



HAL
open science

Opti-Morph User Guide

Megan Cook, Frédéric Bouchette, Bijan Mohammadi, Nicolas Fraysse

► **To cite this version:**

Megan Cook, Frédéric Bouchette, Bijan Mohammadi, Nicolas Fraysse. Opti-Morph User Guide. 2021.
hal-03348827

HAL Id: hal-03348827

<https://hal.science/hal-03348827v1>

Preprint submitted on 20 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Opti-Morph

User Guide

Megan Cook *^{1,3,4}, Frédéric Bouchette †^{1,3}, Bijan Mohammadi ‡^{2,3}, and Nicolas Fraysse§⁴

¹GEOSCIENCES-M, Univ Montpellier, CNRS, Montpellier, France

²IMAG, Univ Montpellier, CNRS, Montpellier, France

³GLADYS, Univ Montpellier, CNRS, Le Grau du Roi, France

⁴BRL Ingénierie, Nîmes, France

September 20, 2021

Abstract

Opti-Morph is a new morphodynamic model based on optimization theory. This user guide explains the theory behind the model as well as the hydrodynamic models currently associated with Opti-Morph. A presentation of the numerical model is described to allow the user to run Opti-Morph on their own configurations. Two basic examples are also provided.

Keywords: optimization, morphodynamics, coastal dynamics, constraints, hydrodynamics

*Corresponding Author: megan.cook@umontpellier.fr

†frederic.bouchette@umontpellier.fr

‡bijan.mohammadi@umontpellier.fr

§nicolas.fraysse@brl.fr

Contents

I	Introduction	3
I.1	About	3
I.2	Expectations and objectives	3
I.3	Target audience	3
II	Processes and theoretical formulation	4
II.1	Domain and definitions	4
II.2	Hydrodynamic model	5
II.2.1	Introduction	5
II.2.2	Shared processes between hydrodynamic models	6
II.2.3	Presentation of hydrodynamic models	8
II.3	Morphodynamic model by wave energy minimization	23
II.3.1	Introduction	23
II.3.2	Governing seabed dynamics	23
II.3.3	Choice of direction of descent	25
II.3.4	Constraints	26
III	Numerical model	28
III.1	Presentation	28
III.1.1	Workflow	28
III.1.2	Algorithm summary	31
III.1.3	Class organisation	31
III.2	Running Opti-Morph	34
III.2.1	Getting started	34
III.2.2	Main file	35
III.2.3	Input data	37
III.2.4	Output data	39
IV	Applications	42
IV.1	Linear Seabed Beach Configuration	42
IV.1.1	Setting	42
IV.1.2	Input files	42
IV.1.3	Results	43
IV.2	Beach Configuration with submerged breakwaters	44
IV.2.1	Setting	44
IV.2.2	Input files	45
IV.2.3	Results	46
A	List of parameters	48

I Introduction

I.1 About

The numerical hydro-morphodynamic model presented here embodies a new approach to coastal morphodynamics, based on optimization theory. This model is based on the assumption that a sandy seabed evolves over time in order to minimize a certain wave-related function, the choice of which depends on what is considered the driving force behind coastal morphodynamics. This numerical model was given the name **Opti-Morph**, and has the advantages of being fast, robust, and requires very few input parameters.

Optimization theory has been widely used in the study of coastal zones, but is mainly applied in the development of protection structures [Isèbe et al., 2008a, Isèbe et al., 2008b] and/or ports [Cook et al., 2021c, Isèbe et al., 2008b]. Continuing the pioneering work of [Bouharguane et al., 2010, Mohammadi and Bouharguane, 2011] and [Mohammadi and Bouchette, 2014], which put in motion the idea that morphodynamic processes can be described using optimization theory, we have developed a numerical model that simulates the evolution of the seabed while taking into account the complex coupling between morphodynamic and hydrodynamic processes. Based on the assumption that the seabed adapts to minimize a certain hydrodynamic quantity, this optimization problem is also subjected to a certain number of constraints, allowing for a more accurate description of the morphodynamic evolution. First results can be found in [Cook et al., 2021b], where this model was applied to a flume configuration and compared with physical data. A comparative analysis was also conducted between Opti-Morph and another numerical model for the purpose of evaluating the performance of Opti-Morph in comparison with existing hydro-morphodynamic models.

I.2 Expectations and objectives

The main goal of Opti-Morph is to demonstrate the potential of using optimal control in the modeling of coastal dynamics by designing an adaptable, easy-to-use numerical model.

A non-exhaustive list of objectives considered during the initial development of Opti-Morph follows.

- Production of logical/naturalistic simulation results
- Adaptable to open-sea or flume configurations
- Low run times
- High robustness
- User-friendly (including inexperienced users)
- Adaptability of:
 - hydrodynamic model
 - cost function
 - constraints
- Possibility of introducing submerged breakwaters or other soft engineering techniques

I.3 Target audience

The Opti-Morph model is a tool intended for any person wishing to simulate the natural evolution of the coastal seabed in response to the incoming wave conditions, and/or to study the effect of man-made submerged devices on the sediment transport. As such this tool can be used by engineers and project managers when planning the deployment of certain wave-reducing structures and their influence on the surrounding sediment.

This model can also be used in an academic setting to study the driving forces behind coastal morphodynamics. The model has been developed such that the mobility of the seabed is driven by the minimization of a cost function. The

choice of cost function determines which coastal hydro-morphodynamic components drive the seabed mobility. The default cost function is wave energy but other functions can be explored if desired.

II Processes and theoretical formulation

The Opti-Morph model operates by pairing a hydrodynamic and a morphodynamic model. The basis of the morphodynamic model is the minimization of a wave-related cost function; this cost function is provided by the hydrodynamic model, which demonstrates the close relation between the hydrodynamic and morphodynamic processes. In this first section, we will begin by describing the domain and definitions used throughout this document. The hydrodynamic model is then described with its various adaptations, followed by the theoretical description of the morphodynamic model. This includes the governing equations, the choice of cost function, and the introduction of the notion of constraints.

II.1 Domain and definitions

We consider a coordinate system composed of a horizontal axis x and a vertical axis z . We denote $\Omega := [0, x_{\max}]$ the domain of the cross-shore profile, where $x = 0$ refers to an arbitrary fixed point in the deep water and x_{\max} an arbitrary fixed point at the shore, beyond the coast. The elevation of the seabed is a one-dimensional positive function, defined by: $\psi : \Omega \times [0, T] \times \Psi \rightarrow \mathbb{R}^+$ where $[0, T]$ is the interval of time considered for the study, and Ψ is the set of physical parameters describing the shape of the seabed. Let h_0 be the mean water level (m) and h be the water depth (m), defined over the cross-shore profile by $h = h_0 - \psi$. We denote Ω_S the sub-domain of Ω over which the waves shoal and Ω_B the sub-domain of Ω over which the waves break. Let x_B denote the location of the breaking of the waves and let x_S be the shoreline position, that is when the seabed intersects with the mean water level h_0 . Note that the breaking location and the shoreline position may vary over the course of the simulation, and as such x_B and x_S are both time-dependent. The same also goes for Ω_B and Ω_S . Time dependency of x_B , x_S , Ω_B and Ω_S have been omitted for clarity from the remainder of this document. The definitions and notations presented above are summarized in Figure 1.

