Tutorials for courses OCEA0096-1 and SPAT0025-1
—
Modeling of sedimentary processes The calcite transition zone

Guy Munhoven

Version 1.2.7
(6th March 2024)

Preamble

The most recent version of this document can be found at my Study Projects workshop site, together with supporting scripts and utilities.

Users building the application used here from Open Source Archives on Zenodo as outlined in Appendix A should also refer to that site for the latest version of the PYTHON script `meduti.py` and more recent helper utilities.

1 Introduction

The abundance of carbonate (calcite) in the surface sediment typically decreases with depth of the seabed from about 80–90 % to shallow to less than 5 % at great depth (Fig. 1). The result is a transition zone whose extent and appearance vary according to different parameters:

- Rate of deposition of refractory materials (clays, sands, detrital matter);
- Calcite deposition rate;
- Deposition rate of organic matter;

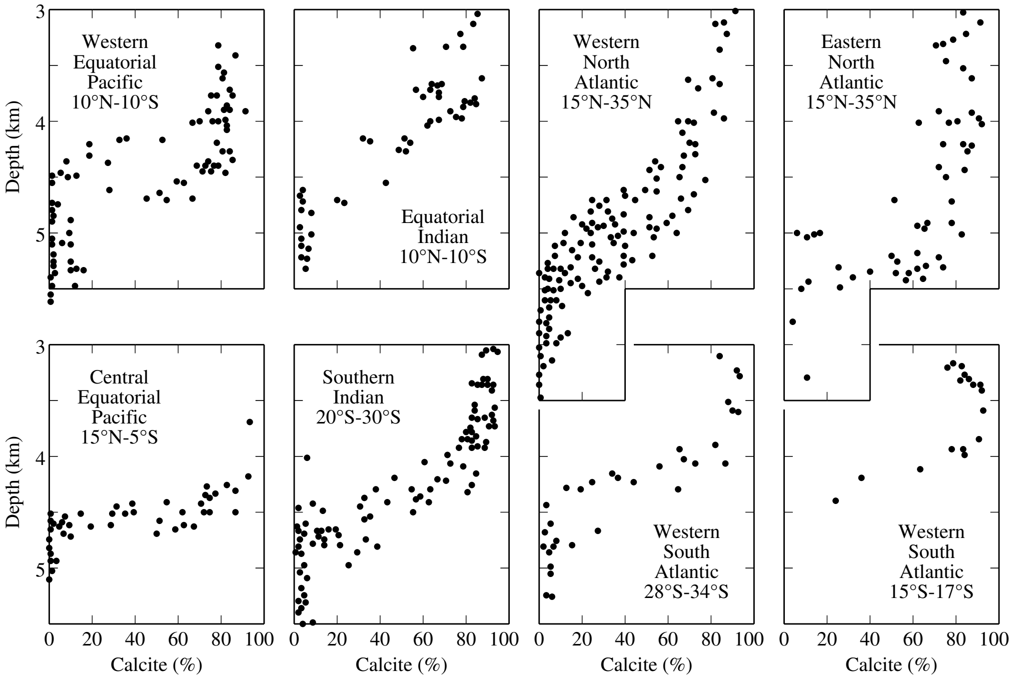

Figure 1: Fraction of $CaCO_3$ in surface sediment as a function of depth, for different regions of the global ocean (Broecker and Peng, *Tracers in the Sea*, 1982)

- Remineralization rate of calcite;
- Oxidation rate of organic matter.

For this project, we will focus on the influence of deposition rates and remineralization rates. For both aspects, the MEDUSA model of early diagenesis in the MEDANA (MEDUSA ANAlysis) version will be used.

The values of all these parameters may vary as a function of sea-floor. For the purpose of the present study, the sea-floor is subdivided into several tens of depth intervals (typically 80 intervals, distributed between 1,000 and 5,000 m depth, each interval thus covering 50 m). Each one of these depth intervals has one sediment column attached. This column covers the topmost 10 cm of the surface sediment where the most important early diagenesis reactions that control the sediment composition take place.

