

Technical Note: Coupling of chemical processes with the Modular Earth Submodel System (MESSy) submodel TRACER

P. Jöckel, A. Kerckweg, J. Buchholz-Dietsch, H. Tost, R. Sander, and A. Pozzer

Max-Planck Institute of Chemistry, Air Chemistry Department, P.O. Box 3060, 55020 Mainz, Germany

Received: 25 October 2007 – Published in Atmos. Chem. Phys. Discuss.: 23 November 2007

Revised: 13 February 2008 – Accepted: 15 February 2008 – Published: 19 March 2008

Abstract. The implementation of processes related to chemistry into Earth System Models and their coupling within such systems requires the consistent description of the chemical species involved. We provide a tool (written in Fortran95) to structure and manage information about constituents, hereinafter referred to as tracers, namely the Modular Earth Submodel System (MESSy) generic (i.e., infrastructure) submodel TRACER. With TRACER it is possible to define a multitude of tracer sets, depending on the spatio-temporal representation (i.e., the grid structure) of the model. The required information about a specific chemical species is split into the static meta-information about the characteristics of the species, and its (generally in time and space variable) abundance in the corresponding representation. TRACER moreover includes two submodels. One is TRACER_FAMILY, an implementation of the tracer family concept. It distinguishes between two types: type-1 families are usually applied to handle strongly related tracers (e.g., fast equilibrating species) for a specific process (e.g., advection). In contrast to this, type-2 families are applied for tagging techniques. Tagging means the artificial decomposition of one or more species into parts, which are additionally labelled (e.g., by the region of their primary emission) and then processed as the species itself. The type-2 family concept is designed to conserve the linear relationship between the family and its members. The second submodel is TRACER_PDEF, which corrects and budgets numerical negative overshoots that arise in many process implementations due to the numerical limitations (e.g., rounding errors). The submodel therefore guarantees the positive definiteness of the tracers and stabilises the integration scheme. As a by-product, it further provides a global tracer mass diagnostic. Last but not least, we present the submodel PTRAC, which

allows the definition of tracers via a Fortran95 namelist, as a complement to the standard tracer definition by application of the TRACER interface routines in the code. TRACER with its submodels and PTRAC can readily be applied to a variety of models without further requirements. The code and a documentation are included in the electronic supplement.

1 Introduction

The Earth climate system is characterised by the cycling of constituents through different domains, such as the carbon, nitrogen, and sulphur cycles. This cycling results in a chemical coupling, which is controlled by a large variety of processes on a wide range of spatial and temporal scales. For the correct representation of this chemical coupling in Earth System Models (ESMs), a consistent treatment of all involved chemical species in the specific processes is required. For instance, the fate of a given species in the atmosphere needs to be tracked from its primary emissions, through the chemical transformations (reactions with other species in the gas, aqueous and aerosol phases), transport through the atmosphere (e.g., advection, convection, diffusion, mixing), up to the removal from the atmosphere, i.e., transition into another domain, e.g., by deposition to the surface (land or ocean) or uptake by the biosphere.

For the implementation of this chemical coupling into ESMs, it is desirable to have a flexible framework for a straightforward and easy expansion by additional constituents (or constituent characteristics). This allows the creation of tailor-made model setups to address specific scientific questions. If the access to the tracer information is standardised for all processes, a high degree of consistency is reached and the code is very transparent for the user. This flexibility and user-friendliness can be achieved without a significant loss of computational performance, since the



Correspondence to: P. Jöckel
(joeckel@mpch-mainz.mpg.de)

model setup including the chemical coupling is determined and fixed during the initialisation phase of the model simulation and does not slow down the time integration.

Here, we provide a comprehensive framework for the implementation of chemical species into ESMs, the Modular Earth Submodel System (MESSy) generic submodel TRACER. The predicate “generic” implies that the submodel provides a part of the overall MESSy infrastructure and is itself structured as a MESSy submodel (Jöckel et al., 2005). The electronic supplement (<http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>) of this technical note contains the complete source code, a simple example application code, and a documentation of the interface routines. The code is written in standard Fortran95 (ISO/IEC 1539-1). We stress that this generic submodel TRACER can be used for other ESMs, general circulation models, chemistry transport models etc., without requiring the complete MESSy implementation, in a similar way as the generic submodel NCREGRID (Jöckel, 2006).

In this technical note we describe the basic ideas behind and the overall implementation of TRACER (Sect. 2). TRACER contains two optional submodels: TRACER_FAMILY (Sect. 3) provides a namelist-controllable implementation of two different tracer family concepts. TRACER_PDEF (Sect. 4) guarantees the positive definiteness of the chemical species by correcting and balancing numerically produced negative values and provides a global tracer mass diagnostic. Furthermore, the MESSy submodel PTRAC, which is also included in the electronic supplement (<http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>), allows the definition of tracers by a Fortran95 namelist (Sect. 5).

Throughout this technical note “tracer” is used in a conceptual manner as a synonym for all kinds of constituents, i.e., reactive chemical species, long-lived species, substances of different phases (e.g., water vapour and liquid water), etc.

2 TRACER

The information about a specific chemical species (a *tracer*) is divided into two parts: the *data*, which describe the abundance of the species at a given location (and time), and the *meta-information*, which describe the characteristics of the species that are independent of space and/or time (see Sect. 2.1 below). The data part requires an advanced memory management, since the structure depends on the underlying *representation* of the location information in the model. Examples of *representations* are 3-dimensional grids (grid-point representation, e.g., latitude, longitude, level), spectral coefficients (spectral representation), or the positions of trajectories (Lagrangian representation). Furthermore, the representation may vary with the domain, e.g., the grids applied in ocean models are usually different from the grids applied in atmospheric models. Therefore, the top-level entity pro-

vided by TRACER is the *tracer set* which contains all tracers of a specific representation. Different co-existing tracer sets can be defined within the model.

2.1 The meta-information structure

The meta-information of the tracers within a tracer set is stored in a concatenated list of nested Fortran95 structures. The stored information is grouped into the following top-level structures:

- *t_ident* contains the information for the identification of a given species.
- *t_meta* contains values, flags and switches related to specific processes.

