A MODEL-INDEPENDENT ELEMENT TRACING SOFTWARE FOR BIOGEOCHEMICAL MODELS (ETRAC)

Fabian Große

Dalhousie University
Department of Oceanography
1355 Oxford Street
Halifax, NS, B3K 3G4
dal.ca/faculty/science/oceanography.html

November 2017

Technical Report No. OCE-2017-01

This technical report and the software it describes were created by Fabian Große. Parts of this report were published in his dissertation (Große, 2017). Those parts are not explicitly marked or referenced in the text. For any questions and access to the software please contact him via: grosse@bafg.de

Summary

The **E**lement **TRAC**ing software (ETRAC) described in this report facilitates the tracing of one or multiple different elements (e.g., nitrogen and/or phosphorus) from individual input sources, such as rivers, throughout the entire biogeochemical cycle represented by the model it is applied to.

This method is often referred to as "trans-boundary nutrient transports" (TBNT; e.g., Blauw et al., 2006; OSPAR, 2010) and was first published by Ménesguen et al. (2006). However, its theory grounds on the work by Ménesguen and Hoch (1997) who showed that each specific property, e.g., the input source or the age, of a certain element can be traced within a biogeochemical model by introducing additional state variables that represent the product of the element and the property.

Up to now, most studies utilizing the TBNT method based on the direct implementation of the method to the applied model (e.g., Ménesguen et al., 2006; Radtke et al., 2012; Radtke and Maar, 2016; Troost et al., 2013). The ETRAC software aims for a step forward by facilitating the elemental tracing without requiring its implementation to the user's model. Instead the user has to provide the relevant information about the model structure (i.e., grid and variable interactions) and specific model output. This report describes the theory of the TBNT method, and the basic functioning and setup of the ETRAC software. In addition, example results from existing ETRAC applications are presented.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Biogeochemical models allow for the domain-wide, spatially and temporally consistent quantification of physical and biochemical processes of the ecosystem they describe. In addition, they can be used to track the dispersal of tracers within an ecosystem. There are different applications for investigating this spatial displacement of tracers using such models. For instance, passive tracers, like rhodamine/dye, which are not affected by internal source or sink processes can be traced using both Lagrangian (e.g., Hainbucher et al., 1987; Lenhart et al., 1995) and Eulerian models (e.g., Zhang et al., 2012). The modelling of radioactive tracers represents a more complex application as it includes simple decay functions (e.g., Dahlgaard, 1995; Harms, 1997). However, in order to quantify the influence of inorganic and organic nutrients from individual sources on the biochemical processes within an ecosystem, a more sophisticated approach is required.

Ménesguen and Hoch (1997) provided the theoretical basis for the tracing of a selected property, e.g., the source of a nitrogen (N) element brought into an ecosystem) throughout the entire biochemical process chain represented by the underlying model (hereafter 'base model'). Since that, several modelling studies made use of their method with various research objectives. For instance, Wijsman et al. (2004) determined the N retention capacity of the Scheldt using this method, while Timmermann et al. (2010) applied it for determining the sources of P inside a Danish estuary. Other studies quantified the amount of N from various riverine sources bound in phytoplankton in French (Ménesguen et al., 2006; Perrot et al., 2014) and Belgian coastal waters (Lacroix et al., 2007). The method was further used for investigating the influence of atmospheric N deposition on the North Sea (Troost et al., 2013) or to determine the dispersal of riverine TN and TP in the Baltic Sea (Neumann, 2007; Radtke et al., 2012). Radtke and Maar (2016) applied the method for the quantification of the total nitrogen (TN) exchange between the North Sea and the Baltic Sea. These applications illustrate the versatility of this method, for which – in the meantime – the term 'trans-boundary nutrient transports' (TBNT) was established (Blauw et al., 2006; OSPAR, 2010). This term will also be used throughout this report when referring to the method itself.

All these studies relied on the direct implementation of the TBNT method into the original model, i.e., significant changes to the model code were required. Radtke et al. (2012) used a so-called Code Generation Tool (CGT; https://ergom.net/index.php/code-generation-tool.html, developed at the Leibniz Institute for Baltic Sea Research (IOW), to implement TBNT into ERGOM (Ecological Regional Ocean Model). This CGT facilitates the implementation of TBNT, however, it still requires building a "new" model.

This report describes a model-independent **E**lement **TRAC**ing software (ETRAC) that enables the realization of TBNT as a post-processing application that only requires basic information on the model structure (e.g., grid and processes) and specific model output. Hence, the only changes/additions may be required for the output of the model ETRAC should be applied to. The following chapters will first describe the basic theory of TBNT. Thereafter, a more technical description of the ETRAC software, its requirements and its application is provided. Finally, some example results from existing ETRAC applications to different biogeochemical models are presented.

Chapter 2

The TBNT method

This chapter describes the concept of the TBNT method used for the tracing of nutrient inputs from multiple sources. First, a theoretical description is provided, followed by a more technical description in relation to numerical models.

2.1 Theory

The basic idea of TBNT is that organic and inorganic matter, that contains a certain chemical element (e.g., N), obtains a unique, source-specific label at the moment of its release into the ecosystem. The labelling of matter based on the element implies that all model state variables containing this element must be labelled accordingly. The labelled state variables experience the same biochemical and physical processes as the overall state variables, i.e., the total amount of labelled and unlabelled material, however, proportional to their relative contribution to this overall amount. In the following, the prefixes 'bulk' and 'fraction' are used to refer to the overall state variables and processes, and their labelled counterparts, respectively.

Following the approach by Ménesguen and Hoch (1997), any property p (e.g., origin/source, age etc.) can be attached to any bulk state variable X and can be traced within an ecosystem model by solving an additional differential equation for the product $X \cdot p$ instead of the bulk state variable X. The product $X \cdot p$ then represents that subset of the bulk variable X with the defined property p. Technically, the product $X \cdot p$ is introduced into the base model as a new fraction state variable $X_p = X \cdot p$ and the related processes are introduced according to the differential equation describing the changes in the corresponding bulk state variable X. Each combination of state variables X and properties p_i, $X_p^i = X \cdot p_i$, therefore requires an additional differential equation to be introduced to the model.

In order to show demonstrate the relation between bulk variables and fraction variables, one has to start with the convection-diffusion equation for the concentration C_X of a bulk state variable X, which reads as:

$$\frac{dC_X}{dt} = \underbrace{\nabla \cdot \left(\overline{\overline{D}} \nabla C_X\right)}_{diffusion} \quad \underbrace{- \nabla \cdot (\vec{v} C_X)}_{convection/advection} \quad \underbrace{+ R_X}_{sources/sinks}. \tag{2.1}$$

The diffusive transport is calculated according to Fick's first law with the second order diffusion tensor (or diffusivity) $\overline{\overline{D}}$. In the convective/advective transport term, $\vec{v}$ represents the 3D velocity vector. R_X represents the sources and sinks of X (i.e., biochemical processes, input from external sources). Assuming that the bulk variable X consists of N fractions $X^i = X \cdot p_i$, with $i = 1, 2, ..., N-1, N$, implies for the concentration C_X:

$$C_X = \sum_{i=1}^{N} C_X^i, \tag{2.2}$$

where C_X^i denotes the concentration of the fraction variable X_i. Dividing Eq. (2.2) by C_X yields:

$$\sum_{i=1}^{N} \frac{C_X^i}{C_X} = 1, \tag{2.3}$$

with $\frac{C_X^i}{C_X}$ representing the relative contribution of the fraction variable X_i. Multiplication of the right-hand side of Eq. (2.1) with 1 results in:

$$\frac{dC_X}{dt} = \left[\nabla \cdot \left(\overline{\overline{D}} \nabla C_X\right) - \nabla \cdot (\vec{v} C_X)\right] \cdot 1 + R_X \cdot 1. \tag{2.4}$$

Combining Eq. (2.4) with Eqs. (2.2) and (2.3) leads to:

$$\frac{d}{dt}\left(\sum_{j=1}^{N} C_X^i\right) = \left[\nabla \cdot \left(\overline{\overline{D}} \nabla C_X\right) - \nabla \cdot (\vec{v} C_X)\right] \cdot \sum_{j=1}^{N} \frac{C_X^i}{C_X} + R_X \cdot \sum_{j=1}^{N} \frac{C_{X_\text{in}}^i}{C_{X_\text{in}}}. \tag{2.5}$$

