

# Travaux pratiques des cours OCEA0096-1 et SPAT0025-1

---

## Modélisation de processus sédimentaires La zone de transition de la calcite

Guy Munhoven

Version 1.2.7  
(6 mars 2023)

### Préambule

La version la plus récente de ce document se trouve sur mon site “atelier” Projets d’étude, avec des scripts et utilitaires de support.

Les utilisateurs qui construisent l’application utilisée ici à partir des archives Open Source sur Zenodo, comme décrit dans l’annexe A, sont invités à consulter ce même site pour obtenir la dernière version du script PYTHON `meduti.py` et d’autres utilitaires plus récents.

### 1 Introduction

L’abondance en carbonate (calcite) dans le sédiment superficiel décroît typiquement avec la profondeur du fond marin de quelque 80–90% à faible profondeur à moins de 5% à grande profondeur (Fig. 1). Il en résulte une zone de transition, dont l’étendue et l’apparence varient en fonction de différents paramètres :

- taux de dépôt de matériaux réfractaires (argiles, sables, matière détritique);
- taux de dépôt de calcite;
- taux de dépôt de matière organique;

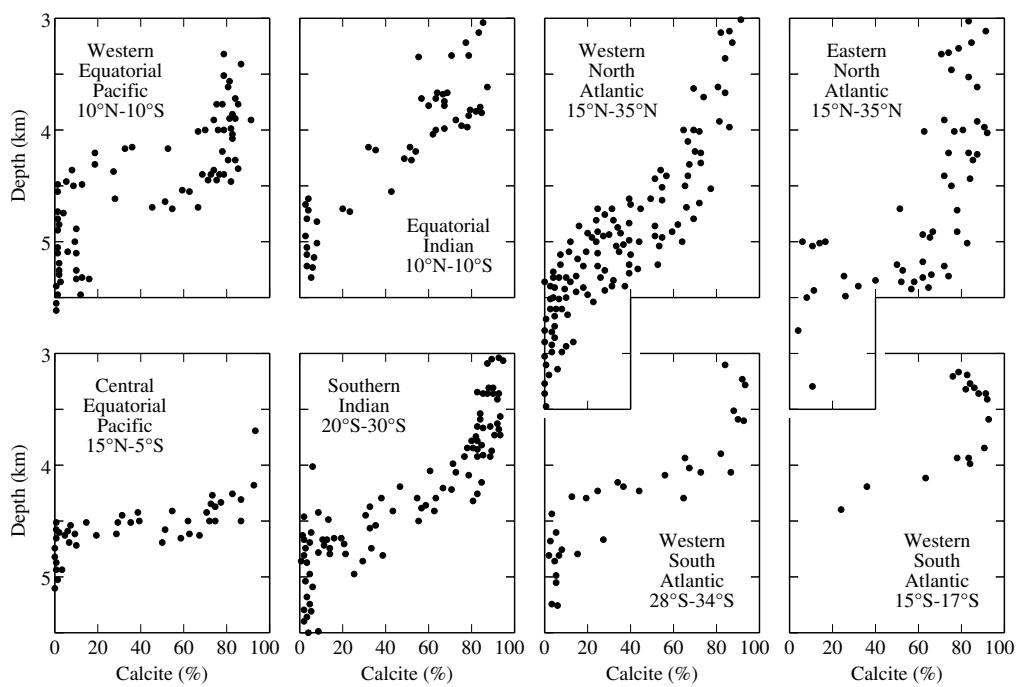


FIG. 1 – Fraction de  $\text{CaCO}_3$  dans le sédiment superficiel en fonction de la profondeur, pour différentes régions de l'océan global (Broecker and Peng, Tracers in the Sea, 1982).

- taux de reminéralisation de la calcite;
- taux d’oxydation de la matière organique.

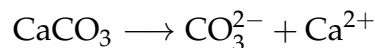
Pour ce projet, nous allons nous focaliser sur l’influence des taux de dépôt et des taux de reminéralisation. Pour les deux aspects, le modèle de diagenèse précoce MEDUSA (Model of Early Diagenesis in the Upper Sediment) en version MEDANA (MEDUSA ANALYSIS) sera utilisé.

Tous ces paramètres peuvent varier en fonction de la profondeur du fond marin. Pour les besoins de cette étude, ce dernier est subdivisé en quelques dizaines d’intervalles de profondeur (typiquement 80 intervalles, distribués de 1000 à 5000 m de profondeur, chaque intervalle couvrant ainsi une tranche de 50 m). À chacun de ces intervalles de profondeur est attaché une colonne sédimentaire, résolvant les 10 cm superficiels de sédiment où ont lieu les transformations diagénétiques précoces les plus importantes qui sont déterminantes pour la composition de sédiment.