The identification of the tracer (*t_ident*) consists of

- a unique *fullname* composed of a *basename* and an additional optional *subname* (e.g., for tagging, isotopic information, phase information, etc.),
- the name of the *submodel* that defined the tracer in the tracer set,
- a unique *index*, which is the number of the tracer in the tracer set that can also be used to address the corresponding data (see Sect. 2.2),
- the *medium* of the tracer (e.g., air, aerosol, cloud, ocean, etc.),
- the *quantity* describing the abundance of the tracer (amountfraction, numberdensity, or concentration), and its *unit*,
- the *type* of the tracer (e.g., single tracer, family, isotopologue).

The additional information in *t_meta* comprises containers for integer, real and string values, respectively (one Fortran95 array of rank 1 for each type). These containers can be associated with specific information about the species (e.g., the molar mass), and/or switches for the corresponding processes (e.g., advection on or off). Additional containers can readily be added and used in those processes where they are required without any interference with already implemented processes/characteristics. More information on the meta-information structure can be found in the electronic supplement (<http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>).

TRACER provides high-level interface routines to define new tracers, to set the tracer meta-information, and to receive information from the meta-information structure (see electronic supplement <http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>).

2.2 The data structure

The data representation of a tracer set can use up to three array ranks (e.g., for 3 spatial dimensions). Every tracer set can provide an arbitrary number (1 or larger; only limited by the available memory) of *data-instances*, e.g., for the different stages of a time-integration scheme. The first three instances can optionally convey the specific meaning of the data at time step t , the tracer tendency, and the data at time step $t - \Delta t$, respectively. All data-instances of one tracer set are internally stored in a contiguous array of rank 6, where

- rank 1, 2 and 4 are the tracer set specific representation dimensions,
- rank 3 is the number of the tracer in the tracer set (*tracer index*),
- rank 5 is of length 1,
- rank 6 is the index of the data-instance in the tracer set.

This implementation of the data as a contiguous array and the chosen distribution of the ranks allows an elegant and efficient addressing of meaningful sub-arrays through Fortran95 pointers.

The meta-information and data structure of a tracer set with 2 tracers is sketched in Fig. 1.

2.3 The setup procedure

Since the model setup including the number of tracers in a specific tracer set is determined at runtime and the chemical information, especially the data, is quite memory consuming, a dynamical memory allocation is desirable. For the sake of an optimal runtime performance, however, storage of all tracers in a tracer set in a contiguous array (for optimal memory access in loops) is also desirable. A hard-coded pre-set maximum number of tracers per tracer set is either a waste of memory, or requires recompilation if a change of the setup changes the number of tracers. In order to avoid both problems, the tracer setup is performed during the *initialisation phase* of the model simulation in the following steps:

1. Definition of tracer sets: The tracer sets are defined by the base model interface layer (BMIL; for definition see Jöckel et al., 2005; see also Fig. 3); each tracer set is identified by a unique name.
2. Tracer definition: The submodels (in the submodel interface layer (SMIL); details in Jöckel et al., 2005) define their tracers within the tracer sets. At this stage only the meta-information is stored.
3. Setup of tracer sets: The tracer sets are fixed (locked) by the BMIL; after this, no more tracers can be added. The memory of each tracer set is allocated according to the number of tracers in the tracer set and the associated

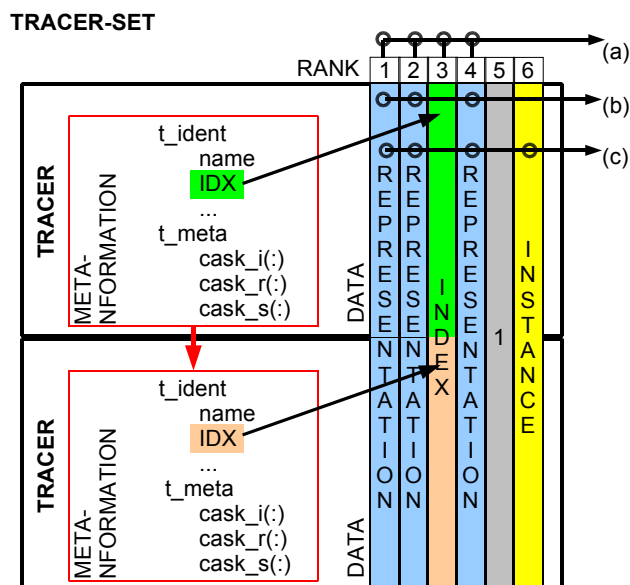


Fig. 1. Diagram showing the context of 2 exemplary tracers (with meta-information and data) stored in a tracer set. The meta-information is contained in a concatenated list (indicated by the red connector) of nested Fortran95 structures. The unique tracer index (IDX) can be used to directly access the data corresponding to the tracer. The contiguous data array allows the straightforward pointer based access to the tracer memory as indicated by 3 examples; the circles indicate the full range along the respective rank; ranks without circles are fixed by a specific index: (a) rank-4 pointer to one instance of all tracers in a tracer set; (b) rank-3 pointer to one instance of one tracer; (c) rank-4 pointer to all instances of one tracer. Rank 5 is currently unused and intended for future use, e.g., for data representations which require 4 indices.

representation. The required pointers to the sub-arrays (e.g., to the various data instances) are set. In addition, the concatenated lists of meta-information structures are transformed into arrays, to allow an efficient access via the tracer index (e.g., for usage within loops).

4. Chemical coupling: The submodels (in the SMIL) can receive information (e.g., availability, index, molar mass, etc.) about specific tracers and set local pointers to the corresponding data.

After the initialisation phase, the chemical coupling between the submodels does not change technically (i.e., w.r.t. the memory/pointer assignment) and the tracer data can be accessed efficiently during the *time integration phase* of the model simulation. In the *finalising phase* of the model simulation data and meta-information of all tracer sets are removed from the memory by the BMIL. TRACER provides the interface routines for these operations (<http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>).

