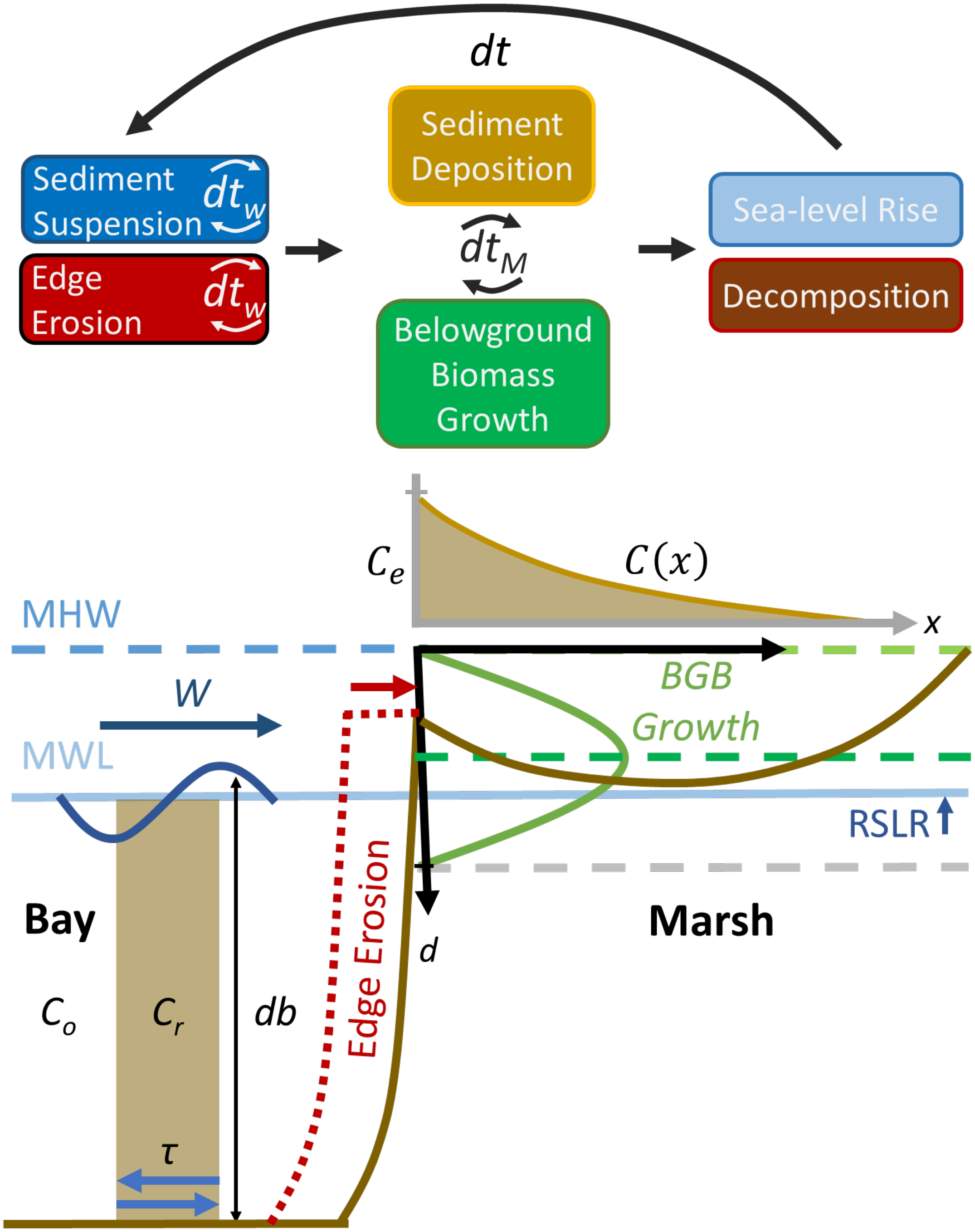
Marsh Transect Model: Installation and Operation Instructions



Contact: Dredging Operations and Environmental Research (DOER) Program, U.S. Army, Engineer Research and Development Center, Vicksburg MS 39180

Table of Contents

INTRODUCTION:

The Marsh Transect Model simulates coastal marsh edge erosion, elevation change, and future migration into upland areas where possible under wave impact and sea level rise. It is a variation of the Bay-Marsh-Forest Transect Carbon Model (BMFTCM) introduced in Kirwan et al. (2016) and simulates wave-induced marsh edge erosion, marsh elevation change due to below-ground mineral and organic material, and migration over decadal timescales. This 1D transect model includes more processes than simplistic representations of marsh elevation but does not require full hydrodynamic model integration to run. The MTM provides a semi-empirical approach to predicting future coastal marsh extent at a point in the landscape. This can allow planners and engineers to determine:

1. if the selected location is viable for the continued persistence of a coastal marsh
2. what (if any) structures may be required to maintain the coastal marsh
3. the capacity of the marsh to reduce erosion and attenuate waves and/or storm surge in the future
4. the maintenance requirements to achieve required coastal marsh geometry

This document guides users through the installing the model, creating input files, and viewing model output. A separate Coastal and Hydraulics Engineering Technical Note (CHETN; Krafft et al., 2021) describes the underlying equations in greater detail.

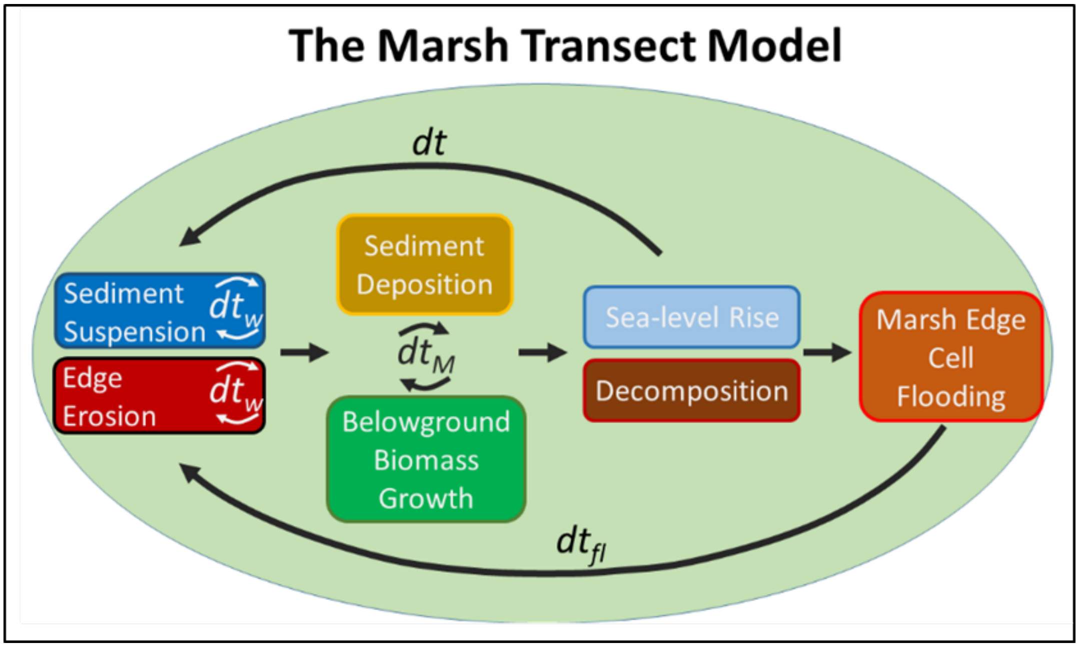
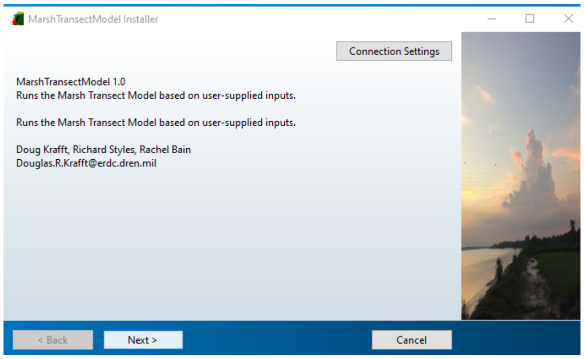


Figure 1: Flow chart of the processes in the MTM and associated time steps. See Krafft et al. (2021) for additional details.

INSTALLATION:

(Note that all instructions are for Windows OS.)

1) Download and run “MTM_installer.exe”. The following screen should appear:



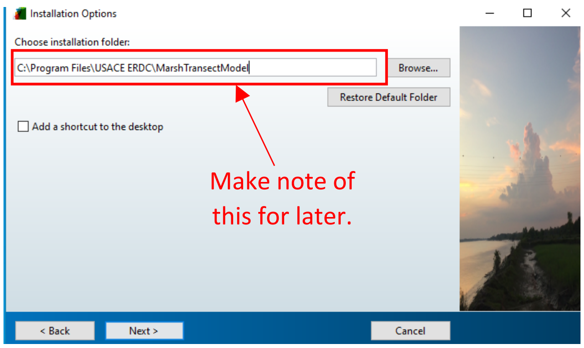
MarshTransectModel 1.0
Runs the Marsh Transect Model based on user-supplied inputs.