## 1.1 Configuration du modèle

La version MEDANA mise à votre disposition est configurée comme suit:

- composition:
  - $\text{CO}_2$ ,  $\text{HCO}_3^-$ ,  $\text{CO}_3^{2-}$ ,  $\text{O}_2$  et  $\text{NO}_3^-$  dans les eaux interstitielles;
  - *clay*, calcite et matière organique (M.O.) en phase solide.
- équilibres:
  - système des carbonates:  $\text{CO}_2 - \text{HCO}_3^- - \text{CO}_3^{2-}$
- processus:
  - dissolution de la calcite ( $\text{CaCO}_3$ ):

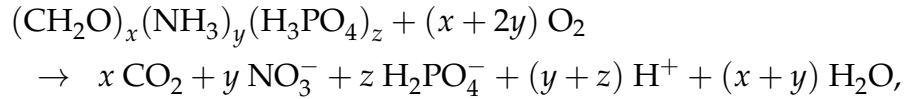


suivant la loi cinétique

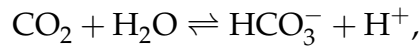
$$T_{\text{calc}} = k_{\text{calc}} [\text{calcite}] (1 - \Omega_{\text{calc}})^n$$

où  $\Omega_{\text{calc}} = [\text{CO}_3^{2-}] / [\text{CO}_3^{2-}]_{\text{sat}}$  est le degré de saturation local de l’eau interstitielle par rapport à la calcite et où  $[\text{CO}_3^{2-}]_{\text{sat}} = K_{\text{satcalc}} / [\text{Ca}^{2+}]$  est la concentration à saturation en  $\text{CO}_3^{2-}$  envers la calcite, calculée à partir de celle en  $\text{Ca}^{2+}$  (paramétrisée à partir de la salinité) et du produit de solubilité de la calcite à saturation,  $K_{\text{satcalc}}$ , calculé, lui, en fonction de la pression, de la température et de la salinité ambiante par

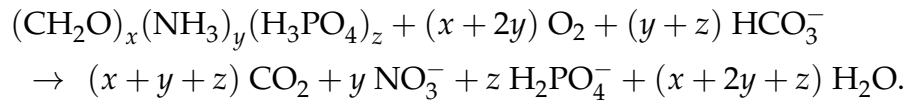
– dégradation aérobie de la matière organique :



réaction reformulée ici en tenant compte de l'équilibre entre  $\text{CO}_2$  et  $\text{HCO}_3^-$ ,



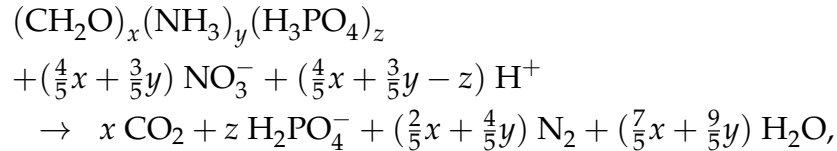
ce qui nous permet d'éliminer  $\text{H}^+$  (qui n'est pas explicitement pris en compte dans le modèle). Ainsi,



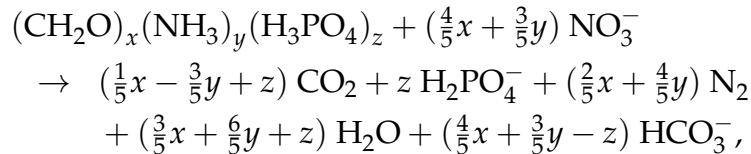
La cinétique de ce processus est régie par une loi de Monod :

$$T_{\text{MOox}} = k_{\text{MOox}} [\text{M.O.}] \frac{[\text{O}_2]}{k_{\text{hsox}} + [\text{O}_2]}.$$

– dégradation de la matière organique par réduction de nitrate et dénitrification complète :



récrit de manière analogue à ci-dessus en prenant en compte l'équilibre entre  $\text{CO}_2$  et  $\text{HCO}_3^-$  pour éliminer  $\text{H}^+$ ,



supposée suivre la loi cinétique

$$T_{\text{MOdn}} = k_{\text{MOdn}} [\text{mat. org.}] \frac{[\text{NO}_3^-]}{k_{\text{hsdn}} + [\text{NO}_3^-]} \left(1 - \frac{[\text{O}_2]}{k_{\text{inox}} + [\text{O}_2]}\right).$$

## 1.2 Tests de sensibilité

### 1.2.1 Taux de reminéralisation

Pour les taux de reminéralisation, deux grandes approches sont envisageables :

1. variation des constantes de réaction
2. différentes lois cinétiques (linéaires, non-linéaires ; en  $\Omega$ , en  $\Delta$  ; ...)

Les constantes de réaction sont contrôlées via le fichier `medusa.rrp` (nom fixe), qui doit se trouver dans le dossier d'où le modèle est exécuté. Les différentes lois cinétiques sont sélectionnées à l'étape de la configuration du code. Pour ce projet, nous allons uniquement modifier les valeurs des constantes de réaction.

### 1.3 Taux de dépôt

Pour les taux de dépôt, nous pouvons encore envisager deux grandes approches :

1. les taux (flux) de dépôt sont identiques aux taux d'exportation (pas de dégradation de la matière organique dans la colonne d'eau) ;
2. les flux diminuent avec la profondeur (dégradation dans la colonne d'eau pendant le transit depuis la surface vers le fond).

La distribution des taux de dépôt est dérivée des informations fournies dans le fichier des forçages (typiquement appelé `medana_forcing.nml` — voir 2.3.4 pour plus de détails).

