

Supplementary material to

Model of Early Diagenesis in the Upper Sediment
with Adaptable complexity – MEDUSA (v. 2):
a time-dependent biogeochemical sediment module for
Earth System Models, process analysis and teaching

Building and Running the Test Case Applications

G. Munhoven

Dépt. d'Astrophysique, Géophysique et Océanographie,
Université de Liège,
B-4000 Liège, Belgium,
eMail: Guy.Munhoven@uliege.be

20th April 2021

1 Getting Started

1.1 System requirements

Building the model requires

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

1.1.1 Fortran compilers

The model has been successfully built with the following Fortran compilers

- Intel Fortran, versions 7.0 and later;
- Portland Group, PGF90;
- GFORTRAN.

Current developments by the author mostly call upon GFORTRAN (version 5.4.0).

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.

1.1.2 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.

1.2 Preparation steps

1.2.1 Getting the source code(s)

The source codes for MEDUSA can be obtained in several ways.

1. The compressed archive `medusa_v2.tar.gz` included in the Supplement of the paper is self-contained and includes the required `LIBTHDYCT` and `MUXML` libraries. The source code tree is thus complete and does not have any external dependencies, except for the NetCDF library. For the sake of clarity, we assume here that the archive is unpacked into a directory called `medusa_v2`:

```
mkdir medusa_v2
cd medusa_v2
tar xvfz MY_PATH_TO_ARCHIVE/medusa_v2.tar.gz
```

Please substitute `MY_PATH_TO_ARCHIVE` by the location where you have stored the code archive after extraction from the supplement's ZIP archive. As a result, three directories are created in `medusa_v2`:

```
libthdyct/  - THDYCT library source code;
medmbm/     - MEDUSA and friends source codes;
muxml/      -  $\mu$ XML library source code.
```

The additional `00readme.txt` file informs about the internal SVN revision numbers of the three codes.

2. The codes contained in the `medusa_v2.tar.gz` archive from the paper's Supplement can also be retrieved from Zenodo, where bug-fix releases and updates will be archived. In order to assemble a complete and functional source tree, you will have to download three different compressed archives:

- `libthdyct.tar.gz` for THDYCT – DOI:10.5281/zenodo.4677790;
- `muxml.tar.gz` for μ XML – DOI:10.5281/zenodo.4677788;
- `medmbm.tar.gz` for MEDUSA and applications – DOI:10.5281/zenodo.4677682.

Please make sure to switch to the most recent versions available. For the sake of clarity, we assume again that these three archives are unpacked into a directory called `medusa_v2`:

```
mkdir medusa_v2
cd medusa_v2
tar xvfz MY_PATH_TO_ARCHIVE/libthdyct.tar.gz
tar xvfz MY_PATH_TO_ARCHIVE/muxml.tar.gz
tar xvfz MY_PATH_TO_ARCHIVE/medmbm.tar.gz
```

Please substitute MY_PATH_TO_ARCHIVE by the location where you have stored the code archives downloaded from Zenodo. As a result, three directories are created in medusa_v2:

```
libthdyct/  – THDYCT library source code
medmbm/     – MEDUSA and friends source codes
muxml/      –  $\mu$ XML library source code
```

The additional 00readme_libthdyct.txt, 00readme_muxml.txt and 00readme_medmbm.txt files inform about the internal SVN revision numbers of the respective codes and also state the licenses under which they are made available.

3. By custom access to the SVN repository that hosts the source code and allows you to manually download it to your computing platform where all the necessary tools (SVN client, ...) for this access means have been installed beforehand – you will have been informed about the required credential information to use and steps to carry out in this case.

When following one of the first two pathways, there is only little configuration work required. The Makefiles have been set up so that the directory tree is respected. On LINUX based systems, the installation can most probably be used as is. Adaptations will only be required in case you prefer to use another Fortran 95 compiler than GFORTRAN or if your NetCDF library is in a non-standard place. The third pathway may require some additional configuration as it gives access to the author's development version.

1.2.2 Compiling the auxiliary libraries

The two auxiliary libraries THDYCT (source code in libthdyct) and μ XML (source code in muxml) are compiled as follows:

```
cd libthdyct
# Please check the Makefile and adapt if necessary
make
cd ../muxml/src
# Please check the Makefile and adapt if necessary
make
# cd back to medusa_v2
cd ../../
```

The produced *.a and *.mod files should be left in place. The INCFLAGS and LDFLAGS in all the Makefiles in the medmbm tree have been set so that all the required dependencies can be found there.