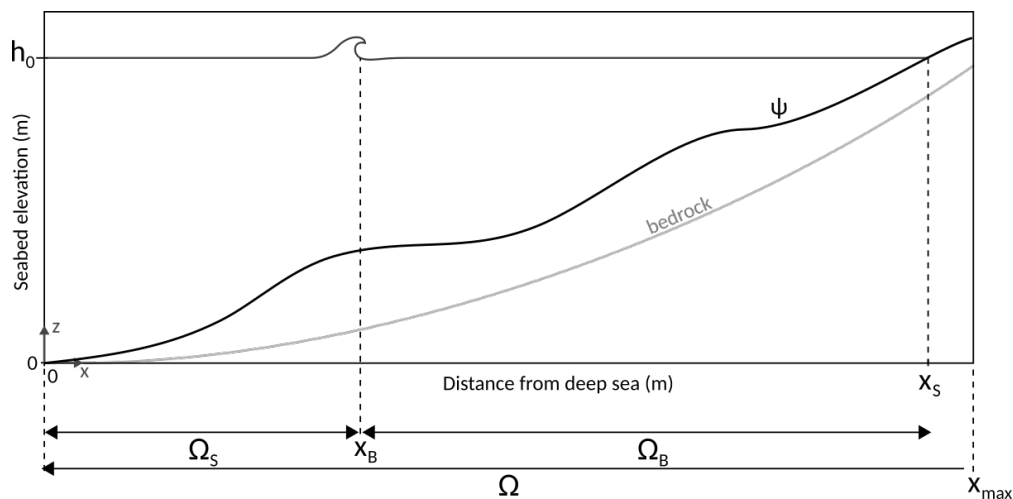


Figure 1: Illustration of the cross-shore profile where breaking occurs once at $x = x_B$

i At the present time, Opti-Morph operates in a one-dimensional setting, depicting either the cross-shore profile of a sandy beach or a flume experiment. However, no assumptions were made regarding the choice of dimension in the theoretical development of this model, and as a result, it is straightforward to extend this theory to a two-dimensional configuration.

II.2 Hydrodynamic model

II.2.1 Introduction

In order to model the evolution over time of the seabed ψ and given the assumption that ψ changes over time in response to the minimization of some hydrodynamic quantity, a hydrodynamic model capable of providing the necessary data is required, here the significant wave height over the cross-shore profile.

There are currently seven different hydrodynamic models associated with the Opti-Morph model. These hydrodynamic models have the advantage of being able to analytically calculate the gradient of wave height with respect to the seabed. Indeed, Opti-Morph relies heavily on this differentiation, and explicit formulas induce quick execution times. Considering that one of the main objectives of Opti-Morph is a rapid output time (cf. Section I.2), it is natural for the associated hydrodynamic models to also have significantly fast execution times. This is achieved by providing wave height analytically with regard to the seabed ψ . Other hydrodynamic models can be integrated with ease, provided that the parameters and methods required by the morphodynamic model are still present. A model using a time-consuming differentiation method can be used but defeats the purpose of Opti-Morph which was designed to be quick and of low-complexity. Another solution would be to perform the calculation of the gradient using automatic differentiation programs [Griewank and Walther, 2008, Hascoët and Pascual, 2004]. This exceeds the initial scope of the Opti-Morph model, but may be implemented at some point in the future if needs be.

The seven hydrodynamic models presented below provide the significant wave height over the cross-shore profile as well as other wave-related data. They range in complexity, from the simplest model in II.2.3.a. to the more complex in II.2.3.g.. In fact, the hydrodynamic models, and the order in which they appear, illustrate the evolution of the model over the course of its development, with the last being the final product. This was carried out in an attempt to address the different issues encountered during the research phase. Each model has its own merits and all have been retained in order to demonstrate the simplicity of replacing one hydrodynamic model with another. This evolution of the hydrodynamic model is showcased in Table 1.

Hydrodynamic model	1	2	3	4	5	6	7
Characteristics	(II.2.3.a.)	(II.2.3.b.)	(II.2.3.c.)	(II.2.3.d.)	(II.2.3.e.)	(II.2.3.f.)	(II.2.3.g.)
Shoaling							
Breaking condition							
Energy conservation							
Multiple wave-breaking							
Handling submerged breakwaters							
Dependency of seaward activity							
Dependency of seaward activity with decreasing influence							
Smooth transition of boundary conditions							
Incorporation of anti-dissipative effect							

Table 1: Defining characteristics of each of the featured hydrodynamic models

In the following sections, hydrodynamic models no. 1 to 7 are presented. All are based on the same principle derived from linear wave theory [Dean and Dalrymple, 1991]. The processes shared by all of the models appear in the first section. Then, individual descriptions of the various hydrodynamic models are provided.

II.2.2 Shared processes between hydrodynamic models

Let H be the significant wave height over the cross-shore profile. We use the partition of $\Omega = \Omega_S \cup \Omega_B$ to define H : waves over the cross-shore profile are either shoaling (over Ω_S) or breaking (over Ω_B).

In order to determine H over the cross-shore profile, the following wave parameters are required:

- wavenumber k (m^{-1}): see Section II.2.2.a.
- phase velocity C ($m.s^{-1}$): see Section II.2.2.b.
- group velocity C_g ($m.s^{-1}$): see Section II.2.2.c.

❗ All models also require the following input data:

- forcing wave height $t \rightarrow H_0(t)$
- forcing wave period $t \rightarrow T_0(t)$ (here assumed constant)
- a wave-breaking index γ

II.2.2.a. Wavenumber

The wavenumber is determined by the linear dispersion equation. Linear dispersion is given by:

$$\sigma^2 = gk \tanh(kh) \quad (1)$$

where $\sigma = \frac{2\pi}{T_0}$ is the wave pulsation (s^{-1}), $g \approx 9.81 \text{ m.s}^{-2}$ is the gravitational acceleration, and k is the wavenumber (m^{-1}). Recalling that $k = \frac{2\pi}{L}$, this equation states that waves with different wavelengths L (m) travel at different speeds C ($m.s^{-1}$). Here, we use it to determine the wavenumber k by using a recursive algorithm such as the Newton-Raphson method.