## 1.4 Alternatives

D'autres aspects peuvent évidemment entrer en compte dans le contrôle de la fraction de carbonate dans le sédiment superficiel des fonds marins :

- mélange de plusieurs classes de carbonate avec des taux de reminéralisation différents
- plusieurs classes de matière organique (approche multi-G), typiquement deux à trois (très labile, labile, réfractaire)
- ...

L'étude de ces aspects demande cependant de reconfigurer le code du modèle et de mettre au point des fichiers de contrôle adéquats. Nous n'allons donc, dans un premier temps, pas poursuivre ces pistes.

## 2 Prise en main du modèle

Afin d'être utilisable, le code source de MEDUSA doit être configuré, généré et complété, puis compilé. Pour les besoins de ce projet, une application particulière, qui ne demande pas d'opération de configuration supplémentaire, a été préparée.

### 2.1 Code source et arborescence des répertoires

Le code source de MEDUSA peut être mis à votre disposition de plusieurs manières :

1. prêt à l'utilisation (pré-installé et pré-compilé) dans une machine virtuelle offrant aussi des outils pour inspecter les résultats (LIBRE-OFFICE, NCBROWSE, OCTAVE) et qui est utilisable sur n'importe quelle plateforme avec VIRTUALBOX installé;
2. par un accès personnalisé au dépôt SVN qui héberge le code source, et qui vous permet de le télécharger manuellement vers votre plateforme de calcul personnelle équipée des outils nécessaires (client SVN, compilateur, ...) – des instructions *ad hoc* vous auront été communiquées dans ce cas.
3. à partir d'un jeu d'archives de codes, à dés-archiver dans un répertoire commun, suivant les instructions fournies en annexe A.

Dans le premier cas de figure, l'arborescence du code de MEDUSA se trouve sous `work/medmbm/trunk4mcg` dans le répertoire principal de l'utilisateur student (`/home/student`). Dans le deuxième cas, elle se trouvera probablement sous `trunk4mcg` dans le répertoire d'installation. Si les instructions données en annexe A sont suivies, elle se trouvera sous `work/medmbm/trunk4mcg` dans le dossier principal de l'utilisateur (de manière similaire au premier cas). Par la suite, nous dénoterons ce répertoire racine par `$MEDREPRACINE`.

Dans `$MEDREPRACINE`, il y a une série de sous-répertoires :

- `src-med` contient le code source du noyau de MEDUSA ;
- `src-mcg` contient le code source et les bibliothèques du configurateur et générateur de code ;
- `docs` contient la documentation générale ;
- `apps/medana_wno3` contient le code source de l'application MEDANA avec dénitrification utilisée pour ce travail<sup>1</sup> ;

---

1. Suivant la version des codes utilisée, le répertoire `apps` peut contenir d'autres sous-répertoires dont les noms commencent par `medana`. Ces derniers incluent des variantes

- `work/medana_wno3` est le dossier de travail par défaut dans lequel les tests de sensibilité sont exécutés.

Au cas où l'exécutable de `medana_wno3` doit être régénéré, il suffit de retourner dans `$MEDREPRACINE/apps/medana_wno3` et d'y effectuer (dans une fenêtre Terminal), les commandes suivantes :

```
make tidy
make
```

Prenez soin de noter le nom de l'exécutable généré, qui contient un tampon temporel caractéristique (date et heure de compilation).

## 2.2 Répertoire de travail

Il est recommandé d'utiliser `$MEDREPRACINE/work/medana_wno3` comme répertoire de travail, même s'il est possible d'utiliser n'importe quel autre répertoire. Ce répertoire contient un jeu complet de fichiers de configuration et d'initialisation pour MEDUSA. Si on désire garder les résultats de plusieurs simulations afin de pouvoir les comparer, on peut cependant y créer des sous-répertoires, et d'adapter les noms des fichiers d'entrée-sortie dans la liste de fichiers (voir section 2.3.1 ci-dessous) afin de les délocaliser vers ces sous-répertoires. Lorsque les opérations de génération et compilation de code lancées par `make` se terminent correctement, l'exécutable se trouve dans `$MEDREPRACINE/work/medana_wno3`.

## 2.3 Fichiers de contrôle

Afin d'effectuer des simulations, le modèle a besoin d'un certain nombre d'informations caractéristiques. Ces informations doivent être fournies dans différents fichiers de configuration et d'initialisation.

- `medusa_medana_files.nml` – noms des fichiers d'entrée et de sortie;
- `medusa.tsi` – contrôle du temps (temps initial, longueur des pas d'intégration, nombre de pas d'intégration);
- `medusa_seafloor_init.nml` – fichier par défaut avec les conditions initiales pour les expériences;
- `medana_forcing.nml` – fichier par défaut pour les conditions aux limites;
- `medana.rrp` – constantes des lois cinétiques des processus.

---

de l'application à utiliser ici. Assurez-vous, s'il-vous-plaît, de bien utiliser `medana_wno3` pour ce projet-ci.

Le répertoire de travail \$MEDREPRACINE/work/medana\_wno3 contient des copies pour tous ces fichiers que vous pourrez adapter suivant besoin. Au cas où vous aurez reçu les codes via la machine virtuelle ou par accès au dépôt SVN, il y a moyen de récupérer facilement les versions originales de ces fichiers. Il suffit d'aller dans ce répertoire et effectuez-y (dans une fenêtre Terminal) les commandes appropriées parmi suivantes, selon le fichier à récupérer en particulier:

```
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
```

Si vous avez par contre assemblé le code à partir des archives de code source ouvert, vous devrez récupérer les versions originales de ces fichiers de l'archive du module medana\_wno3.