1.1 Model configuration

The version of MEDANA made available for this project is configured as follows:

- composition:
  - CO_2, HCO_3^-, CO_3^{2-}, O_2 et NO_3^- in the porewaters;
  - *clay*, calcite and organic matter (O.M.) in the solid phase.
- equilibria:
  - carbonate system: $CO_2 - HCO_3^- - CO_3^{2-}$
- processes:
  - Calcite ($CaCO_3$) dissolution

    $$\mathrm{CaCO_3} \longrightarrow \mathrm{CO_3^{2-}} + \mathrm{Ca^{2+}}$$

    following the kinetic rate law

    $$T_{\mathrm{calc}} = k_{\mathrm{calc}}\,[\mathrm{calcite}]\,(1 - \Omega_{\mathrm{calc}})^n$$

    where $\Omega_{\mathrm{calc}} = [\mathrm{CO_3^{-2}}]/[\mathrm{CO_3^{-2}}]_{\mathrm{sat}}$ is the degree of saturation of the porewaters with respect to calcite and where $[\mathrm{CO_3^{-2}}]_{\mathrm{sat}} = K_{\mathrm{satcalc}}/[\mathrm{Ca^{2+}}]$ is the saturation concentration of CO_3^{2-} with respect to calcite, derived from the Ca^{2+} concentration (which is parametrized as a function of salinity) and the calcite solubility product, K_{satcalc} (which is calculated as a function of local pressure, temperature and salinity);

- Oxic degradation (respiration) of organic matter

$$\begin{aligned}&(\mathrm{CH_2O})_x(\mathrm{NH_3})_y(\mathrm{H_3PO_4})_z + (x+2y)\,\mathrm{O_2}\\ &\quad\rightarrow\; x\,\mathrm{CO_2} + y\,\mathrm{NO_3^-} + z\,\mathrm{H_2PO_4^-} + (y+z)\,\mathrm{H^+} + (x+y)\,\mathrm{H_2O};\end{aligned}$$

this reaction is reformulated here to take into account the thermodynamic equilibrium between CO_2 and HCO_3^-

$$\mathrm{CO_2} + \mathrm{H_2O} \rightleftharpoons \mathrm{HCO_3^-} + \mathrm{H^+}$$

which allows to eliminate H^+ from the original reaction (H^+ is not explicitly represented in the model):

$$\begin{aligned}&(\mathrm{CH_2O})_x(\mathrm{NH_3})_y(\mathrm{H_3PO_4})_z + (x+2y)\,\mathrm{O_2} + (y+z)\,\mathrm{HCO_3^-}\\ &\quad\rightarrow\; (x+y+z)\,\mathrm{CO_2} + y\,\mathrm{NO_3^-} + z\,\mathrm{H_2PO_4^-} + (x+2y+z)\,\mathrm{H_2O}.\end{aligned}$$

The following Monod rate law is adopted for this process

$$T_{\mathrm{OMox}} = k_{\mathrm{OMox}}\,[\mathrm{O.M.}]\,\frac{[\mathrm{O_2}]}{k_{\mathrm{hsox}} + [\mathrm{O_2}]}$$

where k_{OMox} is the rate constant and k_{hsox} the half-saturation constant.

- Organic matter degradation by nitrate reduction and full denitrification

$$\begin{aligned}&(\mathrm{CH_2O})_x(\mathrm{NH_3})_y(\mathrm{H_3PO_4})_z\\ &+(\tfrac{4}{5}x + \tfrac{3}{5}y)\,\mathrm{NO_3^-} + (\tfrac{4}{5}x + \tfrac{3}{5}y - z)\,\mathrm{H^+}\\ &\quad\rightarrow\; x\,\mathrm{CO_2} + z\,\mathrm{H_2PO_4^-} + (\tfrac{2}{5}x + \tfrac{4}{5}y)\,\mathrm{N_2} + (\tfrac{7}{5}x + \tfrac{9}{5}y)\,\mathrm{H_2O},\end{aligned}$$

which we reformulate again as above, taking into account the thermodynamic equilibrium between CO_2 and HCO_3^- to eliminate H^+,

$$\begin{aligned}&(\mathrm{CH_2O})_x(\mathrm{NH_3})_y(\mathrm{H_3PO_4})_z + (\tfrac{4}{5}x + \tfrac{3}{5}y)\,\mathrm{NO_3^-}\\ &\quad\rightarrow\; (\tfrac{1}{5}x - \tfrac{3}{5}y + z)\,\mathrm{CO_2} + z\,\mathrm{H_2PO_4^-} + (\tfrac{2}{5}x + \tfrac{4}{5}y)\,\mathrm{N_2}\\ &\qquad + (\tfrac{3}{5}x + \tfrac{6}{5}y + z)\,\mathrm{H_2O} + (\tfrac{4}{5}x + \tfrac{3}{5}y - z)\,\mathrm{HCO_3^-},\end{aligned}$$

and for which the following kinetic rate law (a Monod part for the nitrate reduction step and a hyperbolic inhibition part with respect to oxygen) is adopted

$$T_{\mathrm{OMdn}} = k_{\mathrm{OMdn}}\,[\mathrm{O.M.}]\,\frac{[\mathrm{NO}_3^-]}{k_{\mathrm{hsdn}} + [\mathrm{NO}_3^-]}\left(1 - \frac{[\mathrm{O}_2]}{k_{\mathrm{inox}} + [\mathrm{O}_2]}\right)$$

where k_{OMdn} is the rate constant, k_{hsdn} the half-saturation constant for the nitrate reduction and k_{inox} the inhibition constant with respect to oxygen.