Runs the Marsh Transect Model based on user-supplied inputs.

Doug Krafft, Richard Styles, Rachel Bain
Douglas.R.Krafft@erdc.dren.mil

Figure 2: Installation screen corresponding to Step 1.

2) Click “Next”. Choose the installation folder (make sure to remember where you put it!) and click “Next” again.



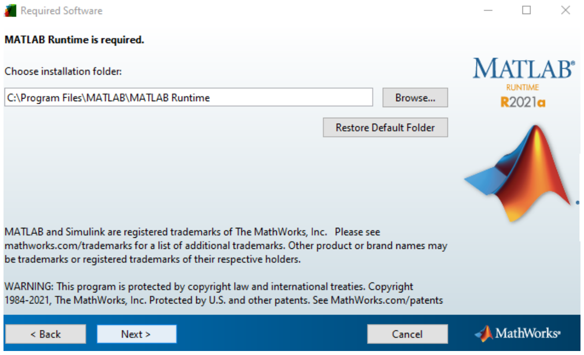
Choose installation folder: C:\Program Files\USACE ERDC\MarshTransectModel

□ Add a shortcut to the desktop

Make note of this for later.

Figure 3: Installation screen corresponding to Step 2.

3) Select an installation folder for MATLAB Runtime, then click “Next”. (Note: if MATLAB Runtime is already installed on your computer, then this step will be omitted during installation.)



MATLAB Runtime is required.

Choose installation folder: C:\Program Files\MATLAB\MATLAB Runtime

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. Please see mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

WARNING: This program is protected by copyright law and international treaties. Copyright 1984-2021, The MathWorks, Inc. Protected by U.S. and other patents. See MathWorks.com/patents

Figure 4: Installation screen corresponding to Step 3.

4) Accept the License Agreement, click “Next” once more, and then click “Install”. Wait for installation to complete. Click “Finish”.

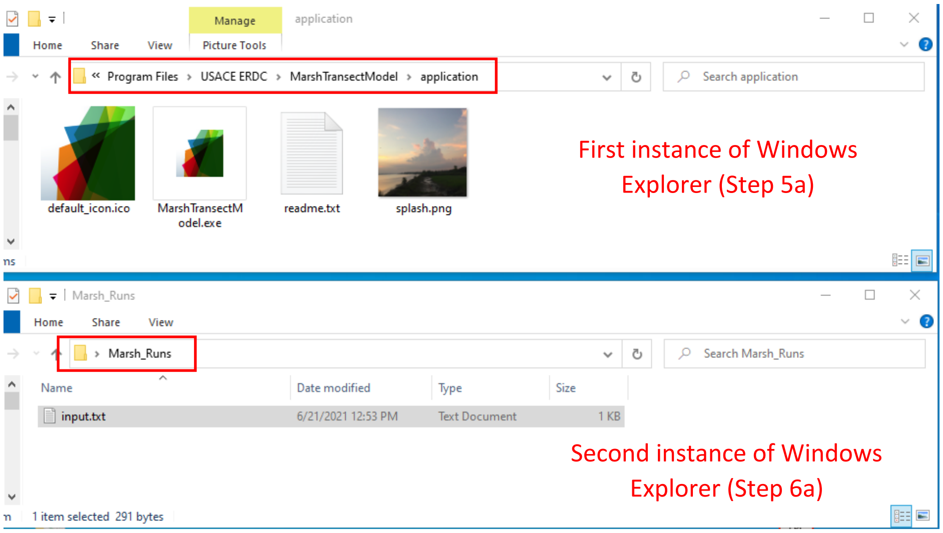
HOW TO RUN (OPTION A—DRAG AND DROP):

5A) In Windows Explorer, navigate to the “Application” directory, which is a subdirectory of the installation directory from Step 2. Using the example from above, you would go to:

```
C:\Program Files\USACE ERDC\MarshTransectModel\application
```

Verify that the directory contains the Marsh Transect Model executable, which should have been installed in Step 4. (Do not double-click or otherwise attempt to open the executable. There is no user interface, so nothing will happen if you try to open it.)

6A) In a second instance of Windows Explorer, navigate to the directory containing the input file. For example, suppose the inputs are stored in “C:\Username\Documents\Marsh_Runs\input.txt”. You must have write permissions in the directory containing the input file because the model output will be written to the same location.

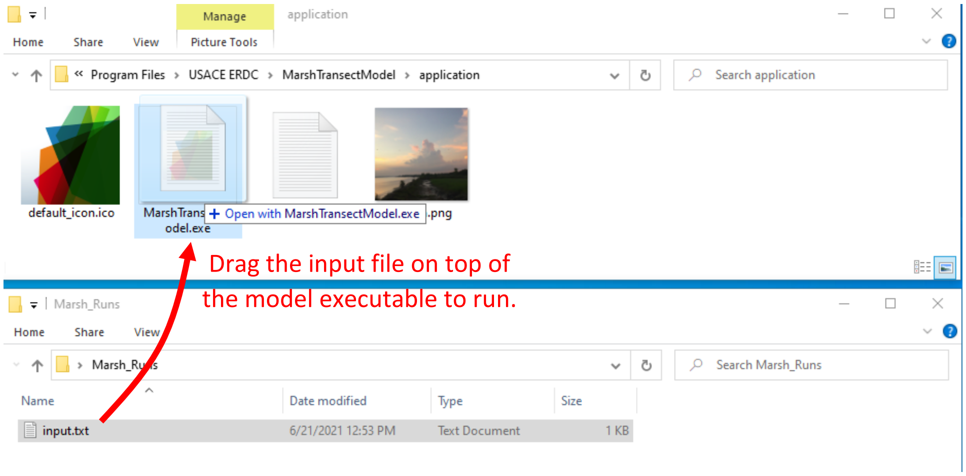


First instance of Windows Explorer (Step 5a)

Second instance of Windows Explorer (Step 6a)

Figure 5: Screenshot of Steps 5A and 6A.

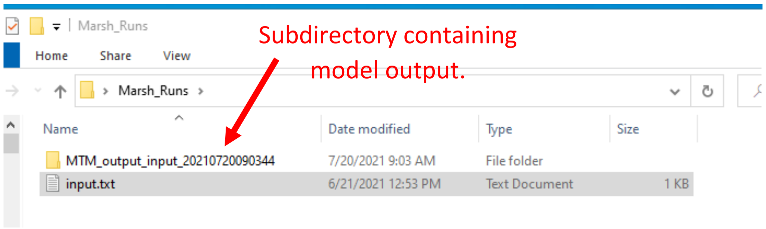
7A) Click and drag the “input.txt” file on top of the Marsh Transect Model executable to run the model. A console screen may appear with several status updates while the model runs, but no additional output will be visible. The calculations may require several minutes before completion.



Drag the input file on top of the model executable to run.

Figure 6: Screenshot of Step 7A.

8A) After the Marsh Transect Model completes its calculations, the output will be written to a subdirectory of the directory containing the input file. The subdirectory will be labeled by appending the run completion time in the format YYYYMMDDhhmmss, so there is no risk of overwriting output from previous runs.



Subdirectory containing model output.

Name	Date modified	Type	Size
MTM_output_input_20210720090344	7/20/2021 9:03 AM	File folder	
input.txt	6/21/2021 12:53 PM	Text Document	1 KB

Figure 7: Screenshot of output corresponding to Step 8A.

HOW TO RUN (OPTION B—COMMAND PROMPT):

5B) From the Command Prompt, navigate to the “Application” directory, which is a subdirectory of the installation directory from Step 2. Using the example from above:

```
cd C:\Program Files\USACE ERDC\MarshTransectModel\application
```

6B) Verify that the Marsh Transect Model executable is in this directory by typing the command **`dir`**. If installation was successful, you should see a list of files in the directory, including “MarshTransectModel.exe”.

7B) Suppose that the inputs are stored in the file “C:\Username\Documents\Marsh_Runs\input.txt”. Note that you must have write permissions to the directory containing the input file or the model will be unable to write output. To run the model, you would type:

```
MarshTransectModel C:\Username\Documents\Marsh_Runs\input.txt
```

into the Command Prompt in the “Application” directory. This command calls the executable to run with the provided input file. Note that the full path to the input file must be explicitly included.

8B) Wait for the model to run. It may take several minutes to complete. While running, you may see several status updates within the Command Prompt, but no other output will appear. After the model finishes, a new command line will become available; *e.g.*, you will see something like

```
C:\Program Files\USACE ERDC\MarshTransectModel\application >
```

At this point, the model has finished running, and no further action is required.

9B) The model output will be written to a subdirectory within the directory containing the input file. The output directory will be labeled with a string of the format YYYYMMDDhhmmss denoting the time of run completion. In this example, the output was written to:

```
C:\Username\Documents\Marsh_Runs\MTM_output_YYYYMMDDhhmmss
```

Note that the name of the output directory will vary according to the name of the input file, but output from previous runs will not be overwritten due to the completion time being appended.

