# MEDMBM: Coupling MBM and MEDUSA

—

# a tentative handbook

(for SVN revision 261ff of MEDMBM)

Guy Munhoven
Université de Liège, Belgium
http://www.astro.ulg.ac.be/~munhoven

28th July 2020

## 1 Introduction

MEDMBM is a coupled model of the carbon cycle in the ocean, atmosphere and surface sediment: the ocean carbon cycle model is the Multi-Box-Model (MBM); the sedimentary processes are described by the Model of Early Diagenesis in the Upper Sediment (A) (MEDUSA).

MBM is an eleven-box model of the ocean-atmosphere subsystem of the global carbon cycle, with ten oceanic and one atmospheric reservoirs. The ocean reservoirs have a realistic depth distribution, based on the hypsometric curve. Reservoir volumes and surface areas of the interfaces between boxes are calculated from sea-level. Sea-level histories are prescribed from a forcing file (see SLV file below). Salinity is derived from the ocean volume boxes. The temperature histories for the different boxes are also prescribed from a forcing file (see TPR file below). MBM includes mass balance equations for DIC, TA, phosphate (chosen as the limiting nutrient) and oxygen in the oceanic reservoirs and $CO_2$ in the atmosphere. $^{13}C$ and $^{14}C$ isotopic characteristics of all carbon bearing tracers are also considered for all of the reservoirs. Biogenic export fluxes from the surface ocean to the intermediate and deep-sea reservoirs include organic matter, calcite and aragonite. The carbonate-carbon/organic-carbon rain ratio may be fixed (see DAT file below) or prescribed from a scenario; similarly for the aragonite fraction in the carbonate rain (see DAT and FF1 files below). Organic matter is partly

remineralized in the water column below 100 m depth, following a power-law approach. The adequate exponent is derived from the fraction of the export rain that gets remineralised between 100 and 1000 m depth (see DAT file below). MBM also considers carbonate accumulation on the continental shelf, both reefal and non-reefal. Coral reefs grow only in water shallower than 100 m and on the continental shelf. They neither build up during sea level drops nor during too fast rises. Coral reef growth is enhanced with an increasing degree of supersaturation of surface seawater with respect to aragonite. Non-reefal carbonate accumulation is taken proportional to the area of flooded continental shelf (see Munhoven and François [1996] and Munhoven [1997] for more details).

MEDUSA is a transient one-dimensional advection-diffusion-reaction model that describes the coupled early diagenesis processes of carbonates and organic matter in the surface sediment. Sediment is represented as a two-phase porous medium, i.e., solids and porewater. In the vertical, it is subdivided into two different zones. Solids raining down from the surface of the ocean are collected by the *reactive mixed layer* at the top. It is always 10 cm thick (this can nevertheless be changed via a parameter in the source code). Here, solids are transported by bioturbation and advection, solutes by molecular diffusion. Chemical reactions are restricted to this zone. Solids that get transported deeper than the bottom boundary of the reactive mixed layer enter the second zone, the *historical zone*, where sediment accumulates. The historical zone is made up by a pile of layers representing a synthetic sediment core. Each layer is 1 cm thick, except for the top one, which acts as a buffer layer that collects the material that leaves the mixed-layer. Once it has grown more than 1 cm thick, it is converted into a normal historical layer. A new buffer layer is created at the top of the pile (just below the mixed layer) and gets all the material left over after the conversion of the old buffer layer into a normal historical layer. As long as the sedimentary material stays in the historical zone, it is not subject to chemical reactions. However, not all of the material that enters the historical zone will necessarily be preserved. MEDUSA actually takes chemical erosion into account. When the sediment column becomes subject to chemical erosion, the advection velocity at the bottom of the mixed-layer bottom changes sign and points towards the sediment surface interface. As bioturbational activity will always extend to 10 cm depth, the topmost material from the historical zone will get mixed back into the reactive mixed layer. Back in the mixed layer, it is again subject to chemical reaction (e.g., carbonate may then dissolve there, possibly thousands of years after it got deposited). When MEDUSA faces chemical erosion, the material from the buffer layer is the first to return to the mixed layer. If the buffer layer runs empty, the most recently created historical

layer is discarded and its contents used to fill up the buffer layer again.

MEDUSA is based upon a time-dependent formulation. This is required in order to take chemical erosion into account. Evolution equations for clay, calcite, aragonite and organic matter (solids) and for $CO_2$, $HCO_3^-$, $CO_3^{2-}$ and $O_2$ (porewater solutes) are included. A complete description of these coupled advection-diffusion-reaction equations is given in Appendix A of Munhoven [2007].

# 2 Getting the source and building the model

## 2.1 System requirements

Building the model requires

- a Fortran 95 compiler;
- the NETCDF library with the FORTRAN77 interface installed;
- a Subversion client.