### 2.3.1 Liste de fichiers de configuration, d'entrée et de sortie

Les noms de la plupart des fichiers d'entrée et de sortie sont précisés dans le fichier medusa\_medana\_files.nml. Le nom de ce fichier ne peut pas être modifié et ce fichier doit se trouver dans le répertoire de travail (celui d'où medana\_wno3 est exécuté). medusa\_medana\_files.nml contient deux listes de noms, qui fixent les noms de différents fichiers d'entrée et de sortie.

```
1 &nml_cfg
2 cfn_nmlin_init      = 'medusa_seafloor_init.nml'
3 ctitle_ncfiles     = 'Offline_with_MEDANA_WNO3_(393,uXML_33) '
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 /
```



L'entrée `ctitle_ncfiles` (ligne 3) permet de préciser un titre descriptif pour chaque simulation, qui sera transcrit dans tous les fichiers de résultats NETCDF.

### 2.3.2 Contrôle des pas temporels

Le temps initial (valeur sans importance), la durée de chaque pas (en années) et le nombre de pas à effectuer sont précisés dans `medusa.tsi` (nom non modifiable). Ce fichier doit, tout comme le précédent, se trouver dans le répertoire d'où `medana_wno3` est exécuté. Le fichier fourni par défaut peut être utilisé tel quel dans un premier temps. Il convient cependant de vérifier si l'état stationnaire est atteint et, le cas échéant, adapter la longueur de l'intégration.

### 2.3.3 Conditions initiales

Les *conditions initiales* sont, pour les besoins de ce projet, lues à partir du fichier `medusa_seafloor_init.nml`. Le fichier fourni par défaut sera utilisé tel quel pour nos simulations.

```
1  ! Template for medusa_seafloor_init.nml, automatically
   ! generated by
2  ! CREATE_MOD_SEAFLOOR_INIT from the MEDUSA configuration
3  ! utility MedusaCoCoGen.
4  ! Solids' (prefix "solid_") are expected to be given in %dry-
   ! weight
5  ! Porewater solutes' (prefix "porew_") are expected to be given
   ! in mol/m3
6  &nml_seafloor_init
7  solid_pcent_clay = 100.00D+00
8  solid_pcent_calc = 0.00D+00
9  porew_molm3_co3 = 81.76D-03
10 porew_molm3_hco3 = 2307.01D-03
11 porew_molm3_co2 = 31.23D-03
12 solid_pcent_om = 0.00D+00
13 porew_molm3_o2 = 100.00D-03
14 porew_molm3_no3= 35.00D-03
15 /
```

### 2.3.4 Forçages: conditions aux limites

Les *conditions aux limites* et leurs variations sont contrôlées à l'aide du fichier de forçage `medana_forcing.nml`. Le nom de ce fichier peut être modifié dans la liste `medusa_medana_files.nml`. Si on décide de d'organiser les simulations par sous-répertoires, on peut prévoir une copie de ce fichier par simulation, de la stocker dans le sous-répertoire, et d'adapter l'entrée `cnml_nmlin_forcing` dans `medusa_medana_files.nml` en conséquence (faire précéder le nom du fichier du nom de sous-répertoire adéquat, les deux concaténés par un '/').

Ce fichier fixe tout d'abord le nombre de colonnes sédimentaires, via la liste de noms `&nml_forcing_general`, suivant la règle explicitée dans les commentaires:

```
1 &nml_forcing_general
2 ! General config
3 ! n_medanacols
4 ! = 0: all parameters change independently
5 !       (a grid with length PRODUCT(xyz_npt) is created)
6 ! > 0: all parameters change in parallel
7 !       (a grid with length <n_medanacols> is created
8 !       and all xyz_npt below must be either equal to
9 !       1 (value is repeated) or to <n_medanacols>.
10 n_medanacols = 0
11 /
12
```

La deuxième liste de noms, `&nml_forcing_enviro`, fixe les conditions environnementales pour chaque colonne:

- profondeur [m] – entrées préfixées par «`wdbsl_`»;
- température [degrés Celsius] – entrées préfixées par «`wtmpdc_`»;
- salinité [—] – entrées préfixées par «`wsalin_`».

Pour chacun de ces paramètres, il faut fournir une valeur minimale (suffixe «`_min`»), une valeur maximale (suffixe «`_max`») et le nombre de valeurs à considérer (suffixe «`_npt`»). Les valeurs  $x_i$  effectivement utilisées sont calculées selon  $x_i = x_{\min} + i(x_{\max} - x_{\min}) / (n - 1), i = 0, \dots, n - 1$ , où  $n$  est le nombre de points. Comme nous nous intéressons à la forme de la zone de transition, nous allons laisser varier uniquement les profondeurs (ici, 80 intervalles de profondeur, référencés par leurs centres, couvrant le fond marin de 1000 à 5000 m en-dessous du niveau de la mer):

```
13 &nml_forcing_enviro
```

```

14 wdbsl_min = 1025.00D+00
15 wdbsl_max = 4975.00D+00
16 wdbsl_npt = 80
17 wtmpdc_min = 2.00D+00
18 wtmpdc_max = 2.00D+00
19 wtmpdc_npt = 1
20 wsalin_min = 34.70D+00
21 wsalin_max = 34.70D+00
22 wsalin_npt = 1
23 /
24