1.2 Sensitivity tests

1.2.1 Remineralization rate

For remineralization rates, two main approaches are possible:

1. variation of reaction rate constants
2. different kinetic laws (linear, non linear, in Ω, in Δ, ...)

The reaction constants are controlled via the `medusa.rrp` file (fixed name), which must be located in the folder from which the model is run. The different kinetic laws are selected in the code configuration step. For this project, we will only modify the values of the reaction rate constants.

1.2.2 Deposition rate

For deposition rates, we can still consider two broad approaches:

1. deposit rates (flows) are identical to export rates (no degradation of organic matter in the water column);
2. fluxes decrease with depth (degradation in the water column during transit from the surface to the bottom).

The distribution of deposition rates is derived from the information provided in the forcing file (typically called `medana_forcing.nml` — see 2.3.4) for more details).

1.2.3 Alternatives

Other aspects may, of course, be taken into account in the control of the carbonate fraction in the surface sediment of the seabed.

- Mixture of several classes of carbonate with different rates of remineralization
- Several classes of organic matter (multi-G approach), typically two to three (very labile, labile, refractory)
- ...

The study of these aspects, however, requires reconfiguring the model code and developing appropriate control files. So we will not, at first, explore these possibilities.

2 Getting Started with the Model

In order to be usable, the source code for MEDUSA must be configured, generated, and compiled. For the purposes of this project, a particular application, which does not require additional configuration operations, has been prepared.

2.1 Source code and directory tree

The source code for MEDUSA can be made available to you in several ways:

1. ready-to-use (pre-installed and pre-compiled) in a virtual machine that further includes a few tools to inspect and analyse the results (LIBREOFFICE, NCBROWSE, OCTAVE) and can be used on any platform that has VIRTUALBOX installed;
2. by custom access to the SVN repository that hosts the source code and allows you to manually download it to your personal computing platform where all the necessary tools have been installed beforehand (SVN client, compiler, ...) – you will have been informed about the steps to carry out in this case;
3. as a series of code and data archives to be unpacked in a common directory.

In the first case, the MEDUSA code tree is located under /home/student/work/medmbm/trunk4mcg. In the second case, it will probably be located under trunk4mcg in the installation directory. If the instructions given in Appendix A are followed, it will be located under work/medmbm/trunk4mcg in the user's home directory in the third case (similarly to the first case). This root directory will be denoted $MEDROOTDIR in the following.

In $MEDROOTDIR, there are several subdirectories:

- src-med contains the kernel source code for MEDUSA;
- src-mcg contains the source code and libraries of the configurator and code generator;
- docs contains the general documentation;
- apps/medana_wno3 contains the source code of the MEDANA application with denitrification used for this study[1];
- work/medana_wno3 is the default working folder in which the sensitivity tests are run.

In case the executable of medana_wno3 must be regenerated, just return to $MEDROOTDIR/apps/medana_wno3 and to perform (in a Terminal window) the following commands:

```
make tidy
make
```

Make sure to write down the generated executable name, which contains a characteristic time stamp (compile date and time).

2.2 Working directory

It is recommended to use $MEDROOTDIR/work/medana_wno3 as the working directory, even if it is possible to use any other directory. This directory contains a complete set of configuration and initialization files for MEDUSA. If we want to keep the results of several simulations so that we can compare them, we can however create subdirectories and adapt the names of the input-output files in the file list (see section 2.3.1 below) to relocate them to these subdirectories. When the code generation and compilation operations run by make terminate correctly, the executable is located in $MEDROOTDIR/work/medana_wno3.

[1] Note that, depending on the code collection that you are using, there may be other subdirectories under apps whose name start with medana. They hold variants of the application to be used here. Please make sure to use medana_wno3 here.

2.3 Control Files

In order to carry out simulations, the model needs a certain number of characteristic information. This information must be provided in various configuration and initialization files.

- `medusa_medana_files.nml` – file names for input and output;
- `medusa.tsi` – time control (initial time, integration time step length, number of time steps);
- `medusa_seafloor_init.nml` – default name of the file providing the initial conditions for the experiments;
- `medana_forcing.nml` – default name of the file providing the boundary conditions;
- `medusa.rrp` – kinetic rate law constants.

