[Water Resources Research]

Supporting Information for

What came first, mud or biostabilizers? Elucidating interacting effects in a coupled model of mud, saltmarsh, microphytobenthos and estuarine morphology

Muriel Z.M. Brückner[1], Lisanne Braat[1], Christian Schwarz[1,2], and Maarten G. Kleinhans[1]

[1 Faculty of Geosciences, Utrecht University, PObox 80115, 3508 TC Utrecht, The Netherlands]

[2 College of Earth, Ocean, and Environment, University of Delaware, Lewes, DE, USA]

Contents of this file

Additional Supporting Information (Files uploaded separately)

Delft3D-files (will be provided under https://github.com/UtrechtRiversEstuariesl)
Matlab code of the dynamic vegetation model and the microphytobenthos model (will be provided under https://github.com/UtrechtRiversEstuaries)

Introduction

The supporting information provides more detailed information on the technical aspects of the dynamic vegetation model, including the structure, pre-processing of files and the usage. All files and the model code are provided on GitHub.

In order to use the dynamic vegetation model, the interface to MATLAB, provided by Deltares and further described in their Delft3D-MATLAB manual, needs to be saved on the computer and pathed to MATLAB. This toolbox contains the functions that are required to import Delft3D FLOW output data into MATLAB.

Preprocessing of the Delft3D-files is necessary to guarantee that all files are readable within the dynamic vegetation model. This means that the file names for the veg1.txt, veg.trv and veg.trd-file do not change and the veg.trv-file needs to be empty before the start of the simulation. All files required to run the Delft3D simulation and MATLAB code need to be located in the same folder. Throughout the simulation additional folders will be created that contain the current simulation ('work'-folder) and the results per year ('results_year').

Model time-scales

As we require a synchronization between the different time-scales of the hydrodynamic, morphodynamic and ecological computations, the choice of the model time-scales requires careful consideration. The total simulation time defined in Delft3D must be the sum of all ecological couplings (ETS) carried out in the simulation including an initial run to

$$SimulationTime_{D3D} = n_{ETS,year} \times t_{ETS} \times n_{morph.years} + t_{ETS}$$

With total hydrodynamic simulation time in Delft3D left of the equal sign, $n_{ETS,year}$ the number of ETS per year, t_{ETS} the hydrodynamic time per ETS and $n_{morph.years}$ the number of morphological years. This means that for instance 50 years of morphodynamic computations with 12 ETS (meaning monthly coupling) per morphological year requires a total simulation time of $50*12*t_{ETS} + t_{ETS}$ in hydrodynamic time. The length of the ETS, meaning the period of the hydrodynamic forcing, can be defined by the user.

The time-scales are based on the assumption that simplified hydrodynamic boundaries represent morphodynamic computations with a morphological acceleration factor of the choice of the user. The user has to predefine the length of the ETS that is an appropriate representation of one morphodynamic/ecological computation time-step and needs to apply the morphodynamic acceleration factor (*morfac* keyword) accordingly.

Similarly, the output interval of the results written to the Dleft3D output-file defines the data points used in the vegetation or microphytobenthos calculations. As a result, the output must be written at intervals that are a divisor of the t_{ETS} at as dense time-scales as possible but keeping in mind the computational feasibility. For instance, for an ETS of an M2-tide of 744 minutes the output interval must be set to a divisor such as 74.4 min (1/10 ETS) or 37.2 min (1/20 ETS). Each of these data points will be considered in the calculation of the vegetation and microphytobenthos distribution in the colonization and mortality-modules leading to a more accurate vegetation distribution with smaller time-steps.

Coupling of the Delft3D and the Matlab-model

To facilitate usability, the MATLAB model is started once and calls the Delft3D-model via a batch-file. As the MATLAB -code is written in a loop at the end of the hydro-morphodynamic computations the code continues and computes the required vegetation parameters (Module 1-5). This process is repeated over the entire simulation period, leading to online coupling at time intervals of 1 ETS.