The index X_in in the source/sink term indicates that the input variable, respectively, the variable, that is consumed, is used. The summation and differentiation on the left-hand side of Eq. (2.5) can be permuted according to the sum rule in differentiation. Then, the separation of the sum terms in Eq. (2.5) finally provides the convection-diffusion equation for the concentration of an individual fraction variable, C_X^i:

$$\begin{aligned} \frac{dC_X^i}{dt} &= \left[\nabla \cdot \left(\overline{\overline{D}} \nabla C_X\right) - \nabla \cdot (\vec{v} C_X)\right] \cdot \frac{C_X^i}{C_X} + R_X \cdot \frac{C_{X_\text{in}}^i}{C_{X_\text{in}}} \\ &= \underbrace{\nabla \cdot \left(\overline{\overline{D}} \nabla C_X\right) \cdot \frac{C_X^i}{C_X}}_{diffusion} \quad \underbrace{- \nabla \cdot (\vec{v} C_X) \cdot \frac{C_X^i}{C_X}}_{convection/advection} \quad \underbrace{+ R_X \cdot \frac{C_{X_\text{in}}^i}{C_{X_\text{in}}}}_{sources/sinks}. \end{aligned} \tag{2.6}$$

Equation (2.6) implies that all physical (diffusion and transport) and biochemical processes (internal sources/sinks) affecting the concentration of the fraction variable, C_X^i, are calculated as the product of the process based on the concentration of the bulk variable, C_X, and the relative contribution of the fraction variable, C_X^i/C_X. Thus, the major assumption of Eq. (2.6) is that all fraction variables X_i are chemically and physically identical. This in turn means that there is no preference for any fraction variable X_i by any (physical or biochemical) process. Equation (2.6) yields the basis for the application of TBNT in this thesis.

2.1.1 The treatment of diffusion

In previous studies (e.g., Ménesguen and Hoch, 1997; Ménesguen et al., 2006; Radtke, 2012; Radtke et al., 2012), the above mentioned major assumption is applied to convective/advective transport and biochemical processes, however, not to diffusive transport. In contrast, these studies calculated an individual diffusive transport for each fraction variable X_i according to Fick's first law:

$$\vec{J}_X^i = \nabla \cdot \left(\overline{\overline{D}} \nabla C_X^i \right) , \tag{2.7}$$

with $\vec{J}_X^i$ representing the 3D diffusive net transport. In case of different spatial gradients in C_X and C_X^i, this results in a diffusive transport flux for X_i different from that described by the diffusion term in Eq. (2.6) and, consequently, in different distributions of the fraction variables. Equation (2.7) furthermore implies that the different fraction variables X_i of the same bulk species (e.g., NO_3^- from Rhine and NO_3^- from Elbe) are chemically and/or physically different, i.e., their virtual discriminability implies chemical disparity.

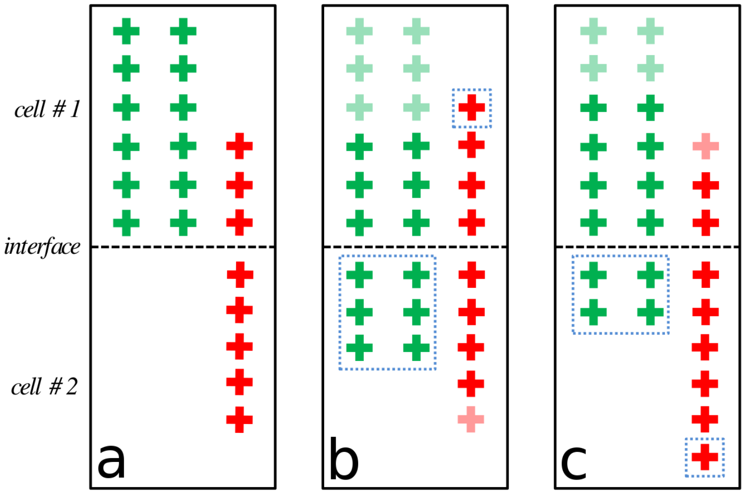

Figure 2.1: Schematic representation of the difference between the treatment of diffusion within the traditional TBNT approach (Ménesguen and Hoch, 1997) and the adapted approach used in this study. The panels show the distribution of two fraction variables of the same species (red/green crosses) at (a) initial state and after instantaneous mixing according to (b) the traditional and (c) the adapted approach. Light-coloured crosses represent the original location of particles which have been diffusively transferred to their target location (blue dotted boxes).

Figure 2.1 illustrates the difference between the two assumptions starting from the same initial distribution (Fig. 2.1a) of two fraction variables of the same species (red and green crosses) in two adjacent water bodies of the same volume. In both cases, the *net* diffusive transport equals five particles from *cell #1* to *cell #2*, i.e., the diffusive net transports of the bulk variable (crosses) are identical. However, according to the traditional approach (Fig. 2.1b; blue dotted boxes) six green particles are transported from *cell #1* to *cell #2* and one red particle is transported from *cell #2* to *cell #1*, while following the approach of this study (Fig. 2.1c; blue dotted boxes) four green and one red particle are transported from *cell #1* to *cell #2*. Thus, for each spatial dimension the traditional approach results in a bidirectional diffusive transport driven by the distribution of the individual variable fractions (in the case of opposing concentration gradients in the different fraction variables), while the new approach results in a unidirectional diffusive transport in the direction of the concentration gradient of the bulk variable.

As (one-dimensional) diffusion in principle is a bidirectional process, the traditional approach intuitively appears to be exact, while the new approach does not. However, it has to be kept in mind that (1) all fraction variables are of the same species (i.e., chemically and physically identical) and only distinguishable by the 'imaginary' label, and (2) that Fick's first law only provides the diffusive *net* transport which implies that there is no knowledge about the two diffusive gross transports contributing to this net transport. For the example in Fig. 2.1a this means that – considering aspect (1) as valid – aspect (2) allows for any final distribution of the particles of the two fraction variables as long as the diffusive net transport is equal to five particles transported from *cell #1* to *cell #2* and as long as the number of particles (colour-independent) in both cells is balanced. This implies that none of the two different assumptions on the diffusive transport – traditional vs. new approach – does necessarily lead to the actual diffusive gross transports, which are not known. Although the two different approaches result in different distributions of the fraction variables, it should be noted that the differences are expected to be generally small and that noticeable differences may only occur in the case of large differences in the spatial gradients of the fraction and bulk variables, i.e., $|\nabla C_X^i| \gg |\nabla C_X|$ or $|\nabla C_X^i| \ll |\nabla C_X|$.

2.2 Discretization for biogeochemical models

Equation (2.6) describes the convection-diffusion equation for the concentration of a fraction variable, C_X^i, as it is used in this study, and which represents the basis for the application of TBNT in combination with a biogeochemical model. In order to apply TBNT to a numerical biogeochemical model with a discrete spatial grid and time step, Eq. (2.6) needs to be discretised as well. Additionally, the change in concentration according to Eq. (2.6) has to be transferred into a change in mass as the particles with property p_i are actually labelled. For a given volume $V = const.$ this change in mass reads as:

$$\frac{dM_X^i}{dt} = \frac{dC_X^i}{dt} \cdot V, \tag{2.8}$$

with M_X^i representing the mass of the fraction variable X_i. However, in the case of a model with a free surface – like ECOHAM – $V = const.$ does not apply to the grid cells in the surface layer. Therefore, the change in mass needs to be calculated

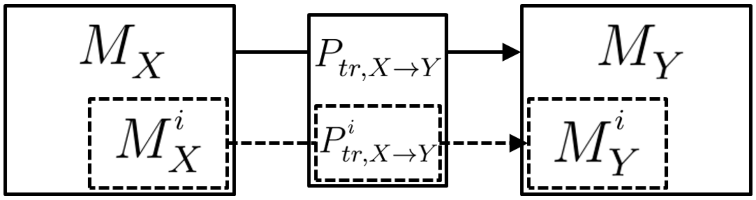

Figure 2.2: Schematic view of relation between model bulk state variables and processes (e.g., NPP) and the corresponding fraction quantities according to the TBNT method. Subscripts indicate bulk quantities, superscript i refers to the corresponding fraction quantities. Process $P_{\mathrm{tr},\,X\to Y}$ transforms mass of state variable M_X into mass of M_Y. The fraction process $P^i_{\mathrm{tr},\,X\to Y}$ is defined as the product of the relative fraction M^i_X/M_X and the bulk process $P_{\mathrm{tr},\,X\to Y}$. Adapted from Wijsman et al. (2004, Appendix D).

for a known volume at a given point in time, $V_0 = V(t_0)$, and can afterwards be transferred into a change in concentration relative to the new volume $V_1 = V(t_1)$. t_0 represents the starting time of the time step of length Δt and $t_1 = t_0 + \Delta t$ represents the ending time of the same time step.

