~~A python library for box modeling:~~ The Earth Science Box Modeling ~~Toolkit~~ToolKit (ESBMTK 0.14.0.11): A Python Library for Research and Teaching

Ulrich G. Wortmann[1], Tina Tsan[1], Mahrukh Niazi[2], Irene A. Ma[1], Ruben Navasardyan[1], Magnus Roland Marun[1], Bernardo S. Chede[4], Jingwen Zhong[1], and Morgan Wolfe[3]

[1]University of Toronto, Canada
[2]Orbio Earth, Cologne, Germany
[3]University of Southampton, United Kingdom
[4]Federal Fluminense University, Brazil

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

Abstract. The Earth Science Box Modeling Toolkit (ESBMTK) is a Python library ~~designed for building and analyzing~~ that streamlines the creation and analysis of box models in ~~Earth science. It uses a~~ the Earth Sciences. With its modular, object-oriented ~~approach to study topics like~~ design, ESBMTK simplifies the study of systems such as the long-term carbon cycle ~~and~~ or the impact of atmospheric CO$_2$ ~~changes on seawater chemistry. ESBMTK allows users to define models in a straightforward and readable way, which also serves as documentation. These model definitions are then converted into equationsand solved using standard~~ 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. ~~The toolkit includes features for common box modeling~~ It also includes built-in functionality for common tasks such as ~~gas exchange between the ocean and atmosphere~~ocean-atmosphere gas exchange, marine carbonate chemistry, ~~and isotope calculations. ESBMTK has been effectively used in both teaching and research settings. While the libraryis continually being improved, its~~ isotope effects, and perturbation scenarios. The library's core interface is stable~~and comes with extensive documentation~~, supported by comprehensive documentation, and available as open source through the pip and conda package management systems.

1 Introduction

Box modeling is a versatile tool to explore a variety of earth ~~systems~~ system processes. Their modest hardware requirements facilitate their use for teaching, or to investigate problems that require long integration times. Prominent examples include e.g., the Harvardton-Bear type models to explore aspects of the 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)~~(Berner, 2006), or the LOSCAR model, which models the atmospheric and marine carbon system components and their C-isotope ratios ~~Zeebe (2012)~~(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; Shields and Mills 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 ~~can limit~~ limits the use of box models in classroom settings. However, 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 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 (ODE), that describe, e.g., the transfer of matter between reservoirs (boxes). To give a trivial example (following Glover et al. 2011), let's consider the concentration of phosphate in a two-box ocean. The concentration change of phosphate in the surface box is simply a function of the phosphate fluxes into and out of the box :

$$\frac{d[PO_4]_S}{dt} = \frac{F_w + F_u - F_d - F_{POP}}{V_S} \tag{1}$$

where $\mathrm{F_w}$ denotes the $\mathrm{PO_4}$ weathering flux, $\mathrm{F_u}$ the $\mathrm{PO_4}$ upwelling flux, $\mathrm{F_d}$ the $\mathrm{PO_4}$ flux related to the thermohaline circulation, $\mathrm{F_{POP}}$ the $\mathrm{PO_4}$ uptake by primary production, and $\mathrm{V_S}$ 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~~shows how to import the classes, create reservoirs and define their relationships~~).

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 (e.g., the atmosphere box) via a connection instance that specifies their relationship (e.g., ~~gas exchange~~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 etc.). 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 https://esbmtk.readthedocs.io/). While there

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 Reservoirs. In this case, it is a flux that depends on the volume of water/time in Sverdrup (1E6m^2/s) and ~~concentration~~ the concentrations of ~~a given~~ the species in the ~~upstream~~ source reservoir. Note that ESBMTK is unit aware and that the name of the `ConnectionProperties` instance is set implicitly.

```python
from esbmtk import Model, Reservoir, ConnectionProperties

M = Model(
    stop="3 Myr",  # end time of model
    max_timestep="1 kyr",  # upper limit of time step
    element=["Phosphor"],  # list of element definitions
    concentration_unit="mol/l")

Reservoir(
    name="S_b",  # box name
    volume="3E16 m**3",  # surface box volume
    concentration={M.PO4: "0 umol/l"})  # initial concentration

Reservoir(
    name="D_b",
    volume="100E16 m**3",
    concentration={M.PO4: "0 umol/l"})

ConnectionProperties(
    source=M.D_b,  # source of flux
    sink=M.S_b,  # target of flux
    scale="30 Sv",  # rate of flux
    ctype="scale_with_concentration",
    id="Thermohaline")  # connection id
```

