi.org/10.1029/2007JD008675, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2007JD008675, 2007.