



The Earth Science Box Modeling Toolkit (ESBMTK 0.14.0.11): a Python library for research and teaching

Ulrich G. Wortmann¹, Tina Tsan¹, Mahrukh Niazi⁴, Irene A. Ma¹, Ruben Navasardyan², Magnus-Roland Marun¹, Bernardo S. Chede⁶, Jingwen Zhong³, and Morgan Wolfe⁵

¹Department of Earth Sciences, University of Toronto, Toronto, Canada

²Department of Mathematics, University of Toronto, Toronto, Canada

³Department of Statistical Sciences, University of Toronto, Toronto, Canada

⁴Orbio Earth, Cologne, Germany

⁵School of Ocean and Earth Science, University of Southampton, Southampton, United Kingdom

⁶Ocean and Earth Dynamics, Institute of Geosciences, Federal Fluminense University, Niterói, Brazil

Correspondence: Ulrich G. Wortmann (uli.wortmann@utoronto.ca)

Received: 18 June 2024 – Discussion started: 28 August 2024

Revised: 28 December 2024 – Accepted: 2 January 2025 – Published: 27 February 2025

Abstract. The Earth Science Box Modeling Toolkit (ESBMTK) is a Python library that streamlines the creation and analysis of box models in earth sciences. With its modular, object-oriented design, ESBMTK simplifies the study of systems such as the long-term carbon cycle or the impact of atmospheric CO₂ variations on ocean chemistry. By standardizing and clarifying how models are defined, the library enhances code readability and serves as a self-documenting tool, making it approachable for undergraduate students and efficient for researchers. ESBMTK automatically translates user-defined models into equations which are solved using established numerical libraries. It also includes built-in functionality for common tasks such as ocean–atmosphere gas exchange, marine carbonate chemistry, isotope effects, and perturbation scenarios. The library’s core interface is stable, supported by comprehensive documentation, and available as open-source software through the pip and conda package management systems.

marine carbonate system (e.g., Broecker et al., 1999); the GEOCARBSULF model, which describes the evolution of the carbon, oxygen, and sulfur biogeochemical cycles over Phanerozoic times (Berner, 2006); or the LOSCAR model, which models the atmospheric and marine carbon system components and their C-isotope ratios (Zeebe, 2012). Even limiting the citations to a specific subject area like paleoceanography results in a long list of publications demonstrating the importance of box modeling (e.g., Sarmiento and Toggweiler, 1984; Tyrrell, 1999; Wallmann, 2003; Ridgwell, 2003; Tyrrell and Zeebe, 2004; Archer, 2005; Wortmann and Chernyavsky, 2007; Slingerland and Kump, 2011; Markovic et al., 2015; Bachan and Kump, 2015; Luo et al., 2016; Rennie et al., 2018; Yao et al., 2018; Boudreau et al., 2018, 2019; Shields and Mills, 2021; Mills et al., 2021; Paytan et al., 2021).

Box models, unlike more complex earth system models, require minimal computational resources. This allows researchers to focus on specific aspects of the earth system, e.g., how carbonate sediment dissolution mitigates ocean acidification. However, many undergraduate and graduate earth science students lack proficiency in traditional coding languages and differential equation solving, which limits the use of box models in classroom settings. The simplicity and widespread adoption of Python, along with the availability of cloud-based computing environments like Jupyter Notebooks, have expanded coding accessibility beyond traditional

1 Introduction

Box modeling is a versatile tool to explore a variety of earth system processes. Modest hardware requirements facilitate its use for teaching or investigating problems that require long integration times. Prominent examples include, e.g., Harvardton-Bear type models for exploring aspects of the

```

1 from esbmtk import Model, Reservoir, ConnectionProperties
2
3 M = Model(
4     stop="3 Myr", # end time of model
5     max_timestep="1 kyr", # upper limit of time step
6     element=["Phosphor"], # list of element definitions
7     concentration_unit="mol/l")
8
9 Reservoir(
10     name="S_b", # box name
11     volume="3E16 m**3", # surface box volume
12     concentration={M.PO4: "0 umol/l"}) # initial concentration
13
14 Reservoir(
15     name="D_b",
16     volume="100E16 m**3",
17     concentration={M.PO4: "0 umol/l"})
18
19 ConnectionProperties(
20     source=M.D_b, # source of flux
21     sink=M.S_b, # target of flux
22     scale="30 Sv", # rate of flux
23     ctype="scale_with_concentration",
24     id="Thermohaline") # connection id

```

Listing 1. Code fragment showing how to import ESBMTK classes and create `Reservoir` objects (instances; see Fig. 1). The `ConnectionProperties` instance defines the relationship between `Reservoir` instances. In this case, it is a flux that depends on the volume of water and time in Sverdrup ($1 \times 10^6 \text{ m}^2 \text{ s}^{-1}$) and the concentrations of the species in the source `Reservoir`. Note that ESBMTK is unit aware and that the name of the `ConnectionProperties` instance is set implicitly.

audiences. Here, we introduce a Python library that separates model geometry (and processes) from the underlying numerical implementation and thus allows students (and researchers) to focus on the conceptual challenges, rather than mathematical theory. We successfully used this library in undergraduate and graduate teaching, as well as for ongoing research projects.

Our approach is best demonstrated by a simple example. Box models are formulated as a system of coupled ordinary differential equations (ODEs) that describe, e.g., the transfer of matter between `Reservoir` instances (boxes). To give a trivial example (following Glover et al., 2011), let us consider the concentration of phosphate in a two-box ocean. The concentration change in phosphate in the surface box is simply a function of the phosphate fluxes into and out of the box:

$$\frac{d[\text{PO}_4]_S}{dt} = \frac{F_w + F_u - F_d - F_{\text{POP}}}{V_S}, \quad (1)$$

where F_w denotes the PO_4 weathering flux, F_u denotes the PO_4 upwelling flux, F_d denotes the PO_4 flux related to the thermohaline circulation, F_{POP} denotes the PO_4 uptake by primary production, and V_S denotes the volume of the surface box.

While conceptually simple, translating the above into computer code is often beyond the coding skills of many earth science students. Furthermore, with increasing model complexity, the reverse process, i.e., deriving the governing relationships from the program code, becomes considerably more difficult. The Earth Science Box Modeling Toolkit (ESBMTK) aims to address both problems by facilitating a declarative model definition that also serves as the model documentation. Modeling objects (instances in Python) are created by importing the respective ESBMTK classes which are then used to create, e.g., `Reservoir` objects (see e.g., Listing 1).

Class instances can then be combined to build a model, e.g., a `Reservoir` instance (say, for the surface ocean box), which can be connected to a second `Reservoir` instance

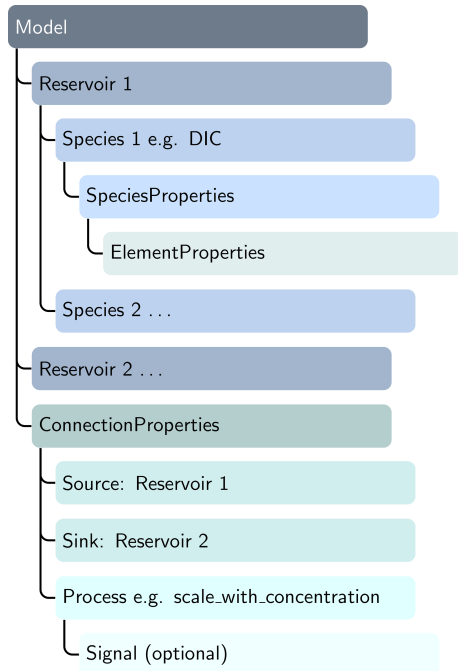


Figure 1. ESBMTK uses a modular hierarchical object structure. `ConnectionProperties` instances are used to establish a source–sink relationship as well as defining the type of transport process (e.g., scaling a flux by concentration; see, e.g., Listing 1). Connection processes can be modulated by external forcings (signal instances) to evaluate the model response to a perturbation. The object-driven model definition reduces code complexity since a change to the number or type of species automatically propagates to the respective connection instances without the need to adapt the code for these changes. The object instances are stateful, which simplifies introspection and debugging. DIC: dissolved inorganic carbon.