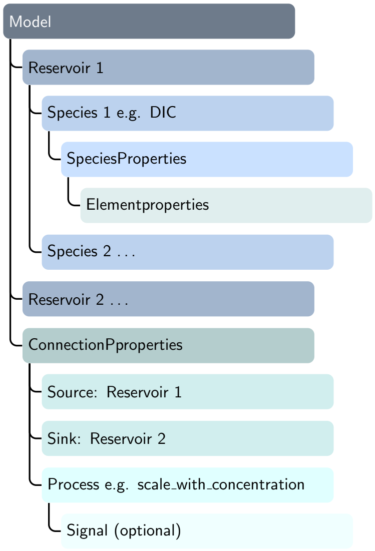

Figure 1. ESBMTK uses a modular hierarchical object structure. ConnectionProperties instances are used to establish a Source/Sink relationship as well defining the type of transport process (e.g., scaling a flux by concentration, see, e.g., Listing 1). Connection processes can be modulated by external forcings (Signal Instances) to evaluate the model response to a perturbation. The object driven model definition ~~, greatly simplifies the model~~ 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.

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 (Hindmarsh, 1992) . Separating 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 ~~or with~~ isotopes, nor are they provisions to consider radiogenic isotopes. Further, isotope effects during ~~Air-Sea~~ air-sea gas exchange are currently only defined for CO_2 and O_2. The online user manual describes~~however~~ , 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 reservoir instance, ESBMTK uses the common delta notation, e.g., for sulfur ^{34}S and ^{32}S we can write:

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

The unit is in permil (i.e., per thousand) or milli Urey, where 1‰ = 1 mUr (Brand and Coplen, 2012). It is customary to combine the unit with the name of the reference standard (e.g., [mUr VCDT]), however, ESBMTK currently does not parse isotope units, rather, delta ~~values a simply given a numeric value where 1 equals 1~~ must be provided in units of mUr.

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

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

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

$$\alpha_{HCO_3^- - OM} = \frac{\left(\frac{^{13}C}{^{12}C}\right)_{HCO_3^-}}{\left(\frac{^{13}C}{^{12}C}\right)_{OM}} \tag{4}$$

Note that the definition of ϵ is independent of the isotope reference standard, and thus the unit is given as mUr only. 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 reservoirs have been initialized with a ~~given~~ defined isotope ratio, and that the connection instance specifies the fractionation factor ~~.~~ ~~Listing #2provides an example~~(see Listing 2).

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  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
```

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)

$$f = A \times f_0 \times \left(\frac{pCO_2}{p_0CO_2}\right)^c \tag{5}$$

where A denotes the area, f_0 the weathering flux at a ~~the~~ given reference pressure p_0CO_2~~,~~. The CO_2 partial pressure at a given time t, is denoted as pCO_2, and c is 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 parametrization of weathering processes.

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)).