GENERAL NOTES ON INPUT FORMATTING:

- Model parameters are always input via text file.

- Input text files will be easiest for most users to write by editing examples. Please see **SUGGESTIONS FOR CREATING INPUT FILES:** on page 18 for help deciding which example to start from, and **EXAMPLE INPUT TXT FILES:** on page 20 for examples. The following information contains a complete description for writing input files, but modifying an example may provide a simpler approach for users getting started.

- Metric units should be used for all parameters. The assumed units are listed in the variable description.

- Parameter names are case sensitive.

- Each line of the text file corresponds to one model parameter. Lines MUST be separated by the "Return" key (i.e., each parameter must be on its own line).

- The three input types are "scalar", "vector", and "function". (However, not all data types are acceptable for a given parameter). Note that matrices containing multiple columns of data are treated as "vectors" within the Marsh Transect Model.

- Input type MUST be explicitly stated in the text file. For example:
  - **`scalar_parameterName1=7`**
  - **`vector_parameterName2=[1 5 4]`**
  - **`function_parameterName3=@(x) 4*x+16`**

- SCALAR INPUTS: these MUST be a single number without brackets. Positive and negative numbers as well as decimals are all acceptable. Exponential notation can also be used. For the number $A \times 10^{B}$, you would type **AeB** in the input file. For $C \times 10^{-D}$, type **`Ce-D`**. Several examples of exponential notation are shown at the end of this document.

- VECTOR INPUTS: there are two options:
  - If the vector has dimensions of 1xN (N>=1), it can be typed directly into the text file in square brackets, *i.e.*, **`vector_parameterName=[9 2 7 4]`**
  - If the vector has dimensions of 2xN, Nx1, or Nx2, it should be stored in a .csv file. (1xN vectors can also be stored in a .csv file if the user prefers this option.) In the input file, this would be entered as:

  `vector_parameterName=C:\Full\Path\NameOfFileContainingVector.csv`

  - 2xN or Nx2 matrices are labeled "vectors" for the purposes of this code.
  - Note that there are no quotation marks surrounding the path to the .csv file.
  - 1x1 vectors can be input as (for example) **`scalar_parameterName=4`** or **`vector_parameterName=[4]`**. DO NOT use **`scalar_parameterName=[4]`** or **`vector_parameterName=4`**, as these formats may not be recognized.

- FUNCTION INPUTS: These MUST be written in vectorized MATLAB syntax, which will take the form of @(x,y,z,…)= {some function of independent variables x, y, z, etc.}. ***All function-type***

inputs have a default within the model code, so writing new functions is not necessary to run the model. Users who are unfamiliar with MATLAB syntax are advised against writing new functions, as incorrect syntax may prevent the model from running and/or produce unexpected output.

- Examples of common operations include:
  - Multiplication: **`function_parameterName=@(x) 3.14.*x`**
  - Division: **`function_parameterName=@(y) 3.14./y`**
  - Exponents: **`function_parameterName=@(z) z.^3`**
  - Multivariate function: **`function_parameterName=@(x,y) 2.*x./y`**
- Note that within the input file, there should be no quotation marks surrounding the function.
- For univariate functions, the name of the independent variable is irrelevant and does not need to be unique among the individual input functions.
- However, for functions with two or more independent variables, the order of the variables will affect the model output. The required variable order is described, if necessary, in the following section.

REQUIRED PARAMETERS:

- **spinup_flag**
  - This flag tells the code whether it should spin up the marsh stratigraphy in advance of running the model. Please see the technical note by Krafft et al. (2021) for details on model spinup.
  - Must be a scalar, either 0 or 1.

- **spinup**
  - If **spinup_flag**==1, this is the number of layers of model spin-up that will be run. Please see the technical note by Krafft et al. (2021) for details on model spin-up.
  - Must be a scalar >0.

- **RSLR**
  - Relative sea level rise in meters per second.
  - Must be a scalar.

- **bfo**
  - Bay fetch in meters.
  - Must be a scalar >0.

- **db**
  - Bay depth in meters.
  - Depth is treated as positive, *i.e.*, the height of the water column above the floor of the bay.
  - Must be a scalar >0.

- **mwo**
  - Marsh width in meters.
  - Must be a scalar >0.

- **amp**
  - Tidal amplitude in meters.
  - May be a scalar (single tidal amplitude) or a vector (multiple tidal amplitudes). For example, tidal amplitude might be represented as **amp**=[1.5 2.0 2.5] for a bay with a neap amplitude of 1.5 meters, a neap-spring transition amplitude of 2.0 meters, and a spring tide amplitude of 2.5 meters.
  - If **amp** is a vector, then a vector of the same length named **tide_dt** is REQUIRED. This is the number of days per year that each amplitude in amp should be run. Continuing the example with three tidal amplitudes, if each amplitude occurs during one-third of the calendar year, then **tide_dt**=[121.75 121.75 121.75]. (Note that 365.25 days / 3 = 121.75 days).
  - If **amp** is a scalar, then **tide_dt** is not required because that single amplitude will be run for the entire year. If **tide_dt** is input for the single-amplitude case, it will be ignored.

- **Co**
  - Sediment concentration in the bay in kg/m^3.
  - Must be a scalar >0.

*One of the following two options is also required: [1] a vector **time**, or [2] a timestep **dt** and a total duration **dur**. Directly inputting a vector for time (option 1) allows for non-uniform time steps. If option 2 is used, then **time** will be constructed from **dt** and **dur**.*

- **time**
  - Must be a vector of monotonically-increasing time values. Time is measured in days.
  - A non-uniform time step is allowed, provided that time is monotonically increasing.
  - Alternatively, **time** may be omitted if variables **dt** (timestep) and **dur** (total model duration) are input. In this case, the time vector will be constructed as **time=dt:dt:(dur+dt)**.

- **dur**
  - Total model duration in days.
  - Must be a scalar.
  - If **time** is included, then **dur** is recalculated as **time**(end) – **time**(1), and the input value of **dur** will be ignored.

- **dt**
  - Model timestep in days
  - Must be a scalar.
  - If **time** is included, then **dt** is calculated by differencing the values in the **time** vector, and the input value of **dt** will be ignored.

OPTIONAL BIOMASS PARAMETERS:

*General note about biomass parameters **BMax**, **Dmin**, **Dmax**, **mui**, and **mki**: All of these can be scalars or vectors, but they all must have the same dimensions. (If one is a scalar, all must be scalars. If one is a vector, all must be same-sized vectors.) In the case of vector input, each element of the vector corresponds to a different plant species. For example, if **BMax**=[2.5 3.2 4.1], this indicates a marsh with three plant species. The first species has a belowground growth rate of 2.5 kg/m^2/yr, the second species has a belowground growth rate of 3.2 kg/m^2/yr, and the third species has a belowground growth rate of 4.1 kg/m^2/yr.*

- **BMax**
    - This is the peak belowground biomass growth rate.
    - Can be a scalar or a vector.
    - If not supplied, the model uses a default value of 2.5 kg/m^2/yr, representative of some *Spartina alterniflora* (smooth cordgrass).

- **Dmin**
    - This is the minimum depth below high water for biomass growth, measured in meters.
    - Can be a scalar or a vector.
    - If not supplied, the model uses a default value of 0 meters, representative of some *Spartina alterniflora.*

- **Dmax**
    - Maximum depth below high water for biomass growth, measured in meters.
    - Can be a scalar or a vector.
    - If not supplied, the model uses a default of 1.4334***amp**-0.483 meters, representative of some *Spartina alterniflora*.

- **mui**
    - Maximum depth below sediment surface for biomass decomposition, measured in meters.
    - Can be a scalar or a vector.
    - If not supplied, the model uses a default of 0.4 meters.

- **mki**
    - Peak rate of nondimensional biomass decomposition.
    - Can be a scalar or a vector.
    - If not supplied, then the model uses a default value of 0.1.

- **biofun1, biofun2**, …
    - These are functions relating belowground biomass growth rate to soil surface elevation.
    - The number of functions corresponds to the length of **mki**. (For example, if **mki** is 1x2, then you will have inputs **biofun1** and **biofun2**. If **mki** is 1x3, then you will have inputs **biofun1**, **biofun2**, and **biofun3**.)
    - If the number of **biofun** inputs does not match the length of **mki**, then all **biofun** inputs will be ignored.

- If the **biofun** functions are not input, then the ith **biofun** will default to a quadratic polynomial fit to abscissa=[**Dmin**(i), (**Dmin**(i)+**Dmax**(i)/2, **Dmax**(i)] and ordinate=[0, **Bmax**(i), 0].

OPTIONAL FORCING INPUTS:

General note about forcing: The model can be forced by inputting [1] wind speed, [2] wave energy flux, or [3] wave height and period. Each of these options can be a scalar (time constant) or a vector (time varying). If no inputs are included, then the model defaults to a time-constant wind speed. If more than one of [1], [2], and [3] are input, they are sorted in descending order. The model can only consider one forcing mechanism, so it will use [3] instead of [2], and [2] instead of [1] if multiple mechanisms are selected.