3 TRACER_FAMILY

The application of tracer families is a well-established approach to overcome numerical issues and unrealistic inconsistencies arising from calculating coupled processes with the common operator splitting technique. For instance, if changing boundary conditions (e.g., sunrise) cause the kinetic operator to establish a sudden disequilibrium among otherwise well equilibrated, related species (i.e., which are exchanged by rapid reactions) within a model time step Δt , large temporal and spatial gradients of short-lived compounds result. A subsequent application of the advection operator results in an unrealistic transport of short-lived species and therefore in an unrealistic initial condition for the kinetic operator in the next time step. In reality these two processes act simultaneously and during the transport the equilibrium between the related species is re-established. One possible solution is to combine the related (fast equilibrating) species into a *tracer family* and to transport only this family keeping the ratio among the family member mixing ratios constant during the model time step Δt (e.g., Sheela et al., 2005; Steil et al., 2003; Carver and Stott, 2000).

The same family concept can also be applied to reduce the stiffness of chemical reaction sets (e.g., Wayne, 2000, pp. 147/148, Austin et al., 2003). If fast chemical reactions lead to a steady state condition among a subset of tracers x_1, x_2, \dots, x_n , these chemical reactions need not be explicitly calculated. Their effect can be imitated by setting the ratios within the tracer family constant:

$$\frac{x_i}{x_j} = \text{const.} \quad \forall i, j \in \{1, 2, \dots, n\} . \quad (1)$$

Besides this concept (further denoted as *type-1 family* concept, Sect. 3.1), another family concept is used to overcome unrealistic slight non-linearities, for instance in *tagging-techniques*, where additional characteristics, not necessarily measurable in reality, are assigned to the tracers. A common example is the tagging of specific species with the region of their primary emissions, or with an isotopic signature. Another example is the joined transport of different aerosol components of one mode or bin. Advection in reality is linear, i.e., the result of advecting a tracer x gives the same result as arbitrarily splitting x into parts (x_1, x_2, \dots, x_n) and advecting these separately. Depending on the implementation and due to numerical limitations, the advection operator in the model might not be linear. As a result the two solutions (advecting x versus advecting the tagged components) will diverge during the integration. One solution is to force the linearity in each model time step by adjusting the values of x_1, x_2, \dots, x_n such that

$$\sum_{i=1}^n x_i = x \quad (2)$$

by concurrently keeping the ratios between the family member mixing ratios. This approach requires the processing of

the original tracer x in addition to its parts x_1, x_2, \dots, x_n and is further denoted as *type-2 family* concept (Sect. 3.2). The MESSy sub-submodel TRACER_FAMILY provides the infrastructure for both concepts.

We will denote the standard mode, in which each tracer (data and meta-information) represents a specific chemical species, as *tracer mode*. In contrast to this, the mode where two or more species are combined into one or more families is called *family mode*. The implementation of TRACER_FAMILY is general w.r.t. the two modes in the sense that conversions between tracer mode and family mode (and vice versa) are possible at any time during a model time step. Thus, the simulated processes (not limited to kinetics or advection) can affect the single tracers and/or the tracer families. For this purpose, the TRACER_FAMILY conversion routines can be called directly from the submodel interface layer (SMIL) of MESSy submodels.

3.1 Type-1 tracer families

In the tracer mode of the type-1 families, the tracer data instances $x_{i,t-\Delta t}$ and $x_{i,t}$ contain the abundance (e.g., the mixing ratio) of species number i at two stages ($t-\Delta t$ and t) of the time integration scheme, respectively. The third data instance, denoted as $\left(\frac{\Delta x_i}{\Delta t}\right)_{t+\tau}$, contains the corresponding total tendency of the species within the operator splitting time integration scheme, i.e., taking into account the tracer tendencies of all processes calculated for the present time step up to the conversion into the family mode, therefore the notation $\tau \leq \Delta t$. For conversion into the family mode (e.g., before the advection operator is applied), first the weighted sum of the *family members* is calculated for each family and stored in a new tracer with index f :

$$x_{f,t-\Delta t} = \sum_{i \in \mathcal{F}} x_{i,t-\Delta t} \cdot \kappa_i \quad (3)$$

$$x_{f,t} = \sum_{i \in \mathcal{F}} x_{i,t} \cdot \kappa_i \quad (4)$$

$$\left(\frac{\Delta x_f}{\Delta t}\right)_{t+\tau} = \sum_{i \in \mathcal{F}} \left(\frac{\Delta x_i}{\Delta t}\right)_{t+\tau} \cdot \kappa_i . \quad (5)$$

Here, \mathcal{F} is the set of tracer indices of the family members. The κ_i are arbitrary weights that can be applied for conservation constraints. For instance, if

$$\kappa_i = 1 ; \quad \forall i \in \mathcal{F} , \quad (6)$$

the number of molecules is conserved. If κ_i is the molar mass of the species, the total mass is conserved. And if κ_i is the number of chlorine atoms per molecule, the total number of chlorine atoms is conserved. These examples illustrate that only one selected property can be conserved.

As second step of the transformation from tracer mode into family mode, the (weighted) fractions ξ_i of the family members in the family are calculated:

$$\xi_{i,t-\Delta t} = \frac{x_{i,t-\Delta t} \cdot \kappa_i}{x_{f,t-\Delta t}} \quad (7)$$

$$\xi_{i,t} = \frac{x_{i,t} \cdot \kappa_i}{x_{f,t}} \quad (8)$$

$$\xi_{i,t+\tau} = \frac{\left(x_{i,t-\Delta t} + \left(\frac{\Delta x_i}{\Delta t}\right)_{t+\tau} \cdot \Delta t\right) \cdot \kappa_i}{x_{f,t-\Delta t} + \left(\frac{\Delta x_f}{\Delta t}\right)_{t+\tau} \cdot \Delta t} \quad (9)$$