```
Species2Species(  # CaCO3 weathering
    source=M.Fw.DIC,  # source of flux
    sink=M.L_b.DIC, # sink of flux
    reservoir_ref=M.CO2_At,  # \DIFdelbegin \DIFdel{pCO2
    }\DIFdelend \DIFaddbegin \DIFadd{pCO
    }\DIFaddend scale=1,  # optional, defaults to 1
    ex=0.2,  # exponent c
    pco2_0="280 ppm",  # reference pCO2
    rate="12 Tmol/a",  # rate at pco2_0
    ctype="weathering",
    id="wca")
```

2.3 Seawater properties and ~~Equilibrium Constants~~equilibrium constants

Provided that the model is specified in units of mol/kg (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 ~~taken from~~ calculated with the pyCO2sys library which ~~unlike hard-coded solutions~~ provides a choice of four different ~~ph-scales~~pH-scales, 18 different parametrizations 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 parametrizations for the dissociation constants.

The solubility of CO_2 is based on the K0 value returned by pyCO2sys, which follows Weiss (1974). ESBMTK reports the CO_2 solubility as `SA_co2` in mol/(t atm) corrected for water vapor pressure at sea level as a function of temperature and salinity ~~following Weiss and Price (1980). The solubility of oxygen follows~~ (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 change 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)~~(Zeebe and Wolf-Gladrow, 2001). However, this is not universally true and remains an important tradeoff between computational efficiency and precision. Future releases will alleviate this shortcoming.

2.4 Carbon Chemistry & Carbonate Dynamics

ESBMTK uses total dissolved inorganic carbon (DIC) and total Alkalinity (TA) as master variables to calculate [H$^+$], and seawater carbonate speciation. While TA and DIC fully determine the state of the marine carbonate system, solving for [H$^+$] is computationally expensive. Follows et al. (2006) ~~where demonstrated~~ demonstrate that if one knows a suitably close estimate for [H$^+$] at t=i, one can estimate [H$^+$] at t=i+1 with sufficient precision from the concentrations of [DIC] and [TA] without computational overhead. Provided that the changes in [H$^+$] between integration time steps is smaller than 3E-11 mol/kg, 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 ~~an initial initial~~ the initial [H$^+$] concentration, and ~~the use iterative algorithm proposed by~~ then use the iterative algorithm of Follows et al. (2006) in subsequent time steps. ESBMTK will print a warning if the change in ~~d~~ [H$^+$] exceeds the above threshold. However~~During~~ , during integration, ESBMTK only carries tracers for boron~~and~~ , [H$^+$] and [CO$_2$]$_{aq}$. 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/dissolution fluxes, one needs a statistical representation of the depth/sediment area relationship in the ocean. ESBMTK approximates this with a hypsometric curve that is based on a 5-minute grid that has been down-sampled from the Global Bathymetry and Topography at 15 Arc Sec (SRTM15+ V2.5.5 dataset, Tozer et al. 2019). The flux calculations use the parametrizations proposed by Boudreau et al. (2010a) and Boudreau et al. (2010b). 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^0_{sp} = 4.29E-7 mol^2/kg^2), a characteristic depth value (z^0_{Sat} = 5078 mbsl), and the calcium and carbonate ion concentrations (see Fig. 2 for equations). In the second step, they provide a parametrization 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 ~~effect of~~ Aragonite dissolution and that their parametrization assumes an idealized mean ocean temperature distribution and homogeneous carbonate ion concentration in the deep ocean box (Boudreau et al., 2010b). However, the scheme is computationally efficient and captures transient changes, i.e., times when the ~~snow-line~~ snow line and carbonate compensation depth are at different depth levels.