The working directory `$MEDROOTDIR/work/medana_wno3` includes copies for all of these files that can be adapted for your particular needs. In case the codes were provided to you via the virtual machine or checked out directly from the SVN repository, pristine versions of these files can be easily recovered. It is sufficient to change to that directory (in a Terminal window) and execute the appropriate commands among the following, depending on which file(s) are required:

```
svn revert medusa_medana_files.nml
svn revert medusa.tsi
svn revert medusa_seafloor_init.nml
svn revert medana_forcing.nml
rm medusa.rrp
cp medusa.rrp_nlcd medusa.rrp
```

If you have assembled the codes from the open source archives, this will not work and the originals will have to retrieved from the archive of the `medana_wno3` module.

2.3.1 Configuration list of input and output files

The names of most input and output files can be changed in the configuration file `medusa_medana_files.nml`. The name of this latter can not be changed and this file must be located in the working directory (where `medana_wno3` is executed). `medusa_medana_files.nml` contains two namelists, which fix the names of different input and output files.

```
1  &nml_cfg
2  cfn_nmlin_init     = 'medusa_seafloor_init.nml'
3  ctitle_ncfiles     = 'Offline␣with␣MEDANA_WNO3␣(393,uXML_32)'
4  cfn_ncout_reaclay  = 'medusa_reaclay.nc'
5  cfn_ncout_reaction = 'medusa_reaction.nc'
6  cfn_ncout_procrate = 'medusa_procrate.nc'
7  cfn_ncout_bc       = 'medusa_bc.nc'
8  cfn_ncout_flx      = 'medusa_flx.nc'
9  !cfn_ncout_sedcore  = 'medusa_sedcore.nc'
10 /
11 &nml_extra
12 cfn_ncout_aux      = '/dev/null'
13 cfn_nmlin_forcing  = 'medana_forcing.nml'
14 cfn_csvout_summary = 'medana_summary.csv'
15 /
```

The entry ctitle_ncfiles (line 3) is used to specify a descriptive title for each simulation, which will be transcribed into all NETCDF results files.

2.3.2 Time step control

The initial time (unimportant value), the duration of each step (in years) and the number of steps to be performed are specified in medusa.tsi. Like medusa_medana_files.nml, this file must be located in the directory where medana_wno3 is executed. Steady-state simulations can be done in one step: set the number of steps to perform to 1 (at line 3) and the time-step length (datime, second line) to 0.0D+00.

2.3.3 Initial conditions

The *initial conditions* are, for the purposes of this project, read from the file medusa_seafloor_init.nml.The file provided by default will be used as is for our simulations.

```
! Template for medusa_seafloor_init.nml, automatically generated by
! CREATE_MOD_SEAFLOOR_INIT from the MEDUSA configuration
! utility MedusaCoCoGen.
! Solids' (prefix "solid_") are expected to be given in %dry-weight
! Porewater solutes' (prefix "porew_") are expected to be given in mol/m3
```

```
&nml_seafloor_init
solid_pcent_clay =  100.00D+00
solid_pcent_calc =    0.00D+00
porew_molm3_co3 =    81.76D-03
porew_molm3_hco3 = 2307.01D-03
porew_molm3_co2 =    31.23D-03
solid_pcent_om =      0.00D+00
porew_molm3_o2 =    100.00D-03
porew_molm3_no3=     35.00D-03
/
```

2.3.4 Forcings: boundary conditions

Boundary conditions and their variations are controlled by the forcing file given in the `cfn_nmlin_forcing` entry in the `medusa_medana_files.nml` list and can be changed there. The default name is `medana_forcing.nml`. If the results of different simulations are distributed in sub-directories, we can provide a copy of this file by simulation, store it in the subdirectory and adapt the entry `cfn_nmlin_forcing` in `medusa_medana_files.nml` (please prefix the file name of the appropriate subdirectory name, both concatenated with a '/').

This file first fixes the number of sediment columns, using the information provided in the `&nml_forcing_general` namelist and following the rule laid out in the comments:

```
&nml_forcing_general
! General config
! n_medanacols
!  = 0: all parameters change independently
!       (a grid with length PRODUCT(xyz_npt) is created)
!  > 0: all parameters change in parallel
!       (a grid with length <n_medanacols> is created
!        and all xyz_npt below must be either equal to
!        1 (value is repeated) or to <n_medanacols>.
n_medanacols = 0
/

```

The second namelist, `&nml_forcing_enviro`, sets the environmental conditions for each column:

- depth [m] - entries prefixed with "`wdbsl_`";

- temperature [degrees Celsius] - entries prefixed with "`wtmpdc_`";
- salinity [—] - entries prefixed with "`wsalin_`".