II.2.2.b. Phase velocity

The phase velocity of a wave C ($m.s^{-1}$) is given by:

$$C(x, t) = C_0(t) \tanh(k(x, t)h(x, t)) \quad \forall (x, t) \in \Omega \times [0, T] \quad (2)$$

where C_0 is the velocity of the forcing waves ($m.s^{-1}$), defined here by $C_0(t) = \frac{g}{2\pi} T_0(t)$ for all $t \in [0, T]$.

II.2.2.c. Group velocity

The group velocity of a wave C_g ($m.s^{-1}$) is given by:

$$C_g = \frac{1}{2} C \left(1 + \frac{2kh}{\sinh 2kh} \right) \quad \forall (x, t) \in \Omega \times [0, T] \quad (3)$$

Let n be the ratio of the wave velocity with respect to the group velocity: $n = \frac{C_g}{C}$.

II.2.2.d. Wave height

For all $x \in \Omega_S$ and $t \in [0, T]$, we define the shoaling wave height H_S as:

$$H_S(x, t) = H_0(t) K_S(x, t) \quad (4)$$

where H_0 is the height of the forcing waves and K_S is the shoaling coefficient (-) defined by Equation (5).

$$K_S(x, t) = \left(\frac{1}{2n(x, t)} \frac{C_0(t)}{C(x, t)} \right)^{1/2} \quad \forall (x, t) \in \Omega_S \times [0, T] \quad (5)$$

II.2.2.e. Breaking wave height

The equations governing breaking wave height (over Ω_B) vary according to the choice of hydrodynamic model. However, all use the breaking condition first established by [Munk, 1949], which states that waves break when their height is too great with respect to the water depth. In other words, waves break when inequality (6) holds, where γ is a wave-breaking index. This parameter is set to 0.55 in the upcoming simulations (cf. Section IV).

$$\frac{H}{h} > \gamma \quad (6)$$

Using this wave-breaking condition, we can define Ω_S and Ω_B as:

$$\Omega_S(t) = \left\{ x \in \Omega, \frac{H(x, t)}{h(x, t)} < \gamma \right\} \quad \text{and} \quad \Omega_B = \left\{ x \in \Omega, \frac{H(x, t)}{h(x, t)} \geq \gamma \right\} \quad (7)$$

❶ The domain over which H is defined as the disjoint union of Ω_S and Ω_B : $[0, x_S] = \Omega_S \cup \Omega_B$ and $\Omega_S \cup \Omega_B = \emptyset$.

II.2.3 Presentation of hydrodynamic models

II.2.3.a. Hydrodynamic model no.1: Shoaling model with decreasing exponential breaking

Technical features

Technical features

- **Code name:** shoaling_1run
- **Use:** Regular seabed
- **Advantages:** Very fast; Known wave-breaking type
- **Inconveniences:** One wave-breaking allowed; wave-breaking type must be known
- **Additional entry parameters:** α : wave-breaking parameter
- **Description:** Shoaling until breaking then decreasing exponentially with the alpha parameters describing the descent.
- **Governing equations:**

$$H(x, t) = \begin{cases} H_0(t)K_S(x, t) & \text{for } x \in [0, x_B] \\ H_S(x_B) \frac{e^{-\alpha(x-x_B)} - e^{-\alpha(x_S-x_B)}}{1 - e^{-\alpha(x_S-x_B)}} & \text{for } x \in [x_B, x_S] \end{cases}$$

for all $(x, t) \in \Omega \times [0, T]$, where α is user-defined.

▲ Time dependency of x_B and x_S have been omitted for clarity.

Detailed description

We assume that waves shoal the length of the cross-shore profile until the breaking condition (6) is first activated. This point is denoted x_B . From then on, waves break up until the shoreline x_S , decreasing in height until $H(x_B) = 0$. Breaking wave height is described as a simple decreasing exponential function from x_B to x_S : for all $x \in \Omega_B = [x_B, x_S]$, wave height is given by:

$$H(x, t) = H_S(x_B) \frac{e^{-\alpha(x-x_B)} - e^{-\alpha(x_S-x_B)}}{1 - e^{-\alpha(x_S-x_B)}} \quad (8)$$

where the parameter α determines the manner in which the waves break (cf. Figure 2).

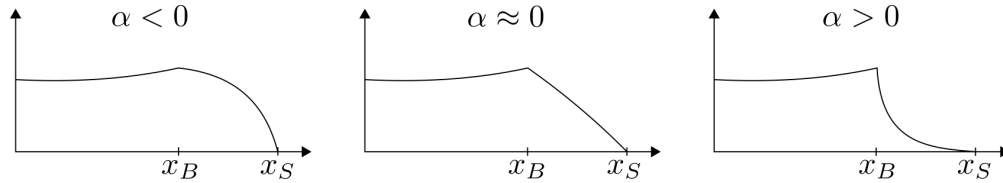


Figure 2: Different values of α alter the behavior of the breaking waves.

❗ This function was designed such that the resulting wave height over the cross-shore profile Ω is a continuous function with zero wave height at the shoreline i.e. $H(x_B) = H_S(x_B)$ and $H(x_S) = 0$.

Combined with the shoaling equation (4) over $\Omega_S = [0, x_B]$, Hydrodynamic model no.1 provides wave height over the cross-shore profile using the following definition:

$$H(x, t) = \begin{cases} H_S(x, t) & \text{for } x \in [0, x_B] \\ H_S(x_B) \frac{e^{-\alpha(x-x_B)} - e^{-\alpha(x_S-x_B)}}{1 - e^{-\alpha(x_S-x_B)}} & \text{for } x \in [x_B, x_S] \end{cases} \quad (9)$$

for all $x \in \Omega$ and $t \in [0, T]$.

Note that breaking may occur only once, and therefore Ω_B and Ω_S are both connected sets.

Illustration

An illustration of the the wave height provided by this model is given in Figure 3 on different types of seabeds.

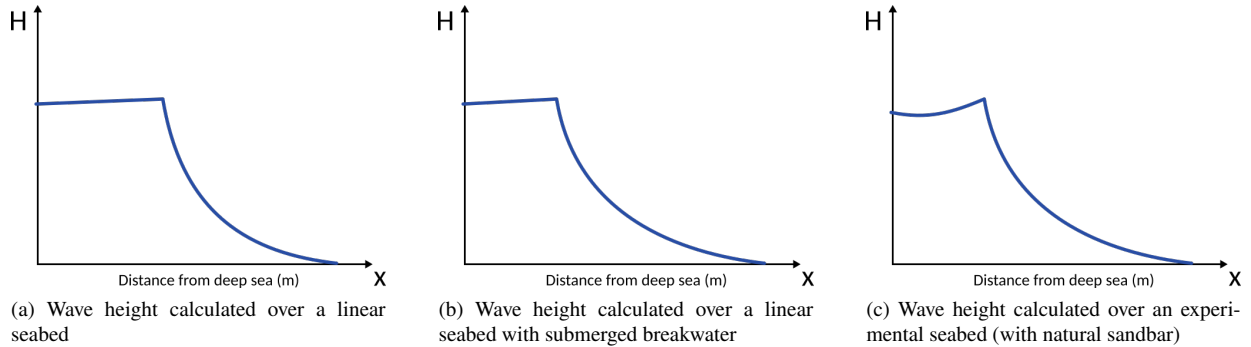


Figure 3: Illustration of Hydrodynamic model no.1 on different seabeds

All three examples of wave height look alike: waves shoal in the deeper waters then decrease exponentially to reach zero at the shore.