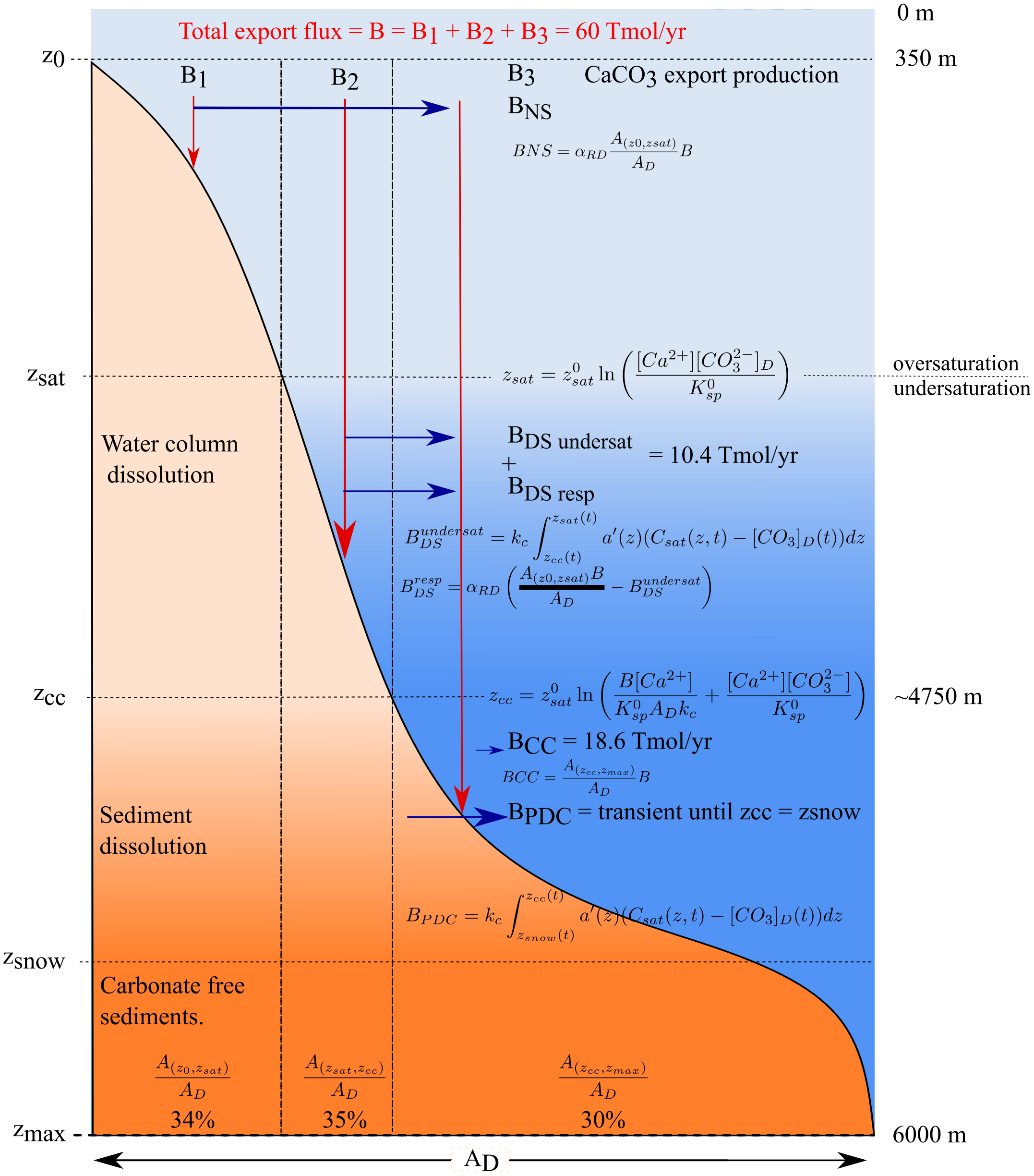
Total export flux = B = $B_1 + B_2 + B_3$ = 60 Tmol/yr
0 m
z_0
350 m
B_1
B_2
B_3
CaCO$_3$ export production
B_{NS}
$BNS = \alpha_{RD} \frac{A_{(z0,zsat)}}{A_D} B$
z_{sat}
$z_{sat} = z^0_{sat} \ln \left(\frac{[Ca^{2+}][CO_3^{2-}]_D}{K^0_{sp}} \right)$
oversaturation
undersaturation
Water column dissolution
B_{DS} undersat
+
B_{DS} resp
= 10.4 Tmol/yr
$B^{undersat}_{DS} = k_c \int_{z_{cc}(t)}^{z_{sat}(t)} a'(z)(C_{sat}(z,t) - [CO_3]_D(t))dz$
$B^{resp}_{DS} = \alpha_{RD} \left(\frac{A_{(z0,zsat)}B}{A_D} - B^{undersat}_{DS} \right)$
z_{cc}
$z_{cc} = z^0_{sat} \ln \left(\frac{B[Ca^{2+}]}{K^0_{sp} A_D k_c} + \frac{[Ca^{2+}][CO_3^{2-}]}{K^0_{sp}} \right)$
~4750 m
B_{CC} = 18.6 Tmol/yr
$BCC = \frac{A_{(z_{cc},z_{max})}}{A_D} B$
Sediment dissolution
B_{PDC} = transient until zcc = zsnow
$B_{PDC} = k_c \int_{z_{snow}(t)}^{z_{cc}(t)} a'(z)(C_{sat}(z,t) - [CO_3]_D(t))dz$
z_{snow}
Carbonate free sediments.
$\frac{A_{(z_0,z_{sat})}}{A_D}$
34%
$\frac{A_{(z_{sat},z_{cc})}}{A_D}$
35%
$\frac{A_{(z_{cc},z_{max})}}{A_D}$
30%
z_{max}
6000 m
A_D

Figure 2. Parametrizations for carbonate burial and dissolution fluxes as proposed by Boudreau et al. (2010a). The letter A denotes cumulative seafloor areas, and the letter 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} the transient dissolution if the depth of z_{cc} and the ~~snowline~~ snow line diverge from each other. α is the fraction of CaCO$_3$ that dissolves above the saturation horizon z_{sat}.

2.5 Gas Reservoirs and Air-Sea Gas Exchange Fluxes

ESBMTK provides a ~~gas-reservoir~~ gas reservoir class that can be used to track concentration changes of, e.g., pCO_2. In its default setting, this class uses a mass of 1.78E20 mol for the earth's atmosphere and tracks a given species as the mol ratio relative to the atmosphere. While this class can be used to track several species (e.g., O_2 and pCO_2), they are currently treated as independent of each other. Further, changes in a given species concentration will not affect the overall mass of the atmosphere. This error associated with typical variations in pCO_2 is however negligible.

Gas exchange between two reservoirs 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)

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

where $[CO_2]_{aq}$ denotes the concentration of CO_2 in solution (in mmol/kg), and pCO_2 denotes the atmospheric CO_2 concentration (in ppm). A denotes the surface area, u the piston velocity, and β 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

$$F_{gas^{13}C} = A \cdot u \cdot \alpha_u \left(\beta \cdot \alpha_{\text{dg}} \cdot p^{13}\text{CO}_2 - \alpha_{\text{db}} \cdot R_{\text{T}} \cdot [CO_2]_{aq} \right) \tag{7}$$

where α_{u} denotes the kinetic fractionation factor during gas exchange (equivalent to and 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 (ϵ=1.076 mUr, Zeebe and Wolf-Gladrow 2001), and α_{db} denotes the equilibrium fractionation between dissolved CO_2 and HCO_3^- (ϵ = 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 connection properties (see Listing 4)

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.