$\xi_{i,t+\tau}$ is, as above, the fraction taking into account all process tendencies of the current model time step up to the conversion, therefore the notation with $\tau \leq \Delta t$. To reduce the memory usage, all three fractions are stored in the memory of the tracer, i.e.,

$$\begin{aligned} x_{i,t-\Delta t} &\leftarrow \xi_{i,t-\Delta t}; \quad x_{i,t} \leftarrow \xi_{i,t}; \\ \left(\frac{\Delta x_i}{\Delta t}\right)_{t+\tau} &\leftarrow \xi_{i,t+\tau}; \quad \forall i \in \mathcal{F}. \end{aligned} \quad (10)$$

At this stage, the process operators (e.g., advection) calculate and add additional tendencies for the family tracer:

$$\left(\frac{\Delta x_f}{\Delta t}\right)_{t+\tau} \leftarrow \left(\frac{\Delta x_f}{\Delta t}\right)_{t+\tau} + \left(\frac{\Delta x_f}{\Delta t}\right)_{\text{process}} \quad (11)$$

The fractions $\xi_{i,t-\Delta t}$, $\xi_{i,t}$ and $\xi_{i,t+\tau}$ are unchanged according to the assumption that the ratios among the family members are constant.

For the backward conversion from the family mode into the tracer mode after the processes affecting the families, the tracer information is reconstructed:

$$x_{i,t-\Delta t} = \frac{x_{f,t-\Delta t} \cdot \xi_{i,t-\Delta t}}{\kappa_i} \quad (12)$$

$$x_{i,t} = \frac{x_{f,t} \cdot \xi_{i,t}}{\kappa_i} \quad (13)$$

$$\begin{aligned} \left(\frac{\Delta x_i}{\Delta t}\right)_{t+\tau} &= \\ \frac{1}{\Delta t} &\left(\xi_{i,t+\tau} \frac{x_{f,t-\Delta t} + \left(\frac{\Delta x_f}{\Delta t}\right)_{t+\tau} \Delta t}{\kappa_i} - x_{i,t-\Delta t} \right). \end{aligned} \quad (14)$$

Note that the sequence of operations is not arbitrary: $x_{i,t-\Delta t}$ must be reconstructed first, because it is used for $\left(\frac{\Delta x_i}{\Delta t}\right)_{t+\tau}$.

One special case w.r.t. the family concept occurs in the sequence of operators. At some stage within the time step the tracer data are output into files for later analysis of the results. At this stage the data should be in tracer mode, however, it is also desirable to consistently output the family tracer. For this, just before the output, only the first step of the conversion into the family mode (i.e., Eqs. 3–5) needs to be applied. Calculation of the fractions (Eqs. 7–9) and storage in the tracer data instances (Eq. 10) must be omitted. The respective conversion routine of TRACER_FAMILY contains a specific switch to account for this special case (see <http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>).

If type-1 families are used as “transport families”, the calculation of the respective transport process (e.g., advection) is not required for the family members. This can be used to reduce the computational burden of a model. To achieve this, the transport algorithm must make use of the tracer meta-information, e.g., the switch for advection (on/off), to skip the respective tracers during the calculation. The corresponding meta-information for the family members is correctly set by the tracer-to-family conversion routines for type-1 families.

3.2 Type-2 tracer families

Implementation of the type-2 family concept (e.g., for tracer tagging) is even simpler, since in contrast to the type-1 family mode, the type-2 family mode does not require a conversion of the family members into fractions of the family. Thus, a conversion into family mode (before a process operator acting on families) consists only of the first conversion step (Eqs. 3–5).

After the process, the backward conversion into tracer mode applies the forced linearisation, whereby the family tracer is the base, and the tagged family members are adjusted (rescaled). For this, first the (weighted) sum of the tracers at the current stage (accounting for the current total tendency) is calculated:

$$\hat{x}_{f,t+\tau} = \sum_{i \in \mathcal{F}} \left(x_{i,t-\Delta t} + \left(\frac{\Delta x_i}{\Delta t}\right)_{t+\tau} \cdot \Delta t\right) \cdot \kappa_i \quad (15)$$

With this sum and the current value of the family,

$$\tilde{x}_{f,t+\tau} = x_{f,t-\Delta t} + \left(\frac{\Delta x_f}{\Delta t}\right)_{t+\tau} \cdot \Delta t, \quad (16)$$

the tracer tendencies are rescaled (denoted by the prime, see Appendix A)

$$\begin{aligned} \left(\frac{\Delta x_i}{\Delta t}\right)'_{t+\tau} &= \frac{x_{i,t+\tau} - x_{i,t-\Delta t}}{\Delta t} = \\ \left(\frac{\Delta x_i}{\Delta t}\right)_{t+\tau} &\cdot \frac{\tilde{x}_{f,t+\tau}}{\hat{x}_{f,t+\tau}} + \frac{x_{i,t-\Delta t}}{\Delta t} \cdot \left(\frac{\tilde{x}_{f,t+\tau}}{\hat{x}_{f,t+\tau}} - 1\right), \end{aligned} \quad (17)$$

such that the time integration scheme yields consistent results between family and family members.

In TRACER_FAMILY the rescaling of the type-2 family members (Eqs. 15–17) is optional. This allows the application of the type-2 family concept for purely diagnostic purposes, e.g., to define a family that contains all chlorine components.

In contrast to type-1 families, type-2 families applied for tagging cannot be used to decrease the computational burden, since it is required that the type-2 family and its members are equally treated by all processes.

3.3 Implementation

The definition of type-1 and type-2 families in TRACER_FAMILY is controlled via a Fortran95 namelist. A family definition consists of

- the tracer set of the family,
- the family type (1 or 2),
- a switch for the rescaling (relevant for type-2 only)
- the name of the family tracer with an optional subname, and
- the list of tracer names of the family members and their weights. The default weight for all members of a family is 1.

The first valid tracer in the family member list determines the characteristics (meta-information) of the family, e.g., the molar mass. Listed family members which are not present in the chosen model setup are ignored. Empty families (i.e., with no members) are also ignored. The following limitations apply to the families and family members:

- Families can only have members of the same tracer set, due to the representation constraints.
- Families cannot be family members.
- A tracer cannot be a member of more than one family in one setup, with only one exception: Type-2 family members of a family without rescaling can be members of type-1 families.