❗ This model is recommended on all seabeds, on the condition that the user is content with having waves only breaking once along the cross-shore profile. It is also the faster of the models.

II.2.3.b. Hydrodynamic model no.2: Shoaling model with decreasing exponential breaking and energy conservation

Technical features

Technical features

- **Code name:** shoaling_2run
- **Use:** Regular seabed
- **Advantages:** Very fast; Guarantees conservation of wave energy between sediment displacement (for a same forcing condition)
- **Inconveniences:** One wave-breaking allowed
- **Additional entry parameters:** $\alpha_{t=0}$: initial breaking decrease parameter
- **Description:** Shoaling until breaking then decreasing exponentially with $\alpha(t)$ chosen for energy conservation.
- **Governing equations:**

$$H(x, t) = \begin{cases} H_0(t)K_S(x, t) & \text{for } x \in [0, x_B] \\ H_S(x_B) \frac{e^{-\alpha(t)(x-x_B)} - e^{-\alpha(t)(x_S-x_B)}}{1 - e^{-\alpha(t)(x_S-x_B)}} & \text{for } x \in [x_B, x_S] \end{cases}$$

for all $(x, t) \in \Omega \times [0, T]$, where α is determined to ensure conservation of wave energy between sediment displacement.

▲ Time dependency of x_B and x_S have been omitted for clarity.

Detailed description

This model is based on the principle of energy conservation. Given two wave height functions H_1 and H_2 originating from the same forcing H_0 , we should have **conservation of energy**, irrespective of the shape of the seabed. Therefore, the energy of the system should be the same before and after applying the morphodynamic model.

To achieve this, we implement the following workflow:

- **Step 1:** Apply Hydrodynamic model with user defined parameters
- **Step 2:** Apply Morphodynamic model
- **Step 3:** Apply Hydrodynamic model with parameters selected to ensure energy conservation

We adopt the previous hydrodynamic model from Section II.2.3.a., but allow variations of the value of the parameter α to guarantee energy conservation. This implies that waves break differently depending on the forcing wave energy.

Step 1: In Step 1, the wave height over the cross-shore profile is set as:

$$H(x, t) = \begin{cases} H_S(x) & \text{for } x \in [0, x_B] \\ H_S(x_B) \frac{e^{-\alpha(x-x_B)} - e^{-\alpha(x_S-x_B)}}{1 - e^{-\alpha(x_S-x_B)}} & \text{for } x \in [x_B, x_S] \end{cases} \quad (10)$$

for all $x \in \Omega$ and $t \in [0, T]$. Here, H_S is once again given by the shoaling equation (4) and α is user-defined as a prediction of the type of breaking waves.

Step 2: The morphodynamic model is then applied which provides the new seabed elevation function ψ in response to the forcing conditions at time t .

Step 3: Here, we consider that waves break as per Equation (10), but α is no longer user-defined. We now need to determine the breaking parameter α such that energy is conserved.

Let $H_1, \mathcal{E}_1, x_{S_1}, x_{B_1}, \Omega_{S_1}$ (resp. $H_2, \mathcal{E}_2, x_{S_2}, x_{B_2}, \Omega_{S_2}$ and Ω_{S_2}) be the wave height, wave energy, shoreline and breaking point, shoaling zone and breaking zone before (resp. after) the morphodynamic changes.

Conservation of energy implies:

$$\mathcal{E}_1 = \mathcal{E}_2 \Rightarrow \frac{1}{16} \int_{\Omega} \rho_w g H_1^2 = \frac{1}{16} \int_{\Omega} \rho_w g H_2^2 \quad (11)$$

$$\Rightarrow \underbrace{\int_{\Omega} H_1^2}_{A} = \underbrace{\int_{\Omega_{S_2}} H_2^2 dx}_{B} + H_2(x_{B_2})^2 \int_{\Omega_{B_2}} \left(\frac{e^{-\alpha(x-x_{B_2})} - e^{-\alpha(x_{S_2}-x_{B_2})}}{1 - e^{-\alpha(x_{S_2}-x_{B_2})}} \right)^2 dx \quad (12)$$

$$\Rightarrow \frac{A - B}{H^2(x_{B_2})} = \int_{\Omega_{B_2}} \left(\frac{e^{-\alpha(x-x_{B_2})} - e^{-\alpha(x_{S_2}-x_{B_2})}}{1 - e^{-\alpha(x_{S_2}-x_{B_2})}} \right)^2 dx \quad (13)$$

▲ Wave height over $\Omega \setminus (\Omega_S \cup \Omega_B)$ is zero.

The quantities A and B are easily calculated. Using a Newton-Raphson method, we can determine α such that Equation (13) holds, and therefore energy is conserved between morphodynamic changes.

To conclude, the Hydrodynamic Model no.2 provides wave height over the cross-shore profile using the following definition: For all $(x, t) \in \Omega \times [0, T]$

$$H(x, t) = \begin{cases} H_S(x) & \text{for } x \in [0, x_B] \\ H_S(x_B) \frac{e^{-\alpha(t)(x-x_B)} - e^{-\alpha(t)(x_S-x_B)}}{1 - e^{-\alpha(t)(x_S-x_B)}} & \text{for } x \in [x_B, x_S] \end{cases} \quad (14)$$

where $\alpha(t)$ is the time-dependent breaking parameter ensuring conservation of energy at time $t \in [0, T]$.

Illustration

An illustration of the the wave height provided by this model is given in Figure 4 on different types of seabeds.