```
\DIFadd{Species2Species(  # Ocean to atmosphere F7 }& \DIFadd{F8 in Fig. 3
    source=M.O2_At,  # Reservoir Species
    sink=M.S_b.O2,  # Reservoir Species
    species=M.O2,
    solubility="1098 mmol/(m^3 * atm)",  # solubility
    a_u=0.9972,  # kinetic fractionation factor
    a_dg=1.00073,  # equilibrium fractionation factor
    piston_velocity="4.8 m/d",
    ctype="gasexchange",
    id="ex_O2",
)
```

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-file. The signal data can either be interpreted 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. ~~Reading signal data from a file, automatically truncates~~ Signal data is automatically truncated and/or ~~pads the data and re-interpolates it~~ padded to match the model time domain, and the data is resampled so that it matches the model time step. However, care must be taken that signal duration is at least four times as long as the model time step. Signal instances are then associated with one or more connection instances. See the code ~~snippet~~ example in listing 5.

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 PO_4 flux. The online manual provides further examples

```
Signal(
    name="CR",  # Signal name
    species=M.PO4,  # SpeciesProperties
    start="1 Myrs",
    shape="pyramid",
    duration="500 kyrs",
    stype="addition"
    mass="45 Pmol")

ConnectionProperties(
    source=M.weathering,  # source of flux
    sink=M.S_b,  # target of flux
    rate=F_w,  # rate of flux
    species=[M.PO4], # species
    id="river",  # connection id
    ctype="regular", # connection type
    signal=M.CR, # associate the CR-Signal with this connection
)
```

2.7 Numerical Implementation

ESBMTK defaults to an implicit backward differentiating ODE solver ~~which~~ that is suitable for the typically stiff problems in the 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 ~~timestep~~ time step of 1 second. While this seems short given geological ~~time scales~~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 ~~complication with variable timestep algorithms is however~~ challenge with variable time step algorithms is that they cannot ~~know~~ account for the nature of episodic events, ~~like driving a model with volcanic~~ such as volcanic eruptions or anthropogenic carbon ~~input~~inputs driving the model. The ESBMTK model class thus provides the `max_timestep` keyword ~~which~~ that limits the solver to time step values that are smaller than this value. While the BDF 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 equals

$$t = 10^{-7} \times v \tag{8}$$

where v denotes a given variable value. In other words, for a concentration value of 28 mM, the solution must be within ±2.8E-6 mM.

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 box for the deep ocean. Additionally, it has a box representing the atmosphere.

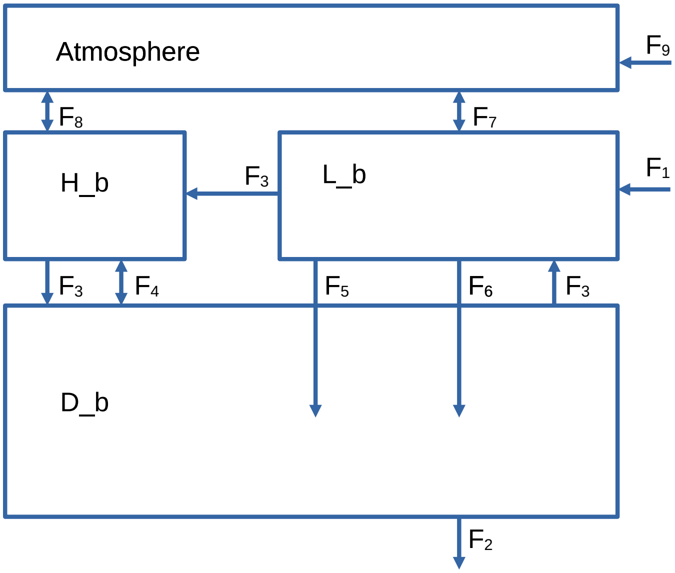

Figure 3. Model geometry used by Boudreau et al. (2010b). See text for flux descriptions, and Tab. 2 for flux values. Note that fluxes can denote more than one species, e.g., F6 stands for the carbonate export flux that will affect dissovced inorganic carbon (DIC) as well as total alkalinity (TA).

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 (~~F53~~F3) is fully remineralized and has no effect on alkalinity. The carbonate export flux (F6) is partly dissolved and partly buried (F2), where the partitioning between F2 and F6 depends on the carbon speciation in the deep box. The model uses a fixed rain ratio where F5/F6 = 0.3. Alkalinity and dissolved organic carbon are replenished via a constant weathering flux (F1). The model does not consider phosphor cycling. Thermohaline circulation (F3) and mixing between the high-latitude and deep ocean boxes (F4) redistribute the dissolved species, and gas exchange with the atmosphere balances the concentration of dissolved CO_2 between the low-latitude and high-latitude boxes (F7 & F8). Model parameters are given in tables 1 to 4.