### 2.1.1 Fortran compilers

The model has been successfully built with the Fortran compilers from

- Intel Fortran, versions 7.0 and later;
- Portland Group, PGF90;
- Silverfrost (Windows, earlier versions of the model only);
- G95;
- GFORTRAN.

Current developments by the author mostly call upon GFORTRAN (version 4.4.5).

The source code aims at strictly conforming to the Fortran 95 standard, except for two subroutine calls: `ABORT()` and `FLUSH(iunit)`, which are common extensions that may, however, require to use special compiler flags (e.g., `-Vaxlib` is required with some versions of the Intel Fortran compiler, while GFORTRAN and PGF90 have built-in versions of these). For compilers that do not provide one or both of these subroutines as intrinsics or extensions to the standard, one may simply add external subroutines: a subroutine `ABORT()` that simply stops the program, and a subroutine `FLUSH(iunit)` that does nothing but continue the program.

### 2.1.2 NetCDF library and utilities

Versions 3.5 and later of the NetCDF library have been used to build the model so far. Earlier version may possibly work as well.

For quick progress checks on NetCDF files during model experiments, ncbrowse or Ferret have proven useful.

### 2.1.3 Subversion clients

Subversion clients compatible with Subversion 1.3 and later can be used. The command line version `svn` is sufficient. GUIs such as RapidSVN (Linux) or TortoiseSVN (Windows) may add some extra comfort.

## 2.2 Source code location

The source code of medmbm is available from the author's subversion repositories. The medmbm repository is currently located at `svn://heracles.astro.ulg.ac.be/medmbm`. Please request a personal user name and password by sending an eMail to the author.

You will furthermore need to get and install the two following libraries:

1. $\mu$xml, a library that provides the interface routines necessary to read the XML description files used to configure and generate the code of the sediment model medusa. The source code of $\mu$xml is in the subversion repository `svn://heracles.astro.ulg.ac.be/minixml`.

2. thdyct, a library that provides procedures to calculate thermodynamic constants. The source code of libthdyct can be found in the subversion repository `svn://heracles.astro.ulg.ac.be/libthdyct`. If you prefer, you may of course use your own preferred set of functions for the thermodynamic constants instead of those from the thdyct library. You will most probably have to build wrappers around your own functions. Please refer to the source code of thdyct to see how to match the names, the functional arguments and units expected by mbm and medusa.

## 2.3 Building MEDMBM

The first stage, i.e., the installation of the thdyct and the $\mu$xml libraries generally needs to be carried out only once, as these libraries rarely if ever change. If you previously installed them and you already have the most recent version, you may immediately proceed to section 2.3.3!

### 2.3.1 The THDYCT library

The typical installation procedure to install THDYCT is as follows.

```
mkdir libthdyct-work
cd libthdyct-work
svn checkout svn://heracles.astro.ulg.ac.be/libthdyct/trunk
cd trunk
# choose one of the Makefiles provided,
# or adapt one of them to your needs
# and rename it to Makefile
make
mv libthdyct.a my_path_to_libthdyct
```

Please substitute *my_path_to_libthdyct* by the location where your libraries are stored (a public directory, such as `/usr/local/lib`, e.g., or a private one).

In the rare event that the THDYCT library needs to be updated, return to the `libthdyct-work/trunk` directory, update your working copy with `svn update`, then continue with the `Makefile` adaptations to moving the built `libthdyct.a` to the library directory.

### 2.3.2 The $\mu$XML library

The typical installation procedure to install $\mu$XML is as follows.

```
mkdir muxml-work
cd muxml-work
svn checkout svn://heracles.astro.ulg.ac.be/minixml/trunk
cd trunk
make libmodmxm.a
mv libmodmxm.a my_path_to_muxml_lib
mv *.mod my_path_to_muxml_incl
```

Please substitute *my_path_to_muxml_lib* by the location where your libraries are stored (a public directory, such as `/usr/local/lib`, e.g., or a private one) and *my_path_to_muxml_incl* by the location where your include files are stored (a public directory, such as `/usr/local/include`, e.g., or a private one).

In case the $\mu$XML library needs to be updated, it is sufficient to return to the `muxml-work/trunk` directory and update your working copy. After an update, it is recommended to perform a clean build of `libmodmxm.a`:

```
cd muxml-work/trunk
svn update
make clean
make libmodmxm.a
mv libmodmxm.a my_path_to_muxml_lib
mv *.mod my_path_to_muxml_incl
```

Make sure that the old library and include files are actually replaced by the new ones.

### 2.3.3 MEDMBM

If you have not previously installed MEDMBM or MEDUSA, please first create a working directory tree head and retrieve the source code:

```
mkdir medmbm
svn checkout \
  svn://heracles.astro.ulg.ac.be/medmbm/branches/trunk4mcg ./
```

If you are only updating the model source code, please replace the "`svn checkout svn://...`" line by "`svn update`" (no other arguments required). Your credentials with the Subversion repository server are normally cached.