- **wind**
  - This is the wind speed in meters per second.
  - Can be a scalar or vector.
  - Default value is 0 m/s.
  - Negative values are interpreted as suspending sediment, but not eroding the marsh edge.

- **wave_time**
  - The time values, measured in days, at which forcing data is applied (i.e., **wind**, or **Hs** and **Tp**, or **W** and **tau**).
  - If the user does not supply a vector, the model will assume that **wave_time** = **time** (i.e., the main time vector).
  - The timestamps in **wave_time** do not need to match the timestamps in **time**, but there should be at least one wind or wave forcing event per time-step in **time**.

- **stress_flag**
  - Flag which indicates that the model should be forced by supplying stress and wave power (instead of wind speed).
  - Options are 1 or 0 (scalar).
  - If **stress_flag**==1, then parameters **W** and **tau** must also be input. Descriptions are below.
  - Default value is 0.

- **W**
  - Wave energy flux, in Watts/meter, that reaches the marsh edge and contributes to edge erosion when **stress_flag**==1.
  - May be a scalar or vector.
  - If **W** is a vector, then a vector of corresponding timestamps named **wave_time** should also be input. If **wave_time** is not input, then the model will assume that the values in **W** correspond to the timestamps in **time**.

- **tau**
    - If **stress_flag**==1, then **tau** is the wave shear stress, in Pascals, that suspends sediment just off of the marsh edge.
    - Can be a scalar or a vector.
    - If **tau** is a vector, then the values of **tau** correspond to the timestamps in **wave_time**. (Note that **W** and **tau** must be the same length and have the same timestamps.)

- **wave_flag**
    - Flag which indicates that the model should be forced by supplying wave height and period.
    - Options are 1 or 0 (scalar).
    - If **wave_flag**==1, then **Hs** and **Tp** must also be input. See descriptions below.
    - Default value is 0.

- **Hs**
    - If **wave_flag**==1, then **Hs** is the wave height, in meters, which forces edge erosion and sediment suspension.
    - **Hs** can be a scalar or a vector.
    - If **Hs** is a vector, then the timestamps corresponding to **Hs** should be supplied in **wave_time**. If **wave_time** is not supplied, then the model will assume that **wave_time** = **time**.

- **Tp**
    - If **wave_flag**==1, **Tp** is the wave period, in seconds, which forces edge erosion and sediment suspension.
    - **Tp** can be a scalar or a vector.
    - If **Tp** is a vector, then the corresponding timestamps should be supplied in **wave_time**. (Note that **Tp** and **Hs** must be the same length and have the same timestamps.)

OTHER OPTIONAL INPUTS:

- **Be**
    - Marsh edge erodibility coefficient in m^2/Watts/s, which relates wave energy flux to marsh edge retreat. The model calculates wave energy flux from wind speed or wave height and period if **Be** is not directly input.
    - Must be a scalar.
    - The default value of **Be** is $3.016x10^{-9}$ m^2/Watt/s, corresponding to 1 meter of lateral edge erosion each year that the wave energy flux reaching the marsh is 10.5 Watts/m (such as a 5 m/s wind blowing over a 2,600 m fetch with a depth of 2 m).
    - Can be tuned to match observations from field sites.

- **P**
    - Tidal period in seconds.
    - Must be a scalar.

  - The model uses a default of 45,000 seconds (i.e., 12.5 hours ≈ M_2 period) if **P** is not provided by the user.
  - Note that all amplitudes in **amp** will have the same period **P**. It is not possible to define different periods for different amplitudes.

- **xz**
  - This is an initial transect of horizontal position (x) and elevation (z), both in meters.
  - The elevation datum for **xz** is related to mean water level by **msl_0**. For example, a z-elevation of 0.1 meters will correspond to mean water level if **msl_0** is also 0.1 meters.
  - **xz** is a Nx2 vector which must be contained within a .csv file. The path to the .csv file should be listed in the input text file. The x-values should be in the first column, and the z-values should be in the second column.

- **flood_check**
  - This input tells the model how often to remove un-vegetated cells on the exterior edge of the marsh. Values are compared to values in the vector "**time**", so units should match.
  - **flood_check** should be a vector.
  - For example, if **time**=[365 730 1095 1460 …] and **flood_check**=[730 1460 …], then the model will remove un-vegetated cells during every second timestep.
  - If not input, the default value is 10^{16} plus the last element of the vector **time**. This arbitrarily large value of **flood_check** prevents the model from checking edge cells while the model is running.
  - Alternatively, a timestep **flood_check_dt** may be included, in which case the model will construct **flood_check** based on the spin-up duration and **flood_check_dt**.

- **flood_check_dt**
  - If **flood_check**=={any arbitrary value} and **flood_check_dt** is also input, then **flood_check** will be constructed using the timestep **flood_check_dt**.
  - IMPORTANT: **flood_check** MUST be included or **flood_check_dt** will be ignored, in which case **flood_check** will be set to the default value of 10^{16}. However, if **flood_check_dt** is included, then **flood_check** can be set to ANY arbitrary value, because it will be overwritten.
  - **flood_check** is calculated as 365***spinup** + (**flood_check_dt**:**flood_check_dt**:(**dur+flood_check_dt**)).

- **rhos**
  - Bulk density of the deposited sediment, including porosity and cohesive effects, in kg/m^3.
  - Must be a scalar >0.
  - If not input, the default value is 2000 kg/m^3.

- **rhoo**
  - Bulk density of organic matter in kg/m^3.
  - Must be a scalar >0.
  - Default value is 85 kg/m^3.

- **of**
  - This is the dimensionless fraction of the initial marsh that is belowground biomass (or can decompose).
  - Must be a scalar >0.
  - Default value is 0.3 if not input.

- **msl_0**
  - Mean sea level in meters at **time**=0.
  - Must be a scalar.
  - Default value is 0.

- **ws**
  - Sediment fall velocity in the marsh (meters per second).
  - Must be a scalar >0.
  - Default value is 5x10^{-5} m/s.

- **tcr**
  - Critical shear stress for sediment suspension just off the marsh edge, in Pascals.
  - Must be a scalar >0.
  - Default value is 0.1 Pa.

- **lambda**
  - Dimensionless erodibility coefficient of the bay area adjacent to the marsh (similar to Kirwan et al. 2016, Mariotti and Carr 2014, and Smith and McLean 1977)
  - Must be a scalar.
  - Default value is 1x10^{-4} (dimensionless).
  - Relates wind or wave forcing to suspended sediment concentrations following the equation:

$$C_r = \rho_s \lambda \frac{\tau - \tau_{cr}}{\tau_{cr}} \left(1 + \lambda \frac{\tau - \tau_{cr}}{\tau_{cr}}\right)$$

where

ρ_s = (rho_s) the bulk sediment density (kg/m^3)

λ = (lambda) a non-dimensional sediment erodibility coefficient

τ = (tau) shear stress (Pa)

τ_{cr} = (tau_cr) critical shear stress (Pa).

C_r = (C_r) locally suspended sediment concentration (kg/m^3)

- **c_flag**
  - Flag which allows the user to directly specify the sediment entering the marsh.
  - Options are 1 or 0 (scalar).
  - Default value is 0.

- **C**
    - If **c_flag**==1, then this is the sediment concentration entering the marsh.
    - Units are kg/m^3
    - Note that **C** is capitalized and that the variable names are case sensitive.
    - **C** must be a scalar (time-constant value).

- **cm1**
    - **cm1** is a nondimensional multiplicative coefficient which scales the computed suspended sediment concentrations before the sediment enters the marsh.
    - For example, **cm1** might be used to produce hypothetical results for a dredged sediment berm that doubles the suspended sediment concentration.
    - The default value is 1.

- **numiterations**
    - This is the number of steps over which to apply sediment deposition and belowground biomass growth.
    - The model supplies a default value of 500, which users are unlikely to need to alter.
    - If used, should be a positive scalar.

- **organic_dep_autoch**
    - Autochthonous carbon matrix, in kg, for initial marsh conditions.
    - Should be positive.
    - Users only need to specify this value if they also intend to specify starting organic content that is not uniform or generated by the spinup function.
    - Should be MxNxL, where M is the number of cells in the domain, N is the number of spinup layers, and L is the number of plant communities.

- **mineral_dep**
    - Mineral sediment matrix, in kg, for the initial marsh conditions.
    - Should be positive.
    - Users are unlikely to need to specify.
    - Should be MxN where M is the number of cells in the domain and N is the number of spinup layers.

- **fetch_table**
    - Nx2 vector of [angle, fetch].
    - Angles are measured in radians.
    - Only relevant when using wind forcing and **theta**. Has not been tested extensively, but should allow a user to specify different fetch distances for different wind angles, with the model interpolating between specified values.

- **theta**
    - Radian angle of wind or wave forcing. Will modify edge erosion by cos(**theta**).
    - Should be a scalar or vector of the same dimensions as the wind or wave forcing data.

    - Angles directed away from the marsh (**theta** < -pi or **theta**>pi) will contribute to suspended sediment but not marsh edge erosion. Angles directed towards the marsh (-pi < **theta** <pi) will both suspend sediment and contribute to marsh edge erosion.
    - Default value is 0, so users only need to specify if they intend to modify edge erosion or are using the vector **fetch_table**.