(e.g., the deep-ocean box) via a connection instance that specifies their relationship (e.g., scale by concentration; see line 12 in Listing 1). This results in a hierarchical structure that, while verbose, explicitly encodes the model geometry and the relationships between the respective model objects (see Fig. 1).

ESBMTK comes with a wide array of predefined processes to connect boxes (e.g., scale a flux relative to another flux, kinetic and equilibrium isotope effects, sediment dissolution, gas exchange). Additionally ESBMTK provides a variety of methods for post-processing and data management (including graphical output) and leverages standard Python methods for introspection and interactive documentation (see the user guide for details, Wortmann, 2025a). While there is no graphical interface similar to Simulink, this approach significantly reduces coding complexity and model development time. Crucially, the model structure is independent of the numerical implementation. Instead, the model is parsed dynamically to create the necessary equation system which is then passed to an ODE solver library like ODEPACK (Hind-

marsh, 1992). Separating the model description from numerical implementation results in well-documented model code and combines the computational efficiency of state-of-the-art numerical libraries with the ease of use of Python. Presently, the resulting ODE is coded as Python, but it is possible to modify the parser to output the ODE system in other languages (e.g., Julia).

2 Methods

The following sections are not meant as a user guide; rather, they describe implementation details and the underlying assumptions. The user guide and code examples are available online (see the “Code availability” section below).

2.1 Isotope ratios

Several ESBMTK classes have the option to perform stable-isotope-related calculations, with the important caveat that presently there is no structure for isotope systems with more than two isotopes, nor are there provisions to consider radiogenic isotopes. Further, isotope effects during air–sea gas exchange are currently only defined for CO_2 and O_2 . The online user manual describes, however, how to supply the relevant parameters for other gases. In the following, we will only describe the most pertinent implementation details.

To specify the initial isotope ratio of a given species instance, ESBMTK uses the common delta notation; e.g., for ^{34}S and ^{32}S (sulfur) we can write

$$\delta^{34}\text{S} = \left(\frac{\left(\frac{^{34}\text{S}}{^{32}\text{S}} \right)_{\text{Sample}}}{\left(\frac{^{34}\text{S}}{^{32}\text{S}} \right)_{\text{Standard}}} - 1 \right) \times 1000 \text{ [mUr VCDT]}. \quad (2)$$

The unit is in per mil (i.e., per thousand) or milliurey, where $1\text{‰} = 1 \text{ mUr}$ (Brand and Coplen, 2012). Although it is customary to combine the unit with the name of the reference standard (e.g., [mUr VCDT]), ESBMTK currently does not parse isotope units; rather, delta must be provided in units of milliurey.

If a connection between two `Reservoir` instances involves a process that changes the isotope ratio, one can specify the enrichment factor ϵ in the `ConnectionProperties`, where ϵ is defined as

$$\epsilon = (\alpha - 1) \cdot 1000 \text{ [mUr]}, \quad (3)$$

where α equals the isotope fractionation factor between two substances like HCO_3^- and organic matter (OM) during photosynthesis:

$$\alpha_{\text{HCO}_3^- - \text{OM}} = \frac{\left(\frac{^{13}\text{C}}{^{12}\text{C}} \right)_{\text{HCO}_3^-}}{\left(\frac{^{13}\text{C}}{^{12}\text{C}} \right)_{\text{OM}}}. \quad (4)$$

Note that the definition of ϵ is independent of the isotope reference standard, and thus the unit is only given as milliurey. As with delta values, the enrichment factor has to be supplied as a number without units. Internally, ESBMTK only tracks the total concentration and the concentration of the dominant isotope species. The respective delta values are computed once integration has finished. Adding, e.g., isotope fractionation to a given connection (transport process) requires that the respective `Reservoir` instances have been initialized with a defined isotope ratio and that the connection instance specifies the fractionation factor (see Listing 2).

2.2 Weathering

ESBMTK provides a connection type that calculates weathering intensity as a function of CO_2 . The implementation is rather simple and follows Walker et al. (1981):

$$f = A \times f_0 \times \left(\frac{p\text{CO}_2}{p_0\text{CO}_2} \right)^c, \quad (5)$$

where A denotes the area and f_0 denotes the weathering flux at a given reference pressure $p_0\text{CO}_2$. The CO_2 partial pressure at a given time t is denoted as $p\text{CO}_2$, and c is a constant that defines the strength of the weathering (see Walker et al., 1981, and Listing 3). It is however easy to add a new weathering class to ESBMTK that adds a more comprehensive parameterization of weathering processes.

2.3 Seawater properties and equilibrium constants

Provided that the model is specified in units of mol kg^{-1} (i.e., substance content; McNaught and Wilkinson, 2019) and that pressure, temperature, and the concentrations of total alkalinity (TA) and total dissolved inorganic carbon (DIC) are known, ESBMTK can calculate a variety of tracers and dissociation constants. The carbonate system dissociation constants are calculated with the PyCO2SYS library, which provides a choice of 4 different pH scales, 18 different parameterizations for the dissociation constants, and various methods to calculate buffer factors (see Humphreys et al., 2022). This approach not only avoids code duplication but also simplifies the comparison between different models. At present ESBMTK supports PyCO2SYS options to select the pH scale and the parameterizations for the dissociation constants.

The solubility of CO_2 is based on the K_0 value returned by PyCO2SYS, which follows Weiss (1974). ESBMTK reports the CO_2 solubility as SA_{CO_2} in $\text{mol L}^{-1} \text{atm}^{-1}$, corrected for water vapor pressure at sea level as a function of temperature and salinity (see Weiss and Price, 1980). Oxygen solubility is based on the parameters listed in Sarmiento and Gruber (2006), and seawater density is calculated using the equation of state given by Zeebe and Wolf-Gladrow (2001).

It should be noted that, presently, the code assumes that neither temperature nor pressure changes during the model

run. Therefore thermodynamic and kinetic constants are not updated during the model run. In many cases, this is of no concern since, e.g., during glacial–interglacial changes, the changes to the carbonate equilibrium constants are almost fully compensated by the change in ocean volume and the resulting variations in alkalinity (Zeebe and Wolf-Gladrow, 2001). However, this is not universally true and remains an important trade-off between computational efficiency and precision. Future releases will alleviate this shortcoming.