Boudreau et al. (2010b) use the equilibrium constants parametrization of Millero et al. (2006), and report their results on the free pH scale. However, they do not report the fractional value used for the dissolution above the saturation horizon (α). Manual tuning of the ESBMTK implementation suggests that a value of 0.6 results in steady-state conditions ~~that are~~ 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 CO_2

Name	Area [m^2]	Volume [m^3]	P [bar]	T [°C]
H_b	0.5E14	1.76E16	17.6	2
L_b	2.85E14	2.85E16	5	21.5
D_b	3.36E14	1.29E18	240	2

Table 1. Geometry and PT conditions for the reservoir boxes in the Boudreau et al. (2010a) model. All boxes use a salinity of 35.

Name	Symbol	Flux
Weathering DIC	F1	12 Tmol/a
Weathering Alkalinity	F1	24 Tmol/a
Burial Fluxes	F2	= F1
Thermohaline circulation	F3	25 Sv
Mixing	F4	30 Sv
Organic Matter Export	F5	200 Tmol/a
$CaCO_3$ export	F6	60 Tmol/a

Table 2. Flux parameters as used by Boudreau et al. (2010a). The DIC and Alkalinity burial flux F2 is a function of the export productivity and CO_3^{2-} concentration in the deep box (see Fig. 2). The gas exchange fluxes F7 and F8 are a function of the dissolved CO_2 concentrations in the surface boxes.

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

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).

Box	L_b	H_b	D_b
K0	3.1106e-02	5.8223e-02	5.8223e-02
K1	1.0590e-06	7.4495e-07	9.6431e-07
K2	7.5417e-10	4.1328e-10	4.9063e-10
KW	3.5299e-14	5.6521e-15	6.9213e-15
KB	1.8545e-09	1.2038e-09	1.6189e-09
DIC [μmol/kg]	~~1940~~ 1941 (1952)	~~2151~~ 2152 (2153)	~~2294~~ 2295 (2291)
TA [μmol/kg]	~~2281~~ 2282 (2288)	~~2347~~ 2349 (2345)	~~2403~~ 2404 (2399)

Table 4. Equilibrium constants K_i as used in each box. These values are computed by pyCO2sys (Humphreys et al., 2022) based on the PT values in Tab. 1, and reported relative to the free pH scale. The concentrations 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 pCO_2 in the ESBMTK model is ~~275~~270 ppm, Boudreau et al. (2010a) do not list their steady-state pCO_2.

pulse (F9) that is based on the IS92a emission scenario (Leggett et al., 1992) but uses a Gaussian evolution after 2100 AD that peaks near the year 2250 AD. The total CO_2 emission equals 4025 Gt C over 600 years, and Boudreau et al. (2010a) assume that there is no terrestrial carbon uptake. Fig. 4 shows a comparison between the ESBMTK-based model implementation and the data reported by Boudreau et al. (2010a).

Both models demonstrate that the CO_2 release (Fig. 4 panel g) ~~increase~~ increases the CO_2 fluxes across the air-sea interface (panel d) and the increase in ocean water acidity due to the dissolution of CO_2. This causes a rapid rise of the saturation horizon (zsat, panel e), a fairly rapid rise of the carbonate compensation depth (zcc, panel e), and a slower rise of the snow line (znow, panel e). Consequently, the carbonate burial flux decreases, and the carbonate dissolution flux increases (panel h), elevating the DIC and TA concentrations in all ocean boxes. The increase in TA ~~, enhances the oceans buffer capacity ,~~ enhances the ocean's buffer capacity leading to a rapid drawdown of atmospheric CO_2 after the year 2320 (panel f). 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 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 (~~12 and 7~~ 11 and 6 μmol respectively, see Tab. 4), which in turn affects the gas exchange fluxes (panel d in Fig. 4). In the deep box, the DIC concentration is ~~2.6~~ 4 μmol higher and the TA concentration is ~~4~~ 5 μmol higher than in the original model, resulting in a slightly higher CO_3^{2-} concentration (~~87~~ 87.4 versus 86 μmol/kg in the original model), deepening the location of the critical horizons by about ~~50~~ 63 meters.