3.4 Examples

For calculating the stratospheric chemistry in a model setup as presented in Jöckel et al. (2006) or similar, it is recommended to apply the type-1 family concept for the reactive chlorine, bromine, and nitrogen compounds, respectively, for the advection operator:

ClO_Z: Cl, ClO, HOCl, OClO, Cl₂ ($\kappa_{\text{Cl}_2} = 2$), Cl₂O₂ ($\kappa_{\text{Cl}_2\text{O}_2} = 2$)

BrO_Z: Br, BrO, HOBr, BrCl, Br₂ ($\kappa_{\text{Br}_2} = 2$)

NO_Z: N, NO, NO₂, NO₃, N₂O₅ ($\kappa_{\text{N}_2\text{O}_5} = 2$)

As indicated in parentheses, the weighting factors are chosen to conserve the number of chlorine, bromine, and nitrogen atoms, respectively.

An example for the application of the type-2 families is to transport aerosol components of a specific mode (or bin) of an aerosol dynamical model together as one family.

The corresponding namelist including both examples is shown in Appendix B.

4 TRACER_PDEF

The abundances of tracers (mixing ratios, concentrations, etc.) are defined as non-negative quantities. A basic requirement of the implementation of processes that change the amounts of tracers in ESMs is therefore the positive definiteness of the corresponding operators. This is for instance a crucial requirement for the design of numerical transport schemes (e.g., Prather, 1986; Bott, 1989; Smolarkiewicz and Rasch, 1991; Chlond, 1994).

Nevertheless, in the operator sequence of comprehensive models, negative values can still occur for several reasons. Some implementations, e.g., for tracer transport, can produce negative values in cases where very strong tracer gradients in time and/or space are present. Approximate parameterisations (e.g., linearised sinks) can produce negative overshoots. In any case there are numerical limitations; negative values can occur due to the limited precision and rounding errors, e.g., as (small) negative overshoots of the numerical solver for the stiff kinetic system. These negative values are unphysical and can lead to instabilities in other process operators (which expect positive tracer values), e.g., resulting in illegal instructions, such as for instance the logarithm or square root of negative values. Furthermore, subsequently applied process operators will spread the problem (even if their implementation is accurately positive definite), e.g., by transporting the negative values.

TRACER_PDEF provides the functionality to correct (i.e., set to zero) and to budget these “technically caused” negative values. As a by-product, TRACER_PDEF provides an online global mass diagnostic for all tracers.

To accomplish this, first the global masses of all tracers x are calculated

$$M_x^{(+)} = \sum x \cdot s_x \cdot m_{\text{air}}; \forall x > 0, \quad (18)$$

$$M_x^{(-)} = \sum x \cdot s_x \cdot m_{\text{air}}; \forall x < 0, \quad (19)$$

where the sum in both equations is over all grid-boxes (according to the representation of the corresponding tracer set). m_{air} is the mass of air in the respective grid-box, and s_x is a tracer-specific factor for the conversion of the unit into a mass mixing ratio. With these definitions, a positive mass $M_x^{(+)}$ and a negative mass $M_x^{(-)}$ for each species x is calculated. If all operators were perfectly positive definite, and the initial condition was $x \geq 0; \forall x$, then the negative mass would be zero. In general, this is not the case. Therefore, as a second step the negative values are (optionally) reset to zero. For this, an additional tendency is added to achieve a value of zero at the end of the time integration scheme.

The procedure so far may mask severe implementation errors of other submodels. However, the ratio f_x of the negative mass to the conserved total mass

$$f_x = \frac{|M_x^{(-)}|}{M_x^{(+)} + M_x^{(-)}} \quad (20)$$

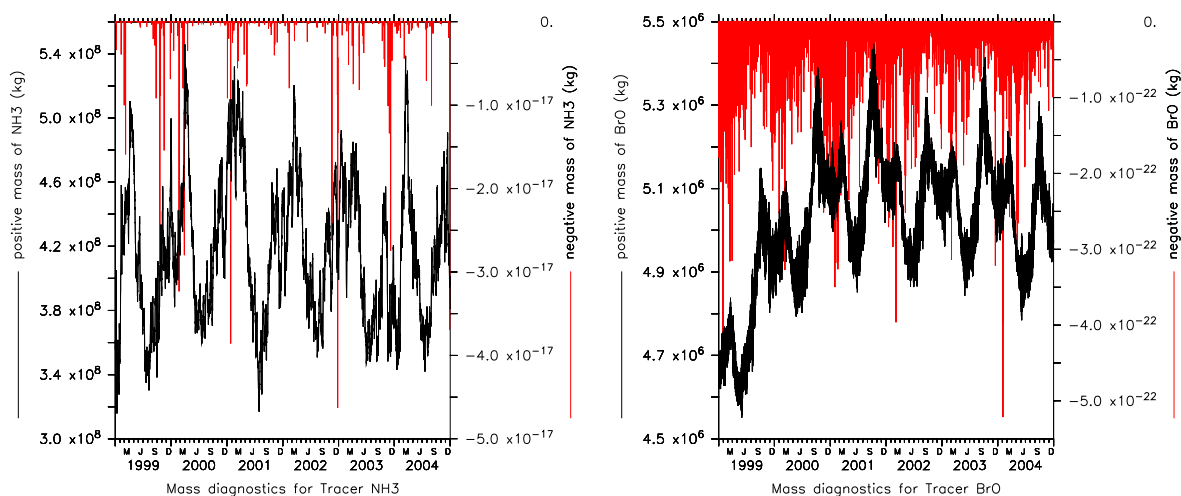


Fig. 2. Positive (black line, left axis) and negative (red line, right axis) global tracer masses in kg for NH_3 (left panel) and BrO (right panel) for a simulation period of six years (1999 to 2004).

provides a measure for the severity of the positive definiteness violation for every species x . This can be used to define a (tracer-specific) threshold above which TRACER_PDEF will terminate the model integration, since the violation is most likely caused by a severe error. If $|M_x^{(-)}|$ is sufficiently small, the positive definiteness violation can be considered to be caused by numerical noise. The tracer-specific threshold and the option to reset the negative values of a specific tracer are set in a Fortran95 namelist (see Appendix B).