2.4 Carbon chemistry and carbonate dynamics

ESBMTK uses total dissolved inorganic carbon (DIC) and total alkalinity (TA) as master variables to calculate $[\text{H}^+]$ and seawater carbonate speciation. While TA and DIC fully determine the state of the marine carbonate system, solving for $[\text{H}^+]$ is computationally expensive. Follows et al. (2006) demonstrate that if one knows a suitably close estimate for $[\text{H}^+]$ at $t = i$, one can estimate $[\text{H}^+]$ at $t = i + 1$ with sufficient precision from the concentrations of [DIC] and [TA] without computational overhead. Provided that the changes in $[\text{H}^+]$ between integration time steps are smaller than $3 \times 10^{-11} \text{ mol kg}^{-1}$, the associated error is too small to be of concern (Follows et al., 2006). We therefore use the PyCO2SYS library during the model initialization to compute the initial $[\text{H}^+]$ concentration and then use the iterative algorithm of Follows et al. (2006) in subsequent time steps. ESBMTK will print a warning if the change in $[\text{H}^+]$ exceeds the above threshold. However, during integration, ESBMTK only carries tracers for boron, $[\text{H}^+]$, and $[\text{CO}_2]_{\text{aq}}$. All other carbon species are calculated once the integration finishes.

Carbonate dissolution in the water column and sediments is a function of the saturation state which changes with depth. To calculate the resulting burial and dissolution fluxes, one needs a statistical representation of the relationship of sediment area versus depth. ESBMTK approximates this with a hypsometric curve that is based on a 5 min grid that has been downsampled from the Global Bathymetry and Topography at 15 arcsec (SRTM15+ V2.5.5 dataset; Tozer et al., 2019). The flux calculations use the parameterizations proposed by Boudreau et al. (2010a, b). Their approach first calculates specific depth boundaries (i.e., the saturation depth for CaCO_3 z_{sat} or the CaCO_3 compensation depth z_{cc}) as a function of the average CaCO_3 solubility product in the surface ocean ($K_{\text{sp}}^0 = 4.29 \times 10^{-7} \text{ mol}^2 \text{ kg}^{-2}$), a characteristic depth value ($z_{\text{sat}}^0 = 5078 \text{ m b.s.l.}$), and the calcium and carbonate ion concentrations (see Fig. 2 for equations). In the second step, they provide a parameterization of the resulting CaCO_3 burial–dissolution fluxes as a function of the carbonate export flux from the surface ocean and the area between the critical depth intervals (e.g., between z_{sat} and z_{cc}). It should be noted that Boudreau et al. (2010a) do not consider the aragonite dissolution and that their parameterization assumes an idealized mean ocean temperature distribution and homogeneous carbonate ion concentration in the deep-

```

1 Reservoir(
2   name="S_b", # box name
3   volume="3E16 m**3", # surface box volume
4   concentration={M.DIC: "0 umol/l"}, # initial concentration
5   delta={M.DIC: 0}) # initial isotope ratio
6
7 ConnectionProperties(
8   source=M.S_b, # source of flux
9   sink=M.D_b, # target of flux
10  ctype="scale_with_concentration", # connection process
11  scale=Redfield * M.S_b.volume / tau,
12  species={M.DIC}, # apply only to DIC
13  epsilon=-28) # fractionation factor in mUr

```

Listing 2. Code fragment showing how to add isotope calculations to a given model (lines 5 and 13). Note that this code will not run by itself. Working examples are provided in the online documentation.

```

1 Species2Species( # CaCO3 weathering
2   source=M.Fw.DIC, # source of flux
3   sink=M.L_b.DIC, # sink of flux
4   reservoir_ref=M.CO2_At, # pCO
5   scale=1, # optional, defaults to 1
6   ex=0.2, # exponent c
7   pco2_0="280 ppm", # reference pCO2
8   rate="12 Tmol/a", # rate at pco2_0
9   ctype="weathering",
10  id="wca")

```

Listing 3. Code fragment showing how to add a connection between a source (F_w) and sink, where the flux is a function of the CO_2 partial pressure (see Eq. 5).

ocean box (Boudreau et al., 2010b). However, the scheme is computationally efficient and captures transient changes, i.e., times when the snow line and carbonate compensation depth are at different depth levels.

2.5 Gas reservoirs and air–sea gas exchange fluxes

ESBMTK provides a `GasReservoir` class that can be used to track concentration changes in, e.g., $p\text{CO}_2$. In its default setting, this class uses a mass of 1.78×10^{20} mol for the earth’s atmosphere and tracks a given species as the mole ratio relative to the atmosphere. While this class can be used to track several species (e.g., O_2 and $p\text{CO}_2$), they are currently treated as independent of each other. Further, changes in a given species’s concentration will not affect the overall mass of the atmosphere. The error associated with typical variations in $p\text{CO}_2$ is however negligible.

Gas exchange between two `Reservoir` instances is implemented as a connection instance that requires a `GasReservoir` and a regular `Reservoir` instance that carries seawater tracers (see above). The gas exchange implementation follows Zeebe (2012):

$$F_{\text{gas}} = A \cdot u (\beta \cdot p\text{CO}_2 - [\text{CO}_2]_{\text{aq}}), \quad (6)$$

where $[\text{CO}_2]_{\text{aq}}$ denotes the concentration of CO_2 in solution (in mmol kg^{-1}) and $p\text{CO}_2$ denotes the atmospheric CO_2 concentration (in ppm). A denotes the surface area, u denotes the piston velocity, and β denotes the solubility of CO_2 . Currently, ESBMTK provides these parameters for CO_2 and O_2 .

Isotope fractionation effects related to the exchange of CO_2 across the air–sea interface assume that the isotope ratios of HCO_3^- and DIC are roughly equal. This simplification introduces a small error of up to 0.3 mUr at 20 °C and a pH between 7.5 to 8.2 (see Zeebe, 2012), and we calculate the gas exchange flux for ^{13}C as