There are two more `Makefile`s that might require some adaptations before the complete model can be compiled.

1. In `Makefile` under `src-mcg`, `INCFLAGS` needs to include the path to the directory where the $\mu$XML `*.mod` files have installed, prefixed by `-I` (please do not delete the `-I.` part that is required as well), and `LDFLAGS` must allow the linker to find the $\mu$XML library. Typically:

   ```
   INCFLAGS = -I. -Imy_path_to_muxml_incl
   LDFLAGS = -Lmy_path_to_muxml_lib -lmodmxm.a
   ```

   where, as previously, *my_path_to_muxml_incl* must be substituted by the path (absolute or relative to `src-mcg`) to the directory where your installed the $\mu$XML `*.mod` files, and *my_path_to_muxml_lib* by the path to the directory where `libmodlib.a` has been installed.

2. In `Makefile` under `src-mbm`:

   - set the `FC` variable to the name of your Fortran 95 compiler.

Check if any of the pre-configurations on lines 23–42 might be suitable. If so, comment out the unwanted parts (put a `#` as the first character in the line) and uncomment the required parts (delete all of the initial `#`'s on the respective lines). Basically,

- the `FFLAGS` variable has to collect all the required compilation flags (related to optimization levels, special processor options, etc.);
- the `LDFLAGS` variable must provide the necessary information for the linker to find the NetCDF and the THDYCT libraries (please notice that $\mu$XML is not required here): `LDFLAGS` must include `-L`*my_path_to_libthdyct* `-lthdyct -L`*path_to_libnetcdf* `-lnetcdf`. Long lines may be split by inserting a backslash character ('\') at the place of your choice and continuing with the rest on the next line.

You will have to substitute *my_path_to_libthdyct* for the place where you put the THDYCT library in the first stage. For NetCDF, there is a large variety of possible installations, and depending on this, you need to adapt adequate `-L...` and `-l...` library switch combinations. So far, the following situations have been encountered:

- the NetCDF library files may be is installed system-wide, in which case you may omit the `-L`*path_to_libnetcdf* part, else you will have to substitute the correct path for *path_to_libnetcdf* (to precede the actual `-l...` library specifications);
- the `libnetcdf.a` or `libnetcdf.so` that is available to you possibly includes all of the objects required for linking with Fortran, in which case `-lnetcdf` is sufficient;
- your `libnetcdf.a/so` possibly only contains bindings for the C language, in which case you may have to add `-lnetcdff` after `-lnetcdf` (a situation typically encountered with recent Debian based systems, but also with some self-compiled NetCDF installations);
- if the NetCDF installation on your system was compiled to provide HDF5 support, you may have to add `-lhdf5` and `-lhdf5_hl` after `-lnetcdf`.

In practice, if you are unsure about the peculiarities of the NetCDF installation on your particular system, you may follow this heuristic procedure:

(a) find out where the `libnetcdf.a` or some `libnetcdf.so.*` that is appropriate for your compiler is located and adopt an adequate `-L...` switch if necessary; include the `-lnetcdf` switch alone and try to build the model (`make` – see below);

(b) if the linker complains about undefined `NF...` symbols after all of the source code successfully compiled, the extra `-lnetcdff` switch is most probably required (check if there is a `libnetcdff.a` or `libnetcdff.so.*` file in the same directory where you found `libnetcdf.*`); add the `-lnetcdff` switch after `-lnetcdf` and recompile;

(c) if the linker complains about undefined `H5...` symbols after all of the source code successfully compiled, the extra `-lhdf5` and `-lhdf5_hl` switches are required; add the `-lhdf5` switch after `-lnetcdf` and recompile; if there remain undefined `H5...` symbols add also `-lhdf5_hl` and recompile;

(d) if the model still does not correctly build after these adaptations, you will have to contact your system administrator for assistance.

You are now ready to build the model. In `src-mbm`, please execute:

```
make
```

Some parts of the model are time-stamped, as is the final executable. If you are to resume a failed compilation attempt, it is recommended to clean the `src-mbm` directory first for left-over `*.o` files that would otherwise carry an inconsistent time-stamp with

```
make clean
```

and then only `make`.

The final executable (`medmbm.`*`timestamp`*) is moved to the `work` or the `work/mbm` directory.

# 3 Running MEDMBM

## 3.1 MBM: Configuration and data files

Except for the name and location of the configuration file, `mbm.cfg`, and the logging (`medmbm.log`, `medmbm-dorpri.log`) and debugging (`medmbm.dbg`) files, the names of the various files read (data, forcings, ...) and written (results, diagnostics, by MBM can be freely chosen. File names used by

MEDUSA are currently hard-coded, but it is planned to change this in a near future.