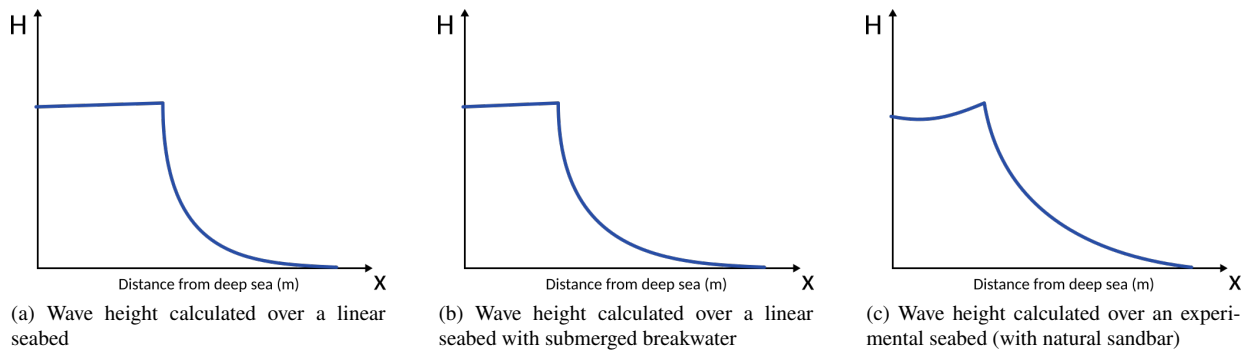


Figure 4: Illustration of Hydrodynamic model no.2 on different seabeds

Similar to before, all three examples of wave height look alike: waves shoal in the deeper waters then decrease exponentially to reach zero at the shore. The only difference is the value of α which differs over time.

❗ This model is recommended on all seabeds, on the condition that the user is content with waves only breaking once across the cross-shore profile.

II.2.3.c. Hydrodynamic model no.3: Global shoaling model with Munk's breaking condition

Technical features

Technical features

- **Code name:** shoaling_throughout
- **Use:** Regular seabed
- **Advantages:** Very fast; Possibility of multiple wave-breakings
- **Inconveniences:** Not suitable for submerged breakwaters
- **Additional entry parameters:** -
- **Description:** Shoaling waves in shoaling zone(s) and waves based on Munk's breaking condition in breaking zones.
- **Governing equations:**

$$H(x, t) = \begin{cases} H_0(t)K_S(x, t) & \text{for } x \in \Omega_S \\ \gamma h(x, t) & \text{for } x \in \Omega_B \end{cases}$$

for all $(x, t) \in \Omega \times [0, T]$.

⚠ Time dependency of Ω_B and Ω_S have been omitted for clarity.

Detailed description

As with the previous two models, the height of the waves over Ω_S is described by the shoaling equation (4). Over Ω_B , we set the height of the wave H as the tipping point condition between breaking and shoaling waves. That is, for all $(x, t) \in \Omega_S \times [0, T]$, breaking waves are defined as:

$$H(x, t) = \gamma h(x, t) \tag{15}$$

❗ The sets Ω_S and Ω_B can now be non-connected, i.e. the shoaling zone (resp. breaking zone) can now be comprised of multiple disjoint sets (cf. Figure 5).

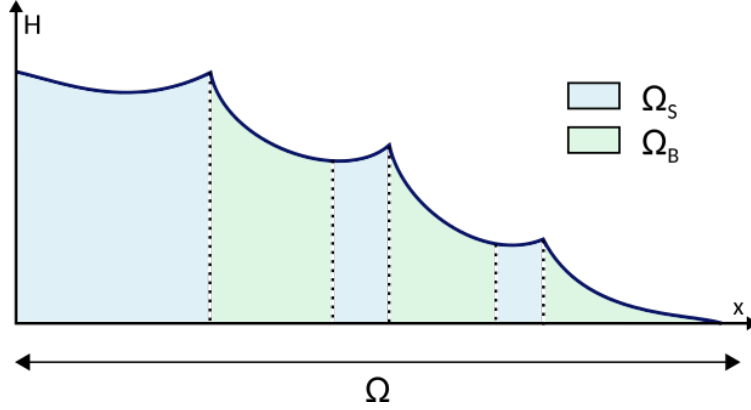


Figure 5: Multiple wave-breakings are now possible, which leads to Ω_S and Ω_B being potentially disconnected

Therefore, Hydrodynamic model no.3 provides wave height over the cross-shore profile using the following definition:

$$H(x, t) = \begin{cases} H_0(t)K_S(x, t) & \text{for } x \in \Omega_S \\ \gamma h(x, t) & \text{for } x \in \Omega_B \end{cases} \quad (16)$$

Illustration

An illustration of the the wave height provided by this model is given in Figure 6 on different types of seabeds.

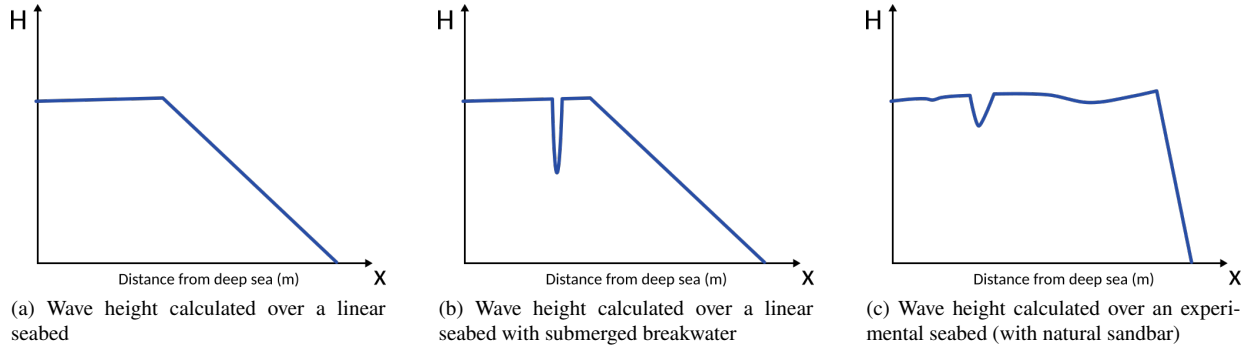


Figure 6: Illustration of Hydrodynamic model no.3 on different seabeds

When waves break, wave height closely follows the profile of the seabed, since over Ω_B , $H = \gamma(h_0 - \psi)$, by definition of h . As such, for a linear seabed, the breaking descent is linear (Fig. 6a). For the configuration with submerged breakwater, the shape of the structure is outlined (Fig. 6b). The structure triggers breaking but wave height quickly resumes it's previous state. This demonstrates that this model is not equipped to manage underwater structures or any irregular seabed. This is also shown in Figure 6c, where the natural sandbar starts the breaking phenomenon, but wave height increases unrealistically once the sandbar has been passed. This model does however have the advantage of allowing multiple breakings. Breaking occurs twice in Figures 6b and 6c, once at the breakwater/sandbar and once further toward the coast.

i This model is recommended on regular seabeds, but is unable to handle irregular seabed such as those with submerged breakwaters or natural sandbars.