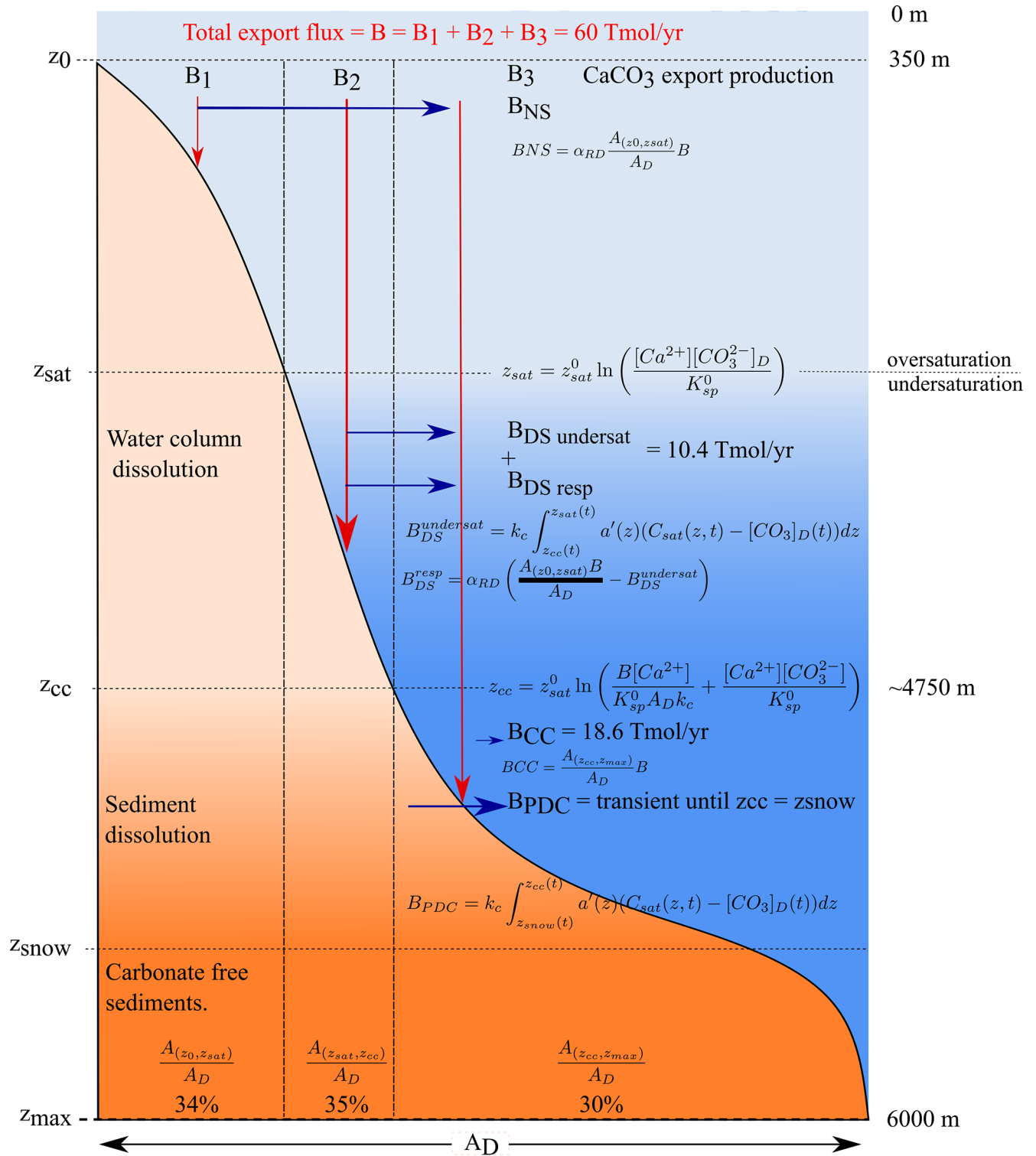


Figure 2. Parametrizations for carbonate burial–dissolution fluxes as proposed by Boudreau et al. (2010a). A denotes cumulative seafloor areas, and B denotes fluxes. The critical depth intervals (z_0 , z_{cc} , z_{snow}) denote the separation between the saturated and undersaturated waters and between carbonate-bearing and carbonate-free sediments. B_{NS} denotes sedimentary calcite dissolution from oxic respiration, B_{DS} denotes the dissolution by respiration in the sediments and in dissolution in the water column, B_{CC} denotes the dissolution below the carbonate dissolution depth, and B_{PDC} denotes the transient dissolution if the depth of z_{cc} and the snow line diverge from each other. α is the fraction of CaCO₃ that dissolves above the saturation horizon z_{sat} .

```

1 Species2Species( # Ocean to atmosphere F7 and F8 in Fig. 3
2     source=M.O2_At, # Reservoir Species
3     sink=M.S_b.O2, # Reservoir Species
4     species=M.O2,
5     solubility="1098 mmol/(m^3 * atm)", # solubility
6     a_u=0.9972, # kinetic fractionation factor
7     a_dg=1.00073, # equilibrium fractionation factor
8     piston_velocity="4.8 m/d",
9     ctype="gasexchange",
10    id="ex_O2",
11 )

```

Listing 4. Example showing how to explicitly specify the equilibrium and kinetic isotope fractionation factors during gas exchange. Note that presently these are not updated during the model run.

$$F_{\text{gas}^{13}\text{C}} = A \cdot u \cdot \alpha_u \left(\beta \cdot \alpha_{\text{dg}} \cdot p^{13}\text{CO}_2 - \alpha_{\text{db}} \cdot R_T \cdot [\text{CO}_2]_{\text{aq}} \right), \quad (7)$$

where α_u denotes the kinetic fractionation factor during gas exchange (equivalent to an ϵ value of 0.8 mUr; Zhang et al., 1995), α_{dg} denotes the equilibrium fractionation factor between CO_2 in solution and CO_2 in gas ($\epsilon = 1.076$ mUr; Zeebe and Wolf-Gladrow, 2001), and α_{db} denotes the equilibrium fractionation between dissolved CO_2 and HCO_3^- ($\epsilon = 9.36$ mUr; Zeebe and Wolf-Gladrow, 2001). ESBMTK provides the respective fractionation factors for CO_2 and O_2 . For other gases, these factors can be specified in the `ConnectionProperties` (see Listing 4).

2.6 Model perturbations

A key element in box modeling studies is to force one or more model boundary conditions, e.g., CO_2 emissions. ESBMTK provides the signal class that implements methods to create square, pyramidal, and bell-shaped signals, as well as a method to read forcing data from a CSV (comma-separated values) file. The signal data can be interpreted either as an absolute flux that is added to an existing flux or as a multiplier that is used to increase/decrease a given flux. Furthermore, signal instances can be added together to create arbitrarily complex shapes. Signal data are automatically truncated and/or padded to match the model time domain, and the data are resampled so that they match the model time step. However, care must be taken that signal duration is at least 4 times as long as the model time step. Signal instances are then associated with one or more connection instances. See the code example in Listing 5.

2.7 Numerical implementation

ESBMTK defaults to an implicit backward differentiating ODE solver that is suitable for the typically stiff

problems in earth sciences. Specifically, we use the `scipy.integrate.BDF` solver as provided by the SciPy library which builds on the algorithms by Byrne and Hindmarsh (1975), Hairer et al. (1993), and Shampine and Reichelt (1997). This algorithm uses a variable time step and automatically increases the time step until the solution becomes unstable. ESBMTK defaults to an initial time step of 1 s. While this seems short given geological timescales, setting this value to a longer time interval has no perceptible influence on the execution time since the solver rapidly increases the integration interval. Conversely, however, setting this value too high can affect the stability of the carbonate system solution. This is particularly true for small-scale models that, e.g., model the acidification of distilled water in a beaker.

A challenge with variable-time-step algorithms is that they cannot account for the nature of episodic events, such as volcanic eruptions or anthropogenic carbon inputs driving the model. The ESBMTK model class thus provides the `max_timestep` keyword that limits the solver to time step values that are smaller than this value. While the BDF (backward differentiation formula) solver is not sensitive to scaling problems (i.e., differences between variables that are very small and those that are very large), its convergence criterion needs to be adjusted for variables that differ by orders of magnitudes. ESBMTK does this based on the initial values of the respective species so that the absolute tolerance value t is

$$t = 10^{-7} \times v, \quad (8)$$

where v denotes a given variable value. In other words, for a concentration value of 28 mM, the solution must be within $\pm 2.8 \times 10^{-6}$ mM.

```

1 Signal(
2   name="CR", # Signal name
3   species=M.PO4, # SpeciesProperties
4   start="1 Myrs",
5   shape="pyramid",
6   duration="500 kyrs",
7   stype="addition"
8   mass="45 Pmol")
9
10 ConnectionProperties(
11   source=M.weathering, # source of flux
12   sink=M.S_b, # target of flux
13   rate=F_w, # rate of flux
14   species=[M.PO4], # species
15   id="river", # connection id
16   ctype="regular", # connection type
17   signal=M.CR, # associate the CR-Signal with this connection
18 )

```

Listing 5. Example on how to create a signal and associate it with a given connection instance to create a transient pulse in the riverine PO₄ flux. The online manual provides further examples.

3 Proof of concept

In order to show the versatility of ESBMTK and to test the model results, we implement the Boudreau et al. (2010a) model using the ESBMTK library. The model code and associated scripts are available online (see the “Code availability” section below). The Boudreau et al. (2010a) model consists of three ocean boxes, one for the low-latitude ocean areas, one for the high-latitude ocean areas, and one for the deep ocean. Additionally, it has a box representing the atmosphere (see Fig. 3).