For each of these parameters, a minimum value (suffix "`_min`"), a maximum value (suffix "`_max`") and the number of values to considered (suffix "`_npt`"). The values x_i actually used are computed according to $x_i = x_{\min} + i(x_{\max} - x_{\min})/(n-1), i = 0, \ldots, n-1$, where n is the number of points. As we are interested in the shape of the transition zone, we will let only the depths vary (here 80 depth intervals, referenced by their centres, covering the seabed from 1000 to 5000 m below the level of the sea) :

```
&nml_forcing_enviro
wdbsl_min  = 1025.00D+00
wdbsl_max  = 4975.00D+00
wdbsl_npt  = 80
wtmpdc_min =    2.00D+00
wtmpdc_max =    2.00D+00
wtmpdc_npt = 1
wsalin_min =   34.70D+00
wsalin_max =   34.70D+00
wsalin_npt = 1
/

```

Boundary concentrations for solutes (dissolved inorganic carbon – `dic`, alkalinity – `alk`, dissolved oxygen – `o2`, nitrate – `no3`) are derived from the data from the third namelist, `&nml_forcing_wconc`:

```
&nml_forcing_wconc
wconc_min_dic = 2.380D+00
wconc_max_dic = 2.380D+00
wconc_npt_dic = 1
wconc_min_alk = 2.480D+00
wconc_max_alk = 2.480D+00
wconc_npt_alk = 1
wconc_min_o2  = 0.160D+00
wconc_max_o2  = 0.160D+00
wconc_npt_o2  = 1
wconc_min_no3 = 0.035D+00
wconc_max_no3 = 0.035D+00
wconc_npt_no3 = 1
/

```

It is again necessary to provide the minimum and maximum values, as well as the number of points to consider, for each dissolved component. The values provided in the sample `medana_forcing.nml` are typical for the Pacific Ocean.

The distribution of solid fluxes for the different columns is determined from the information provided in a fourth namelist, `&nml_forcing_xpflx`. For each solid component, a group of four inputs must be given provided: minimum and maximum values, number of points and scale of variation (linear or logarithmic). Since the *rain ratio* is supposed to play an important role, an additional group suffixed '`_rr`' was added at the end, so as to be able to bind the calcite fluxes directly to the organic matter fluxes. This group is only taken into account if `xpflx_npt_rr` is different from zero (see comment on line 86).

```
40 &nml_forcing_xpflx
41 ! Export flux forcings
42 ! If xpflx_npt_xyz == 1:
43 !   xpflx_xyz = xpflx_min_xyz
44 ! If xpflx_npt_xyz > 1:
45 !   xpflx_xyz varies from xpflx_min_xyz to xpflx_max_xyz
46 !   depending on the value of xpflx_scl_xyz:
47 !     - xpflx_scl_xyz = 0: linear variation
48 !     - xpflx_scl_xyz = 1: logarithmic variation
49 ! Archer (1991), Sarmiento and Gruber (2006):
50 ! Clay flux    = 10% of the calcite mass flux
51 !              = 10% of 0.018 kg/m2/yr (see below)
52 !              = 0.0018 kg/m2/yr ~ 0.002 kg/m2/yr
53 xpflx_min_clay =  0.002D+00
54 xpflx_max_clay =  0.002D+00
55 xpflx_npt_clay = 1
56 xpflx_scl_clay = 0
57 ! Archer (1991), Sarmiento and Gruber (2006):
58 ! Calcite flux = 18 micromolC/cm2/yr
59 !              = 18 * 1e-6 mol / (1e-4 m2) / yr
60 !              = 0.18 molCaCO3/m2/yr
61 !              = 0.018 kg/m2/yr (@ 0.1 kg/mol)
62 xpflx_min_calc =  0.018D+00
63 xpflx_max_calc =  0.018D+00
64 xpflx_npt_calc = 1
65 xpflx_scl_calc = 0
66 ! Archer (1991), Sarmiento and Gruber (2006):
```

```
67 ! OrgMatter flux = 12 micromolC/cm2/yr
68 !               = 12 * 1e-6 mol / (1e-4 m2) / yr
69 !               = 0.12 molC/m2/yr
70 !               = 0.001132 molOrgM/m2/yr
71 !               = 0.004 kg/m2/yr (@ 3.553 kg/mol)
72 !xpflx_min_om   =  0.004D+00
73 !xpflx_max_om   =  0.004D+00
74 ! We reference it at 3175m, the depth where
75 ! DeltaCO3=0 for DIC=2.380 and ALK=2.480.
76 ! The following ones are export fluxes for Martin's n = -0.858
77 ! At 3175m, it will be 0.004D+00
78 xpflx_min_om   =  0.0777242D+00
79 xpflx_max_om   =  0.0777242D+00
80 xpflx_npt_om   = 1
81 xpflx_scl_om   = 0
82 !xpflx_min_rr =  0.666666666666666667D+00
83 !xpflx_max_rr =  1.50D+00
84 xpflx_min_rr =  0.0343093
85 xpflx_max_rr =  0.0771960
86 ! Set xpflx_npt_rr to zero or a negative value to use the raw
      calcite flux data above
87 xpflx_npt_rr = 0
88 xpflx_scl_rr = 1
89 /
90
```