4.1 Examples

The sub-submodel TRACER_PDEF has been applied in the simulations described by Jöckel et al. (2006). As an example, Fig. 2 shows the global tracer masses (positive and negative) for NH_3 and BrO . NH_3 occurs mainly in the troposphere, while BrO has higher mixing ratios in the stratosphere compared to the troposphere.

Both masses of both tracers show strong variations. The variations in the positive tracer mass result from the variation of sources (emissions, chemistry) and sinks (chemistry, dry and wet deposition) depending on the meteorological conditions. For both species, a seasonal cycle is evident. The negative values result from numerical limitations. While for NH_3 (left panel of Fig. 2) the negative tracer mass occurs only sporadically, for BrO negative values are calculated almost continuously. This is a consequence of the higher photochemical activity (shorter lifetime) of BrO and its strong changes during sunrise and sunset. In contrast to this, NH_3 in the applied kinetic setup is only indirectly influenced by the photochemistry.

For both species, the negative values are considered to result only from numerical noise, because the negative mass is 25 (NH_3) and 28 (BrO) orders of magnitude lower than

the positive mass. We therefore conclude that the correction hardly affects the global budget, but indeed increases the stability of the integration scheme and guarantees the overall positive definiteness for other process operators. We further point out that negative values such as those seen in the examples do not occur for all species. For longer-lived species, like O_3 , but also for short-lived tracers, like H , negative values never occurred in the simulation and the correction was not required.

4.2 Discussion

We emphasise that a meaningful application of TRACER_PDEF requires a proper positive definite implementation of all tracer related processes. TRACER_PDEF is solely designed to correct and budget technically caused negative overshoots such as rounding errors. It is neither suited to correct for process-specific misconceptions, nor capable of correcting positive overshoots, which can anyway not be fixed a posteriori.

Based on this prerequisite, the locally occurring negative values are simply reset to zero, accepting a small mass conservation violation due to the artificially created mass. Methods to redistribute this mass either globally or locally are not implemented since such attempts likely introduce other problems such as unphysical instantaneous long-range transport, violation of the monotonicity constraints etc. Nevertheless, the mass conservation violation is diagnosed, thus providing a quality control measure. And indeed, as the examples above show, the global mass conservation violation is negligible.

The local impact of such a correction procedure, however, is not controlled by the simple “global mass based” threshold mechanism. Yet, with the prerequisite above, only negligible

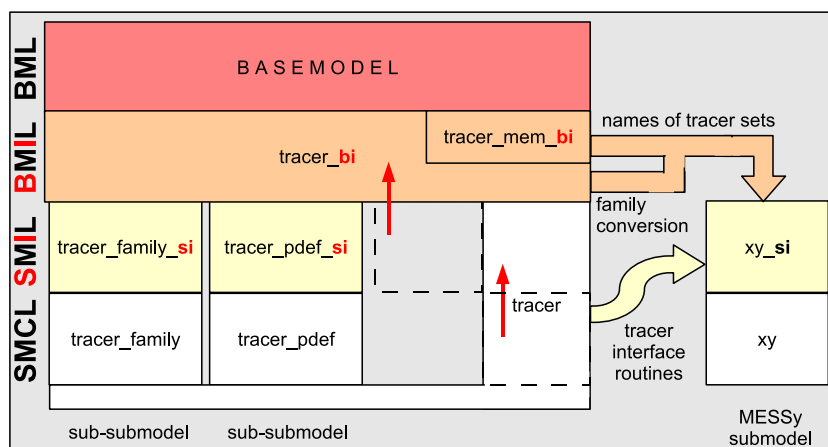


Fig. 3. Module structure of the generic submodel TRACER with its submodels TRACER_FAMILY and TRACER_PDEF. The names of the corresponding Fortran95 modules are *messy_main_X.f90*, where the “X” is a placeholder for the names in the boxes. The suffixes *si* and *bi* denote modules of the submodel interface layer (SMIL) and the base model interface layer, respectively. The meaning of “generic submodel” is indicated by the dashed boxes and red arrows: A generic submodel is itself coded as an ordinary submodel (i.e., split into SMCL and SMIL), but it provides a part of the MESSy infrastructure (BMIL) and therefore can be used (in the Fortran95 sense) directly by ordinary submodels (here the submodel “xy”). Thus, the SMCL and SMIL of TRACER are shifted to the SMIL and to the BMIL, respectively.

negative overshoots are to be expected. Nevertheless, alternative, more local threshold conditions can easily be implemented into TRACER_PDEF.

5 PTRAC

The standard method of defining tracers is the application of the TRACER interface routines (see <http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>) from within the SMIL of submodels. Any modification, however, requires recompilation, which is not in all cases desirable. The MESSy submodel PTRAC (for *prognostic tracers*) allows in addition the alternative definition of tracers via a Fortran95 namelist. The namelist entries (as shown in more detail in Appendix C) correspond directly to the tracer meta-information as explained in Sect. 2.1. Tracers can be defined independently for different tracer sets.

If a tracer is an aerosol species, its evolution depends largely on the aerosol dynamical model operator which defines the corresponding aerosol properties (e.g., radius, density, radius standard deviation, etc.). Thus, for aerosol tracers the name of the aerosol dynamical submodel, the corresponding method (modal or bin), the corresponding mode (or bin) number and the density of the specific aerosol component are stored in the tracer meta-information. This information can then be used by the aerosol dynamical models (and by other aerosol process operators) to select the tracers they process. An example for such an application is described in detail by Kerkweg et al. (2007, and references therein).