The model assumes that there is no organic and inorganic export flux from the high-latitude to the deep-ocean box and that the particulate organic matter flux from the low-latitude to the deep-ocean box (F_3) is fully remineralized and has no effect on alkalinity. The carbonate export flux (F_6) is partly dissolved and partly buried (F_2), while the partitioning between F_2 and F_6 depends on the carbon speciation in the deep box. The model uses a fixed rain ratio, where $F_5/F_6 = 0.3$. Alkalinity and dissolved organic carbon are replenished via a constant weathering flux (F_1). The model does not consider phosphor cycling. Thermohaline circulation (F_3) and mixing between the high-latitude and deep-ocean boxes (F_4) redistribute the dissolved species, and gas exchange with the atmosphere balances the concentration of dissolved CO₂ between the low-latitude and high-latitude boxes (F_7 and F_8). Model parameters are given in Tables 1–4.

Boudreau et al. (2010b) use the equilibrium constants parameterization of Millero et al. (2006) and report their re-

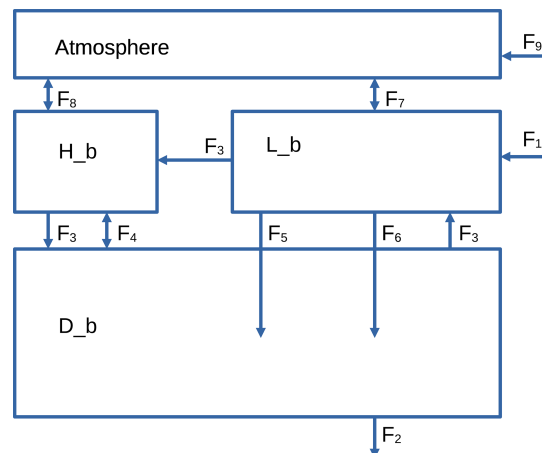


Figure 3. Model geometry used by Boudreau et al. (2010b). See text for flux descriptions and Table 2 for flux values. Note that fluxes can denote more than one species; e.g., F_6 stands for the carbonate export flux that will affect dissolved inorganic carbon (DIC) as well as total alkalinity (TA).

sults on the free pH scale. However, they do not report the fractional value used for the dissolution above the saturation horizon (α_{RD}). Manual tuning of the ESBMTK implementation suggests that a value of 0.6 results in steady-state conditions similar to the values reported by Boudreau et al. (2010b) (see Fig. 4). We then use these steady-state conditions to force the model with a CO₂ pulse (F_9) that is based

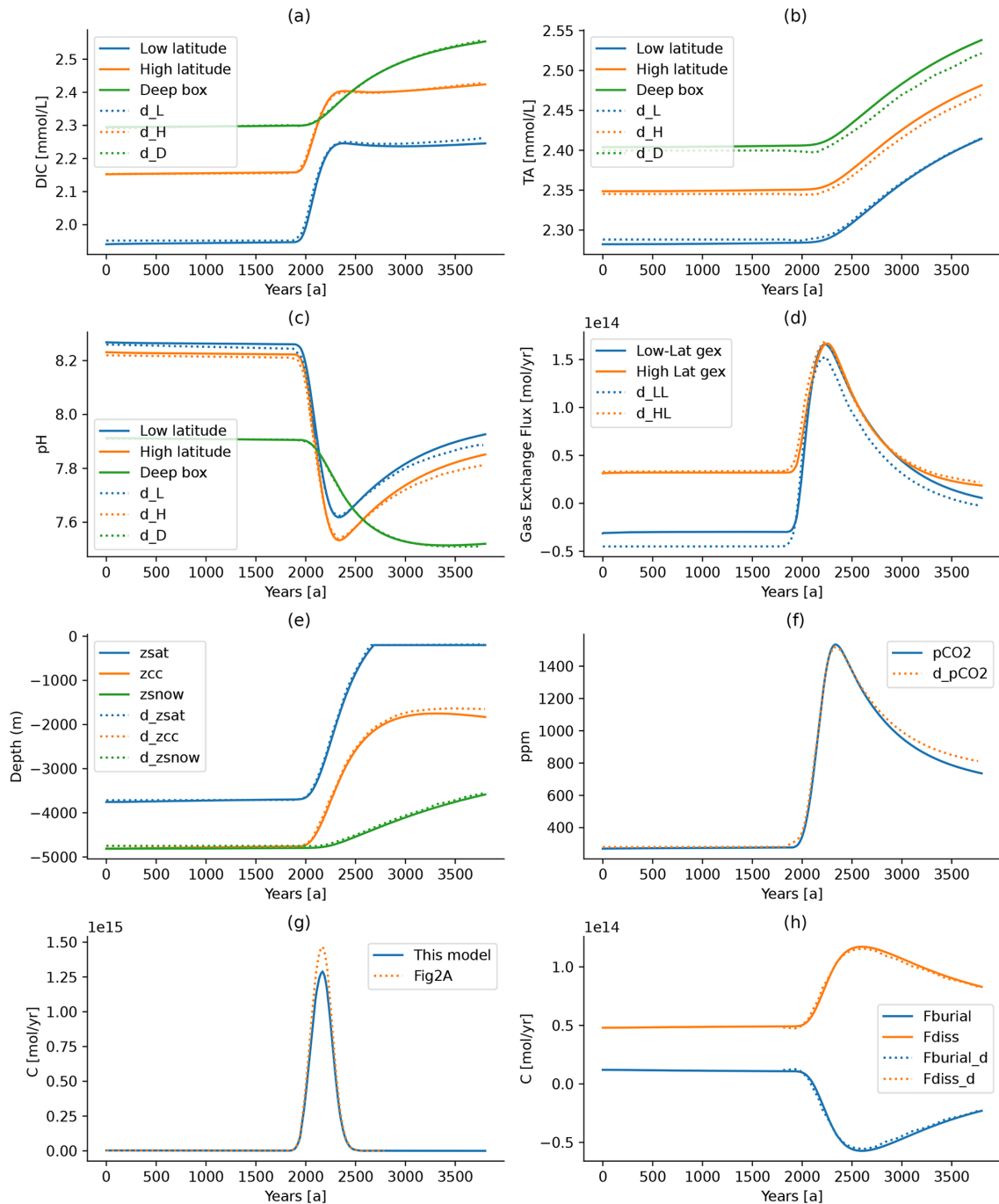


Figure 4. Comparison between the model results reported by Boudreau et al. (2010a) and the ESBMTK-based implementation. Solid lines denote the ESBMTK results, and dotted lines denote data that have been digitized from the figures in Boudreau et al. (2010a). See text for discussion. gex in panel (d) stands for gas exchange.

on the IS92a emission scenario (Leggett et al., 1992) but uses a Gaussian evolution after 2100 that peaks near the year 2250. The total CO₂ emission equals 4025 GtC over 600 years, and Boudreau et al. (2010a) assume that there is no terrestrial carbon uptake. Figure 4 shows a comparison

between the ESBMTK-based model implementation and the data reported by Boudreau et al. (2010a).

Both models demonstrate that the CO₂ release (Fig. 4g) increases the CO₂ fluxes across the air–sea interface (Fig. 4d) and the increase in ocean water acidity due to the dissolution of CO₂. This causes a rapid rise in the saturation

Table 1. Geometry and P – T conditions for the Reservoir boxes in the Boudreau et al. (2010a) model. All boxes use a salinity of 35.