Discretisation of Eq. (2.8) for a given ECOHAM time step $[t_0, t_0 + \Delta t]$ leads to:

$$\frac{\Delta M^i_X}{\Delta t} = \frac{\Delta C^i_X}{\Delta t} \cdot V(t_0). \tag{2.9}$$

According to Eqs. (2.1) and (2.6) the change in mass of a bulk X and fraction variable X^i is calculated as the sum of all transport processes, and source and sink processes. Thus, a model process must be mass-conservative and is either defined as the (physical) **exchange of mass** of a single state variable X between two adjacent grid cells, j and k, or as the (biochemical) **transformation of mass** between two state variables, X and Y, within a single grid cell j. For a certain fraction variable X_i this change in mass over the time step $[t_0, t_0 + \Delta t]$ reads as:

$$\frac{\Delta M^i_X}{\Delta t} = \sum_{n=1}^{N_{\mathrm{ex}}} P^i_{\mathrm{ex},\,n}(\Delta t) + \sum_{n=1}^{N_{\mathrm{tr}}} P^i_{\mathrm{tr},\,n}(\Delta t). \tag{2.10}$$

Here, N_{ex} and N_{tr} represent the numbers of the exchange processes P^i_{ex} and of the transformation processes P^i_{tr}, respectively. These processes on the fraction variable X_i can in principle be calculated as:

$$P^i_{\mathrm{ex}} = P_{\mathrm{ex}} \cdot \frac{M^i_X}{M_X}, \quad P^i_{\mathrm{tr}} = P_{\mathrm{tr}} \cdot \frac{M^i_X}{M_X}, \tag{2.11}$$

with P_{ex} and P_{tr} representing the corresponding bulk processes and the relative fraction $M^i_X/M_X = C^i_X/C_X$ (see Eq. (2.6)) for a known constant volume V. This relation between fraction and bulk process and the related fraction variables is illustrated in Fig. 2.2.

An **exchange process** $P_{\mathrm{ex},\,X}$ reduces the mass of the transferred state variable X in the originating grid cell j and increases its mass in the target grid cell k by the process' value. Advective and diffusive transports are considered as exchange processes. For a time step of length Δt, the changes of mass of a bulk state variable, M_X, in two adjacent grid cells, j and k, due to an exchange process are calculated as:

$$\begin{aligned} M_X(t_0 + \Delta t, j) &= M_X(t_0, j) - P_{\mathrm{ex},\,X}(\Delta t, I_{jk}), \qquad (2.12)\\ M_X(t_0 + \Delta t, k) &= M_X(t_0, k) + P_{\mathrm{ex},\,X}(\Delta t, I_{jk}). \end{aligned}$$

Here, M_X refers to the mass of the bulk state variable X being transferred from input grid cell j into output grid cell k. $P_{\mathrm{ex},\,X}$ indicates the bulk exchange process at the interface I_{jk} between the two grid cells j and k.

A **transformation process** $P_{\mathrm{tr},\,X\to Y}$ reduces the mass of the input variable M_X and increases the mass of the output variable M_Y by its value. Thus, for a time step of length Δt the changes of masses of two bulk state variables, M_X and M_Y, due to a transformation process are calculated as:

$$\begin{aligned} M_X(t_0 + \Delta t, j) &= M_X(t_0, j) - P_{\mathrm{tr},\,X\to Y}(\Delta t, j), \qquad (2.13)\\ M_Y(t_0 + \Delta t, j) &= M_Y(t_0, j) + P_{\mathrm{tr},\,X\to Y}(\Delta t, j). \end{aligned}$$

Here, M_X and M_Y refer to the mass of the bulk input state variable and the bulk output state variable, respectively. $P_{\mathrm{tr},\,X\to Y}$ indicates the bulk transformation process within the grid cell j. t_0 refers to the starting time of the time step of length Δt, i.e., $t_0 + \Delta T$ indicates the end of the considered time step.

The definitions of the two types of bulk processes, Eqs. (2.12) and (2.13):, apply accordingly to fraction processes:

$$\begin{aligned} M_X^i(t_0 + \Delta t, j) &= M_X^i(t_0, j) - P_{\mathrm{ex},\,X}^i(\Delta t, I_{jk}), \qquad (2.14)\\ M_X^i(t_0 + \Delta t, k) &= M_X^i(t_0, k) + P_{\mathrm{ex},\,X}^i(\Delta t, I_{jk}),\\ M_X^i(t_0 + \Delta t, j) &= M_X^i(t_0, j) - P_{\mathrm{tr},\,X\to Y}^i(\Delta t, j), \qquad (2.15)\\ M_Y^i(t_0 + \Delta t, j) &= M_Y^i(t_0, j) + P_{\mathrm{tr},\,X\to Y}^i(\Delta t, j). \end{aligned}$$

In the first set of equations, Eq. (2.14), M_X^i indicates the mass of the fraction variable transferred from the originating cell j into the target cell k by the fraction exchange process $P_{\mathrm{ex},\,X}^i$. In the second set of equations, Eq. (2.15), M_X^i and M_Y^i refer to the masses of the input and output fraction variables of a transformation process $P_{\mathrm{tr},\,X\to Y}^i$.

Following Eq. (2.11), fraction transformation processes $P_{\mathrm{tr},\,X\to Y}^i$ are proportional to the relative amount R_X^i of mass of the input fraction variable M_X^i compared to the bulk mass M_X at the beginning of the time step t_0 in the considered grid cell j and the value of the corresponding bulk process $P_{\mathrm{tr},\,X\to Y}$ for the time step Δt. Accordingly, fraction exchange processes $P_{\mathrm{ex},\,X}^i$ are proportional to the relative amount R_X^i of mass of the transferred fraction variable M_X^i compared to the bulk mass M_X at the beginning of the time step t_0 in the originating grid cell j and the value of the bulk process $P_{\mathrm{ex},\,X}$ at the interface I_{jk} between originating cell j and target cell k for the time step Δt. Thus, the two types of fraction processes are calculated as follows:

$$\begin{aligned} P_{\mathrm{ex},\,X}^i(\Delta t, I_{jk}) &= R_X^i(t_0, j) \cdot P_{\mathrm{ex},\,X}(\Delta t, I_{jk}), \qquad (2.16)\\ P_{\mathrm{tr},\,X\to Y}^i(\Delta t, j) &= R_X^i(t_0, j) \cdot P_{\mathrm{tr},\,X\to Y}(\Delta t, j). \qquad (2.17) \end{aligned}$$

The relative fraction R_X^i is calculated as:

$$R_X^i(t_0, j) = \frac{M_X^i(t_0, j)}{M_X(t_0, j)}, \tag{2.18}$$

with j indicating either the originating grid cell in the case of an exchange process (Eq. (2.16)) or j indicating the grid cell in which the transformation takes place in the case of a transformation process (Eq. (2.17)).

By applying Eqs. (2.14)–(2.18) the fraction processes and state variables can be diagnostically calculated using the corresponding bulk processes and state variables. Technically speaking, an additional set of diagnostic differential equations is introduced for each labelled source and element meaning an increase in the number of state variables and processes proportional to the number of labelled input sources.

Chapter 3

The ETRAC software

Existing TBNT studies (Wijsman et al., 2004; Blauw et al., 2006; Ménesguen et al., 2006; Lacroix et al., 2007; Neumann, 2007; Timmermann et al., 2010; Radtke et al., 2012; Troost et al., 2013; Perrot et al., 2014; Radtke and Maar, 2016) are based on the direct implementation of the TBNT method into the biogeochemical base model. That means, additions to the model code were required before being able to calculate TBNT within the model, and consequently, that each application of TBNT is completely model-dependent. In addition, the increasing number of model variables and processes causes an increase in memory usage and computation time. Radtke et al. (2012) circumvented the manual implementation using the so-called 'code generation tool' (http://www.ergom.net/index.php/code-generation-tool.html) to automatically generate a model code including labeled quantities. However, this software requires model-dependent code templates and detailed abstracted descriptions of the processes represented in the base model.

The simple linear relation between fraction processes and state variables and their corresponding bulk quantities (see Eqs. (2.15)–(2.18)) allows for the calculation of fraction processes and state variables using a post-processing software. The only information needed is the initial distribution of the bulk state variables and the magnitude of the bulk processes as well as the input rates of the labeled state variables (e.g., daily rate of riverine NO_3^- input). Große (2017), for the first time, presents the implementation of the TBNT method as a post-processing approach using the here described ETRAC software. The advantage of this approach is that different setups of labeled input sources can be applied to the same base model results, providing a flexible method to investigate the impact of different sources depending on the research question that has to be addressed.