Configuration, data and forcing files come in three different types:

- fixed-order text files: all information must be presented in a set order;
- structured text files, either as tagged mark-up-type files, or Fortran 95-style namelist files (the provided namelist files generally include comments, which became only standard with Fortran 95);
- NetCDF files.

The most flexible of these are of course NetCDF files, as they may be extended without disturbing the currently present information.

### 3.1.1 `mbm.cfg`

MEDMBM expects to find a file called `mbm.cfg` in the working directory. `mbm.cfg` provides names and paths for all other files that provide required or optional data.

`mbm.cfg` is a structured text file, and its mark-up-type structure is as follows:

1. a freely usable header section, that ends before the first line starting with `##`;
2. a model ID section, delimited by lines saying `## MODEL_ID` and `## END MODEL_ID` — the model ID is written to the RES file (see below);
3. an experiment ID section, delimited by lines saying `## EXPERIMENT_ID` and `## END EXPERIMENT_ID` — the experiment ID is written to the NetCDF files that trace the evolution of the sediment;
4. a files section, providing names of files for various purposes (initial state, general configuration, geometry of the basins, forcing data etc.) identified by three-letter codes; this section is initiated with a `## FILES` line, and closed with a `## END FILES` line. The syntax of each line of the file list is

   *COD* `= '`*filename_with_path*`'`

   Currently known codes (*COD* above) are

   **DAT** "DATa" file: basic data on model structure, MBM parameters, etc. [mandatory entry, Input];

**INI** "INItial state" of MBM [mandatory entry, Input];

**RES** "RESults": human-readable summary diagnostics of MBM [optional, Output];

**CSL** "ConSoLe" file: main variables and diagnostics of MBM, as a function of time [optional, Output];

**CMS** "Complete Model State": more comprehensive evolution of MBM [optional, Output];

**ELV** "ELeVations": hypsometry (depth profiles) for the five ocean regions [mandatory entry, Input];

**SLV** "Sea-LeVel" evolution for MBM [optional, Input];

**FLX** "FLuX" file: human-readable summary diagnostics of MBM [optional, Output];

**FF1** "Forcing Function 1", currently not used [optional, Input];

**PUR** "Phosphate Utilization Ratio", currently not used – reserved for a future revision [optional, Input];

**CRR** "Carbonate Rain Ratio": evolution of the carbonate-C/organic-C ratios in the surface reservoirs (evolutions of `rhocac`) [optional, Input];

**AFC** "Aragonite Fraction of Carbonate": evolution of the global fraction of aragonite in the carbonate rain (evolution of `frcara`) [optional, Input];

**SHF** "SHelf Fluxes" evolution (`coralo` and `shbnko`) [optional, Input];

**WEA** "WEAthering" flux scenarios [optional, Input];

**SDO** "SeDiment Output": obsolete, not used any more [optional, Output];

**SDI** "SeDiment Input": obsolete, not used any more [optional, Input];

**CON** "CONtinuation" file for MBM – results file that can be used as an `INI` file in subsequent runs – currently disabled in the source code [optional, Output];

**CBF** "Continental Biospheric Forcings": forcings related to sources and sinks of atmospheric $CO_2$ related to biospheric processes [optional, Input];

**TPR** "TemPeRature" evolution (in all of the ocean reservoirs) [optional, Input].

The following rules and limitations apply:

- Blank lines are ignored and skipped.
- Comments may be started at any place (not inside a character constant, of course) with a bang (character '`!`').
- The maximum length of the **model ID** character string is set in the declaration of `model_idstr` in the module `mod_mbm_info.F` in `src-mbm`. The default maximum length is 80 characters.
- The maximum length of the **experiment ID** character string is set in the declaration of `experiment_idstr` in the module `mod_mbm_info.F` in `src-mbm`. The default maximum length is 80 characters.
- The maximum length of file names is set by the integer parameter `gnrl_maxfilename_len` in `mod_generalparams.F` in `src-mbm`. The default length is 255 characters.
- Unknown file codes found in `mbm.cfg` produce a warning message, without abortion.

A sample `mbm.cfg` file is provided in the `work/mbm-template` directory, along with sample `DAT` and `INI` files, and a few others related to MEDUSA.

Some entries are mandatory (as mentioned); for others `mbm` has default information.

### 3.1.2 DAT file

The `DAT` entry in `mbm.cfg` is mandatory!

The DAT file provides fundamental parameter values for processes considered in MBM. Please refer to the fully commented sample `mbm.dat` provided in `work/mbm-template` for details. This is a fixed-format, fixed-order text file. No lines must be added or deleted, except in the water flux section, if the line numbers are consistently adapted; blank lines may be significant. Data are read in free-format mode, with one item per line. Any text after the numerical data (one to five items) on each line is ignored and can be used for commenting purposes.