1.2.3 Compiling MEDUSACOCOGEN

Before MEDUSA can be used the configuration and code generation tool MEDUSACOCOGEN must be compiled. It is furthermore best to pre-compile the MODLIB library libmodlib.a right away as well. This is not indispensable as that library will be re-compiled during any model build process if necessary. Doing this manually however ensures that the make process uses your preferred compiler.

```
cd medmbm/src-mcg
# Please check the Makefile and adapt if necessary
make
cd lib
# Please check the makefile (sic!) and adapt if necessary
make
```

```
# cd back to medmbm
cd ../../
```

The apps directory under medmbm holds the source codes for COUPSIM and JEASIM; the source code for MBM is in src-mbm.

2 Building and Running the Test Case Applications

2.1 Coupled Simulation Experiment With Resolved Sediment Records: MEDMBM-PT

The source code for the application used in this section is located in src-mbm (not in the apps tree, for historical reasons). The Makefile in src-mbm offers two top targets:

1. medmbm_default, which is the default target and which produces a model version that is equivalent to MBM coupled to MEDUSA-v1 (Munhoven, 2007);
2. medmbm_calc_pt, which builds the enhanced version with a production time concentration tracer attached to calcite used in this study.

To build the application, it is thus sufficient to

```
cd src-mbm
make tidy
make medmbm_calc_pt
```

The produced executables are time-tagged and moved to work/mbm. The medmbm_default target produces an executable with the base name medmbm; with medmbm_calc_pt the executable has the base name is medmbm_pt. The data required to complete the simulation experiment (weathering scenario, etc.) are located under data/MBM. More information about the various configuration, initialization and forcing files required to run run MBM-MEDUSA (or MEDMBM) can be found in section 3 of the *MEDMBM: Coupling MBM and MEDUSA – a tentative handbook* guide (medmbm.pdf).

2.2 The Coupling Simulator COUPSIM

2.2.1 Building the executable

There are four different flavours of the COUPSIM_BEC application. Source code can be found in

- apps/coupsim_bec for the normal single-processor version;
- apps/coupsim_bec_mpi_1D for an MPI version, where the host model has no particular spatial grid (or the grid configuration is ignored) and the sediment columns to process are simply ordered sequentially, allowing the workload to be distributed almost evenly over the available processors;
- apps/coupsim_bec_mpi_2D for an MPI version, where the host model has a rectangularly ordered two-dimensional grid, that is partitioned in equally dimensioned rectangular sub-grids and to be processed by one processor each;
- apps/coupsim_bec_mpi_2DT2D for an MPI version, where the host model grid is organised as a set of tiles and tile-sets (the tile-sets being distributed in a 2D array – the MITgcm uses such a grid) such that each MPI processor works on a set of tiles.

Each flavour comes with two variants: one with a non-linear (default) and one with the linear calcite dissolution rate law. To build the variant with the non-linear calcite dissolution rate law, change directory to its source directory and

```
make tidy
make coupsim_nlcd
```

or `make tidy`; `make` as this is the default target; to build the variant with the linear calcite dissolution rate law

```
make tidy
make coupsim_lcd
```

2.2.2 Preparing and Running the Experiment

The working directory, where the executables are moved to by the `make` process is `work/coupsim` (for all versions, all variants).

Before the model can be used, a few extra data-sets that cannot be provided here for copyright and also space reasons must be downloaded and pre-processed to extract data and derive annual averages. Instructions about how to do this are provided in the `00readme.txt` file in the `data/US-JG0FS` directory from the archive: please `cd` to that directory and execute the commands outlined in that file. Please notice that `NETCDF OPERATORS (NCO)` are required for the pre-processing stage. After these steps have been completed, the application is ready to be run. The working directory `work/coupsim` can be used for the serial and all of the MPI versions as it contains all the necessary configuration and set-up files.

2.3 Complex Site-Oriented Application: JEASIM

Although the primary usage of `MEDUSA` is for coupling to ocean biogeochemical cycle models, it can also be used for the analysis of site data.