3.1 Setup and data requirements

The ETRAC software applies a one-dimensional (1D) indexing scheme, which means that, independent of the base model grid structure, all wet cells of the model domain are sorted into a 1D vector. This would allow for the application to different types of grids, including regular cubic grids as well as unstructured grids, however, it also makes the provision of a list of cell neighbours per grid cell as well as information about whether a cell is a surface or bottom cell inevitable. The software furthermore applies the simple variable-process relations given by Eqs. (2.15) and (2.14). Thus, information about the input and output state variables related to the different model processes is required. By providing this information the software is able to interpret and use the results of the base model.

Besides this model-related information the user has to provide information defining the TBNT setup, i.e., definition of source groups to be labeled including information about the source type (*river*, *atmospheric* or *open boundary*), number of sub-sources (e.g., in case of multiple rivers grouped together), and input locations input. If required, so-called *target variables* and *areas* can be defined to obtain regionally aggregated information from the TBNT calculation. Target variables can defined as: (1) individual state variables or (2) the sum of various state variables, explicitly included in the base model (e.g., TN: sum of all N-containing state variables). Target areas are defined as certain sub-regions of the model domain and are described by the model grid cells which have to be aggregated to the corresponding target area. Information on the relative contributions by different of the defined target variables will be produced for each target area.

In order to conduct a TBNT analysis with the ETRAC software the user has to provide the following information and data:

(a) *general user information (mandatory):*

- run identifier for TBNT calculation
- year of calculation
- time step on which base model results are stored
- time step on which TBNT calculation shall be conducted
- starting and ending time (of year) of the TBNT calculation
- switch if initialisation file shall be used for fraction variables
- selection of output to be created:
  - relative contributions of fraction variables (incl. output time step)
  - absolute fraction variables and processes (incl. output time step)
  - target variables and areas

(b) *bulk process- and variable-related (mandatory):*

- list of the names of all model bulk processes and related state variables containing the labeled element (e.g., N)
- netCDF (.nc; **?**) data of all model bulk processes and related bulk state variables containing the labeled element

(c) *grid-related (mandatory):*

- list of indices of neighbours for each grid cell according to 1D indexing scheme
- list of indices of bottom and surface cells according to 1D indexing scheme
- list of all model rivers including name and indices of input cells according to 1D indexing scheme

(d) *source-related (mandatory):*

- list of all source groups, including names of sub-sources in case of riverine source groups
- list of grid cell indices of open boundaries and – in case of N labeling – atmospheric input

(e) *fraction variable-related (optional):*

- data of all relative contributions of all fraction variables

(f) *target-related (optional):*

- list of target variables including contributing model state variables
- list of grid cell indices for all target areas according to 1D indexing scheme

In the following, the setup and the data requirements are described in more detail.

3.1.1 The ETRAC setup file: `tbnt_set.nml`

The main setup file for the ETRAC software is the `tbnt_set.nml` file. In this file, all the user-defined input is specified, by providing basic information directly in this file (e.g., run identifier and time setup) and by linking the relevant input files (e.g., the files containing grid information and bulk variables and fluxes). The file is subdivided into multiple `FORTRAN` namelists sorted according to the information they contain. `FORTRAN` namelists start with an ampersand (‘`&`’) followed by the namelist’s name, and end with a slash (‘`/`’). The individual namelists are explained in this section. Each subsection provides a listing of an individual namelist and a table with a description of the function of each namelist entry. The following type abbreviations are used in the tables: `str` → string, `int` → integer.

3.1.1.1 Namelist #1: `tbnt_run_nml`

The namelist `tbnt_run_nml` (see Listing 3.1) provides the basic setup of the TBNT calculation. Its entries are explained in Table 3.1.

```
&tbnt_run_nml
runID         = 'roms854-NGoMex-TBNT'
isWarmStart   = 1
continueWrite = 0
tbnt_set_dir  = '/scratch/grosse/miTBNT_run_tmp/tbnt_setup/'
bulk_nc_dir   = '/scratch/grosse/roms854_TBNT-N_NGoMex_PERFECT_RST/'
bulk_nc_file  = 'tbnt_mch_bio_2001.nc'
init_nc_dir   = '/scratch/grosse/TBNT_roms854_NGoMex'
init_nc_file  = 'roms854-NGoMex-TBNT_2000_relative_fractions.nc'
/
```

Listing 3.1: Namelist: `tbnt_run_nml`. See Table 3.1 for information on individual entries.

Table 3.1: Description of namelist entries: `tbnt_run_nml` (see Listing 3.1).

entry name	type	description
`runID`	`str`	user-defined identifier of TBNT calculation
`isWarmStart`	`int`	0 → new calculation: automatic initial tracer distribution depending on source setup (see Sect. 3.1.1.6) 1 → continued calculation: relative fractions read from initialization file
`continueWrite`	`int`	0 → create new output files (required for new year) 1 → write into existing output files (required for continuing a year)
`tbnt_set_dir`	`str`	directory containing TBNT setup files
`bulk_nc_dir`	`str`	directory containing input netCDF file with grid cell volumes, bulk variables and fluxes
`bulk_nc_file`	`str`	filename of bulk input netCDF file
`init_nc_dir`	`str`	directory containing input netCDF file with relative fractions for initialization (only needed if `isWarmStart`=1)
`init_nc_file`	`str`	filename of input netCDF file with relative fractions for initialization (only needed if `isWarmStart`=1) if `continueWrite`=1: `init_nc_dir` and `init_nc_file` must point to result file from the previous calculation

3.1.1.2 Namelist #2: `time_setup_nml`

The namelist `time_setup_nml` (see Listing 3.2) provides the time setup of the TBNT calculation. Its entries are explained in Table 3.2.

```
&time_setup_nml
year        = 2004
timeStep    = 1440
startStep   = 1
endStep     = 366
subTimeStep = 1440
/
```

Listing 3.2: Namelist: `time_setup_nml`. See Table 3.2 for information on individual entries.

Table 3.2: Description of namelist entries: `tbnt_run_nml` (see Listing 3.2).

entry name	type	description
`year`	`int`	year of calculation
`timeStep`	`int`	time step of bulk netCDF input data (in minutes)
`startStep`	`int`	starting time step of calculation
`endStep`	`int`	ending time step of calculation if `startStep`=0 or `endStep`=0, TBNT calculation is done for entire year
`subTimeStep`	`int`	user-defined fixed time step for TBNT calculation; must be proper divider of `timeStep`

3.1.1.3 Namelist #3: `area_vol_nml`

The namelist `area_vol_nml` (see Listing 3.3) provides the names of the netCDF state variables containing the grid cells' volume and area. Its entries are explained in Table 3.3.

```
&area_vol_nml
areaVar = 'area'
volVar  = 'volume'
/
```

Listing 3.3: Namelist: `area_vol_nml`. See Table 3.3 for information on individual entries.

Table 3.3: Description of namelist entries: `area_vol_nml` (see Listing 3.3).

entry name	type	description
`areaVar`	`str`	name of netCDF variable containing grid cell area (only needed if code compiled with `#undef TBNTmass_fluxes`, i.e., processes are stored as rate of mass change **per area**)
`volVar`	`str`	name of netCDF variable containing grid cell volume

3.1.1.4 Namelist #4: `model_nml`

The namelist `model_nml` (see Listing 3.4) provides the directory and names of the files containing the information on the model grid and state variable interactions, i.e., processes. Its entries are explained in Table 3.4.

```
&model_nml
model_set_dir         = '/scratch/grosse/roms854_NGoMex/model_setup/
   '
model_grid_file       = 'model_grid_ROMS_NGoMex.txt'
model_fluxes_file     = 'model_fluxes_ROMS.txt'
model_dummy_vars_file = 'model_dummy_vars_ROMS.txt'
model_rivers_file     = 'model_rivers_ROMS_NGoMex.txt'
/
```

Listing 3.4: Namelist: `model_nml`. See Table 3.4 for information on individual entries.