If parameter values are to be specified for each of the five surface reservoirs, the order is always SNATL – SEATL/SLATL – SANT – SEI-P/SLI-P – SNPAC (following the denominations from Munhoven [1997]/Munhoven [2007] where relevant).

The following might be of interest for changes or may not be changed under any circumstances

- as of revision 222, the contents of lines 6 and 7 change meaning

- line 6 now holds `ORGM_C`, the molar carbon content in the Redfield organic matter formula (before rev. 222, this line held `RHOPC`, which is equal to the inverse of `ORGM_C`, as we assume that the formula is normalized to a phosphorus content of 1 mol P per mol of organic matter);
- line 7 now holds `ORGM_N`, the molar nitrogen content in the Redfield organic matter formula (before rev. 222, this line held `RHONC`, which is equal to the ratio `ORGM_N/ORGM_C`).

If older `DAT` files are re-used, please amend as necessary.

- `COREM2` on line 8, five values (one for each surface reservoir) — the fraction of POC remineralized between 100 and 1000 m depth depth.

  **NB**: Munhoven [1997] only considered a single such parameter (denoted $r_{\mathrm{oxyd}}$), taken to be the same under all five surface reservoirs.

  **CAVEAT**: all values within the `COREM2` array must be **different from 0.9**! The implementation of the organic matter flux calculation as a function of depth actually actually leads to a $(x_1^0 - x_2^0)/0$ indetermination for 0.9. This degeneration is currently not caught and will lead to floating point exceptions.

- `COREM3` on line 9 — this is a place-holder and not used any more. Leave as is, please.

- `RHOCAC` on line 10, five values (one for each surface reservoir) — the default carbonate carbon/organic carbon rain-ratio.

  **NB**: Munhoven [1997] only considered three parameters for the rain ratios: $r^{\mathrm{C}}_{\mathrm{polar}}$, $r^{\mathrm{C}}_{\mathrm{SEATL}}$ and $r^{\mathrm{C}}_{\mathrm{SEI-P}}$. To mimick that choice, the five values to be provided here would thus have to be
  $r^{\mathrm{C}}_{\mathrm{polar}}$ $r^{\mathrm{C}}_{\mathrm{SEATL}}$ $r^{\mathrm{C}}_{\mathrm{polar}}$ $r^{\mathrm{C}}_{\mathrm{SEI-P}}$ $r^{\mathrm{C}}_{\mathrm{polar}}$
  This **entry may become obsolete in a near future** and we will have its value always prescribed by a forcing scenario from a dedicated file.

- `RHOPO4` on line 11, five values (one for each surface reservoir) — the phosphate uptake (utilisation) efficiency, expressed as the fraction of the phosphate that enters the respective reservoir by the water circulation. This efficiency sets the export production (i.e., the production that leaves the 100 m depth horizon). In the high-latitude reservoirs, which are 1000 m deep, a large fraction of the export production gets remineralized within the reservoir (between 100 and 1000 m depth –

see COREM2 above) and the remineralized phosphate may be used for organic production within the reservoir (which is supposed to be well-mixed). Accordingly, the "efficiency" value may be greater than 1. Do not let you get misled by the name "export" production: this does not represent the flux of organic matter that gets exported out of the reservoir, but rather the flux fo organic matter that gets exported out of the euphotic zone (typically the surface 100 m). In any surface reservoir $i$, PO4UTI is thus equal to $u_i/(1-r_i)$, where $u_i$ is the actual fraction of phosphate brought into reservoir $i$ by the water circulation and $r_i$ is the average fraction of the organic export production remineralized within the reservoir (if the reservoir is more than 100 m deep), close to, but not equal to the respective COREM2 value, because of the deposition and burial of organic matter in the surface sediment between 100 and 1000 m depth.

**NB1**: Munhoven [1997] only considered three different values for this parameter: $u^{\mathrm{P}}_{\mathrm{SN}}$ $u^{\mathrm{P}}_{\mathrm{SE}}$ and $u^{\mathrm{P}}_{\mathrm{SANT}}$. To mimick that choice, the five values to be provided here would thus have to be
$u^{\mathrm{P}}_{\mathrm{SN}}$ $u^{\mathrm{P}}_{\mathrm{SE}}$ $u^{\mathrm{P}}_{\mathrm{SANT}}$ $u^{\mathrm{P}}_{\mathrm{SE}}$ $u^{\mathrm{P}}_{\mathrm{SN}}$

**NB2**: Munhoven [1997] did not consider remineralization of organic matter within the surface reservoirs. Accordingly, the $u^{\mathrm{P}}_{\mathrm{xx}}$ parameters above indeed controlled the organic matter flux actually exported out of the reservoir (at 1000 m depth in high-latitude, and at 100 m depth in low-latitude reservoirs.