```

Les concentrations aux limites pour les solutés (carbone inorganique dissous – dic, alcalinité – alk, oxygène dissous – o2, nitrate – no3) sont dérivées des données de la troisième liste de noms, &nml\_forcing\_wconc :

```

25 &nml_forcing_wconc
26 wconc_min_dic = 2.380D+00
27 wconc_max_dic = 2.380D+00
28 wconc_npt_dic = 1
29 wconc_min_alk = 2.480D+00
30 wconc_max_alk = 2.480D+00
31 wconc_npt_alk = 1
32 wconc_min_o2 = 0.160D+00
33 wconc_max_o2 = 0.160D+00
34 wconc_npt_o2 = 1
35 wconc_min_no3 = 0.035D+00
36 wconc_max_no3 = 0.035D+00
37 wconc_npt_no3 = 1
38 /
39

```

De nouveau, il faut fournir les valeurs minimales et maximales, ainsi que le nombre de points à considérer, pour chaque composant dissous. Les valeurs fournies sont typiques pour l’Océan Pacifique.

La distribution des flux de solides pour les différentes colonnes est déterminée à partir des informations fournies dans la quatrième liste de noms, &nml\_forcing\_xpflx. Pour chaque composant solide, un groupe de quatre entrées est prévu : les valeurs minimale et maximale, le nombre de points et l’échelle de variation (linéaire ou logarithmique). Puisque le *rain ratio* est censé jouer un rôle important, un groupe supplémentaire suffixé ‘\_rr’ a été ajouté à la fin, de manière à pouvoir lier les flux de calcite directement aux flux de matière organique. Ce groupe est seulement pris en

considération si xpflx\_npt\_rr est différent de zéro (voir commentaire à la ligne 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 = 0.66666666666666667D+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

```

Enfin, les variations des flux des solides en fonction de la profondeur dans la colonne d'eau peuvent être contrôlées par la dernière liste de noms, &nml\_forcing\_wcprof. Les valeurs de paramètres éventuellement requis (exposants pour le profil de Martin devront suivre cette liste dans l'ordre solides concernés, tels que donnés ci-dessous. Il faut uniquement fournir les valeurs de paramètres requis effectivement.

```

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)
98 wcprof_id_clay = 0
99 wcprof_id_calc = 0
100 wcprof_id_om = 2
101 /
102 0.858

```

### 2.3.5 Constantes de réaction

Les constantes à adopter pour les lois cinétiques adoptées sont fixées dans le fichier `medusa.rrp` (*Reaction Rate Parameters*). Le nom de ce fichier est imposé et le fichier doit se trouver au même endroit d'où le modèle est

exécuté. Il inclut une liste de nom par processus inclus. Les commentaires servent de guide pour la signification des différentes entrées.