Table 3.4: Description of namelist entries: `model_nml` (see Listing 3.4).

entry name	type	description
`model_set_dir`	`str`	directory with model setup files
`model_grid_file`	`str`	name of `ASCII` file with information on neighbouring, surface and bottom cells
`model_fluxes_file`	`str`	name of `ASCII` file with information on model state variable interactions (i.e., processes)
`model_dummy_vars_file`	`str`	name of `ASCII` file with information on model dummy variables (i.e., involved in processes but not included in the output)
`model_rivers_file`	`str`	name of `ASCII` file with information on model river locations

3.1.1.5 Namelist #5: `indexing_nml`

The namelist `indexing_nml` (see Listing 3.5) provides information on the original indexing of the base model. Its entries are explained in Table 3.5.

```
&indexing_nml
dimName_x       = ’xi_rho’
dimName_y       = ’eta_rho’
dimName_z       = ’s_rho’
gridName_x      = ’lon_rho’
gridName_y      = ’lat_rho’
gridName_z      = ’s_rho’
n_x             = 128
n_y             = 64
n_z             = 20
offset_xStart   = 1
offset_xEnd     = 1
offset_yStart   = 1
offset_yEnd     = 1
model_idep_file = ’model_iDep_NGoMex_MCH.txt’
/
```

Listing 3.5: Namelist: `indexing_nml`. See Table 3.5 for information on individual entries.

Table 3.5: Description of namelist entries `indexing_nml` (see Listing 3.5).

entry name	type	description
`dimName_x`	`str`	name of netCDF dimension in x-direction
`dimName_y`	`str`	name of netCDF dimension in y-direction
`dimName_z`	`str`	name of netCDF dimension in z-direction
`gridName_x`	`str`	name of netCDF grid variable for x-direction
`gridName_y`	`str`	name of netCDF grid variable for y-direction
`gridName_z`	`str`	name of netCDF grid variable for z-direction
`n_x`	`int`	number of indices in x-direction
`n_y`	`int`	number of indices in y-direction
`n_z`	`int`	number of indices in z-direction
`offset_xStart`	`int`	starting index offset in x-direction
`offset_xEnd`	`int`	ending index offset in x-direction
`offset_yStart`	`int`	starting index offset in y-direction
`offset_yEnd`	`int`	ending index offset in y-direction
		(no calculation outside offset region; offsets>0 required if mass conservation not guaranteed at open boundaries)
`model_idep_file`	`str`	name of `ASCII` 2D map with number of vertical levels per (x,y) location

3.1.1.6 Namelist #6: fraction_numbers_nml

The namelist fraction_numbers_nml (see Listing 3.6) provides information on the number of elements to be labeled, and the number and types of source groups. Its entries are explained in Table 3.6.

```
&fraction_numbers_nml
nTracedElements = 2
nRiverFractions = 2
nOpenBFractions = 1
nAtmosFractions = 0
initialFraction = 1
/
```

Listing 3.6: Namelist: fraction_numbers_nml. See Table 3.6 for information on individual entries.

Table 3.6: Description of namelist entries: fraction_numbers_nml (see Listing 3.6).

entry name	type	description
nTracedElements	int	number of labeled elements (e.g., 1 if only N or P, 2 if both at once)
nRiverFractions	int	number of river source groups
nOpenBFractions	int	number of open boundary source groups
nAtmosFractions	int	number of atmospheric source groups
initialFraction	int	index of source group to which initial mass in the system is attributed (only needed if isWarmStart=0; see Sect. 3.1.1.1): 0 → *untraced* sources if existent; equally distributed over all source groups otherwise >0 → selected source group; source group order: (1) rivers, (2) open boundaries, (3) atmospheric); order within these groups as defined by user

3.1.1.7 Namelist #7: `element_nml`

The namelist `element_nml` (see Listing 3.7) defines which elements have to be labeled. Its entries are explained in Table 3.7.

```
&element_nml  tracedElement = 'N' /
&element_nml  tracedElement = 'P' /
```

Listing 3.7: Namelist: `element_nml`. See Table 3.7 for information on individual entries.

Table 3.7: Description of namelist entries: `element_nml` (see Listing 3.7).

entry name	type	description
`tracedElement`	`str`	name of labeled element as used in `model_fluxes_file` (see Sect. **??**)

The ETRAC software expects the namelist `element_nml` as often in `tbnt_set.nml` as defined by `nTracedElements` in the namelist `fraction_numbers_nml` (see Sect. 3.1.1.6). If the namelist is defined more often, only the first `nTracedElements` occurrences are evaluated. In the here shown example (Listing 3.6 and Listing 3.7) the number of labeled elements is 2. Therefore, `element_nml` is defined twice, with the first one switching on nitrogen (`'N'`) labeling and the second one switching on phosphorus (`'P'`) labeling.

3.1.1.8 Namelist #8: `linked_fluxes_nml`

The namelist `linked_fluxes_nml` (see Listing 3.8) controls the treatment of ‘linked fluxes’ and provides the name of the file in which those fluxes and their linking state variables are defined. Its entries are explained in Table 3.8. For detailed information on linked fluxes see Sect **??**.

```
&linked_fluxes_nml
linkedFlxOutStep = -1
linkedFlx_file   = 'linked_fluxes_ROMS.txt'
/
```

Listing 3.8: Namelist: `linked_fluxes_nml`. See Table 3.8 for information on individual entries.

Table 3.8: Description of namelist entries: `linked_fluxes_nml` (see Listing 3.8).

entry name	type	description
`linkedFlxOutStep`	`int`	number of TBNT steps between each writing of output for linked fluxes; may vary between −1 and the overall number of TBNT steps (= `endStep` − `startStep` + 1): −1 → no output (i.e., no treatment of linked fluxes) 0 → output only at beginning and end of calculation; cumulated over entire calculation period 1 → output for every TBNT step; cumulated over each step >1 → output written at intervals of user-defined number of TBNT steps; cumulated over each interval
`linkedFlx_file`	`str`	name of `ASCII` file containing information on linked fluxes

3.1.1.9 Namelist #9: `river_groups_nml`

The namelist `river_groups_nml` (see Listing 3.9) defines the names of the river groups and the number of contributing sub-sources. Its entries are explained in Table 3.9.

```
&river_groups_nml
tracedGroupName = ’Mississippi’
nTracedRivers   =  43
/
&river_groups_nml
tracedGroupName = ’Atchafalaya’
nTracedRivers   =  12
/
```

Listing 3.9: Namelist: `river_groups_nml`. See Table 3.9 for information on individual entries.

Table 3.9: Description of namelist entries: `river_groups_nml` (see Listing 3.9).

entry name	type	description
`tracedGroupName`	`str`	user-defined name of river group
`nTracedRivers`	`int`	number of group members, i.e., different river (x,y)-locations

The ETRAC software expects the namelist `river_groups_nml` to be defined as often in `tbnt_set.nml` as defined by `nRiverFractions` in `fraction_numbers_nml` (see Sect. 3.1.1.6). If the namelist is defined more often, only the first `nRiverFractions` occurrences are evaluated. In the here shown example (Listing 3.6 and Listing 3.9) the number of different river groups is 2. Therefore, `river_groups_nml` is defined twice, with the first one defining the *’Mississippi’* as a source group with 43 members and the second one defining the *’Atchafalaya’* group consisting of 12 members.

3.1.1.10 Namelist #10: `river_fraction_nml`

The namelist `river_fraction_nml` (see Listing 3.10) defines the sub-sources (i.e., rivers) of the different river groups. Its entries are explained in Table 3.10.

```
&river_fraction_nml
tracedRivName = ’Mississippi-01’
rivGroupID    =  1
/
[...]
&river_fraction_nml
tracedRivName = ’Mississippi-43’
rivGroupID    =  1
/
&river_fraction_nml
tracedRivName = ’Atchafalaya-01’
rivGroupID    =  2
/
[...]
&river_fraction_nml
tracedRivName = ’Atchafalaya-12’
rivGroupID    =  2
/
```

Listing 3.10: Namelist: `river_fraction_nml`. See Table 3.10 for information on individual entries. ‘`[...]`’ are placeholders for additional entries for each river group.

Table 3.10: Description of namelist entries: `river_fraction_nml` (see Listing 3.10).

entry name	type	description
`tracedRivName`	`str`	user-defined name of contributing river; must match name in `model_rivers_file`
`rivGroupID`	`int`	river group identifier of contributing river

For each river group, the ETRAC software expects the namelist `river_fraction_nml` to be defined as often in `tbnt_set.nml` as defined by `nTracedRivers` in the corresponding `river_groups_nml` (see Sect. 3.1.1.9). If the namelist is defined more often, only the first `nTracedRivers` (total number of all river groups) occurrences are evaluated. In the here shown example (Listing 3.9 and Listing 3.10) the two source groups – *Mississippi* and *Atchfalaya* – consist of 43 and 12 river locations, respectively. Therefore, `river_fraction_nml` is 55 times, with the first 43 entires defining the the members of the *Mississippi* group, and the 12 last entries defining the members of the *Atchafalaya* group.