Name	Area [m ²]	Volume [m ³]	P [bar]	T [°C]
H_b	0.5×10^{14}	1.76×10^{16}	17.6	2
L_b	2.85×10^{14}	2.85×10^{16}	5	21.5
D_b	3.36×10^{14}	1.29×10^{18}	240	2

Table 2. Flux parameters as used by Boudreau et al. (2010a). The DIC and alkalinity burial flux F_2 is a function of the export productivity and CO_3^{2-} concentration in the deep box (see Fig. 2). The gas exchange fluxes F_7 and F_8 are a function of the dissolved CO_2 concentrations in the surface boxes.

Name	Symbol	Flux
Weathering DIC	F_1	12 Tmol a ⁻¹
Weathering alkalinity	F_1	24 Tmol a ⁻¹
Burial fluxes	F_2	= F_1
Thermohaline circulation	F_3	25 Sv
Mixing	F_4	30 Sv
Organic matter export	F_5	200 Tmol a ⁻¹
CaCO ₃ export	F_6	60 Tmol a ⁻¹

horizon (z_{sat} ; Fig. 4e), a fairly rapid rise in the carbonate compensation depth (z_{cc} ; Fig. 4e), and a slower rise in the snow line (z_{snow} ; Fig. 4e). Consequently, the carbonate burial flux decreases and the carbonate dissolution flux increases (Fig. 4h), elevating the DIC and TA concentrations in all ocean boxes. The increase in TA enhances the ocean's buffer capacity, leading to a rapid drawdown of atmospheric CO_2 after the year 2320 (Fig. 4f). However, returning to preindustrial steady-state values requires the re-equilibration of the marine alkalinity pool, a process that occurs over hundreds of thousands of years. For a detailed interpretation of the model results, refer to the original publication by Boudreau et al. (2010a).

4 Discussion

The steady-state results of the ESBMTK model not only broadly match the data of Boudreau et al. (2010a) but also show noticeable differences. This is particularly true for the low-latitude ocean, where both the DIC and TA steady-state concentrations are lower than those in the original model (11 and 6 μmol , respectively; see Table 4), which in turn affects the gas exchange fluxes (Fig. 4d). In the deep box, the DIC concentration is 4 μmol higher and the TA concentration is 5 μmol higher than in the original model, resulting in a slightly higher CO_3^{2-} concentration (87.4 versus 86 $\mu\text{mol kg}^{-1}$ in the original model), deepening the location of the critical horizons by about 63 m.

The differences between the low-latitude surface box and the deep ocean are mainly controlled by the export productivity and the burial–dissolution fluxes as well as the thermohaline upwelling. Productivity and upwelling velocity are known constants, and the burial–dissolution fluxes equations are known as well. However, the fraction of carbonate dissolution (α_{RD}) above the saturation horizon is not mentioned by Boudreau et al. (2010a). Increasing α until the surface DIC and TA values better match the original model, however, increases the respective differences in the deep box. This in turn increases the CO_3^{2-} concentrations and deepens the depth of the z_{sat} , z_{cc} , and z_{snow} horizons by another 50 m and further reduces the steady-state $p\text{CO}_2$. Carbon speciation in the deep box would also be affected by the choice of dissociation constants, but it is also conceivable that the differences are caused by the underlying hypsographic data.

We cannot exclude the possibility that there is a numerical error in the ESBMTK library, but it is more likely that the observed variations are caused by small differences in the dissociation constants and/or hypsometric data. Both ESBMTK and Boudreau et al. (2010a) use the carbon dissociation constants parameterization of Millero et al. (2006); however, both rely on third-party libraries (PyCO2SYS and AquaEnv, respectively) to calculate the k values, and we were unable to compare the constants used in our model with the constants used in the original model.

Boudreau et al. (2010a) provide a non-steady-state case to test the response of the system against the release of 4025 Gt over 600 years. We digitized the forcing function for our model from Fig. 2 in Boudreau et al. (2010a). Integration of the digitized data yields a total carbon mass of 4590 Gt C instead of 4025 Gt. We, therefore, scale the digitized data by a factor of 0.877, which results in the differences shown in Fig. 4g. Using the solid line in Fig. 4g as a forcing function, our model yields results that are similar to the original model. While the CaCO₃ burial and dissolution fluxes are similar, the long-term response in the deep-ocean alkalinity is among the more visible differences. However, overall, the ESBMTK implementation replicates the result of the original model well.

5 Conclusions

ESBMTK started as a teaching tool, with the idea of emphasizing model geometry and processes over coding details. This is particularly true for conceptually simple models in combination with Jupyter Notebooks, an approach that has been successfully used in classes of undergraduates that had no previous coding experience. Advanced students with basic Python skills benefit from using ESBMTK by being able to focus on the inherent complexities of model definition, rather than being sidetracked by numerical issues. This approach significantly reduces model development time and ensures that the object-based modeling results in well-documented

Table 3. Biogeochemical rate parameters as used in the ESBMTK version of the Boudreau et al. (2010a) model. With the exception of α , all parameters after Boudreau et al. (2010a).

Parameter	Symbol	Value	Units
Piston velocity	v_G	4.8	m d^{-1}
CaCO_3 dissolution coefficient	k_c	8.84	m yr^{-1}
CaCO_3 solubility at $z = 0$	K_{sp}	4.29×10^{-7}	$\text{mol}^2 \text{kg}^{-1}$
Characteristic depth	z_{sat}^0	5078	m
Ca^{2+} concentration	$[\text{Ca}^{2+}]$	0.0103	mol kg^{-1}
CaCO_3 inventory	I_{CaCO_3}	529	mol m^{-2}
Fraction of CaCO_3 dissolution above z_{sat}	α_{RD}	0.6	

Table 4. Equilibrium constants as used in our implementation of the Boudreau et al. (2010a) model (where K0 denotes the solubility coefficient for CO_2 in seawater, K1 and K2 are the first and second dissociation constants of carbonic acid, KW is the ion product of water, and KB is the dissociation constant of boric acid). These values are computed by PyCO2SYS (Humphreys et al., 2022) based on the P – T values in Table 1 and reported relative to the free pH scale. The concentration values for DIC and TA are the steady-state concentrations in the ESBMTK version of Boudreau et al. (2010a). The steady-state values of the original model are in brackets. The steady-state $p\text{CO}_2$ in the ESBMTK model is 270 ppm; Boudreau et al. (2010a) do not list their steady-state $p\text{CO}_2$.

Box	L_b	H_b	D_b
K0	3.1106×10^{-02}	5.8223×10^{-02}	5.8223×10^{-02}
K1	1.0590×10^{-06}	7.4495×10^{-07}	9.6431×10^{-07}
K2	7.5417×10^{-10}	4.1328×10^{-10}	4.9063×10^{-10}
KW	3.5299×10^{-14}	5.6521×10^{-15}	6.9213×10^{-15}
KB	1.8545×10^{-09}	1.2038×10^{-09}	1.6189×10^{-09}
DIC [$\mu\text{mol kg}^{-1}$]	1941 (1952)	2152 (2153)	2295 (2291)
TA [$\mu\text{mol kg}^{-1}$]	2282 (2288)	2349 (2345)	2404 (2399)

code that is easy to read with a basic understanding of Python syntax. The hierarchical, object-oriented program structure provides a robust framework for experienced Python programmers to adapt or extend the ESBMTK library. These features are also attractive in a research environment, significantly improving readability and reproducibility without incurring major performance penalties.