Finally, changes in solids fluxes as a function of depth in the water column can be controlled by the last namelist, `&nml_forcing_wcprof`. Additional parameter values that may be required (e.g., scale lengths for exponentially decreasing profiles or exponents for Martin's profiles) should follow this namelist in the same order as the profiles on lines 98–100 reuqire them. Only the required parameter values actually used must be given. No comments are allowed after the last namelist.

```
91 &nml_forcing_wcprof
92 ! Definitions of Water Column PROFiles for solids' fluxes
93 ! wcprof_id_xyz:
94 !  = -1: custom (hard-coded)
95 !  =  0: unchanged (default)
96 !  =  1: exponentially decreasing (give exponential factor
      after namelist)
97 !  =  2: Martin-type power-law (give power-law exponent after
    namelist)
wcprof_id_clay =  0
wcprof_id_calc =  0
wcprof_id_om   =  2
/
0.858
```

2.3.5 Reaction rate constants

The constants to use for the adopted kinetic rate laws are fixed in the file `medusa.rrp` (*Reaction Rate Parameters*). The name of this file is imposed and the file must be in the same place from where the executable is called. It includes one namelist for each process included. The comments serve as a guide for the meaning of the different entries.

```
! Process 1 [CalcDissolution]
! ===========================
! Expression: k_1 * [Calcite] * (1 - [CO3]/ksat_1)**n_1
&nml_CalcDissolution
k_1 = 365.25D+00
n_1 = 4.5D+00
/


! Process 2 [OrgMatterOxicDegrad]
! ===============================
! Expression: k_2 * [OrgMatter] * [O2]/(khs_2 + [O2])
&nml_OrgMatterOxicDegrad
k_2 = 3.2D-02
khs_2 = 0.020D+00
/


! Process 3 [OrgMatterFullDenit]
! ==============================
! Expression: k_3 * [OrgMatter] * ([NO3]/(khs_3 + [NO3])) * (
    kih_3/(kih_3 + O2))
&nml_OrgMatterFullDenit
k_3 = 3.2D-03
khs_3 = 0.005D+00
kih_3 = 0.020D+00
```

2.4 Sensitivity Tests

We will begin with the most complete experience and will thereafter reduce the complexity more and more, in order to isolate and compare the effects of the factors that are most important for the abundance of carbonate in the surface sediment a a function the depth of the sea floor, and therefore control of the thickness of the transition zone. The supplied model configuration includes the following components:

- CO_2, HCO_3^-, CO_3^{2-}, O_2 and NO_3^- as solutes;
- clay (inert matter), calcite and organic matter as solids.

The included processes are:

- the dissolution of calcite, according to a law in $(1 - \Omega)^n$;
- the aerobic degradation of organic matter, with limitation by oxygen concentration according to a Monod law;
- the anaerobic degradation of organic matter by nitrate reduction, with
  - an inhibition by oxygen according to a complementary Monod law;
  - a limitation by the concentration of nitrate according to a Monod law.

The carbonate system is assumed to be in thermodynamic equilibrium.

Before starting, it is advisable to make a backup copy of the two input files that we will modify during the tests of sensitivity below (`medusa.rrp` and `medana_forcing.nml`, and possibly also `medusa_medana_files.nml`).

2.4.1 Complete Experience

The complete model as described above is used and in addition we adopt a flux distribution for organic matter that decreases with depth according to a Martin profile, with an exponent of -0.858 (see forcing file, lines 91–102, and especially lines 100 and 102 — attention, on line 102, we must indicate *the opposite of the actual exponent*, the profile always being assumed decreasing, hence requiring a negative exponent).

2.4.2 Anaerobic degradation of organic matter

Adopt more or less important reaction rates for this process, by modifying the value of `k_3` in the `medusa.rrp` file (line 27):

- increase the value by a factor of 2, 5 or 10;
- decrease the value by a factor of 2, 5 or 10;
- remove the effect of this process by setting the value to zero.

2.4.3 Aerobic degradation of organic matter

Adopt more or less important reaction rates for this process, by modifying the value of `k_2` in the `medusa.rrp` file (line 16):

- increase the value by a factor of 2, 5 or 10;
- decrease the value by a factor of 2, 5 or 10;
- combine these modifications with the removal of anaerobic degradation.

2.4.4 Dissolution of calcite

Adopt more or less important reaction rates for this process, by modifying the value of `k_1` in the file `medusa.rrp` (line 5):

- increase the value by a factor of 2, 5 or 10;
- decrease the value by a factor of 2, 5 or 10.