3.1.1.11 Namelist #11: `openb_fraction_nml`

The namelist `openb_fraction_nmll` (see Listing 3.11) sets the input file with information on open boundary sources. Its entry is explained in Table 3.11.

```
&openb_fraction_nml
openb_file = 'tbnt_openb_source_NGoMex_MCH.txt'
/
```

Listing 3.11: Namelist: `openb_fraction_nml`. See Table 3.11 for information on individual entries.

Table 3.11: Description of namelist entries: `openb_fraction_nml` (see Listing 3.11).

entry name	type	description
`openb_file`	`str`	name of `ASCII` file containing information on open boundary locations and treatment

3.1.1.12 Namelist #12: atmos_fraction_nml

The namelist openb_fraction_nmll (see Listing 3.11) sets the input file with information on atmospheric sources. Its entry is explained in Table 3.12.

```
&atmos_fraction_nml
atmos_file = 'tbnt_atmos_source_NGoMex_MCH.txt'
/
```

Listing 3.12: Namelist: atmos_fraction_nml. See Table 3.12 for information on individual entries.

Table 3.12: Description of namelist entries: atmos_fraction_nml (see Listing 3.12).

entry name	type	description
atmos_file	str	name of ASCII file containing locations of atmospheric input sources

3.1.1.13 Namelist #13: `bulk_bud_nml`

The namelist `bulk_bud_nml` (see Listing 3.13) defines variable name and location for the calculation of a bulk variable mass balance. Its entries are explained in Table 3.13. This is only used when ETRAC was compiled with `define TBNTonly_bulk_bud`.

```
&bulk_bud_nml
balVarName = 'LdetritusN'
iBal       = 50
jBal       = 50
kBal       = 20
/
```

Listing 3.13: Namelist: `bulk_bud_nml`. See Table 3.13 for information on individual entries.

Table 3.13: Description of namelist entries: `bulk_bud_nml` (see Listing 3.13).

entry name	type	description
`balVarName`	`str`	name of variable for which mass balance shall be calculated; name as used in `bulk_nc_file`
`iBal`	`int`	index in x-direction of grid cell for mass balance
`jBal`	`int`	index in y-direction of grid cell for mass balance
`kBal`	`int`	index in z-direction of grid cell for mass balance

3.1.1.14 Namelist #14: output_nml

The namelist output_nml (see Listing 3.13) defines the output to be generated by ETRAC. Its entries are explained in Table 3.14.

```
&output_nml
output_dir       = '/scratch/grosse/roms854_ETRAC/'
relFracOutStep   = 1
absFracOutStep   = 1
targetOutput     = 0
target_area_file = 'target_areas_NGoMex_MCH.txt'
target_vars_file = 'target_variables_ROMS.txt'
/
```

Listing 3.14: Namelist: output_nml. See Table 3.14 for information on individual entries.

Table 3.14: Description of namelist entries: output_nml (see Listing 3.14).

entry name	type	description
output_dir	str	directory to which output will be written
relFracOutStep	int	number of TBNT steps between each writing of output for relative fraction variables (snapshots); may vary between −1 and the overall number of TBNT steps (= endStep − startStep + 1): −1 → no output 0 → output only at beginning and end of calculation 1 → output for every TBNT step >1 → output written at intervals of user-defined number of TBNT steps
absFracOutStep	int	same as relFracOutStep but for absolute fractions (variables & fluxes); fluxes are cumulated over output interval
targetOutput	int	switch for enabling (1) or disabling (0) spatially aggregated output for target variables (see Sect. ??)
target_area_file	str	name of ASCII file containing list grid cell indices of individual target areas
target_vars_file	str	name of ASCII file containing list of target variables (incl. contributing state variables)

3.1.2 ETRAC input files

In this section the different input files, defined in the previously described setup file (`tbnt_set.nml`), are described in more detail. The files will be described in the order of their occurrence in the setup file. A set of example file can be found in the supplementary material for the 'Northern Gulf of Mexico (NGoMex)' test case (see Appendix **??**).

3.1.2.1 The netCDF input files

The bulk variables and fluxes file: `bulk_nc_file`

The ETRAC software expects a netCDF file generated by the model for which the TBNT analysis shall be conducted. This file must contain the temporally and spatially resolved information on all state variables and fluxes (or processes) of the selected elemental cycle(s)(see Sects. 3.1.1.6 and 3.1.1.7) as well as grid cell volumes. In addition, the grid cell area needs to be part of the stored data if the fluxes are stored per unit area. If the treatment of linked fluxes is enabled (see Sect. 3.1.1.8), these fluxes must also be included in the input file. Furthermore, the output has to fulfil the requirements implied by Eqs. (2.12) and (2.13).

In the current version of the model, that only works for rectangular, three-dimensional grids, the grid structure of the input file must correspond with the grid dimensions defined in namelist `indexing_nml` (`n_x`, `n_y`, `n_z`; see Sect. 3.1.1.5). The time range covered by the file must correspond with that defined by `timeStep`, `startStep` and `endStep` in namelist `time_setup_nml` (see Sect. 3.1.1.2).

The initialization file: `init_nc_file`

In case of running a 'warm-started'/'hot-started' TBNT analysis, the ETRAC software expects a netCDF file generated by a previous ETRAC run. This file is generated when `relFracOutStep` ≥ 1 in namelist `output_nml` (see Sect. 3.1.1.14). This file must contain the spatially resolved information on all relative variable fractions of the selected elemental cycle(s) and for the defined source setup (see Sects. 3.1.1.6 and 3.1.1.10–3.1.1.12) and initial time (`startStep`). If the warm start is done at some point in time during the course of a year, the file must contain output on the same interval as defined by `timeStep`. Otherwise, i.e., if an entire year is shall be calculated an input file with output for the last time step of the previous year only is sufficient (`relFracOutStep` $= 0$).

3.1.2.2 Model grid files

The model grid file: `model_grid_file`

The ETRAC software expects an `ASCII` file containing the information on the model grid structure (see Sect. 3.1.1.4). These information include:

- number of wet (i.e., not on land) cells in the model domain,
- number of spatial dimensions (default: 3),

- lists of grid indices of neighbouring grid cells for each grid cell and spatial dimension,
- number and list of surface grid cell indices,
- number and list of bottom grid cell indices.

The maximum depth index file: `model_idep_file`

The ETRAC software expects an `ASCII` file containing a two-dimensional map with the index of the maximum depth level of each model water column (see Sect. 3.1.1.5). This file contains an integer indicating whether the maps latitudinal indexing corresponds with the indexing of the netCDF input files (0; first line of map corresponds with latitudinal index 1 of netCDF file, i.e., bottom of file matrix) or not ($\neq 0$). This value is followed by an integer map with the maximum depth index of each grid cell. A value of 0 implies that a grid cell is on land.

3.1.2.3 Model fluxes and variables files

The variable interactions file: `model_fluxes_file`

The ETRAC software expects an `ASCII` file containing the information on the model state variable interactions, i.e., the model fluxes (see Sect. 3.1.1.4). For each flux, the following information must be provided:

- element of whose cycle the flux is part of, and by which the fluxes relevant to the TBNT analysis are identified (see Sect. 3.1.1.7),
- name of the flux as used in the netCDF input file (see Sect. 3.1.2.1),
- type of the flux used during TBNT calculation (for details see Table 3.15),
- directional component of the flux: 0 for fluxes within a single grid cell; >0 for fluxes between two grid cells: (1) first component, (2) second etc.
- name and type of input and output variables as used in the netCDF input file (see Sect. 3.1.2.1; for details see Table 3.16),
- name, type and conversion factor of auxiliary flux (only required if flux has to be calculated from flux with different element).

The 'dummy' variables file: `model_dummy_vars_file`

In case one or more model fluxes provided in `model_fluxes_file` have "open ends" (i.e., input/output variables not stored in `bulk_nc_file`), the ETRAC software expects an `ASCII` file containing the information about these 'dummy' variables (see Sect. 3.1.1.4). This file contains a list of all dummy variables with the names of the corresponding auxiliary variables and conversion factors used to calculate each dummy variable.