```

1  ! Process 1 [CalcDissolution]
2  ! =====
3  ! Expression: k_1 * [Calcite] * (1 - [CO3]/ksat_1)**n_1
4  &nml_CalcDissolution
5  k_1 = 365.25D+00
6  n_1 = 4.5D+00
7  /
8
9
10 ! Process 2 [OrgMatterOxicDegrad]
11 ! =====
12 ! Expression: k_2 * [OrgMatter] * [O2]/(khs_2 + [O2])
13 &nml_OrgMatterOxicDegrad
14 k_2 = 3.2D-02
15 khs_2 = 0.020D+00
16 /
17
18
19 ! Process 3 [OrgMatterFullDenit]
20 ! =====
21 ! Expression: k_3 * [OrgMatter] * ([NO3]/(khs_3 + [NO3])) * (
      kih_3/(kih_3 + O2))
22 &nml_OrgMatterFullDenit
23 k_3 = 3.2D-03
24 khs_3 = 0.005D+00
25 kih_3 = 0.020D+00
26 /

```

## 2.4 Tests de sensibilité

Nous allons commencer par l'expérience la plus complète et nous allons par la suite réduire de plus en plus la complexité, afin d'isoler et de comparer les effets des facteurs les plus importants pour l'abondance de carbonate dans le sédiment de surface en fonction de la profondeur du fonds marin, et donc le contrôle de l'épaisseur de la zone de transition. La configuration de modèle fournie inclut les composants suivants:

- $\text{CO}_2$ ,  $\text{HCO}_3^-$ ,  $\text{CO}_3^{2-}$ ,  $\text{O}_2$  et  $\text{NO}_3^-$  comme solutés;
- clay (matière inerte), calcite et matière organique comme solides.

Les processus inclus sont :

- la dissolution de la calcite, suivant une loi en  $(1 - \Omega)^n$  ;
- la dégradation aérobie de la matière organique, avec une limitation par la concentration en oxygène suivant une loi de Monod ;
- la dégradation anaérobie de la matière organique par réduction de nitrate, avec
  - une inhibition par l’oxygène suivant une loi complémentaire de Monod ;
  - une limitation par la concentration en nitrate suivant une loi de Monod.

Le système des carbonates est supposé en équilibre thermodynamique.

Avant de commencer, il vaut mieux faire une copie de sauvegarde des deux fichiers d’entrée que nous allons modifier au cours des tests de sensibilité ci-dessous (`medusa.rrp` et `medana_forcing.nml`, et éventuellement aussi `medusa_medana.cfg`).

#### 2.4.1 Expérience complète

Le modèle complet tel que décrit ci-dessus est utilisé et de plus, nous adoptons une distribution du flux de matière organique qui décroît avec la profondeur suivant un profil de Martin, avec un exposant de  $-0.858$  (voir fichier de forçage, lignes 91–102, et plus spécialement les lignes 100 et 102 — attention, à la ligne 102, il faut indiquer *l’opposé de l’exposant* effectif, le profil étant toujours supposé décroissant, donc à exposant négatif).

#### 2.4.2 Dégradation anaérobie de la matière organique

Adopter des taux de réaction plus ou moins importants pour ce processus, en modifiant la valeur de `k_3` dans le fichier `medusa.rrp` (ligne 27) :

- augmenter la valeur d’un facteur 2, 5 ou 10 ;
- diminuer la valeur d’un facteur 2, 5 ou 10 ;
- supprimer l’effet de ce processus en mettant une valeur nulle.

#### 2.4.3 Dégradation aérobie de la matière organique

Adopter des taux de réaction plus ou moins importants pour ce processus, en modifiant la valeur de `k_2` dans le fichier `medusa.rrp` (ligne 16) :

- augmenter la valeur d’un facteur 2, 5 ou 10 ;
- diminuer la valeur d’un facteur 2, 5 ou 10 ;

- combiner ces modifications avec la suppression de la dégradation anaérobie.

#### 2.4.4 Dissolution de la calcite

Adopter des taux de réaction plus ou moins importants pour ce processus, en modifiant la valeur de `k_1` dans le fichier `medusa.rrp` (ligne 5):

- augmenter la valeur d'un facteur 2, 5 ou 10;
- diminuer la valeur d'un facteur 2, 5 ou 10.

#### 2.4.5 Flux de matière organique constants

Désactiver l'utilisation du profil de Martin

- en mettant la valeur de `wcprof_id_om` à 0 (ligne 100),
- en enlevant le `'!` en début des lignes 72 et 73, et
- en insérant un `'!` en début des lignes 78 et 79.

Les deux dernières opérations servent à avoir des valeurs de flux identiques dans les cas à flux variable et à flux constant à 3175 m de profondeur, c'est-à-dire, à la profondeur de l'horizon de saturation de la calcite pour les concentrations de DIC et d'alcalinité renseignées dans le fichier des forçages (`medana_forcing.nml`).

Répéter les tests 1 à 4 précédents.

## 2.5 Fichiers de résultats

Lors de son exécution, MEDUSA génère une série de fichiers résultats et de fichiers journal (*log files*). Les plus importants pour les besoins de ce projet sont:

- le fichier `REACLAY`, dont le nom est fixé à l'entrée `cfm_ncout_reaclay` dans `medusa_medana.cfg` (ligne 4), et qui contient les résultats de tous les profils des composants pris en considération;
- le fichier `CSV`, dont le nom est fixé à l'entrée `cfm_csvout_summary` dans `medusa_medana.cfg` (ligne 13), et qui fournit un sommaire de toutes les informations requises pour les besoins de cette étude, dont notamment les fractions moyennes des différents constituants solides dans la couche sédimentaire pour chaque colonne de sédiment.



## A Assemblage du modèle à partir des archives de code source ouvert

Avec la publication de MEDUSA (G. Munhoven, *Geoscientif. Model Dev.* 14(6):3603–3631, DOI: 10.5194/gmd-14-3603-2021) le code source du modèle et de ses dépendances ont été mis à disposition en code source ouvert. Le supplément de l'article inclut une archive avec les codes de la version spécifique utilisée pour les applications de démonstration présentées dans l'article.

Les mêmes codes ont été archivés sur Zenodo, quoique dans trois archives séparées, en raison de licences différentes; les révisions corrigeant des bogues seront aussi mises à disposition sur Zenodo. Ici, c'est la version 2.0.2 de MEDUSA qui est utilisée. En général, il est toutefois recommandé de vérifier sur la page de référence de MEDUSA page sur Zenodo (DOI: 10.5281/zenodo.4677681, où on peut aussi trouver les liens vers les librairies THDYCT et  $\mu$ XML) si des versions plus récentes de ces différents logiciels sont disponibles, et d'utiliser plutôt ces dernières.

Un script BASH est disponible pour télécharger les archives nécessaires depuis Zenodo et le module `medana_wno3` depuis le site web de l'auteur à l'université de Liège afin de réaliser une installation fraîche de la configuration du modèle requise pour effectuer les simulations de ce projet. Lorsqu'il est exécuté depuis un répertoire appelé `work` dans votre répertoire principal, l'arborescence de sous-répertoires créée est compatible avec la description introduite à la section 2.1.

Il suffit de télécharger ce script (version en anglais) dans un sous-répertoire `work` de votre répertoire principal (à créer à l'aide de `mkdir ~/work` s'il n'existe pas encore), de le rendre exécutable et de l'exécuter:

```
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 Utiliser MEDUSA sous Microsoft Windows

En principe il devrait être possible de compiler et d'exécuter MEDUSA de manière native sous Microsoft Windows. Le code en soi est conforme au

standard et dès lors il devrait être suffisant d'avoir à disposition un compilateur Fortran 95 et tous les pré-requis nécessaires (bibliothèques NETCDF, etc.). La chaîne de construction basée sur `make` est cependant essentiellement centrée sur UNIX/LINUX et pourrait éventuellement demander des adaptations extensives afin de pouvoir être utilisée dans un environnement de programmation purement Microsoft Windows. Cela n'a pas encore été essayé et cette manière de procéder ne sera donc pas traitée ici pour l'instant.