Rather than implementing our parameterizations for the various equilibrium constants, we use the well-tested PyCO2SYS library, which provides access to a wide range of published equilibrium constants and a choice of different pH scales. At present, carbonate chemistry computations are based on previously published algorithms that are suitable for the modern ocean but will require adaptations for conditions where the deep ocean is warmer than today. Likewise, at present the model is only valid for modern ocean Ca and Mg concentrations and considers calcium carbonate but not aragonite. We also note that the current 0.14.x version of the library does not update kinetic and thermodynamic constants during model execution. Re-implementing a previously pub-

lished model that uses the same carbon chemistry algorithms, we find that the results of both models are in good agreement. We do observe, however, small differences that we attribute to minor variations in the underlying carbon species equilibrium constants.

Code availability. The ESBMTK version that was used to produce the results used in this paper is archived on Zenodo (see Wortmann et al., 2024a, <https://doi.org/10.5281/zenodo.14549407>). The model definition, input data, and the scripts to run the model and produce the plots for the figures used in this paper are available as Wortmann et al. (2024b) (<https://doi.org/10.5281/zenodo.14528185>), and the archived online documentation is available as Wortmann (2025a) (<https://doi.org/10.5281/zenodo.14895031>). The current online documentation is available at <https://esbmtk.readthedocs.io> (esbmtk, 2025), and current versions of ESBMTK are available through the conda and pip package managers as well as from the project website at <https://github.com/uliw/> (Wortmann, 2025b). ESBMTK is published under the GNU Lesser General Public License v3.0.

Data availability. The SRTM15 dataset was published by Tozer et al. (2019). To the best of the authors' knowledge, there is not a persistent URL for this dataset.

Author contributions. UGW initiated this project, secured funding, and wrote the manuscript. TT and MN implemented the carbonate chemistry module and used ESBMTK to implement the Boudreau et al. (2010a) model. RN wrote the first version of the ODE backend, JZ and MW developed the hypsometry code, and BSC and MRM developed the phosphorous feedback module. IAM revised and updated the signal class code.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical rep-

resentation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. Jörg Bollmann provided helpful comments on an early draft of the manuscript, and we thank Shihan Li and an anonymous reviewer for their helpful reviews that improved our paper.

Financial support. This research has been supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant to Ulrich G. Wortmann (DG no. 2018-05873) and fellowship at the Hanse-Wissenschaftskolleg Institute for Advanced Study to Ulrich G. Wortmann. Tina Tsan was supported through a Centre for Global Change Science (CGCS) internship, Magnus-Roland Marun was supported through an NSERC Undergraduate Student Research Award (USRA), Jingwen Zhong was supported through a University of Toronto Excellence Award (UTEA), and Bernardo S. Chede was supported through a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) Programa de Doutorado-sanduiche no Exterior (PDSE) scholarship award (process: CAPES-PrInt, Programa Institucional de Internacionalização; no. 88887.935155/2024-00).

Review statement. This paper was edited by Slimane Bekki and reviewed by Shihan Li and one anonymous referee.

References

- Archer, D.: Fate of Fossil Fuel CO₂ in Geologic Time, *J. Geophys. Res.*, 110, C09S05, <https://doi.org/10.1029/2004jc002625>, 2005.
- Bachan, A. and Kump, L. R.: The Rise of Oxygen and Siderite Oxidation During the Lomagundi Event, *P. Natl. Acad. Sci. USA*, 112, 6562–6567, <https://doi.org/10.1073/pnas.1422319112>, 2015.
- Berner, R. A.: Geocarbsulf: a Combined Model for Phanerozoic Atmospheric O₂ and CO₂, *Geochim. Cosmochim. Ac.*, 70, 5653–5664, <https://doi.org/10.1016/j.gca.2005.11.032>, 2006.
- Boudreau, B. P., Middelburg, J. J., Hofmann, A. F., and Meysman, F. J. R.: Ongoing Transients in Carbonate Compensation, *Global Biogeochem. Cy.*, 24, GB4010, <https://doi.org/10.1029/2009gb003654>, 2010a.
- Boudreau, B. P., Middelburg, J. J., and Meysman, F. J. R.: Carbonate Compensation Dynamics, *Geophys. Res. Lett.*, 37, L03603, <https://doi.org/10.1029/2009gl041847>, 2010b.
- Boudreau, B. P., Middelburg, J. J., and Luo, Y.: The Role of Calcification in Carbonate Compensation, *Nat. Geosci.*, 11, 894–900, <https://doi.org/10.1038/s41561-018-0259-5>, 2018.
- Boudreau, B. P., Middelburg, J. J., Sluijs, A., and van der Ploeg, R.: Secular Variations in the Carbonate Chemistry of the Oceans Over the Cenozoic, *Earth Planet. Sc. Lett.*, 512, 194–206, <https://doi.org/10.1016/j.epsl.2019.02.004>, 2019.
- Brand, W. A. and Coplen, T. B.: Stable Isotope Deltas: Tiny, yet robust signatures in Nature, *Isotop. Environ. Health Stud.*, 48, 393–409, <https://doi.org/10.1080/10256016.2012.666977>, 2012.
- Broecker, W., Lynch-Stieglitz, J., Archer, D., Hofmann, M., Maier-Reimer, E., Marchal, O., Stocker, T., and Gruber, N.: How Strong Is the Harvardton-Bear Constraint?, *Global Biogeochem. Cy.*, 13, 817–820, <https://doi.org/10.1029/1999gb900050>, 1999.
- Byrne, G. D. and Hindmarsh, A. C.: A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations, *ACM T. Math. Softw.*, 1, 71–96, <https://doi.org/10.1145/355626.355636>, 1975.
- esbmtk: The Earth Science Box Modeling Toolkit ESBMTK, <https://esbmtk.readthedocs.io> (last access: 18 February 2025), 2025.
- Follows, M. J., Ito, T., and Dutkiewicz, S.: On the Solution of the Carbonate Chemistry System in Ocean Biogeochemistry Models, *Ocean Model.*, 12, 290–301, <https://doi.org/10.1016/j.ocemod.2005.05.004>, 2006.
- Glover, D. M., Jenkins, W. J., and Doney, S. C.: Modeling Methods for Marine Science, in: 1st Edn., Cambridge University Press, ISBN 978-0-521-86783-2, ISBN 978-0-511-97572-1, <https://doi.org/10.1017/CBO9780511975721>, 2011.
- Hairer, E., Wanner, G., and Nørsett, S. P.: Solving Ordinary Differential Equations I, in: Springer Series in Computational Mathematics, Springer, Berlin, Heidelberg, <https://doi.org/10.1007/978-3-540-78862-1>, 1993.
- Hindmarsh, A. C.: ODEPACK. A Collection of ODE System Solvers, OSTI.GOV, <https://www.osti.gov/biblio/145725> (last access: 19 December 2024), 1992.
- Humphreys, M. P., Lewis, E. R., Sharp, J. D., and Pierrot, D.: Pyco2sys V1.8: Marine Carbonate System Calculations in Python, *Geosci. Model Dev.*, 15, 15–43, <https://doi.org/10.5194/gmd-15-15-2022>, 2022.
- Leggett, J., Pepper, W., Swart, R., Edmonds, J., Meira Filho, L., Mintzer, I., Wang, M., and Watson, J.: Climate Change 1992. the Supplementary Report To the IPCC Scientific Assessment, in: chap. Emissions Scenarios for the IPCC: an Update, Cambridge University Press, 68–95, ISBN 0 521 43829 2, 1992.
- Luo, Y., Boudreau, B. P., and Mucci, A.: Disparate Acidification and Calcium Carbonate Desaturation of Deep and Shallow Waters of the Arctic Ocean, *Nat. Commun.*, 7, 12821, <https://doi.org/10.1038/ncomms12821>, 2016.
- Markovic, S., Paytan, A., and Wortmann, U. G.: Pleistocene Sediment Offloading and the Global Sulfur Cycle, *Biogeosciences*, 12, 3043–3060, <https://doi.org/10.5194/bg-12-3043-2015>, 2015.
- McNaught, A. D. and Wilkinson, A. (Eds.): IUPAC. Compendium of Chemical Terminology, in: 2nd Edn., Blackwell Scientific Publications, Oxford, ISBN 0-9678550-9-8, <https://doi.org/10.1351/goldbook>, 2019.
- Millero, F. J., Graham, T. B., Huang, F., Bustos-Serrano, H., and Pierrot, D.: Dissociation Constants of Carbonic Acid in Seawater As a Function of Salinity and Temperature, *Mar. Chem.*, 100, 80–94, <https://doi.org/10.1016/j.marchem.2005.12.001>, 2006.
- Mills, B. J., Donnadiou, Y., and Godd eris, Y.: Spatial Continuous Integration of Phanerozoic Global Biogeochemistry and Climate, *Gondwana Res.*, 100, 73–86, <https://doi.org/10.1016/j.gr.2021.02.011>, 2021.
- Paytan, A., Griffith, E. M., Eisenhauer, A., Hain, M. P., Wallmann, K., and Ridgwell, A.: A 35-million-Year Record of Seawater Stable Sr Isotopes Reveals a Fluctuating Global Carbon Cycle, *Science*, 371, 1346–1350, <https://doi.org/10.1126/science.aaz9266>, 2021.