The linked fluxes file: `linkedFlx_file`

Table 3.15: Flux types defined within the ETRAC software. '3D' refers to three-dimensional fluxes defined on the entire grid (i.e., each single wet grid cell), '2D' refers to two-dimensional fluxes only defined at the corresponding interface (i.e., sea surface or sea floor). Atmospheric, air-sea and sediment fluxes are usually 2D but may be 3D (with zeros everywhere except at the interface).

type ID	description
1	3D prognostic pelagic flux (from one 3D variable to another 3D variable, inside a single grid cell)
2	2D atmospheric deposition (i.e., uni-directional atmospheric input)
3	3D atmospheric deposition (i.e., uni-directional atmospheric input)
4	2D sediment flux (between sediment and deepest pelagic layer)
5	3D sediment flux (between sediment and deepest pelagic layer)
6	advective transport (from one 3D variable to the same 3D variable, exchange between two cells)
7	diffusive transport (from one 3D variable to the same 3D variable, exchange between two cells)
8	river input
9	precipitation flux
10	2D air-sea flux (gain from/loss to atmosphere)
11	3D air-sea flux (gain from/loss to atmosphere)
12	2D diagnostic flux (i.e., derived from 2D auxiliary flux)
13	3D diagnostic flux (i.e., derived from 3D auxiliary flux)

The ETRAC software expects an `ASCII` file containing the information on fluxes that are linked to defined labeled state variables if linked fluxes are switched on (`linkedFlxOutStep`≥0). (see Sect. 3.1.1.8). For instance, such flux could be oxygen production during primary production which is linked to labeled phytoplankton nitrogen. The file contains a list of all such fluxes incl. their names, types (according to Table 3.15) and linking state variable.

3.1.2.4 The input source files

The river location file: `model_rivers_file`

The ETRAC software expects an `ASCII` file containing the information about the river input locations (see Sect. 3.1.1.4). This file contains the number of (x,y) river locations in the domain and the maximum number of (vertical) grid cells per river location, followed by a list of all model rivers incl. their input locations and names. The names must correspond to those names used in the namelist(s) `river_fraction_nml`, written in capital letters.

The open-boundary sources file: `atmos_fraction_file`

The ETRAC software expects an `ASCII` file containing the information about the open-boundary input sources (see Sect. 3.1.1.11). This file contains the number of different open-boundary source groups, and number of input locations per group. Open-boundary sources make use of the horizontal exchange fluxes (advection and diffusion), hence, for each individual input location the following information must

Table 3.16: Variable types defined within the ETRAC software. '3D' refers to three-dimensional variables defined on the entire grid (i.e., each single wet grid cell), '2D' refers to two-dimensional variables only defined at sea surface or sea floor (e.g., benthic organic matter).

type ID	description
1	3D prognostic variable
2	2D prognostic variable
3	3D diagnostic variable (i.e., derived from 3D auxiliary variable)[1]
4	2D diagnostic variable (i.e., derived from 2D auxiliary variable)[1]
5	3D 'dummy' variable (i.e., not existent in output of applied model but open end of one or more fluxes; created internally during TBNT calculation)
6	2D 'dummy' variable (i.e., not existent in output of applied model but open end of one or more fluxes; created internally during TBNT calculation)

[1] obsolete

be provided: originating and target grid cell indices (according to the 1D indexing scheme), directional component of the flux between the two cells and sign of the flux relative to the positive flux direction.

The atmospheric sources file: `atmos_fraction_file`

If at least one of the labeled elements' cycles includes atmospheric fluxes, the ETRAC software expects an `ASCII` file containing the information about the atmospheric input sources (see Sect. 3.1.1.12). This file contains the number of different atmospheric source groups, and number of input locations per group, followed by the input grid cell indices of each group according to the 1D indexing scheme.

3.1.2.5 The target areas and variables files

The target areas file: `target_area_file`

If the spatially aggregated output for defined target variables and areas is enabled (`targetOutput`=1), the ETRAC software expects an `ASCII` file containing the information about the different target areas (see Sect. 3.1.1.14). The file contains the number of different target areas, their names and numbers of corresponding grid cells, and the list of grid cell indices corresponding to each target area (according to the 1D indexing scheme).

The target variables file: `target_vars_file`

If the spatially aggregated output for defined target variables and areas is enabled (`targetOutput`=1), the ETRAC software expects an `ASCII` file containing the information about the different target variables (see Sect. 3.1.1.14). The file contains the number of different target variables, their names and abbreviations, number and names of contributing variables (the latter as used in the bulk netCDF file), and the time period per year for which the output should be stored.

3.2 Program sequence

The above described user-defined input defines the general setup of the TBNT calculation, i.e., the element(s) being labeled, the different input sources etc. The program sequence of the TBNT software is subdivided into three phases: (1) initialization, (2) calculation, and (3) finalization.

3.2.1 The initialization phase

During the initialization phase, the information provided by the user are read from the input files and the corresponding setup is generated internally. Namely, the following actions are done:

- 'construction' of the model grid according to the 1D indexing scheme,
- creation of variable and flux fields containing all information on variable interactions,
- creation of user-defined source groups containing all input locations of all subsources,
- creation of an additional *untraced* source group; only if one or more river sources are not assigned to a user-defined source group, or one or more surface grid cells are not assigned to an atmospheric source group (if atmospheric fluxes are involved in a labeled element's cycle), and
- creation of masks for target areas and fields for target variables; only if target treatment is switched on.

The introduction of the additional *untraced* source group is required in the described case in order to ensure mass conservation relative to the base model. This is necessary as the overall mass of each state variable is calculated as the sum of all corresponding variable fractions during the TBNT calculation. Thus, if individual input sources are not considered, a mass deficit would evolve. In addition, this solution also allows for the labeling, e.g., of only one source, collecting all non-labeled sources in the *untraced* pool.

After generating the basic setup, the ETRAC software creates the fields used for the fraction variables and fluxes during the TBNT calculation. Those are the fields for the absolute bulk variables and fluxes, as well as those for absolute and relative fraction variables, and absolute fraction fluxes. The absolute bulk variable masses and the relative fractions of the fraction variables are used to calculate the initial distribution of mass attributed to the different fraction variables within the system at the beginning of each TBNT calculation.

In the case of a new calculation (i.e., not a succession of a previous calculation), a spin-up has to be conducted starting from an arbitrary initial distribution of the fraction variables. Depending on the user-defined setup (see Sect. 3.1.1.6), all initial mass within the system may be attributed to a user-defined source group, to the *untraced* source group (if existent) or equally distributed among all source groups (otherwise).

Starting from this initial distribution the actual TBNT calculation is conducted.

3.2.2 The calculation phase and output

During the calculation phase, matter assigned to the different source groups successively enters the system at the input grid cells of each source group according to the related bulk fluxes (e.g., riverine input, horizontal advective/diffusive transport across open boundaries, atmospheric deposition). These fluxes are read from the netCDF bulk input file (see Sect. 3.1.2.1) at the beginning of each main time step (`timeStep`).

By the successive input of mass attributed to the fraction variables at the input locations, the relative contributions of these fraction variables change which affects the related fraction fluxes. Consequently, the signal of each source group introduced at its input locations propagates through the model domain depending on the corresponding input fluxes, as well as the advective and diffusive transports (see Eqs. (2.14) and (2.16)) inside the domain. During the propagation, this signal is transferred between interacting fraction variables of different species of the same source group (e.g., from nitrate to phytoplankton-N; see Eqs. (2.15) and (2.17)). Sinking of organic matter or loss of an element, e.g., of N due to benthic denitrification, may further influence the spatial dispersal of the labeled element within the domain. At the open boundaries, fraction variables of all source groups (user-defined and *untraced*) may leave the model domain depending on their relative contribution and the cross-boundary transport.

3.2.2.1 ETRAC output

During the calculation, the selected output is written with the user-defined output time steps, except for the target output which is written with the time step of the netCDF bulk input file. Relative contributions of all fraction variables are written as instantaneous values into a single file which can later be used as initialization file for a subsequent TBNT calculation. For absolute fraction variables and fluxes an individual output file is created for each source group (including the *untraced* group) containing all fraction variable masses and mass fluxes of the corresponding source group. These individual files can then be used, e.g., to calculate mass balances of individual fraction variables etc. If switched on, the results for the linked fluxes for all source groups are written into a single file. All these files are written in netCDF format, and their data structure is identical to that of the bulk input file.

The target output is written as comma-separated `ASCII` format (`.csv`). For the latter, a single file is created for each user-defined target variable, containing the overall mass of a variable and the relative contributions by all source groups for each time step and target area.

3.2.3 The finalization phase

During the finalization phase, the created output files are properly closed, and importantly, a mass conservation check is conducted comparing the final sum of the masses of all fraction variables of each species (i.e., the calculated bulk variable) with the corresponding bulk mass read from the bulk input file of the base model. By this, the TBNT software provides a quantitative check whether the software ran properly.