2.4.5 Constant flux of organic matter

Disable the use of Martin's profile by

- setting the value of `wcprof_id_om` to 0 (line 100),
- removing '`!`' at the beginning of lines 72 and 73, and
- inserting a '`!`' at the beginning of lines 78 and 79.

The last two operations serve to have identical flux values at 3175 m depth in both experiment series with variable and constant fluxes, i.e., at the depth of the saturation horizon of calcite for the DIC and alkalinity concentrations specified in the forcing file (`medana_forcing.nml`).

Repeat previous tests 1–4.

2.5 Results files

When running, MEDUSA generates a series of results and log files. The most important for the purposes of this project are:

- the REACLAY file, whose name is fixed at the `cfn_ncout_reaclay` in `medusa_medana_files.nml` (line 4), and which contains the results of all profiles of the components considered;
- the CSV file, whose name is fixed at the `cfn_csvout_summary` entry in `medusa_medana_files.nml` (line 13) and which provides a summary of all information required for the purposes of this study, including the average fractions of the various solid constituents in the sedimentary layer for each sediment column.

A Assembling a Working Model Instance From Open Source Archives

With the publication of MEDUSA (G. Munhoven, *Geoscientif. Model Dev.* 14(6):3603–3631, DOI: 10.5194/gmd-14-3603-2021) the source code of the model and of its dependencies have been made available as open source software. The Supplement of that paper includes an archive with the exact version used for the showcase applications presented in the paper.

The same codes have been archived on Zenodo, albeit in three separate archives, due to different licences; subsequent bug-fix releases will also be uploaded to Zenodo. Here, version 2.0.2 of MEDUSA is used. In general it is recommended to refer to the MEDUSA page on Zenodo (DOI: 10.5281/zenodo.4677681, which also has links to the required THDYCT and μXML libraries) to check if more recent versions of the different uploads are available and use those instead.

There is a BASH script available that downloads the required archives from Zenodo, complements it with the `medana_wno3` application from the author's web site at the University of Liège and then does a clean install of the model configuration required to carry out the simulation experiments for this project. When run from a directory called `work` in your home directory, the created directory tree structure is compatible with that introduced in section 2.1. Please download it to a directory named `work` in your home directory (to be created with `mkdir ~/work` if it does not exist), make it executable and run it:

```
cd ~/work
```

```
wget http://www.astro.uliege.be/~munhoven/fr/cours/projects/
   sediment_themes/medana_ctz_install.sh
chmod +x medana_ctz_install.sh
./medana_ctz_install.sh
```

B Using MEDUSA on Microsoft Windows

In principle it should be possible to compile and run MEDUSA natively on Microsoft Windows. The code itself is standard compliant and it would thus be sufficient to have a Fortran 95 compiler and all the necessary prerequisites (NETCDF libraries, etc.) available. The make based building chain is, however, essentially UNIX/LINUX oriented and may possibly require extensive adaptations in order to be used in a pure Microsoft Windows programming environment. This has not been tried so far and will thus not be considered here.

B.1 Cygwin: UNIX for Windows

The CYGWIN programming environment provides an alternative to the native compilation on a computer running Microsoft Windows.[2] CYGWIN allows to run UNIX code on platforms operating under Microsoft Windows and provides a comparatively light-weight means to use MEDUSA on such a platform. Its usage actually does not require any modification of the original code or Makefiles. After installation of the main system (whose installer can be downloaded from www.cygwin.com) the following packages will have to be installed to make the various required tools (make, wget, Fortran compiler, NETCDF development tools, ...) available:

- gcc-core
- gcc-fortran
- make
- netcdf
- libnetcdf-devel

[2] Tests with the similar MSYS2 have unfortunately not been successful, due to the required Fortran interfaces for the NETDCF library are not provided and compilation of the source code failed for unclear reasons.

- `libnetcdf-fortran-devel`
- `wget`

For each one of these packages, it is possible and advisable to install the most recent version available. In case it is necessary to access the SVN repository and to get a means for transferring files to other platforms, the following packages are useful:

- `subversion`
- `openssh`

The resulting installation requires about 1 GB of disk space (and much less to download though) and allows to use MEDUSA from a CYGWIN terminal exactly the same way as in a terminal emulator on a UNIX or LINUX platform.

For an enhanced user experience, a graphical desktop environment such as GNOME, KDE, XFCE or similar can be added. For XFCE, one could thus further install

- `xfce4-session`
- `xfce4-terminal`
- `Thunar`
- `tumbler`
- `mousepad`
- `geany`

which provides a minimalistic but nevertheless functional desktop, including the powerful programming editor GEANY. Please notice that this requires about 1 GB of extra disk space though.