- Water fluxes, lines 12–19 for the vertical fluxes, and lines 20–31 for the horizontal water fluxes in the sample mbm.dat. Line 12 gives the following general parameters relative to the vertical water flow:

```
NBR_VERT  EFFIC_VERT_ADV  EFFIC_VERT_MIX
```

The INTEGER NBR_VERT indicates the number of vertical water fluxes that follow on the next lines. The REAL EFFIC_VERT_ADV sets the factor (in %) that must multiply each vertical advective flux; the REAL EFFIC_VERT_MIX sets the factor (in %) that multiplies each one of the vertical mixing fluxes. Each one of the NBR_VERT next lines sets the water exchange between distinct couples of reservoirs, by the means of four entries of type INTEGER each time:

```
IBOX_FROM  IBOX_TO  VERT_ADV  VERT_MIX
```

Here, VERT_ADV sets the net water flux from reservoir IBOX_FROM to reservoir IBOX_TO (in tenths of Sv), while VERT_MIX sets the mixing (exchange of) water between the two reservoirs (in tenths of Sv)

Similarly to line 12, line 20 has the following general parameters for the horizontal fluxes

```
NBR_HORI  EFFIC_HORI_ADVECT  EFFIC_HORI_MIX
```

and the NBR_HORI next lines provide the required horizontal net and mixing fluxes between couples of reservoirs (in tenths of Sv).
EFFIC_HORI_ADV EFFIC_HORI_MIX again allow to reduce or increase the intensity of the fluxes (efficiency factors in %).

**NB:** Munhoven [1997] only considered one single efficiency parameter ($r_{\mathrm{mix}}$, applied to the mixing components of the circulation only. To mimick this choice, set both EFFIC_VERT_ADV and EFFIC_VERT_ADV to 100, and both EFFIC_VERT_MIX and EFFIC_VERT_MIX to $r_{\mathrm{mix}}$

- FRCARA on line 39 – the global aragonite fraction in the carbonate export production
  This **entry may become obsolete in a near future** and we will have its value always prescribed by a forcing scenario from a dedicated file.

- Constant weathering fluxes on lines 40–43 — **do not change, please; use a WEA scenario file instead!**

- Constant Coral-reef accumulation fluxes on line 44
  Constant Shelf-and-Bank carbonate fluxes on line 45. These entries are mainly used for the purpose of factor analyses. Four items are expected on each of the two lines:

  ```
  FLX_SEATL  FLX_SEI-P  TIME_SWITCHON  TIME_SWITCHOFF
  ```

  where the REAL FLX_SEATL and FLX_SEI-P are the constant flux values to adopt (in units of $10^{18}$ molC/yr) in the SEATL and SEI-P reservoirs, respectively, and TIME_SWITCHON and TIME_SWITCHOFF are the time instants (not ages!) during the simulation experiment when the constant accumulation rates are switched on and off, respectively, thus overriding/disabling the interactive modules for coral-reef and shelf-and-bank carbonate accumulation. The interactive modules remain active outside that time-interval only. The two fluxes may be switched constant and variable independently of each other.

### 3.1.3 INI file

The `INI` entry in `mbm.cfg` is mandatory!

The INI file provides the initial state for MBM. Please refer to the fully commented sample `mbm.ini` provided in `work/mbm-template` for details. This is a fixed-format, fixed-order text file. No lines must be added or deleted, blank lines may be significant and possibly make it impossible to be correctly readable. Data are read in free-format mode, with one item per line. Any text after the first numerical datum on each line is ignored and can be used for commenting purposes.

The main things that might be of interest for changes are

- the starting time of the simulation experiment (line 5);
- the final time of the simulation experiment (line 6).

If you are to recalibrate the model, the following three might be of interest as well:

- additional amounts of DIC (or Total CO2, at line 73);
- additional amounts of Total Alkalinity (line 74);
- additional amounts of Phosphate (line 75).

### 3.1.4 ELV file

The `ELV` entry in `mbm.cfg` is mandatory!

The ELV file provides the information about the depth distribution in the `mbm` reservoirs. It is a NetCDF file and is typically produced with the `bathym.f90` utility from `uti`, that uses the RAND 1°×1° bathymetric data. Currently, the source code of `mbm` is to some extent dependent on depth-intervals with 81 nodes, and sea-levels not exceeding the present-day level (0 m). The cleaning of the code for this constraint is "work in progress". In the long-term, it should be possible to deal with sea-level rises above the pre-industrial level.

**NB**: sea-level files provided in `DataRepository/MBM/Sealevel` (available upon request from the author) have all been truncated at 0 m.