II.2.3.d. Hydrodynamic model no.4: Local shoaling model with Munk's breaking condition

Technical features

Technical features

- **Code name:** shoaling_incremental
- **Use:** Regular or irregular seabeds
- **Advantages:** Very fast; Possibility of multiple wave-breakings
- **Inconveniences:** Unstable (due to the iterative nature of the model)
- **Additional entry parameters:** -
- **Description:** Local shoaling waves in shoaling zone(s) and waves based on Munk's breaking condition in breaking zones.
- **Governing equations:**

$$H(x, t) = \begin{cases} H(x - \varepsilon)K_S(x, t) & \text{if } x \in \Omega_S \\ \gamma h(x, t) & \text{if } x \in \Omega_B \end{cases}$$

for all $(x, t) \in \Omega \times [0, T]$.

▲ Time dependency of Ω_B and Ω_S have been omitted for clarity.

Detailed description

Instead of considering that a wave is spatially dependent on only the initial wave height, this model considers that wave height depends on the seaward activity of the waves. In a configuration with a local sandbar or wave-breaking structure, waves determined by the previous model (Section II.2.3.c.) shoal up until the structure and then break when the structure is detected. However, once the waves move beyond the wave-breaking device, they resume a wave height similar to that before the structure, i.e. disregarding the encounter of the wave-breaking structure. In other words, the model doesn't register the loss of energy that took place at the submerged wave-breaker. This is due to the fact that the only spacial component influencing the wave height across the cross-shore profile is at $x = 0$ (by way of the term $H_0(t)$). This model amends this.

Instead of using wave height at the entry of the domain at each point x of the cross-shore profile Ω , we now use the previous seaward point of the domain discretization, located at $x - \varepsilon$. Therefore, shoaling waves, which were previously described by (4) is now defined by:

$$H(x, t) = H(x - \varepsilon)K_S(x, t) \tag{17}$$

for all $x \in \Omega_S$ and $t \in [0, T]$, where $x - \varepsilon$ is the previous point of the discretization.

With the same breaking process as before over Ω_B , i.e. Equation (15), wave height is now described by the following equation:

$$H(x, t) = \begin{cases} H(x - \varepsilon)K_S(x, t) & \text{for } x \in \Omega_S \\ \gamma h(x, t) & \text{for } x \in \Omega_B \end{cases} \tag{18}$$

for all $(x, t) \in \Omega \times [0, T]$.

▲ Taking the previous point of the discretization guarantees that submerged breakwaters are taken into account. However, the iterative nature of the model leads to unstable results. Increasing ε would render the model more stable, but results in a poor management of the submerged breakwaters once again.

Illustration

An illustration of the the wave height provided by this model is given in Figure 7 on different types of seabeds.

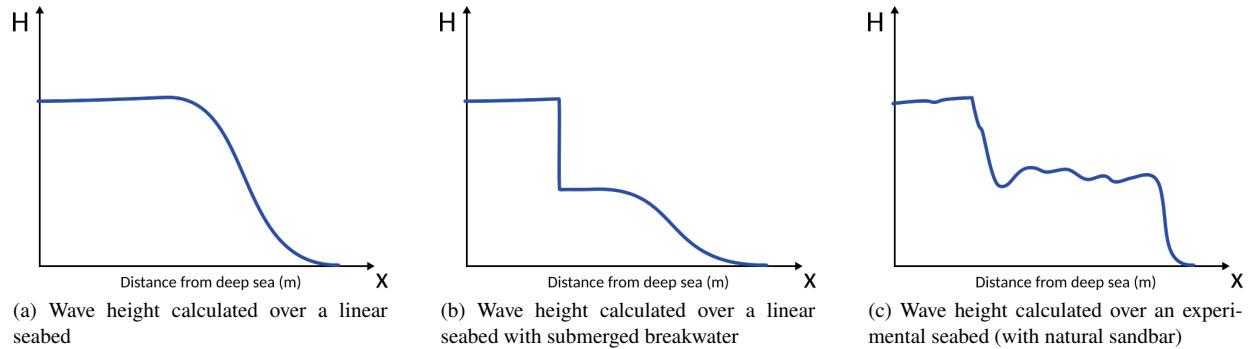


Figure 7: Illustration of Hydrodynamic model no.4 on different seabeds

Figure 7 shows that this model can handle the introduction of geotubes. The structure causes the waves to break prematurely. Wave height drops and shoaling resumes. This is also true for natural sandbars. Once the waves break, waves can once again shoal, allowing for multiple breakings if necessary.

❗ This model is not recommended for Opti-Morph because of the unstable effect it has on the seabed.

II.2.3.e. Hydrodynamic model no.5: Weighted window local shoaling model

Technical features

Technical features

- **Code name:** shoaling_window
- **Use:** Regular or irregular seabeds
- **Advantages:** Fast; Possibility of multiple wave-breakings; Takes into account the effect of the seawards waves; Handles submerged breakwaters
- **Inconveniences:** Poor management of deep sea conditions
- **Additional entry parameters:** d_w : maximal distance of local spatial dependency of a wave
- **Description:** Seaward waves influence wave height with decreasing effect
- **Governing equations:**

$$H(x, t) = \begin{cases} H_0^w(x, t)K_S(x, t) & \text{for } x \in \Omega_S \\ \gamma h(x, t) & \text{for } x \in \Omega_B \end{cases}$$

for all $(x, t) \in \Omega \times [0, T]$, where H_0^w is the weighted average of the seaward waves.

▲ Time dependency of Ω_B and Ω_S have been omitted for clarity.

Detailed description

Instead of considering that waves depend solely on offshore wave height H_0 as in II.2.3.c., or a nearby seaward point as in II.2.3.d., this model suggests that shoaling waves are decreasingly influenced by seawards waves. The greater the distance, the less effect it has of the present wave height. Let $d_w > 0$ be the maximal distance of local spatial dependency of a wave. Wave height at $x \in \Omega_S$ depends on the behavior of the wave height over the interval $[x - d_w, x)$ with a strong influence at the upper-bound and little to no influence at the lower-bound. As such, we introduce a weighting function w , defined over $[0, d_w]$ and quantifies the influence of the seawards waves on the current wave. It is defined such that $w(0) = 1$, $w(d_w) = 0$ and decreases exponentially:

$$\begin{aligned} w : [0, d_w] &\longrightarrow \mathbb{R}^+ \\ x &\longmapsto \exp\left(\ln(0.01) \left(\frac{x}{d_{win}}\right)^2\right) \end{aligned} \quad (19)$$