The model structure

The dynamic vegetation model consists of several code modules to allow for vegetation colonization, growth, mortality, as well as for generation of the in- and output-files required to communicate between Delft3D FLOW and the dynamic vegetation model (Fig. 1).

The initialization module automatically reads the parameters defined in Delft3D by reading the main definition file (*mdf*-file). At the same time, the vegetation definition files (veg.trd) is created that is necessary to run vegetation effects in Delft3D. The creation of the latter is linked to the parameters defined in the veg.txt-file that includes all relevant data of the present vegetation. This data includes physical dimensions of the vegetation throughout season and life-stages, resilience parameters to flow velocity and hydroperiod, as well as the seasonal time-steps such as growth period, colonization time-steps and life time.

Lastly, a first hydromorphodynamic computation (ETS 0) is launched to generate the output data necessary to compute the first vegetation distribution.

The output-data is imported into MATLAB to compute the vegetation distribution (Module 1). The hydro-morphodynamic results are processed to determine spatial variation of the water levels, velocities, mud distribution and erosion and deposition throughout the ETS. A sufficient number of saved time-steps within each ETS is crucial to cover the variability in the hydrodynamics over one ETS. This data is used both in the colonization and in the mortality-module. All data is stored and saved in a matrix for post-processing of the results in each result-folder (*d3dparameters.mat*).

Establishment of the vegetation is computed based on the requirements of the vegetation species (colonization strategy 1 and 2 in veg.txt-file) that depends on the inundation period (strategy 1) or both inundation period and mud fraction (strategy 2). All cells that meet the requirements are colonized by the initial fraction defined in the veg.txt-file.

Growth is computed by aging of the plants using an increasing ID every ETS as

$$ID_{ETS} = ID_{ETS-1} + 1$$

that is linked to the vegetation definition file (veg.trd-file). The properties related to each ID were pre-defined during initialization where each ID was linked to physical properties such as height, density, stem diameter and drag coefficient. Hence, the growth curve of the vegetation is reflected in the trd-file.

To compute mortalities in Module 4 the pressures extracted from the Delft3D-output in Module 1 are used. Per cell, all vegetation fractions are separately compared to the strength of the pressures and the according mortalities are computed. First, the critical stem height and root length of each vegetation fraction are compared to the erosional and depositional depth and if exceeded the vegetation fraction is removed. Second, an absolute mortality fraction is computed for each inundation period and flow velocity that is relative to the initial fraction defined in the cells. The mortality fraction depends on the defined mortality thresholds per life-stage in the veg.txt-file and linearly on the strength of each of the pressure. As a result, in each ETS an absolute mortality is computed for all cells and vegetation fractions that are subsequently removed from the cells. The remaining plant fractions are written to the veg.trv-file (Module 5) that is used as input for the trachytope-module in Delft3D.