In addition to this, PTRAC can also be used as a simplified modal aerosol model, in which the aerosol properties are

constant in time and space. For this it provides a number of modes with the corresponding radii, radii standard deviations (sigma) and aerosol densities. These mode properties are defined in the namelist along with the specification of the tracers (more details are explained in Appendix C). This feature can be used in simplified model setups to test the coupling to processes which operate on aerosol tracers, such as aerosol scavenging (Tost et al., 2006).

6 The implementation in the MESSy structure

TRACER with its submodels TRACER_FAMILY and TRACER_PDEF is implemented according to the MESSy standard (Jöckel et al., 2005), i.e., within the four layer structure. TRACER constitutes a *generic submodel*, i.e., a submodel providing a part of the MESSy infrastructure. The module structure is sketched in Fig. 3.

As such, the MESSy submodels (within their submodel interface layer (SMIL)) are allowed to directly use routines from the submodel core layer (SMCL) of TRACER. Direct access to the sub-submodel routines of TRACER_FAMILY and/or TRACER_PDEF is not required. The interface routines for conversions of tracer families (explained in Sect. 3) are provided by the TRACER BMIL.

PTRAC is implemented as an “ordinary” (as opposed to “generic”) MESSy submodel, in the same way as sketched for the submodel “xy” in Fig. 3. Details about the implementation of submodels have been documented by Jöckel et al. (2005).

7 Summary

The generic submodel TRACER provides the framework for the implementation of chemical species into Earth System Models. A tracer is split into its meta-information (characteristics that do not change with space and/or time) and its data (the abundance of the tracer in space and time). A multitude of tracer sets can be defined and associated with different representations (grid, spectral coefficients, etc.) of the ESM. Each tracer set can store an arbitrary number of data instances (limited only by the physical memory), e.g., for different stages of the time integration scheme of the model.

TRACER includes two submodels, one for the application of tracer families (TRACER_FAMILY) and one for correcting numerical negative overshoots to enforce the positive definiteness of tracer-related processes (TRACER_PDEF). As by-product, TRACER_PDEF also provides a global tracer mass diagnostic.

TRACER_FAMILY comprises two different concepts, type-1 families for strongly related species (e.g., fast equilibrating species during advection), and type-2 families for the linearisation of tagging techniques. Both concepts can be applied simultaneously and the conversion routines can be used by every MESSy submodel. In the current implementation, tracer families for tracer sets with three data instances representing the tracer at time step $t - \Delta t$, the tracer tendency, and the tracer at time step t , respectively, are handled.

The MESSy submodel PTRAC can be applied to easily define prognostic tracers via a Fortran95 namelist. Furthermore, it can serve as a highly simplified aerosol model (with static aerosol properties) for the technical testing of aerosol-related processes (such as dry deposition or scavenging).

The code with documentation is part of the electronic supplement (<http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>). It can be applied to ESMs, general circulation models and chemistry transport models without further adaptations (such as other parts of the MESSy infrastructure).

Appendix A

The rescaling for the type-2 family tracers (Eq. 17) results from two conditions. The first condition is that the (weighted) sum of the rescaled family members must be equal to the family (cf. Eqs. 16 and 17):

$$\begin{aligned} \tilde{x}_{f,t+\tau} &= a \cdot \hat{x}_{f,t+\tau} \\ x_{f,t-\Delta t} + \left(\frac{\Delta x_f}{\Delta t} \right)_{t+\tau} \cdot \Delta t &= \\ a \cdot \sum_{i \in \mathcal{F}} (x_{i,t-\Delta t} + \left(\frac{\Delta x_i}{\Delta t} \right)_{t+\tau} \cdot \Delta t) \cdot \kappa_i, \end{aligned} \quad (\text{A1})$$

where the prime denotes adjusted quantities. The family members are all rescaled by the same factor a , in order

to keep the ratios between the family members unchanged. Only the tendency on the left side is primed, since the second condition is that the values at $t - \Delta t$ must not be changed. Substituting $x_{f,t-\Delta t}$ with its definition (Eq. 3) on the left side, replacing the primed tendency with Eq. (5), i.e., with a sum of primed tracer tendencies, and solving for the sum over the primed tracer tendencies yields Eq. (17), since the equation must be valid for each family member x_i independently.

Appendix B

Figure B1 shows an example namelist file as required by the generic MESSy submodel TRACER. In the CTRL namelist, the submodels TRACER_FAMILY and TRACER_PDEF are switched on (T) or off (F), respectively.

The CTRL_FAMILY namelist controls the submodel TRACER_FAMILY. Most important are the definitions of the tracer families (TF(.)) as explained in Sect. 3.3. The number n of the family is arbitrary ($1 \leq n \leq 50$ in the current implementation), but it must be unique. Here, numbers 1 to 3 correspond to the examples of type-1 families in Sect. 3.4. The weights are defined after the colon. Numbers 10 and 11 are examples of type-2 tracer families, one with and one without rescaling, respectively.

The CTRL_PDEF namelist controls the submodel TRACER_PDEF. The TPD_DEFAULT(.) entries (one for each tracer set) determine the default settings for all species of a tracer set. The numbers are arbitrary, but must be unique. In the example, the default for the tracer set “gp” is to reset the negative values (T) and to terminate (T) the simulation above the threshold ($f_x=0.01$). This default value can then be overwritten for individual tracers with the TPD(.) entries (HNO3_nat and N in the example). The enumeration of TPD(.) is also arbitrary (500 per tracer set), but the numbers need to be unique.

Appendix C

Figure C1 shows an example namelist for the MESSy submodel PTRAC for the definition of tracers. Since “ordinary” submodels must access the infrastructure from their SMIL, the namelist is - following the MESSy standard naming convention - named CPL. Depending on the type of the information, the corresponding variable names begin with I_ for Fortran95 integer values, with R_ for real values and C_ for character (string) values, respectively. In the example a tracer with the name SF6 (C_NAME(1)) in units of “mol/mol” (C_UNIT(1)) is defined for the tracer set “gp” (C_SETS(1)). The identification of the tracer (i.e., the structure t_ident, see Sect. 2.1) is completed by specifying 0 (single tracer) as the type (I_TYPE(1)), 1 (air) as the medium (I_MEDIUM(1)) and 1 (amountfraction) as the quantity (I_QUANTITY(1)) of the tracer.