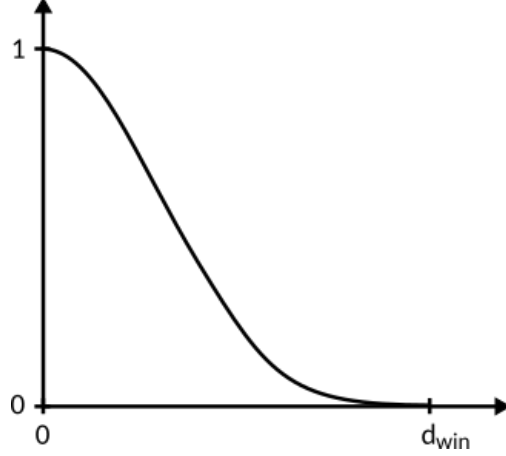


Figure 8: Weighting function w , equal to 1 closest to the present wave and decreases exponentially as the distance seaward increases

An illustration of the weighting function w is given in Figure 8. However, if breaking occurs, the history of the wave prior to breaking should not be relevant. This is to ensure that once the energy of the waves is lost due to breaking, it cannot be regained. The term H_0 in equation (4) (or the term $H(\cdot - \varepsilon)$ in (17)) is now replaced by a weighted average of the seaward wave height, denoted H_0^w and defined as:

$$H_0^w(x, t) = \frac{1}{\int_{x-X}^x w(x-y)dy} \int_{x-X}^x w(x-y)H(y)K(y)dy \quad (20)$$

where $X = \min(x, d_w, \text{dist}(x, x_B))$. Introducing X ensures wave history is taking into account over the appropriate zone: $X = x$ indicates that the zone of local spatial dependency is cut off by the lower-bound of the domain, $X = d_w$ depicts wave dependency over the maximal distance, and $X = \text{dist}(x, x_B)$ indicates that the zone of local spatial dependency is interrupted by waves breaking.

Shoaling waves are therefore described by:

$$H(x, t) = H_0^w(x, t)K_S(x, t) \quad (21)$$

for all $x \in \Omega_S$ and $t \in [0, T]$.

As such, wave height over the cross-shore profile Ω is given by:

$$H(x, t) = \begin{cases} H_0^w(x, t)K_S(x, t) & \text{for } x \in \Omega_S \\ \gamma h(x, t) & \text{for } x \in \Omega_B \end{cases} \quad (22)$$

Illustration

An illustration of the the wave height provided by this model is given in Figure 9 on different types of seabeds.

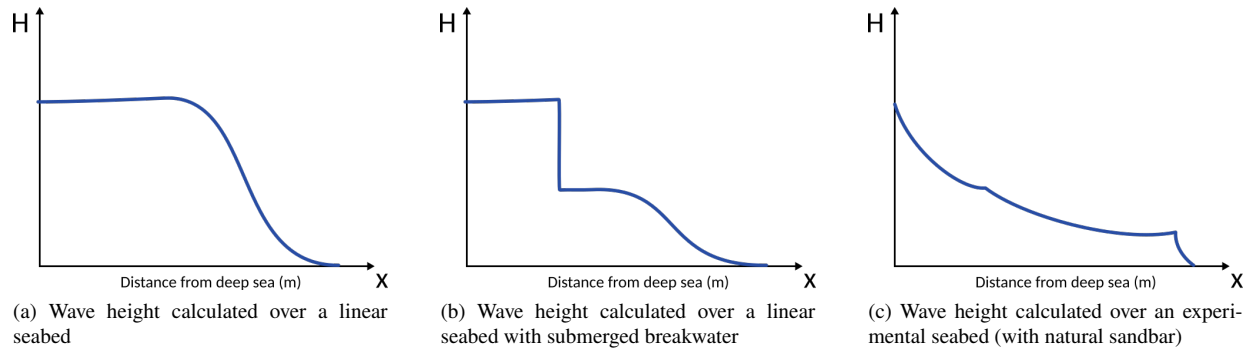


Figure 9: Illustration of Hydrodynamic model no.5 on different seabeds

Like before, this model can handle multiple breakings and the introduction of submerged breakwaters. In addition, it produces stable results. However, in some configurations, the poorly managed boundary conditions may cause a drop in wave height at deep-sea border (e.g. Fig. 9c). This may happen when $x = 0$ doesn't correspond to deep-sea conditions, a condition which is not always possible when applying the model to a flume configurations.

i This model is recommended on regular seabeds, with and without submerged breakwaters. However, the user must proceed with caution when applying to an experimental setting.

II.2.3.f. Hydrodynamic model no.6: Weighted window local shoaling model and improved boundary conditions

Technical features

Technical features

- **Code name:** shoaling_window_LC
- **Use:** Regular or irregular seabeds
- **Advantages:** Fast; Possibility of multiple wave-breakings; Takes into account the effect of the seawards waves; Handles submerged breakwaters; Proper handling of deep sea conditions
- **Inconveniences:** May lead to overly dissipative waves
- **Additional entry parameters:** d_w : maximal distance of local spatial dependency of a wave
- **Description:** Previous hydrodynamic model with linear combination of 6 equations for smoother boundary conditions
- **Governing equations:**

$$H(x, t) = \begin{cases} [(1 - \alpha_w(x))H_0(t) + \alpha_w(x)H_0^w(x, t)] K_S(x, t) & \text{if } x \in \Omega_S \text{ and } x < d_w \\ H_0^w(x, t)K_S(x, t) & \text{if } x \in \Omega_S \text{ and } x \geq d_w \\ \gamma h(x, t) & \text{if } x \in \Omega_B \end{cases}$$

for all $(x, t) \in \Omega \times [0, T]$, where H_0^w is the weighted average of the seaward waves and $\alpha_w(x) = \frac{x}{d_w}$ over $[0, d_w]$.

▲ Time dependency of Ω_B and Ω_S have been omitted for clarity.

Detailed description

The boundary conditions of the previous model lead to unrealistic wave height over the lowermost part of the domain Ω . The weighting function w is provided with a very small number of points at the beginning of Ω , which may lead to an unreasonable drop in wave height in deeper waters. To allow a smooth transition between offshore and nearshore-dependent waves, we adopt a linear combination of the initial shoaling equation (4) and the weighted shoaling equation (21) over the lowermost part of Ω .

Let α_w be the linear combination parameter combining both shoaling physics, defined by:

$$\begin{aligned} \alpha_w : [0, d_w] &\longrightarrow [0, 1] \\ x &\longmapsto \alpha_w(x) = \frac{x}{d_w} \end{aligned} \quad (23)$$

Over the deepest portion of Ω_S , more specifically, the interval $[0, d_w]$, shoaling wave height is defined such that the initial shoaling equation (4) prevails for x close to 0 and the weighted shoaling equation (21) dominates for x close to d_w . A smooth transition between models is achieved via the parameter α_w . The weighted shoaling equation (21) is also adopted over the remaining portion of Ω_S , beyond $x = d_w$.