## B.1 Cygwin: UNIX pour Windows

L'environnement de programmation CYGWIN fournit une alternative efficace à la compilation en mode natif sur un ordinateur fonctionnant sous Microsoft Windows.<sup>2</sup> CYGWIN permet d'utiliser des codes destinés à un environnement UNIX sur des plateformes tournant sous Microsoft Windows et fournit un moyen assez léger pour utiliser MEDUSA sur une telle plateforme. Les codes source originaux ainsi que les `Makefile` peuvent en effet être utilisés sans aucune modification sous CYGWIN. Après l'installation du système principal (dont l'installateur peut être téléchargé depuis [www.cygwin.com](http://www.cygwin.com)) les paquets suivants devront être installés afin de disposer des différents outils nécessaires (`make`, `wget`, compilateur Fortran, outils de développement NETCDF, ...):

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

Pour chacun de ces paquets, il est possible et recommandé de sélectionner la version la plus récente disponible. Au cas où il était nécessaire de pouvoir accéder au dépôt SVN et de disposer d'un moyen de transférer des fichiers avec d'autres plateformes, les paquets suivants seront utiles:

- `subversion`
- `openssh`

---

2. Des essais avec l'environnement similaire MSYS2 n'ont malheureusement pas été couronnées de succès, étant donné que l'interface Fortran pour l'utilisation de la bibliothèque NETCDF n'est pas incluse dans le dépôt du système et que la compilation à partir du code source a échoué pour des raisons inexplicables.

L'installation résultante demande environ 1 Go d'espace disque (et beaucoup moins en volume de téléchargement) et permet d'utiliser MEDUSA à partir d'un terminal CYGWIN terminal exactement de la même manière qu'à partir d'un émulateur de terminal sur une plateforme UNIX/LINUX.

Afin d'améliorer la facilité d'usage, un environnement de bureau graphique tel que GNOME, KDE, XFCE ou similaire peut être ajouté. Pour XFCE, on pourrait ainsi installer en plus

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

fournissant un environnement minimaliste mais toutefois fonctionnel, incluant le puissant éditeur de programmation GEANY. Veuillez noter que l'installation de cet environnement requiert environ 1 Go d'espace disque supplémentaire.

### **Problèmes potentiels:**

#### **Avez-vous des espaces dans votre nom d'utilisateur Windows?**

CYGWIN ne pourra pas fonctionner correctement si votre nom d'utilisateur Windows, dont le nom de votre répertoire principal sous CYGWIN sera dérivé contient des espaces. Si c'est le cas pour vous, il faudra effectuer quelques opérations supplémentaires. J'ai suivi avec succès la procédure suivante<sup>3</sup> par le passé pour résoudre ce problème pour un nom d'utilisateur `Guy_Munhoven` sous Windows (où `_` simplement indique une espace), à redéfinir sous le nom d'utilisateur sans espaces `guy` sous CYGWIN – veuillez adapter suivant vos besoins:

1. Fermer toutes instances de CYGWIN en cours d'exécution.
2. Ouvrir un Terminal CYGWIN
3. Vérifier si `/etc/passwd` existe – si c'est le cas, effacez-le à l'aide de  
`rm /etc/passwd`
4. Ensuite, effectuez les commandes suivantes:

```
cd /home
mv "Guy_Munhoven" guy
ln -s guy "Guy_Munhoven"
```

---

3. See <https://stackoverflow.com/a/47901556>.

```
mkpasswd -l > /etc/passwd  
sed -i 's/Guy_Munhoven/guy/g' /etc/passwd
```

en prenant soin d'inclure les apostrophes et doubles guillemets (' , resp. ") comme indiqué et en faisant attention aux majuscules et minuscules.

5. Fermez le Terminal CYGWIN – les changements vont devenir effectifs au prochain démarrage de CYGWIN.

Une fois que l'environnement CYGWIN est installé et en état de marche, il suffit de suivre les instructions pour récupérer le code de MEDUSA à partir du dépôt SVN ou d'assembler l'arborescence de code suivant la procédure décrite dans l'annexe A ci-dessus.

## B.2 Post-Traitement: Fortran, Tableurs et Python

MEDUSA inclut une série d'utilitaires pour le post-traitement des résultats. Leur codes source Fortran sont fournies dans le répertoire `uti`. Le script BASH ci-dessus inclut les instructions pour les compiler. Leurs exécutables peuvent être transférés vers un emplacement central où le shell peut les trouver, ce qui permet de les utiliser de partout.