- **Xgrid**
    - Vector of monotonically-increasing x positions in meters.
    - Users would only need to specify as **Xgrid** if they intend a non-uniform grid with the built-in spinup function.
    - Alternatively, **Xgrid** can be constructed from "**dx**".

- **dx**
    - If **dx** is input and **Xgrid** is not input, then **Xgrid** is calculated as (1:length(**elevation**))***dx**.
    - If **dx** is input and **Xgrid** is also input, then **dx** is ignored.
    - dx should be in meters

- **elevation**
    - Matrix of elevations for each sediment cohort in each cell.
    - Users should only need to specify if they want to specify non-uniform starting stratigraphy (i.e. the vertical distribution of decomposable organic material in each cell).

- **marsh_edge**
    - Indicates the x-location of the marsh edge.
    - If not included, then **marsh_edge** location is assigned to the position along the marsh profile with the maximum dz/dx for each timestep.

ADDITIONAL INPUT OPTIONS FOR THE SPINUP FUNCTION:

- **upland_width**
    - This is the width of non-inundating land inland of the marsh (in meters) for the spinup function. As relative sea level rises during the simulation, the marsh will migrate into upland areas.
    - Must be a scalar >0.
    - Default value is 0 meters.

- **slope**
    - Slope of upland area for the spinup function
    - Must be a positive scalar
    - Default value is 0.005 (which corresponds to 1:200)

- **m**
    - slope of underlying layer of entirely mineral sediment for spinup function
    - should be a positive scalar
    - Default value is 0.00075 (which corresponds to 3:4,000)

- **RSLR_spinup**
    - Relative sea level rise during the spinup evaluation
    - Must be a positive scalar in m/s
    - Is set to match RSLR in the rest of the model unless specified

USER-DEFINED EDGE EROSION EQUATIONS:

The model includes the capacity to calculate marsh edge erosion from a user defined function, if required. Due to the existence of a pre-defined equation within the code, this feature is entirely optional; the model will still operate without any user-defined functions. THE FUNCTION MUST BE WRITTEN IN MATLAB SYNTAX, SO MATLAB FAMILIARITY IS STRONGLY RECOMMENDED AS A PRE-REQUISITE FOR WRITING USER-DEFINIED EDGE EROSION EQUATIONS. Users who are unfamiliar with MATLAB syntax are advised against inputting new functions. An incorrectly-formatted function may cause the model to fail and/or produce unexpected output. The model contains default functions which will be used automatically if user-defined equations are not input.

The implementation of user defined edge erosion functions is somewhat simplistic, and does not meet the same standard of user-friendliness that some features provide, but does allow the user to specify a unique edge erosion equation as a function of a variety of variables. Because of the implementation strategy VARIABLES THAT ARE USED IN THE FUNCTION HANDLE MUST BE DEFINED IN THE **CORRECT ORDER** IN THE INPUT. As a result, users intending to specify a non-standard edge erosion equation MUST begin the input line with

```
function_edge_erosion_eqn=@(W,tau,Be,theta,bgb,e_m,db,amp,rho_m,C_e,v1,v2)
```

where:

- **W** is wave power in Watts/m
- **tau** is wave shear stress in Pascals
- **Be** is the marsh edge erodibility coefficient in m^2/Watts/s
- **theta** is wind or wave angle in radians
- **bgb** is recent belowground biomass growth in kg/m^3
- **e_m** is marsh edge elevation in meters in the local reference coordinate system
- **db** is bay depth (positive down) in meters relative to the local reference coordinate system after an adjustment for relative sea level rise
- **amp** is tidal amplitude in meters
- **rho_m** is marsh edge cell density in kg/m^3
- **C_e** is local suspended sediment concentration in kg/m^3
- **v1** allows the user to specify an additional time varying parameter in the marsh edge erosion equation.
- **v2** allows the user to specify another additional time varying parameter in the marsh edge erosion equation.

As an example, to prescribe an edge erosion equation that matches the default, a user would enter

```
function_edge_erosion_eqn=@(W,tau,Be,theta,bgb,e_m,db,amp,rho_m,E_r,E_t,C_e,v1,v2) max([Be*W*cos(theta),0])
```

Placing the erosion rate result inside of the max function after concatenating with 0 ensures that negative edge erosion rates will not be calculated.

Four example input files named

- "input_simple_example_mod.txt",
- "input_simple_example_mod_bgb.txt",
- "input_simple_example_mod_SSC.txt",
- and "input_advanced_mod.txt"

are provided at the end of this document. These show several possibilities for the edge erosion equation and the appropriate syntax for entering the function.

The only specific error checking ensures that the user input function handle can accommodate the correct number of variables and return a numeric scalar. If any of those conditions are not met, the model will revert to the default function and alert the user in the error message file.

RESOLVING ERRORS:

The model and wrapper have the capacity to resolve some errors, but users may encounter errors not directly addressed. For a list of warnings and error messages encountered in a particular model run, see out_message_file_YYYYMMDDhhmmss.txt. Unanticipated errors which cause the model to crash are represented near the end of the out message file as:

'MATLAB returned the error message "<error message>" in line <line number> of <MATLAB file>.m'

To see what input the model corrected or supplied, see adjusted_input_file_ preview_YYYYMMDDhhmmss.txt

SUGGESTIONS FOR CREATING INPUT FILES:

Most users should start from an example and modify values to match the intended use. Users may select an example using the following logic:

- Is the elevation known?
    - Yes:
        - Do you need a variety of tidal datums? Is a single tidal amplitude inadequate?
            - Yes:
                - Try starting with input_advanced.txt
            - No:
                - Try starting with input_intermediate_example.txt

    - No:
        - Try starting with input_even_simpler_example.txt
- What forcing information do you have?
    - Wind speed:
        - Use input_even_simpler_example.txt to look at wind forcing
    - Wave Height and Period:
        - Use input_simple_wave_example.txt to look at forcing the model with wave height and period.
    - Wave Power and Stress:
        - Use input_simple_stress_example.txt to look at forcing the model with wave power and stress.

The model has some capability to supply default values for variables which the user has not specified. After running an example, see adjusted_input_file_preview_<time>.txt for values the model used (or had access to). Data that are input in an acceptable format are passed to the model and will be displayed in the adjusted input file preview. This adjust input file preview should help adapting examples for intended use and exploring additional model capabilities.

EXAMPLE INPUT TXT FILES:

Example 1: input_even_simpler_example.txt

- This is an example in which an elevation profile is unavailable. However, the depth of bay (2 meters), bay fetch (5000 meters), and marsh width (1000 meters) are provided by the user. Based on these values, the model will spin-up the initial stratigraphy.
- Forcing is with a constant wind speed directed towards and perpendicular to the marsh edge. Tidal amplitude is constant.

```
scalar_spinup_flag=1
scalar_spinup=25
scalar_RSLR=3.1710e-11
scalar_bfo=5000
scalar_db=2
scalar_mwo=1000
scalar_upland_width=500
scalar_amp=0.625
scalar_P=45000
scalar_Co=0.03
scalar_Be=1.5316e-09
scalar_Bmax=2500e-03
scalar_Dmin=0
scalar_wind=5
scalar_dur=15000
scalar_dt=365
```

Example 2: input_simple_wave_example.txt

- This example is geometrically identical to Example 1, but the model is forced by waves of a constant height (0.2 m) and period (2 seconds).

```
scalar_spinup_flag=1
scalar_spinup=25
scalar_RSLR=3.1710e-11
scalar_bfo=5000
scalar_db=2
scalar_mwo=1000
scalar_upland_width=500
scalar_amp=0.625
scalar_P=45000
scalar_Co=0.03
scalar_Be=1.5316e-09
scalar_Bmax=2500e-03
scalar_Dmin=0
scalar_dur=15000
scalar_dt=365
scalar_wave_flag=1
scalar_Hs=0.2
scalar_Tp=2
```