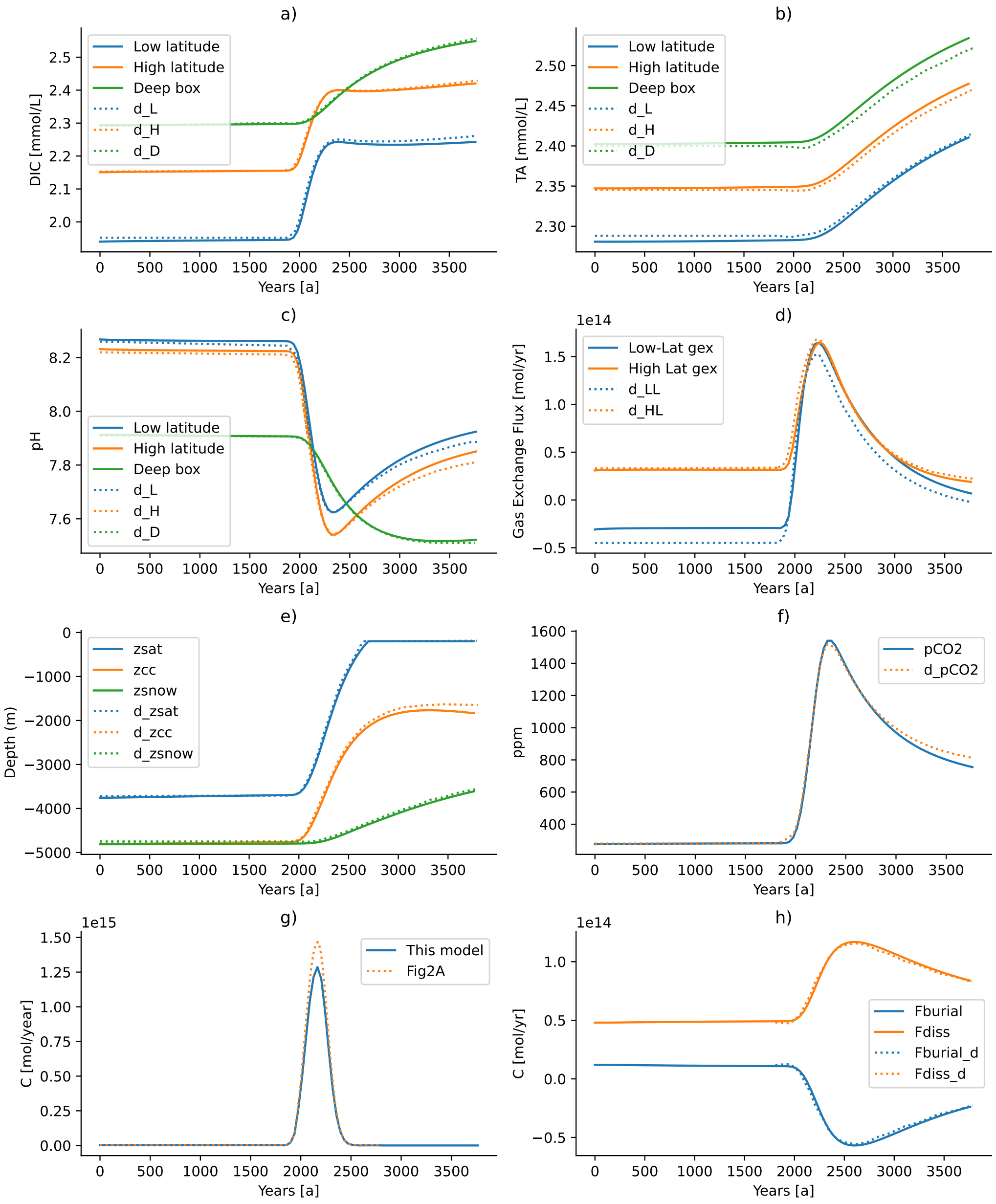

Figure 4. Comparison between the models results reported by Boudreau et al. (2010a) and the ESBMTK-based implementation. Solid lines denote the ESBMTK results, dotted lines denote data that has been digitized from the figures in Boudreau et al. (2010a). See text for discussion.

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) above the saturation horizon is not mentioned by Boudreau et al. (2010a). Increasing α until the surface DIC and TA values ~~are a better match with~~ better match the original model, increases however 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 meters, and further reduces the steady-state pCO_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 parametrization 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 panel g) of Fig. 4. Using the solid line in panel g) as a forcing function, our model yields results that are similar to the original model. While the $CaCO_3$ 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 to emphasize 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 undergraduate classes 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 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 parametrizations 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 adaptions 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 only 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 published 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 ~~which~~ that we attribute to minor variations in the underlying carbon species equilibrium constants.

6 Code availability

~~The current ESBMTK version~~ ESBMTK is available through the conda and pip package managers and from the project website: https://github.com/uliw/esbmtk under the GNU Lesser General Public License v3.0. The documentation is available at https://esbmtk.readthedocs.io and example scripts including the model described in this paper are available at https://github.com/uliw/ESBMTK-Examples. The ~~exact version of the model~~ ESBMTK version that was used to produce the results used in this paper (0.14.0.11) is archived on Zenodo ~~(), as are~~ at https://doi.org/10.5281/zenodo.14549407. The model definition, input data and the scripts to run the model and produce the plots for ~~all the simulations presented~~ the figures used in this paper ~~()~~are available at https://doi.org/10.5281/zenodo.14528185.

7 Author Contribution Statement

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 2010 model. RN wrote the first version of the ODE backend, JZ and MW developed the hypsometry code, BSC and MRM the phosphorous feedback module. IAM, revised and updated the signal class code.

8 Competing Interests

The authors declare that they have no conflict of interest.

9 Acknowledgments

UGW was supported through an NSERC Discovery Grant and a fellowship at the Hanse Institute for Advanced Studies. TT was supported through ~~an NSERC USRA award, MRM through~~ a CGCS internship, and MRM by a NSERC USRA award, JZ through a UofT UTEA award, BSC CAPES PDSE scholarship award (Process: CAPES-PRINT - 88887. 935155/2024-00). Jörg Bollmann provided helpful comments on an early draft of thus manuscript, and we thank Shihan Li and an anonymous reviewer for their helpful reviews that improved our manuscript.

References

Archer, D.: Fate of Fossil Fuel CO_2 in Geologic Time, Journal of Geophysical Research, 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, Proceedings of the National Academy of Sciences, 112, 6562–6567, https://doi.org/10.1073/pnas.1422319112, 2015.

Berner, R. A.: Geocarbsulf: a Combined Model for Phanerozoic Atmospheric O_2 and CO_2, Geochimica et Cosmochimica Acta, 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 Biogeochemical Cycles, 24, https://doi.org/10.1029/2009gb003654, 2010a.