`column_extract2csv` et `sedcore_extract` permettent d'extraire des résultats spécifiques des fichiers NETCDF et de les sauvegarder dans des fichiers CSV, qui peuvent être lus par des tableurs courants.

Traiter un nombre important de fichiers CSV avec un tableur devient cependant rapidement fastidieux et une approche basée sur des scripts serait préférable. PYTHON est un outil efficace et facilement disponible pour ce faire. Les distributions courantes de LINUX fournissent tous les modules PYTHON nécessaires pour le post-traitement des fichiers de résultats de MEDUSA via leurs gestionnaires de paquets respectifs. Malheureusement, le dépôt de paquets de CYGWIN ne comprend pas le module NETCDF4, indispensable, et toutes les tentatives d'installation à l'aide de l'installateur de paquets interne de PYTHON `pip` ont échoué jusqu'à présent. Cette installation est toutefois facile sous Windows. Puisque le système de fichiers de CYGWIN est accessible depuis Windows de manière complète et transparente, l'utilisation de PYTHON depuis Windows est donc tout à fait appropriée.

La *Python Software Foundation* met à disposition toute une série de versions de PYTHON pour Windows. Actuellement, les versions stables les plus récentes sont les 3.10.0 pour Windows 8 et plus récentes, et 3.8.10 pour Windows 7 et plus récentes. Le module `netCDF4` n'est cependant pas encore disponible sous forme de paquet Wheel pour la version 3.10.0. Il

est donc recommandable d'utiliser la 3.8.10 pour toutes les versions de Windows. Il est bien sûr possible de continuer à utiliser une version antérieure de PYTHON 3 déjà installée, pour autant qu'elle inclue les paquets `numpy`, `netCDF4` et `matplotlib`.

Les paquets PYTHON requis peuvent être ajoutés à l'installation de PYTHON à l'aide des commandes suivantes (à effectuer dans une Invite de commandes Windows, `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
```

La première de ces commandes met à jour le gestionnaire de paquets lui-même et la deuxième, optionnelle, installe IPYTHON, un shell interactif PYTHON; les autres installent quatre paquets requis pour le traitement des fichiers NETCDF produits par MEDUSA (le dernier – `pandas` – est optionnel pour l'instant, mais pourrait devenir nécessaire à l'avenir). Le répertoire `uti` contient un module PYTHON, `meduti.py`, qui permet de réaliser rapidement des graphiques de profils de concentrations ou de taux de réaction pour des colonnes individuelles d'une application MEDUSA (à partir des fichiers de résultats NETCDF) ou des transects de la composition moyenne du sédiment superficiel (à partir de `medana_summary.csv`). Après avoir copié `meduti.py` vers un répertoire, où PYTHON peut le localiser, il suffit d'exécuter les instructions suivantes dans votre interpréteur PYTHON préféré depuis le répertoire des résultats :

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

Dans ces instructions, le premier argument à l'appel de `mu.profile(...)` indique le nom du fichier NETCDF de MEDUSA, le second le numéro de la colonne et le troisième le nom de la variable dans le fichier pour lequel un profil doit être tracé. Ce sera toujours le dernier enregistrement temporel qui sera considéré. Les colonnes sont numérotées de (la moins profonde) à 80 (la plus profonde). L'exemple fournit produit ainsi un profil pour la concentration de  $\text{NO}_3^-$  pour le colonne de sédiment numéro 10. Les noms exacts des variables dans les fichiers NETCDF ainsi que leurs métadonnées peuvent être facilement inspectées à l'aide de `ncdump -h filename.nc` depuis une fenêtre terminal LINUX. Les deux fonctions produisent des fichiers graphiques PNG par défaut, mais ce type peut être adapté en modi-

fiant la valeur de `_graphfileext` au début de `meduti.py`.

## Index

\$MEDREPRACINE, 6

arborescence des répertoires, voir répertoires

code source, 6

conditions aux limites, voir fichiers, `medana_forcing.nml`

conditions initiales, voir fichiers, `medusa_seafloor_init.nml`

configuration du modèle, 3

- équilibres, 3

- composition, 3

- processus, 3

  - dégradation aérobie, 4

  - dissolution de la calcite, 3

  - réduction des nitrates, 4

fichiers, 7

- `medana_forcing.nml`, 7, 10–13

- `medusa.rrp`, 7, 13–14

- `medusa.tsi`, 7, 9

- `medusa_medana_files.nml`, 7–9

- `medusa_seafloor_init.nml`, 7, 9

- résultats, 16

- résultats CSV, 16, 20

fichiers de contrôle files, voir fichiers

forçages, voir fichiers, `medana_forcing.nml`

\$MEDREPRACINE, 6

noms des fichiers, voir fichiers, `medusa_medana_files.nml`

paramètres de lois cinétiques, voir fichiers, `medusa.rrp`

pas temporels, voir fichiers, `medusa.tsi`

post-traitement, 20

- fichiers CSV, 20

- Python, 20–22

Python, voir post-traitement, Python

répertoire de travail, 7

répertoires, 6

- `apps/medana_wno3`, 6

- `docs`, 6

src-mcg, 6  
src-med, 6  
work/medana\_wno3, 7

Windows, 17  
  Cygwin, 18–20