### 3.1.5 SLV file

The optional SLV entry points to a file that is in our standard structured text format. Those files have the following setup:

**Line 1:** Quoted information string (up to 80 characters long), written to the RES file (if any)

**Line 2:** `NDATA PERIOD`
two entries, with the INTEGER `NDATA` denoting the number of data entries on the lines below, and the REAL `PERIOD` specifying the period to be applied for periodic replication of the data, in case the data do not cover the complete experiment duration. If `PERIOD` is set to a negative number (typically $-1.$) replication is not allowed and in case of insufficient time coverage, the model experiment aborts with the invitation to provide a SLV file with sufficiently long data coverage. If `NDATA` is equal to 1, the single provided datum is used as a constant. The periodicity information and the age information are meaningless in this case and ignored, as a consequence.

**Next `NDATA` lines:** `AGE SEALEVEL`
two entries per line, with the REAL `AGE` being the age in years before present (i.e., before 1950 A.D.) and the REAL `SEALEVEL` the corresponding sea-level, in meters relative to present-day, negative values indicating lower-then-present-day levels.

Data do not necessarily have to be regularly spaced in time.

Please check out sample files provided in `DataRepository/MBM/Sealevel`.

### 3.1.6 WEA file

The weathering flux file provides the scenario for the evolution of $CO_2$ consumption rates and river $HCO_3^-$ production rates, by different weathering processes. The WEA file is expected to be in NetCDF format. This file is expected to include the following

- a dimension `age` and its concomitant coordinate variable of the same name (type DOUBLE, units: yr before 1950 A.D.)
- one DOUBLE array `biccc(age)` providing the evolution of the global riverine $HCO_3^-$ input to the ocean resulting from carbonate weathering by carbonic acid (from atmospheric/soil $CO_2$)
- one DOUBLE array `bicsc(age)` providing the evolution of the global riverine $HCO_3^-$ input to the ocean resulting from silicate weathering by carbonic acid

- one DOUBLE array `biccs(age)` providing the evolution of the global riverine $HCO_3^-$ input to the ocean resulting from carbonate weathering by sulphuric acid (from pyrite weathering)
- one DOUBLE array `co2cs(age)` providing the evolution of the global atmospheric $CO_2$ uptake during equilibration with waters arising from carbonate weathering by sulphuric acid
- one DOUBLE array `co2oo(age)` providing the evolution of the global production of atmospheric $CO_2$ from kerogen oxidation
- one DOUBLE array `po4rw(age)` providing the evolution of the global riverine input of dissolved phosphate (from rock weathering)

All fluxes are expected to be expressed in Tmol/yr.

Please notice furthermore that

- the `age` variable may include an attribute of type DOUBLE called `replicate_period`, that indicates if the data can be replicated or not: a positive number indicates the period of replication; a negative one indicates that the data must not be replicated;
- if the `age:replicate_period` attribute is absent, MBM assumes that the data *can* be replicated, and uses a default period of 120000 years;
- if the `age` dimension has a length of `1`, any `age:replicate_period` attribute is ignored and the data is supposed to be replicated with infinite length;
- any one of the flux data arrays an be omitted, in which case MBM uses default constant values (beware of consistency problems, though);
- all of the flux data arrays present must include an `average` attribute of type DOUBLE, that gives the average flux value over the replication period (if `age:replicate_period` is absent, or if it is present and positive) or over the complete time interval covered by the data (if `age:replicate_period` is present and negative).

The CSL output file only includes only the combinations of the above that are relevant for the mass balance equations. The arrays in the CSL file are related to the above as follows:

$$\begin{aligned}
\texttt{swcits} &= \texttt{biccs} - 2 \times \texttt{co2cs} \\
\texttt{swcitc} &= \texttt{biccc} + 2 \times \texttt{co2cs} \\
\texttt{iwcitt} &= \texttt{bicsc} \\
\texttt{volctt} &= \texttt{bicscm}/2 - (\texttt{biccsm}/2 - \texttt{co2csm})
\end{aligned}$$

where

```
bicscm  =  bicsc:average
biccsm  =  biccs:average
co2csm  =  co2cs:average.
```

These are the terms used in the mass balance equations in `eqnoce-medusa.F`.

### 3.1.7 TPR file

This entry is optional; `mbm` uses default temperatures if it is not provided. The format of the file whose name can be provided here is NetCDF. See sample files under `DataRepository/MBM/Temperatures` more more information.

### 3.1.8 FF1 file

This entry is optional and may be used to specify a scenario for a forcing function.