2.3.1 Building the executable

The source code for the `JEASIM` application is located in `apps/jeasim`. The `Makefile` therein offers two top targets: `volumeless` (default) and `normal`. The model version produced by the `volumeless` target is most compatible with the model designed by Jourabchi et al. (2008). In that version all solids, except for the non-reactive clay, are supposed to be volumeless. This way, chemical reactions do not have any influence on the advection rate profile, as supposed by Jourabchi et al. (2008). With the `normal` target, solids are considered as usual in `MEDUSA` and the influence of chemical reactions in the calculation of the advection rate profiles is taken into account.

Please notice that this application requires a finer grid than usual `MEDUSA` versions that are coupled to biogeochemical models. For the purpose of this paper a grid without a `DBL` and a `REACLAY` covered by a 321-point grid was used (all bioturbated, as the adopted biodiffusion profile goes to zero continuously). This must be changed in `src-med/mod_gridparam.F` before building the model. The overall extent of 82 cm is set at run-time via the `medusa_grid_config.nml` file.

Before building the default application, the number of grid-points must first be adapted in `src-med/mod_gridparam.F`. For the experiments presented and discussed in the paper, we did not consider any `DBL` (thus leaving `ndn_w2s = 0` at line 15) and increase the number of grid nodes in the bioturbated part in the `REACLAY` realm to 321 (amend line 21 to read `ndn_s2z = 321` and leave `ndn_z2b = 0` at line 28). After this change, the application is ready for building:

```
cd apps/jeasim
make tidy
make
```

The executable's name is time-tagged and moved to `work/jeasim` where it can be run.

2.3.2 Fitting to O₂ and pH profile data

The data (boundary conditions etc.) required to run the model must be provided in the file `jeasim.csv` in the work directory `work/jeasim`. Appropriate instances for `jeasim.csv` can, e. g., be easily derived from the OpenDocument Spreadsheet file `jeasim_v1.ods` (most conveniently processed by OPENOFFICE or LIBREOFFICE. The original data from Jourabchi et al. (2008) are collected in the 'Original' sheet.

The two GDL scripts (also usable with IDL[®]) that were used to carry out the data fitting for the different also provided in `work/jeasim`:

- `medusa_oxygen_fit_script.pro` and `medusa_oxygen_fitfun.pro` for the fits to O₂ profile data;
- `medusa_h_fit_script.pro` and `medusa_h_fitfun.pro` for the fits to pH profile data.

These scripts require that the MPFIT package (see <http://purl.com/net/mpfit> for the most recent version) are installed. `work/jeasim` includes two additional scripts (`read_jeasim_csv.pro` and `write_jeasim_csv.pro`) that are required by the two fitting scripts.

Initialisation files can be generated from the `jeasim_v1.ods`: it is sufficient to save a copy of one of the sheets whose names start with "`jeasim_ini`" in CSV format, making sure

- to use an ASCII/US or ISO-8859-1 character encoding,
- to use commas (,) as field delimiters;
- to use double quotes (") as text delimiters;
- to save the cell content as shown;
- and to quote all text cells.

The `jeasim_ini0` sheet is for the oxygen profile fits only; `jeasim_ini1_4_5`, `jeasim_ini1_2_0` and `jeasim_ini1_1_0` are for the pH profile fits, with calcite dissolution kinetics of order 4.5, 2.0 and 1.0, respectively. `00experiment.log` reports some peculiarities to consider in order to ensure convergence of the fitting process.

The variable `cfn_csvin_jeasim` in the `medusa*_fit_script.pro` scripts holds the name of the CSV file to be used – please amend if necessary. In those scripts, set `i_vl = 1` if the volume-less variant of the application is to be used; other values indicate that the normal version is to be used. The names of the application executables are taken from the `cmd_medusa_exe` variable in the `medusa*_fitfun.pro` files. They are set to

- `./medusa_jeasim_v1.0_321_0` for the volume-less and to
- `./medusa_jeasim.0_321_0` for the normal variants,

respectively.

Finally, please make sure that `cfn_csvin_jeasim` is set to '`jeasim_adj.csv`' in the namelist file with the file names `medusa_jeasim_files.nml` as this is the name that the fitting procedures use for the (modified) data files as they proceed with their iterations.

Profile data for the fits are located under `data/Jourabchi-et al. 2008`.

Results discussed in the paper are included in the `jeasim_v1.ods` sheets whose names start with `jeasim_adj`, one for each `jeasim_ini` sheet.