Example 3: input_simple_stress_example.txt

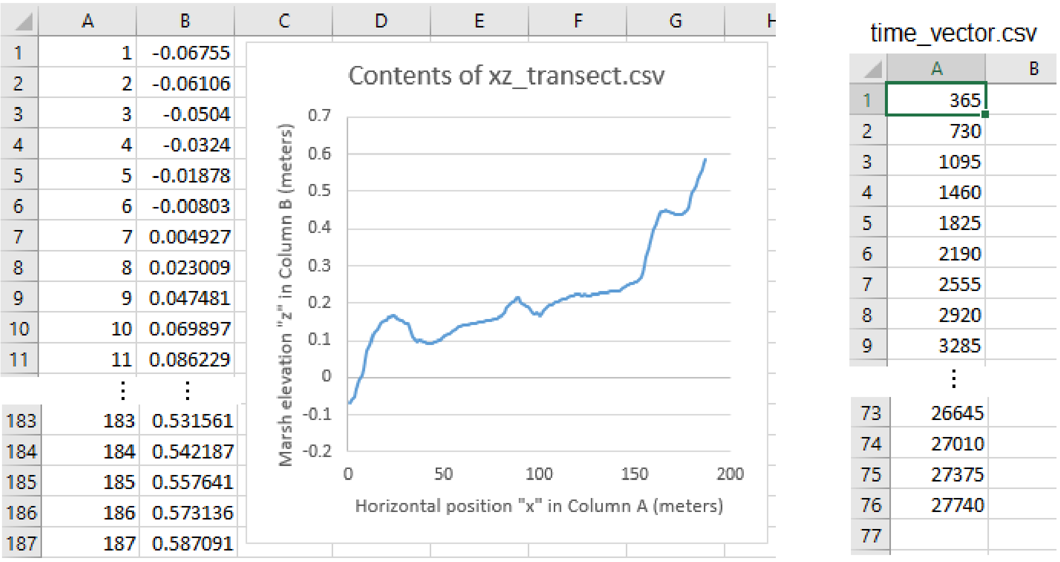
- This example is geometrically identical to Examples 1 and 2, but the model is forced by a time-constant wave energy flux of 21 Watts/meter and a shear stress of 0.16 Pascals.

```
scalar_spinup_flag=1
scalar_spinup=25
scalar_RSLR=3.1710e-11
scalar_bfo=5000
scalar_db=2
scalar_mwo=1000
scalar_upland_width=500
scalar_amp=0.625
scalar_P=45000
scalar_Co=0.03
scalar_Be=1.5316e-09
scalar_Bmax=2500e-03
scalar_Dmin=0
scalar_dur=15000
scalar_dt=365
scalar_stress_flag=1
scalar_W=21
scalar_tau=0.16
```

Example 4: input_intermediate_example.txt

- This is an example with a bay fetch of 1990 meters, a bay depth of 1.2 meters. The initial marsh transect is input as a vector, which is stored as a .csv file. Note that the full path to the .csv file appears in the input file.
- The model timestamps are also input as a vector in this example.
- There are two biomass fractions, which are indicated by the individual values in “mki”, “mui”, “Dmax”, “Dmin”, “Bmax”, and the two biomass functions “biofun1” and “biofun2”.

```
scalar_spinup_flag=0
scalar_spinup=5
scalar_RSLR=1.4269e-10
scalar_bfo=1990
scalar_db=1.2
scalar_Co=0.015
scalar_wind=5
vector_time=C:\Users\Username\Documents\Path\To\File\time_vector.csv
scalar_dur=18250
scalar_dt=365
scalar_amp=0.5
vector_xz=C:\Users\Username\Documents\Path\To\File\xz_transect.csv
vector_mki=[0.1 0.1]
vector_mui=[0.4 0.4]
function_biofun1=@(z) 4*(-4.4804e+05*z.^5+2.6947e+05*z.^4-1.1732e+03*z.^3-
1.8056e+04*z.^2+824.2899*z+571.4103)/1000
function_biofun2=@(d) (-6*(0.22*100-d*100-25).^2+375.7*(0.22*100-d*100-25)-
3315.8)/1000
vector_Dmax=[0.42 -0.18]
vector_Dmin=[-0.18 -0.48]
vector_Bmax=[2.5 2.5]
scalar_flood_check=10e+16
scalar_rhos=500
scalar_of=0.3
```


Contents of xz_transect.csv
Marsh elevation "z" in Column B (meters)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
50
100
150
200
Horizontal position "x" in Column A (meters)
C
D
E
F
G
H

	A	B
1	1	-0.06755
2	2	-0.06106
3	3	-0.0504
4	4	-0.0324
5	5	-0.01878
6	6	-0.00803
7	7	0.004927
8	8	0.023009
9	9	0.047481
10	10	0.069897
11	11	0.086229
	⋮	⋮
183	183	0.531561
184	184	0.542187
185	185	0.557641
186	186	0.573136
187	187	0.587091

time_vector.csv

	A	B
1	365	
2	730	
3	1095	
4	1460	
5	1825	
6	2190	
7	2555	
8	2920	
9	3285	
	⋮	
73	26645	
74	27010	
75	27375	
76	27740	
77		

Figure 8: Contents of input .csv files.

Example 5: input_advanced.txt

- This example is geometrically identical to Example 4, but it uses three tidal amplitudes rather than one. The initial elevation profile is input as a .csv file, which is shown in the figures for Example 4.
- In contrast to Example 4, the time vector is constructed from “dur” and “dt”.

```
scalar_spinup_flag=0
scalar_spinup=5
scalar_RSLR=1.4269e-10
scalar_bfo=1990
scalar_db=1.2
scalar_Co=0.015
scalar_wind=5
scalar_dur=18250
scalar_dt=365
vector_amp=[0.16 0.22 0.28]
vector_tide_dt=[121.6667 121.6667 121.6667]
vector_xz= C:\Users\Username\Documents\Path\To\File\xz_transect.csv
vector_mki=[0.1 0.1]
vector_mui=[0.4 0.4]
function_biofun1=@(z) 4*(-4.4804e+05*z.^5+2.6947e+05*z.^4-1.1732e+03*z.^3-
1.8056e+04*z.^2+824.2899*z+571.4103)/1000
function_biofun2=@(d) (-6*(0.22*100-d*100-25).^2+375.7*(0.22*100-d*100-25)-
3315.8)/1000
vector_Dmax=[0.42 -0.18]
vector_Dmin=[-0.18 -0.48]
vector_Bmax=[2.5 2.5]
scalar_rhos=500
scalar_of=0.3
scalar_test=1
```

Example 6: input_simple_example_mod.txt

- This example is identical to Example 1, except the edge erosion function has been input by the user.
- Notice that all independent variables (W,tau,Be,theta,bgb,e_m,db,amp,rho_m,C_e,v1,v2) must be listed ***in that order*** or the model will revert to the default edge erosion function. However, it is not necessary to *use* all variables from the list in the function. See the example below.

```
scalar_spinup_flag=1
scalar_spinup=25
scalar_RSLR=3.1710e-11
scalar_bfo=5000
scalar_db=2
scalar_mwo=1000
scalar_upland_width=500
scalar_amp=0.625
scalar_P=45000
scalar_Co=0.03
scalar_Be=1.5316e-09
scalar_Bmax=2500e-03
scalar_Dmin=0
scalar_wind=5
scalar_dur=15000
scalar_dt=365
function_edge_erosion_eqn=@(W,tau,Be,theta,bgb,e_m,db,amp,rho_m,C_e,v1,v2)
max([Be*W*cos(theta),0])
```

Example 7: input_simple_example_mod_bgb.txt

- This example applies an edge erosion equation that prevents edge erosion in wave conditions that are below a threshold and modifies edge erosion rates when erosion does occur.
- The threshold is dependent on belowground biomass growth.
- This equation is adapted from Mariotti and Fagherazzi (2010) as,

  $$E = B_e(W - W_{cr})\left(1 + k_v \frac{bgb}{BMax}\right)$$

  when $W > W_{cr}\left(1 + k_v \frac{bgb}{BMax}\right)$

  and 0 when $W < W_{cr}\left(1 + k_v \frac{bgb}{BMax}\right)$

  where
  - E is marsh edge erosion rate in m/s
  - Be is the edge erodibility coefficient (B_e)
  - W is wave power
  - W_{cr} is the critical wave power above which wave power erodes the marsh edge, set to 3 W/m to match the low end of the range listed in Mariotti and Fagherazzi (2010)
  - k_v is a dimensionless coefficient set to 5 to match Mariotti and Fagherazzi (2010)
  - bgb is the recent belowground biomass growth rate in kg/m^3/year
  - BMax is the peak belowground biomass growth rate in kg/m^3/year but must be specified directly in the function handle call because the model is written to accommodate belowground biomass growth rates specified as functions as well. The value of 2.5 kg/m^3 is used in this example.
- The modifications to the edge erosion equation are shown in the example .txt file which follows. Note that the conditional statement (W>3*(1+5*bgb/2.5)) evaluates as 1 if wave power exceeds the threshold and 0 if it does not.

```
scalar_spinup_flag=1
scalar_spinup=25
scalar_RSLR=3.1710e-11
scalar_bfo=5000
scalar_db=2
scalar_mwo=1000
scalar_upland_width=500
scalar_amp=0.625
scalar_P=45000
scalar_Co=0.03
scalar_Be=1.65e-09
scalar_Bmax=2500e-03
scalar_Dmin=0
scalar_wind=5
scalar_dur=15000
scalar_dt=365
function_edge_erosion_eqn=@(W,tau,Be,theta,bgb,e_m,db,amp,rho_m,C_e,v1,v2)
max([Be*(W-3*(1+5*bgb/2.5)) * (W>3*(1+5*bgb/2.5)),0])
```

Example 8: input_simple_example_mod_SSC.txt

- This example applies an edge erosion equation that balances erosion due to wave power with marsh progradation due to suspended sediment deposition.
- This equation is modified from Kirwan et al. 2016 and may be expressed as,

$$E = B_e W - \frac{k_a w_{sf} C_r}{\rho_s}$$

  where
  - E is marsh edge erosion rate in m/s
  - B_e is the edge erodibility coefficient (B_e)
  - W is wave power in W/m
  - k_a is the progradation coefficient (set to 2 to match Kirwan et al. 2016)
  - w_{sf} is settling velocity just outside of the marsh in m/s (set to $0.5x10^{-3}$ to match Kirwan et al. 2016)
  - C_r is suspended sediment concentration in kg/m^3
  - ρ_s is sediment density in kg/m^3