### 3.1.9 AFC file

This entry is optional; it may be used to provide a history of the aragonite fraction of the carbonate export production. The AFC file must be in NetCDF format and is expected to include the following information:

- a dimension `age` and its concomitant coordinate variable of the same name (type DOUBLE, units: yr before 1950 A.D.)
- one DOUBLE array `frcara(age)` providing the evolution of the global aragonite fraction of the carbonate export production (unitary, i.e., units of mol/mol)

Please notice that

- the variable `age` may bear a `replicate_period` attribute, just like in the WEA file (see above for implications);
- if the array `frcara` is not present, a constant default value is used by MBM.

Sample files are provided in `DataRepository/MBM/BioParams/frcara`.

### 3.1.10 CRR file

This entry is optional; it may be used to provide a history of the carbonate-C/organic-C ratio in the biogenic export production in each one of the surface reservoirs. The CRR file must be in NetCDF format and is expected to include the following information:

- a dimension `ori` (Ocean Reservoir ID), whose length must match the number of surface reservoirs in the current model configuration, and its concomitant coordinate variable of the same name (type INT, no units);
- a dimension `age` and its concomitant coordinate variable of the same name (type DOUBLE, units: yr before 1950 A.D.);
- a DOUBLE array `rhocac(age, ori)` providing the evolution of the aragonite fraction of the carbonate export production (mol/mol).

Please notice that

- the variable `age` may bear a `replicate_period` attribute, just like in the WEA file (see above for implications);
- if the array `rhocac` is not present, constant default values are used by MBM.

Sample files are provided in `DataRepository/MBM/BioParams/rhocac`.

### 3.1.11 SHF file

The shelf flux file provides the scenario for the evolution of the carbonate accumulation fluxes on the shelves (`coralo` and `shbnko` in subroutine `SECMEM`. The SHF file is expected to be in NetCDF format. If no file name is provided for the SHF keyword, default values are adopted (from `mod_shelffluxes.F95`). Sample files are provided under `DataRepository/MBM/ShelfFluxes`.

## 3.2 Additional Input Files

MEDUSA requires only two additional files when coupled with MBM.

### 3.2.1 medusa.rrp

This file is used to specify the reaction rate parameters. Its name must always be `medusa.rrp`. It must contain a series of four Fortran 95 namelists. A sample `medusa.rrp` file can be found in `work/mbm-template` or under `src-med/gen` (file `medusa.rrp_template`), once the MEDUSA code for MBM has been generated for the first time. All four namelists must be present, but one or more may be left empty. Empty namelists will trigger the use of default values. All actually adopted constants' values will be reported in the RES file.

### 3.2.2 medusa_seafloor_init.nml

This file (whose name is also fixed) may be used to provide a namelist that defines a globally uniform initial state of the sediment. A sample can be found in `src-med/gen/template` (file `medusa_seafloor_init.nml_template`) once the MEDUSA code for MBM has been generated for the first time. Please refer to that sample and the comments that it includes for more information.

## 3.3 Output Files

Output files are either human-readable text files (diagnostics) or NETCDF files. With the latter, you may use `ncdump -h` *`filename.nc`* to get more information, about variables' names, units, meanings and dimensions.

# References


G. Munhoven and L. M. François. Glacial-interglacial variability of atmospheric $CO_2$ due to changing continental silicate rock weathering : A model study. *J. Geophys. Res.*, 101(D16):21423–21437, 1996. doi: 10.1029/96JD01842.

Guy Munhoven. *Modelling Glacial-Interglacial Atmospheric $CO_2$ Variations : The Role of Continental Weathering.* PhD thesis, Université de Liège, Liège, November 1997. URL `http://www.astro.ulg.ac.be/~munhoven/en/PhDThesis.pdf`.

Guy Munhoven. Glacial-interglacial rain ratio changes : Implications for atmospheric $CO_2$ and ocean-sediment interaction. *Deep-Sea Res. II*, 54 (5-7):722–746, 2007. doi: 10.1016/j.dsr2.2007.01.008.

# Document history

- 03APR2012: first version, relevant to revision 97 of MEDMBM – not introduced into the Subversion repository
- 24APR2012: revised to reflect changes introduced at revision 98 of MEDMBM; minor rephrasing.
- 21MAY2012: revised to reflect changes from revisions 98 to 99 of MEDMBM; minor rephrasing.
- 12JUL2012: improved NetCDF related information, described potential issues and troubleshooting; corrected a few minor text errors.
- 29JAN2014: revised to reflect the changes related to revision 108 (disconnection of the FF1 forcing, inclusion of the AFC and CRR forcings); also improved description of the contents of the WEA file.
- 01DEC2015: revised to reflect the location of the development tree under `branches` in the SUBVERSION repository, to take into account the requirements for the new MEDUSA code configuration and generation procedure for MEDUSA and the modified content of MBM's `DAT` file.
- 28JUL2020: updated to the MEDMBM directory structure valid from rev. 261 onwards; corrected some typos and language errors; corrected for overfull \hbox'es