- Rennie, V. C. F., Paris, G., Sessions, A. L., Abramovich, S., Turchyn, A. V., and Adkins, J. F.: Cenozoic Record of $\delta^{34}\text{S}$ in Foraminiferal Calcite Implies an Early Eocene Shift to Deep-Ocean Sulfide Burial, *Nat. Geosci.*, 11, 761–765, <https://doi.org/10.1038/s41561-018-0200-y>, 2018.
- Ridgwell, A. J.: Carbonate Deposition, Climate Stability, and Neoproterozoic Ice Ages, *Science*, 302, 859–862, <https://doi.org/10.1126/science.1088342>, 2003.
- Sarmiento, J. L. and Gruber, N.: *Ocean Biogeochemical Dynamics*, Princeton University Press, ISBN 9780691017075, 2006.
- Sarmiento, J. L. and Toggweiler, J. R.: A New Model for the Role of the Oceans in Determining Atmospheric $p\text{CO}_2$, *Nature*, 308, 621–624, <https://doi.org/10.1038/308621a0>, 1984.
- Shampine, L. F. and Reichelt, M. W.: The Matlab Ode Suite, *SIAM J. Sci. Comput.*, 18, 1–22, <https://doi.org/10.1137/s1064827594276424>, 1997.
- Shields, G. A. and Mills, B. J.: Evaporite Weathering and Deposition As a Long-Term Climate Forcing Mechanism, *Geology*, 49, 299–303, <https://doi.org/10.1130/g48146.1>, 2021.
- Slingerland, R. and Kump, L.: *Mathematical Modeling of Earth's Dynamical Systems*, Princeton University Press, Princeton, ISBN 9780691145143, 2011.
- Tozer, B., Sandwell, D. T., Smith, W. H. F., Olson, C., Beale, J. R., and Wessel, P.: Global Bathymetry and Topography At 15 Arc Sec: Srtm15+, *Earth Space Sci.*, 6, 1847–1864, <https://doi.org/10.1029/2019ea000658>, 2019.
- Tyrrell, T.: The Relative Influences of Nitrogen and Phosphorus on Oceanic Primary Production, *Nature*, 400, 525–531, <https://doi.org/10.1038/22941>, 1999.
- Tyrrell, T. and Zeebe, R. E.: History of Carbonate Ion Concentration Over the Last 100 Million Years, *Geochim. Cosmochim. Ac.*, 68, 3521–3530, <https://doi.org/10.1016/j.gca.2004.02.018>, 2004.
- Walker, J. C. G., Hays, P. B., and Kasting, J. F.: A Negative Feedback Mechanism for the Long-term Stabilization of Earth's Surface Temperature, *J. Geophys. Res.-Oceans*, 86, 9776–9782, <https://doi.org/10.1029/jc086ic10p09776>, 1981.
- Wallmann, K.: Feedbacks Between Oceanic Redox States and Marine Productivity: a Model Perspective Focused on Benthic Phosphorus Cycling, *Global Biogeochem. Cy.*, 17, 1084, <https://doi.org/10.1029/2002GB001968>, 2003.
- Weiss, R.: Carbon Dioxide in Water and Seawater: the Solubility of a Non-Ideal Gas, *Mar. Chem.*, 2, 203–215, [https://doi.org/10.1016/0304-4203\(74\)90015-2](https://doi.org/10.1016/0304-4203(74)90015-2), 1974.
- Weiss, R. and Price, B.: Nitrous Oxide Solubility in Water and Seawater, *Mar. Chem.*, 8, 347–359, [https://doi.org/10.1016/0304-4203\(80\)90024-9](https://doi.org/10.1016/0304-4203(80)90024-9), 1980.
- Wortmann, U. G.: ESBMTK Documentation, v0.14.0.11, Zenodo [code], <https://doi.org/10.5281/zenodo.14895031>, 2025a.
- Wortmann, U. G.: Ulrich Wortmann uliw, <https://github.com/uliw/> (last access: 18 February 2025), 2025.
- Wortmann, U. G. and Chernyavsky, B. M.: Effect of Evaporite Deposition on Early Cretaceous Carbon and Sulphur Cycling, *Nature*, 446, 654–656, <https://doi.org/10.1038/nature05693>, 2007.
- Wortmann, U. G., Tsan, T., Niazi, M., Ma, I. A., Navasardyan, R., Roland Marun, M., Zhong, J., Chede, B. S., and Wolfe, M.: The Earth Science Box Modeling Toolkit ESBMTK, v0.14.0.11, Zenodo [code], <https://doi.org/10.5281/zenodo.14549407>, 2024a.
- Wortmann, U. G., Tsan, T., Niazi, M., and Zhong, J.: Example code for the Earth Science Box Modeling Toolkit (ESBMTK), v0.0.0.3, Zenodo [code], <https://doi.org/10.5281/zenodo.14528185>, 2024b.
- Yao, W., Paytan, A., and Wortmann, U. G.: Large-Scale Ocean Deoxygenation During the Paleocene-Eocene Thermal Maximum, *Science*, 361, 804–806, <https://doi.org/10.1126/science.aar8658>, 2018.
- Zeebe, R. E.: LOSCAR: Long-term Ocean-atmosphere-Sediment Carbon cycle Reservoir Model v2.0.4, *Geosci. Model Dev.*, 5, 149–166, <https://doi.org/10.5194/gmd-5-149-2012>, 2012.
- Zeebe, R. E. and Wolf-Gladrow, D. A.: CO_2 in Seawater: Equilibrium, Kinetics, Isotopes, in: no. 65 in *Oceanography Book Series*, Elsevier, ISBN 9780444509468, 2001.
- Zhang, J., Quay, P., and Wilbur, D.: Carbon Isotope Fractionation During Gas-Water Exchange and Dissolution of CO_2 , *Geochim. Cosmochim. Ac.*, 59, 107–114, [https://doi.org/10.1016/0016-7037\(95\)91550-d](https://doi.org/10.1016/0016-7037(95)91550-d), 1995.