- The syntax for inputting this equation into the model is shown in the following text file.
- Because this edge erosion equation does not specify that erosion values must be positive, marsh edge progradation could be predicted if the progradation term is larger than erosion, which would result in marsh cell generation.

```
scalar_spinup_flag=1
scalar_spinup=25
scalar_RSLR=3.1710e-11
scalar_bfo=5000
scalar_db=2
scalar_mwo=1000
scalar_upland_width=500
scalar_amp=0.625
scalar_P=45000
scalar_Co=0.03
scalar_Be=2.3e-09
scalar_Bmax=2500e-03
scalar_Dmin=0
scalar_wind=5
scalar_dur=15000
scalar_dt=365
function_edge_erosion_eqn=@(W,tau,Be,theta,bgb,e_m,db,amp,rho_m,C_e,v1,v2)
Be*W*cos(theta)-2*0.5e-03/1000*C_e
```

Example 9: input_advanced_mod.txt

- This is an example which combines the more complicated features of all previous examples in order to specify an initial elevation profile, multiple tidal amplitudes, multiple biomass fractions, and a user-specified edge erosion equation.
- An example of the .csv file containing the xz transect appears in Example 4.

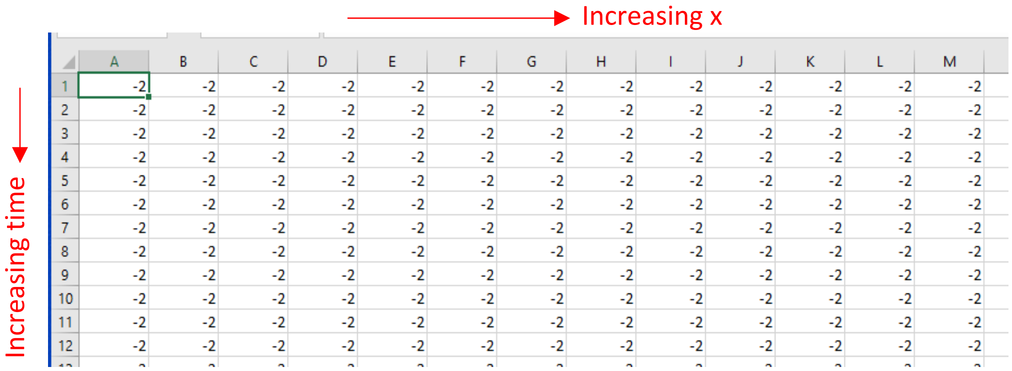
```
scalar_spinup_flag=0
scalar_spinup=5
scalar_RSLR=1.4269e-10
scalar_bfo=1990
scalar_db=1.2
scalar_Co=0.015
scalar_wind=5
scalar_dur=18250
scalar_dt=365
vector_amp=[0.16 0.22 0.28]
vector_tide_dt=[121.6667 121.6667 121.6667]
vector_xz= C:\Users\Username\Documents\Path\To\File\xz_transect.csv
vector_mki=[0.1 0.1]
vector_mui=[0.4 0.4]
function_biofun1=@(z) 4*(-4.4804e+05*z.^5+2.6947e+05*z.^4-1.1732e+03*z.^3-
1.8056e+04*z.^2+824.2899*z+571.4103)/1000
function_biofun2=@(d) (-6*(0.22*100-d*100-25).^2+375.7*(0.22*100-d*100-25)-
3315.8)/1000
vector_Dmax=[0.42 -0.18]
vector_Dmin=[-0.18 -0.48]
vector_Bmax=[2.5 2.5]
scalar_rhos=500
scalar_of=0.3
function_edge_erosion_eqn=@(W,tau,Be,theta,bgb,e_m,db,amp,rho_m,C_e,v1,v2)
max([Be*W*cos(theta),0])
```

DESCRIPTION OF MODEL OUTPUT:

The model will write files containing elevation profiles, fetch, stratigraphy, and biomass content. The contents of each file are described below. Sample output from Example 1 in the previous section ("input_even_simpler_example.txt") is also displayed for each output file.

<file_descriptor>_output_elevation_YYYYMMDDhhmmss.csv

Contains elevation profiles. Each row corresponds to a timestamp in the time vector. The columns correspond to the x-values, with the bay on the left and the marsh on the right. Elevation values are relative to the initial reference and do not update with relative sea level rise. Different rows represent the marsh elevation at different times throughout the model simulation.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
3	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
4	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
5	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
6	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
7	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
8	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
9	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
10	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
11	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
12	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2

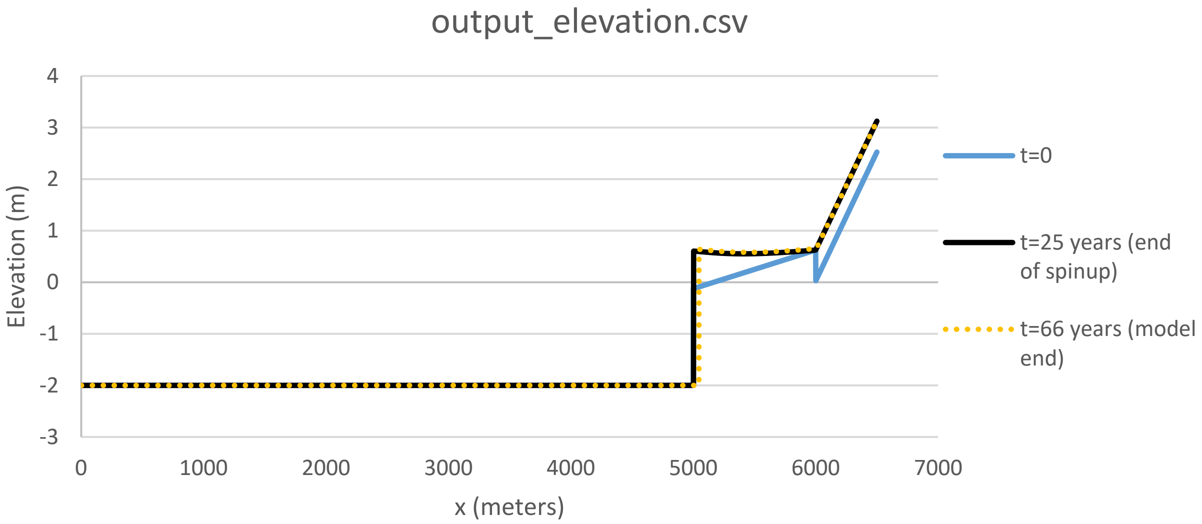


Figure 9: Example of output elevation for the model run with "input_even_simpler_example.txt".

<file_descriptor>_output_fetch_YYYYMMDDhhmmss.csv

Contains a time series of bay fetch, which changes through time due to the marsh edge eroding.

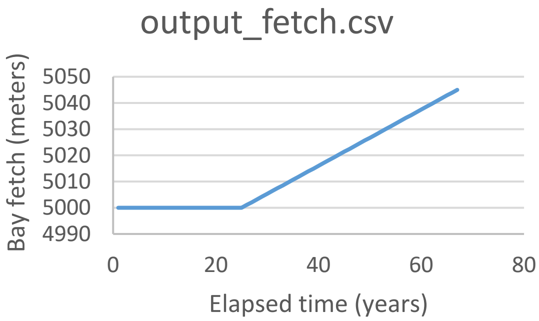


Figure 10: Example of output fetch for the model run with "input_even_simpler_example.txt".

<file_descriptor>_output_msl_YYYYMMDDhhmmss.csv

This file contains a time series of mean sea level relative to the reference elevation for the duration of the simulation.

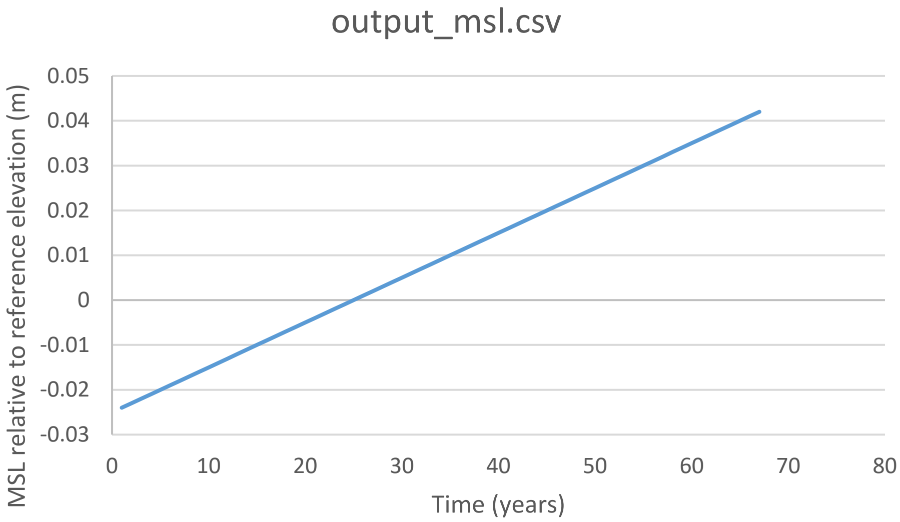


Figure 11: Example of output mean sea level for the model run with "input_even_simpler_example.txt".