3.3 Source code and compilation

The ETRAC software is written in Fortran and consists of five `.F90` source code files – one program file and four modules. A schematic overview of the dependencies between these files is provided in Fig. 3.1. In the module file `tbnt_common`, variables and subroutines used within all other source files are defined, indicated by the dashed arrows between these files and the `tbnt_common` module. The main program file (`miTBNT`) only uses the top-level subroutines defined in the other three module files: `tbnt_init`, `tbnt_main`, and `tbnt_output`. In `tbnt_init`, all subroutines used during the initialization process are defined. The module `tbnt_main` contains all subroutines used for the actual calculation, while the module `tbnt_output` contains the subroutines required for the creation and writing of the ETRAC output.

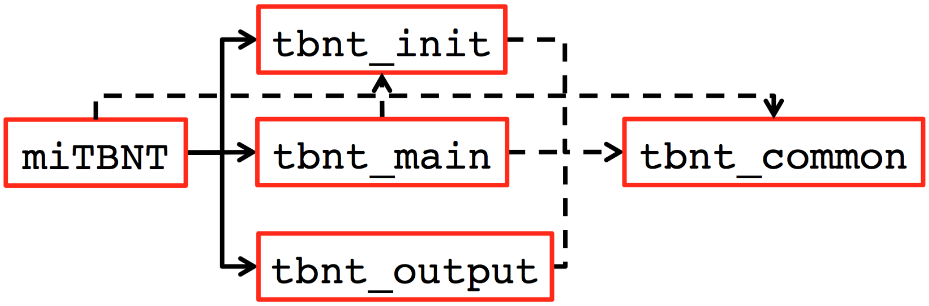

Figure 3.1: Code dependencies between the main program file (`miTBNT`) and the different module files of the ETRAC software. Arrows point from dependent file to required file. Solid arrows indicate the use of the top-level subroutines of a module within the dependent source file, dashed arrows indicate the use of one or more lower-level subroutines, or variables defined within the module.

3.3.1 Compilation

For the compilation of the ETRAC software a `makefile`, using compiler- and machine-dependent configuration files, and a shell script (`compile.sh`) are used. The `makefile` must be located in the same directory as the `.F90` source files, and the configuration files must be located in a subdirectory of that folder named `make-config`. As ETRAC utilizes the netCDF-Fortran library (requiring also the netCDF library) it is necessary to define the correct paths to these libraries, and to load the libraries before compilation. The definition of the paths to the netCDF-Fortran library files is done in the configuration file for the selected compiler. In the `compile.sh` script, the Fortran compiler needs to be selected and the netCDF-Fortran library needs to be loaded. The same applies to the definition of the environment variable containing the path to the configuration file of the netCDF library (`nc-config`). If no `nc-config` file exists on the machine on which ETRAC shall be applied the netCDF libraries need to be defined and loaded manually.

In addition, the correct path of the source code must be defined in the `compile.sh` script. If all this is set up, the source code is compiled by running in a terminal:

```
./compile.sh
```

The compiled code and the executable are stored in the user-defined build directory, also set in the `compile.sh` script.

3.4 Running ETRAC

Chapter 4

Example applications of ETRAC

Bibliography

A.N. Blauw, K. van de Wolfshaar, and H. Meuwese. Transboundary nutrient transports in the North Sea. *WL—Delft Hydraulics Reports*, Z4188, 2006. URL http://publicaties.minienm.nl/download-bijlage/17327/wl-report-z4188-transboundary-nutrient-transorts-in-the-north-sea.pdf. Last accessed: 16-09-2016.

H. Dahlgaard. Transfer of european coastal pollution to the arctic: radioactive tracers. *Marine Pollution Bulletin*, 31(1-3):3–7, 1995.

F. Große. *The influence of nitrogen inputs on the oxygen dynamics of the North Sea.* PhD thesis, Universität Hamburg, Von-Melle-Park 3, 20146 Hamburg, Germany, 2017. http://ediss.sub.uni-hamburg.de/volltexte/2017/8524.

D. Hainbucher, T. Pohlmann, and J.O. Backhaus. Transport of conservative passive tracers in the north sea: first results of a circulation and transport model. *Continental Shelf Research*, 7(10):1161–1179, 1987.

I.H. Harms. Modelling the dispersion of ^{137}cs and ^{239}pu released from dumped waste in the kara sea. *Journal of Marine Systems*, 13(1-4):1–19, 1997.

G. Lacroix, K. Ruddick, N. Gypens, and C. Lancelot. Modelling the relative impact of rivers (Scheldt/Rhine/Seine) and Western Channel waters on the nutrient and diatoms/Phaeocystis distributions in Belgian waters (Southern North Sea). *Cont. Shelf Res.*, 27(10):1422–1446, 2007. doi: 10.1016/j.csr.2007.01.013.

H.J. Lenhart, G. Radach, J.O. Backhaus, and T. Pohlmann. Simulations of the north sea circulation, its variability, and its implementation as hydrodynamical forcing in ersem. *Netherlands Journal of Sea Research*, 33(3-4):271–299, 1995.

A. Ménesguen and T. Hoch. Modelling the biogeochemical cycles of elements limiting primary production in the English Channel. I. Role of thermohaline stratification. *Mar. Ecol.-Prog. Ser.*, 146(1):173–188, 1997. doi: 10.3354/meps146173.

A. Ménesguen, P. Cugier, and I. Leblond. A new numerical technique for tracking chemical species in a multi-source, coastal ecosystem, applied to nitrogen causing Ulva blooms in the Bay of Brest (France). *Limnol. Oceanogr.*, 51(1.2):591–601, 2006. doi: 10.4319/lo.2006.51.1_part_2.0591.

T. Neumann. The fate of river-borne nitrogen in the Baltic Sea–An example for the River Oder. *Estuar. Coast. Shelf S.*, 73(1):1–7, 2007. doi: 10.1016/j.ecss.2006.12.005.

OSPAR. *Results of the 2009 ICG-EMO Workshop on transboundary nutrient transport*, volume 540/2011. OSPAR Commission, London, 2010. ISBN 978-1-907390-81-4. HASEC 11/6/Info.1-E.

T. Perrot, N. Rossi, A. Ménesguen, and F. Dumas. Modelling green macroalgal blooms on the coasts of Brittany, France to enhance water quality management. *J. Marine Syst.*, 132:38–53, 2014. doi: 10.1016/j.jmarsys.2013.12.010.

H. Radtke. Einfluss biologischer Prozesse auf die Ausbreitungswege von Nährstoffen in der Ostsee, 2012. URL http://rosdok.uni-rostock.de/file/rosdok_disshab_0000000981/rosdok_derivate_0000005088/Dissertation_Radtke_2013.pdf. Ph.D. thesis. In German only. Last accessed: 16-09-2016.

H. Radtke and M. Maar. Estimating the effective nitrogen import: An example for the North Sea-Baltic Sea boundary. *J. Geophys. Res.-Biogeo.*, 121(10):2562–2575, 2016. ISSN 2169-8961. doi: 10.1002/2016JG003516.

H. Radtke, T. Neumann, M. Voss, and W. Fennel. Modeling pathways of riverine nitrogen and phosphorus in the Baltic Sea. *J. Geophys. Res.-Oceans (1978–2012)*, 117(C9), 2012. ISSN 2156-2202. doi: 10.1029/2012JC008119.

K. Timmermann, S. Markager, and K.E. Gustafsson. Streams or open sea? Tracing sources and effects of nutrient loadings in a shallow estuary with a 3D hydrodynamic–ecological model. *J. Marine Syst.*, 82(3):111–121, 2010. doi: 10.1016/j.jmarsys.2010.04.008.

T.A. Troost, M. Blaas, and F.J. Los. The role of atmospheric deposition in the eutrophication of the North Sea: a model analysis. *J. Marine Syst.*, 125:101–112, 2013. doi: 10.1016/j.jmarsys.2012.10.005.

J. Wijsman, H. Los, and J. van Beek. The filtering capacity of an estuary for nutrients. *WL—Delft Hydraulics Reports*, Z2836:51pp., 2004.

X. Zhang, R.D. Hetland, M. Marta-Almeida, and S.F. DiMarco. A numerical investigation of the Mississippi and Atchafalaya freshwater transport, filling and flushing times on the Texas-Louisiana Shelf. *Journal of Geophysical Research: Oceans*, 117(C11), 2012. ISSN 2156–2202. doi: 10.1029/2012JC008108. URL http://dx.doi.org/10.1029/2012JC008108. C11009.