Boudreau, B. P., Middelburg, J. J., and Meysman, F. J. R.: Carbonate Compensation Dynamics, Geophysical Research Letters, 37, https://doi.org/10.1029/2009gl041847, 2010b.

Boudreau, B. P., Middelburg, J. J., and Luo, Y.: The Role of Calcification in Carbonate Compensation, Nature Geoscience, 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 and Planetary Science Letters, 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, Isotopes in Environmental and Health Studies, 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 Biogeochemical Cycles, 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 Transactions on Mathematical Software, 1, 71–96, https://doi.org/10.1145/355626.355636, 1975.

Follows, M. J., Ito, T., and Dutkiewicz, S.: On the Solution of the Carbonate Chemistry System in Ocean Biogeochemistry Models, Ocean Modelling, 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, Cambridge University Press, first edn., ISBN 978-0-521-86783-2 978-0-511-97572-1, https://doi.org/10.1017/CBO9780511975721, 2011.

Hairer, E., Wanner, G., and Norsett, S. P.: Solving Ordinary Differential Equations I, 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, https://www.osti.gov/biblio/145725, 1992.

Humphreys, M. P., Lewis, E. R., Sharp, J. D., and Pierrot, D.: Pyco2sys V1.8: Marine Carbonate System Calculations in Python, Geoscientific Model Development, 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, chap. Emissions Scenarios for the IPCC: an Update, pp. 68–95, Cambridge University Press, 1992.

Luo, Y., Boudreau, B. P., and Mucci, A.: Disparate Acidification and Calcium Carbonate Desaturation of Deep and Shallow Waters of the Arctic Ocean, Nature Communications, 7, 12 821, 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, Blackwell Scientific Publications, Oxford, 2 edn., 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, Marine Chemistry, 100, 80–94, https://doi.org/10.1016/j.marchem.2005.12.001, 2006.

Mills, B. J., Donnadieu, Y., and Goddéris, Y.: Spatial Continuous Integration of Phanerozoic Global Biogeochemistry and Climate, Gondwana Research, 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 δ34S in Foraminiferal Calcite Implies an Early Eocene Shift to Deep-Ocean Sulfide Burial, Nature Geoscience, 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, 2006.

Sarmiento, J. L. and Toggweiler, J. R.: A New Model for the Role of the Oceans in Determining Atmospheric pCO_2, Nature, 308, 621–624, https://doi.org/10.1038/308621a0, 1984.

Shampine, L. F. and Reichelt, M. W.: The Matlab Ode Suite, SIAM Journal on Scientific Computing, 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, 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 and Space Science, 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, Geochimica et Cosmochimica Acta, 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, Journal of Geophysical Research: 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 Biogeochemical Cycles, 17, 1084, http://dx.doi.org/10.1029/2002GB001968, 2003.

Weiss, R.: Carbon Dioxide in Water and Seawater: the Solubility of a Non-Ideal Gas, Marine Chemistry, 2, 203–215, https://doi.org/10.1016/0304-4203(74)90015-2, 1974.

Weiss, R. and Price, B.: Nitrous Oxide Solubility in Water and Seawater, Marine Chemistry, 8, 347–359, https://doi.org/10.1016/0304-4203(80)90024-9, 1980.

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.

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, Geoscientific Model Development, 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, no. 65 in Oceanography Book Series, Elsevier, 2001.

Zhang, J., Quay, P., and Wilbur, D.: Carbon Isotope Fractionation During Gas-Water Exchange and Dissolution of CO_2, Geochimica et Cosmochimica Acta, 59, 107–114, https://doi.org/10.1016/0016-7037(95)91550-d, 1995.