<file_descriptor>_output_stratigraphy_YYYYMMDDhhmmss.csv

Each row is a timestamp, and each column is an x location. The values represent the elevation of each packet of mineral sediment (contained in the file labeled *output_mineral_*) and belowground biomass (contained in the file labeled *output_organic_autoch*) at the end of the simulation. Elevation values are relative to the initial reference and do not update with relative sea level rise. The timestamp represents the time-step during which the sediment was deposited and belowground biomass grew. The elevations listed in the stratigraphy file (*_output_stratigraphy*) differ from the elevations in the elevation file (*_output_elevation*) by adjusting elevations from earlier time-steps as belowground biomass decomposes and the marsh edge erodes. Two of the most important uses for output stratigraphy files are 1) specifying the elevations that correspond to the mineral and belowground biomass information, and 2) continuing model runs by feeding the output into an input file.

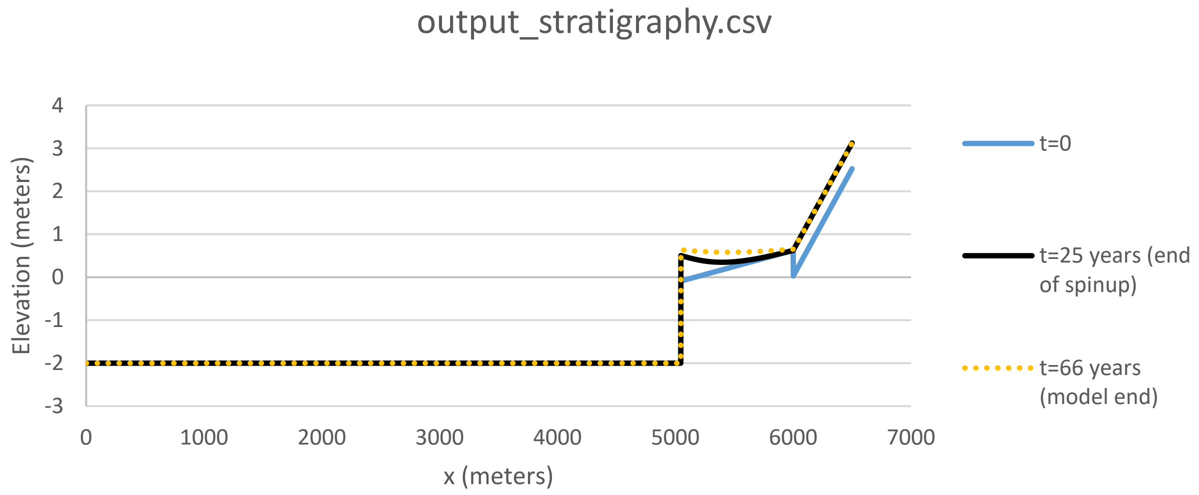


Figure 12: Example of output stratigraphy for the model run with "input_even_simpler_example.txt".

<file_descriptor>_output_mineral_YYYYMMDDhhmmss.csv

Each row of the file corresponds to a timestamp, and each column corresponds to an x-value. The numbers represent kilograms of mineral sediment deposited in each cell, during each timestep.

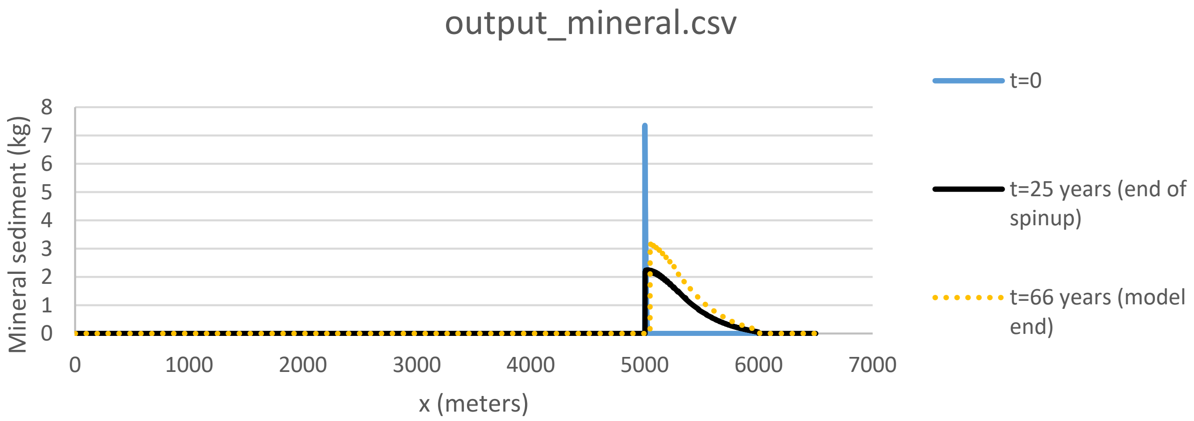


Figure 13: Example of output mineral deposition for the model run with "input_even_simpler_example.txt".

<file_descriptor>_output_organic_autoch_YYYYMMDDhhmmss.csv

Each row is a timestamp, and each column corresponds to an x-value. Values represent kg of belowground biomass in each sediment packet at the end of the simulation. Belowground biomass which grew but also decomposed is not included in the output values. Simulations which consider multiple types of plants will write a '.csv' file of the mass of autochthnous organic material for each individual plant type. As an example, the input file 'input_advanced.txt' specifies growth characteristics for two types of plants. The output files include both '<file_descriptor>_output_organic_autoch1_YYYYMMDDhhmmss.csv' and '<file_descriptor>_output_organic_autoch2_YYYYMMDDhhmmss.csv'.

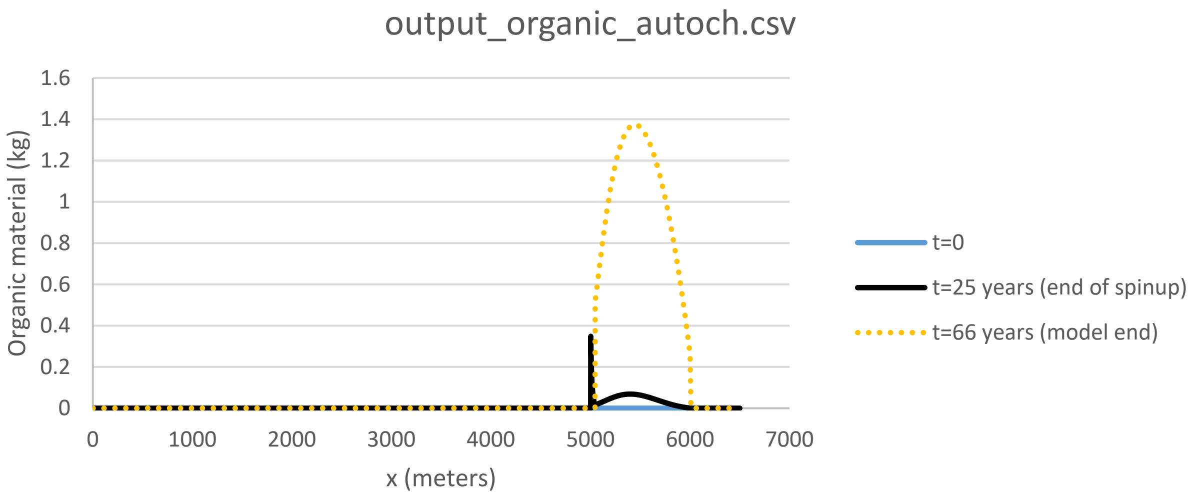


Figure 14: Example of output autochthonous organic material for the model run with "input_even_simpler_example.txt".

WORKS CITED:

Kirwan, M. L., D. C. Walters, W. G. Reay, and J. A. Carr. 2016. "Sea Level Driven Marsh Expansion in a Coupled Model of Marsh Erosion and Migration." *Geophysical Research Letters* 43(9): 4366–4373.

Krafft, D. R., R. Styles, J.Z. Gailani, C.D. Piercy, T.A. Keys, and M.L. Kirwan. 2021. "The Marsh Transect Model: A Simple-One-Dimensional Model for Decade-Scale Coastal Marsh Response to Waves, Sediment Supply, and Sea Level Rise." ERDC CHETN-##-##, 17 pp. (in management review).

Mariotti, G., and J. Carr. 2014. "Dual Role of Salt Marsh Retreat: Long-Term Loss and Short-Term Resilience." *Water Resour. Res.* 50: 2963–2974,

Smith, J. D., and S. R. McLean. 1977. "Spatially Averaged Flow over a Wavy Surface." *J. Geophys. Res.* 82: 1735–1746.