The entries `CASK_X(1, :)` (with $X=I, R, S$ for Fortran95 integer, real or string values, respectively) specify the further characteristics (i.e., the contents of the meta-information containers of `t_meta`) for a specific tracer (here for tracer number 1) as explained in Sect. 2.1 and the supplement (<http://www.atmos-chem-phys.net/8/1677/2008/acp-8-1677-2008-supplement.zip>). Note that for a given Fortran95 type the order of the entries must correspond to the defined container numbers. In the example here, the sequences of the entries (e.g., the switches for the processes like advection, convection, vertical diffusion etc.) are indicated in the comment lines of the namelist. The molar mass of the tracer SF₆ is 146.07 g/mol (`CASK_R(1, 1)`).

All container numbers (number at 2nd rank of the `CASK_X`) used here (see below) are provided as Fortran95 `PARAMETERS` for usage in the code, thus the molar mass of tracer number 1 will be accessed as `CASK_R(1, R_MOLARMASS)`.

For `I_MEDIUM(.)=2`, the tracer will reside in the medium “aerosol”. In this case, the corresponding entries determine which aerosol dynamical model should handle the tracer (`CASK_S(. , 1)`), the density of this aerosol component, (`CASK_R(. , 5)`), whether it is a modal or a bin aerosol dynamical model (`CASK_I(. , 13)`), and which mode or bin this tracer should reside in (`CASK_I(. , 14)`).

In contrast to this, the entries `R_AEROSOL_SIGMA(.)` and `R_AEROSOL_RADIUS(.)` do not specify tracer characteristics, but rather define a radius standard deviation (sigma) and a radius for the mode `CASK_I(. , 14)`, when PTRAC is applied as a simplified modal aerosol model. Since sigma, radius and the aerosol density are common for a given mode, they are defined by the first tracer in the list (tracer enumeration) for a specific mode. In this case the aerosol density is set to 2/3 of the aerosol component density of this tracer (`CASK_R(. , 5)`), providing an approximate ambient aerosol density.

Acknowledgements. The authors wish to acknowledge the use of the Ferret program for analysis and graphics in this paper. Ferret is a product of NOAA’s Pacific Marine Environmental Laboratory (Information is available at <http://ferret.pmel.noaa.gov/Ferret/>). We further thank two anonymous referees and S. Burrows for their valuable comments on this manuscript.

Edited by: M. Dameris

References

Austin, J., Shindell, D., Beagley, S. R., Brühl, C., Dameris, M., Manzini, E., Nagashima, T., Newman, P., Pawson, S., Pitari, G., Rozanov, E., Schnadt, C., and Shepherd, T. G.: Uncertainties and assessments of chemistry-climate models of the stratosphere, *Atmos. Chem. Phys.*, 3, 1–27, 2003, <http://www.atmos-chem-phys.net/3/1/2003/>.

Bott, A.: A positive definite advection scheme obtained by non-linear renormalization of the advective fluxes, *Mon. Weather Rev.*, 117, 1006–1015, 1989.

Carver, G. D. and Stott, P. A.: IMPACT: an implicit time integration scheme for chemical species and families, *Ann. Geophys.*, 18, 337–346, 2000, <http://www.ann-geophys.net/18/337/2000/>.

Chlond, A.: Locally Modified Version of Bott’s Advection scheme, *Mon. Weather Rev.*, 122, 111–125, 1994.

Jöckel, P.: Technical Note: Recursive discretisation of geoscientific data in multiple dimensions in the Modular Earth Submodel System (MESSy) data import interface, *Atmos. Chem. Phys.*, 6, 3557–3562, 2006, <http://www.atmos-chem-phys.net/6/3557/2006/>.

Jöckel, P., Sander, R., Kerkweg, A., Tost, H., and Lelieveld, J.: Technical Note: The Modular Earth Submodel System (MESSy) – a new approach towards Earth System Modeling, *Atmos. Chem. Phys.*, 5, 433–444, 2005, <http://www.atmos-chem-phys.net/5/433/2005/>.

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, *Atmos. Chem. Phys.*, 6, 5067–5104, 2006, <http://www.atmos-chem-phys.net/6/5067/2006/>.

Kerkweg, A., Sander, R., Tost, H., Jöckel, P., and Lelieveld, J.: Technical Note: Simulation of detailed aerosol chemistry on the global scale using MECCA-AERO, *Atmos. Chem. Phys.*, 7, 2973–2985, 2007, <http://www.atmos-chem-phys.net/7/2973/2007/>.

Prather, M. J.: Numerical advection by conservation of second order moments, *J. Geophys. Res.*, 91, 6671–6681, 1986.

Sheela, V., Lal, S., and Brühl, C.: Validation of a 2D model using observed profiles of SF₆ with main focus on the tropics, *Atmos. Environ.*, 39, 1255–1261, 2005.

Smolarkiewicz, P. K. and Rasch, P. J.: Advection on the Sphere: An Eulerian Versus Semi-Lagrangian Approach, *J. Atmos. Sci.*, 48, 793–810, 1991.

Steil, B., Brühl, C., Manzini, E., Crutzen, P., Lelieveld, J., Rasch, P. J., Roeckner, E., and Krüger, K.: A new interactive chemistry climate model. I: Present day climatology and interannual variability of the middle atmosphere using the model and 9 years of HALOE/UARS data, *J. Geophys. Res.*, 108, 4290, doi:10.1029/2002JD002971, 2003.

Tost, H., Jöckel, P., Kerkweg, A., Sander, R., and Lelieveld, J.: Technical Note: A new comprehensive SCAVenging submodel for global atmospheric chemistry modelling, *Atmos. Chem. Phys.*, 6, 565–574, 2006, <http://www.atmos-chem-phys.net/6/565/2006/>.

Wayne, R. P.: *Chemistry of Atmospheres*, Oxford University Press, 2000.