Potential issue: Do you have spaces in you Windows logon name?

CYGWIN will run into much trouble if your Windows logon name, from which your home directory name on CYGWIN will be derived, contains spaces. If this is the case for you, there are a few extra steps to take. I have successfully used the following procedure[3] in the past to solve this issue for a Windows logon name `Guy␣Munhoven` (where ␣ simply highlights a space character) to be mapped onto the spaceless CYGWIN user name guy – please adapt as required:

[3] See https://stackoverflow.com/a/47901556.

1. Close all running instances of CYGWIN.

2. Open a CYGWIN Terminal

3. Check if /etc/passwd exists – if so, please remove it with

   ```
   rm /etc/passwd
   ```

4. Then issue the following commands:

   ```
   cd /home
   mv "Guy␣Munhoven" guy
   ln -s guy "Guy␣Munhoven"
   mkpasswd -l > /etc/passwd
   sed -i ’s/Guy␣Munhoven/guy/g’ /etc/passwd
   ```

   making sure to include quotes (" or ’) as indicated and noticing that it is important to distinguish between upper- and lowercase letters.

5. Close the CYGWIN Terminal – the changes will become effective at the next start of CYGWIN.

Once CYGWIN is set up and running, it is sufficient to launch a Terminal in it and to follow the instructions for getting the MEDUSA code from the SVN repository or to assemble the code tree as described in appendix A above.

B.2 Post-Processing: Fortran, Spreadsheets and Python

MEDUSA includes a series of post-processing utilities. Their Fortran source codes can be found in the `uti` directory. The BASH script above includes the necessary commands to compile them. The resulting executables can be moved to a central location where the shell will be able to find them, so that they can be invoked at any place. Most interesting for our purpose here are `column_extract2csv` and `sedcore_extract`, which extract specific results from the NETCDF files and store them in CSV files, which can be opened with common spreadsheet programs.

Processing a large number of CSV files with a spreadsheet program becomes, however, rapidly tedious and a script based approach more efficient. PYTHON is a straightforward and readily available tool for this purpose. Common LINUX distributions provide all the necessary PYTHON modules for this post-processing via their respective package management systems. Unfortunately, the CYGWIN package repository lacks the indispensable NETCDF4 module and attempts to install it with PYTHON's internal package installer pip failed so far. This is, however, straightforward under Windows. Since the CYGWIN file-system tree is fully and transparently accessible from Windows, a Windows-based PYTHON installation is thus convenient.

The Python Software Foundation offers a large collection of PYTHON releases for Windows. At the time of writing, the most recent stable releases are 3.10.0 for Windows 8 and later, and 3.8.10 for Windows 7 and later. There is, however, no Wheel package version of netCDF4 available in release 3.10.0. It is therefore preferable to use 3.8.10 on all versions of Windows. You may, of course, continue using any prior PYTHON 3 release that you might already have installed on your computer and that include the numpy, netCDF4 and matplotlib packages.

The required PYTHON packages can be added to the PYTHON installation with the following sequence of commands (to be issued in a Windows Command Prompt, cmd.exe):

```
py -m pip install --upgrade pip
py -m pip install ipython
py -m pip install numpy
py -m pip install netCDF4
py -m pip install matplotlib
py -m pip install pandas
```

The first of these commands updates the package manager itself and the (optional) second one installs the interactive PYTHON command shell IPYTHON; the others install four packages that are most commonly required for successful processing of MEDUSA NETCDF files (the last one – pandas – is optional at this time, but might become necessary in the future). The uti directory contains a PYTHON script (meduti.py) which can be used to rapidly produce concentration and reaction rate profiles for individual MEDUSA columns (using the NETCDF results files) or depth transects of the average surface sediment compositions (using medana_summary.csv). Once meduti.py has been copied to a directory where PYTHON may locate it is sufficient to issue the following commands in your preferred PYTHON interpreter from the results directory:

```
import meduti as mu
mu.profile("medusa_reaclay.nc", 10, "xconc_no3")
mu.transect_wno3_csv()
```

In the previous, the first argument to mu.profile(...) gives the name of the MEDUSA NETCDF file, the second argument gives the column number and the third one the name of the variable in the file for which a profile should be drawn. It is always the last time record in the file that is used. Columns are numbered from 1 (the shallowest) to 80 (the deepest). The example would thus produce a NO_3^- concentration profile for sediment column 10. The exact names of the variables and metadata in the NETCDF files can be easily inspected with `ncdump -h` *`filename.nc`* in a LINUX terminal. Both functions generate PNG graphics files by default. That type can nevertheless be changed by modifying the value of `_graphfileext` at the beginning of `meduti.py`.

Index