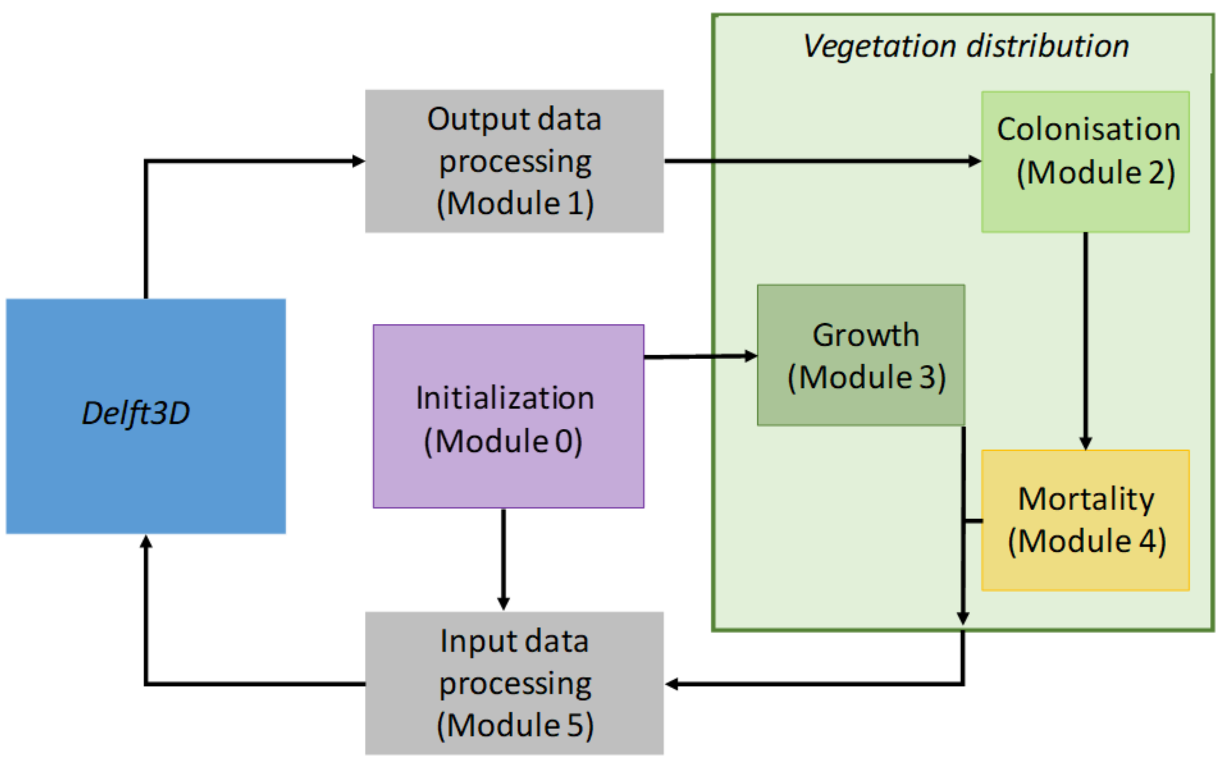

Figure S1. Model structure showing the interaction between Delft3D and the modules of the dynamic vegetation model in the Matlab. The initialization creates the initial input for the dynamic vegetation model and Delft3D (Module 0). The output is loaded into the vegetation code and used to compute colonization and mortality. The combined vegetation data from the growth and mortality modules are processed into Delft3D format (Module 5) and fed into the hydro-morphodynamic computations.

The microphytobenthos (MPB) model

The MPB-model is similarly coupled to the Delft3D model but calculates the MPB distribution by determining microphytobenthos presence when mud content and inundation period are within a specific range. These thresholds are defined in the phyto.txt-file together with growth period and new biotic critical bed shear stress. The spatially varying critical bed shear stress is saved as the *.tce-file that is called by Delft3D, leading to spatially varying erosion properties of the mud fraction.

Preprocessing of the mdf-file

The set-up of the mdf-file is crucial.

The initial starting conditions can either be read from a restart or trim-file using the *restid*-keyword or from initial constants. Importantly, the initial conditions must be written in subsequent order starting with the water level keyword *zeta0* and ending with the initial concentration of the last sediment fraction *Cn*. This is necessary as the removal of the initial conditions is hardcoded in MATLAB between these two values, meaning that all keywords between these two values will be deleted after the initial ETS 0. When the *restid*-option is used, this step is forfeited.

Preprocessing of the veg.txt-file

The veg.txt-file contains all relevant parameters for the computations in MATLAB. It is important to not alter the number of lines and locations of the parameters as MATLAB looks up specific lines in the file to read the data. The first line represent the constant values of the vegetation, including age, colonization strategy, number of life-stages initial plant properties and growth period. The second line includes colonization time-steps. The rest of the lines represent life-stage specific data defining the maximum plant dimensions, the number of years that the plant remains in the life-stage, density and resilience parameters. Each line represent one subsequent life-stage (meaning that line 1 is first life-stage, line 2 second life-stage, etc).

Starting the model

After pre-processing all the files and folders, the model is started via the MATLAB-GUI running the code "*general_input.mat*". The path to the directory has to be added as well as to the files of the interface to MATLAB ("*delft3d_matlab*").