Wave height over the shoaling zone Ω_S is therefore defined for all $x \in \Omega_S$ and $t \in [0, T]$ by:

$$H(x, t) = \begin{cases} [(1 - \alpha_w(x))H_0(t) + \alpha_w(x)H_0^w(x, t)] K_S(x, t) & \text{if } x < d_w \\ H_0^w(x, t)K_S(x, t) & \text{if } x \geq d_w \end{cases} \quad (24)$$

Adopting the same wave height equation over Ω_B as before, with Equation (15), wave height over the cross-shore domain is defined for all $(x, t) \in \Omega \times [0, T]$ by:

$$H(x, t) = \begin{cases} [(1 - \alpha_w(x))H_0(t) + \alpha_w(x)H_0^w(x, t)] K_S(x, t) & \text{for } x \in \Omega_S \text{ and } x < d_w \\ H_0^w(x, t)K_S(x, t) & \text{for } x \in \Omega_S \text{ and } x \geq d_w \\ \gamma h(x, t) & \text{for } x \in \Omega_B \end{cases} \quad (25)$$

Illustration

An illustration of the the wave height provided by this model is given in Figure 10 on different types of seabeds.

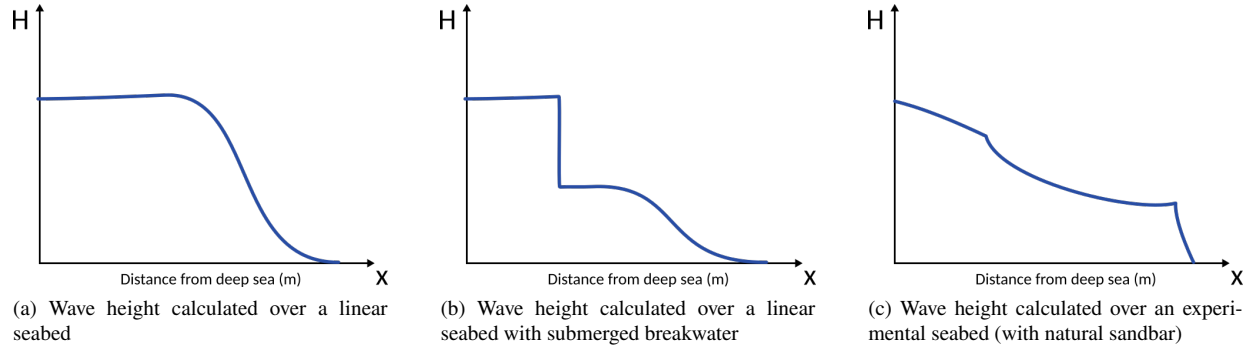


Figure 10: Illustration of Hydrodynamic model no.6 on different seabeds

Now that the boundary condition has been properly dealt with, we no longer observe the unusual drop at the beginning of the domain. Acceptable wave height is produced in all three settings.

❗ This model is recommended on regular seabeds, with and without submerged breakwaters, as well as in experimental settings.

II.2.3.g. Hydrodynamic model no.7: Weighted window shoaling model with anti-dissipative effect

Technical features

Technical features

- **Code name:** shoaling_window_LC_ADT
- **Use:** Regular or irregular seabeds
- **Advantages:** Fast; Possibility of multiple wave-breakings; Takes into account the effect of the seawards waves; Handles submerged breakwaters; Proper handling of deep sea conditions; Control of the dissipative effect of the waves;
- **Inconveniences:** TBA
- **Additional entry parameters:** d_w : maximal distance of local spatial dependency of a wave; a_{AD} and b_{AD} : anti-dissipative parameters
- **Description:** Previous hydrodynamic model with the possibility of including an anti-dissipative term.
- **Governing equations:**

$$H(x, t) = \chi_{AD}(x) \begin{cases} [(1 - \alpha_w(x))H_0(t) + \alpha_w(x)H_0^w(x, t)] K_S(x, t) & \text{for } x \in \Omega_S \text{ and } x < d_w \\ H_0^w(x, t) K_S(x, t) & \text{for } x \in \Omega_S \text{ and } x \geq d_w \\ \gamma h(x, t) & \text{for } x \in \Omega_B \end{cases}$$

for all $(x, t) \in \Omega \times [0, T]$, where H_0^w is the weighted average of the seaward waves, $\alpha_w(x) = \frac{x}{d_w}$ over $[0, d_w]$ and χ_{AD} is an anti-dissipative term.

▲ Time dependency of Ω_B and Ω_S have been omitted for clarity.

Detailed description

Depending on the required wave behavior, it may be necessary to limit the dissipation of the waves. Indeed, one may discover that the hydrodynamic model of section II.2.3.f. dissipates too much energy over the cross-shore profile. This is especially relevant when comparing the numerical wave height with experimental data. For the purposes of allowing the user to calibrate the dissipation of the shoaling waves, we introduce the following anti-dissipative term χ_{AD} :

$$\chi_{AD}(x) = \begin{cases} \left(1 + a_{AD} \frac{x}{x_{\max}}\right)^{b_{AD}} - \left(1 + a_{AD} \frac{x_{\Omega_S^-}(x)}{x_{\max}}\right)^{b_{AD}} + 1 & \text{for } x \in \Omega_S \\ 1 & \text{for } x \in \Omega_B \end{cases} \quad (26)$$

where $x_{\Omega_S^-}(x)$ is the lower-bound of the connected subset of Ω_S where x is found. The parameters a_{AD} and b_{AD} allow the user to define the manner in which the waves dissipate; a_{AD} determines the slope of χ_{AD} and b_{AD} its quadratic behavior. An example of the anti-dissipative term is given in Figure 11, with $b_{AD} > 1$.

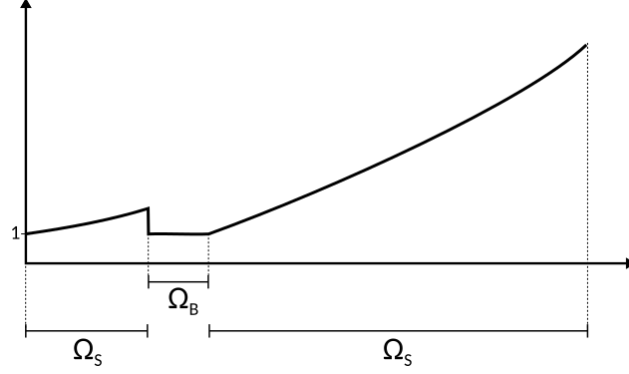


Figure 11: Example of the anti-dissipative term

❶ Setting $(a_{AD}, b_{AD}) = (0, 1)$ disables the anti-dissipative effect.

Wave height over the cross-shore profile Ω is defined by:

$$H(x, t) = \chi_{AD}(x) \begin{cases} [(1 - \alpha_w(x))H_0(t) + \alpha_w(x)H_0^w(x, t)] K_S(x, t) & \text{for } x \in \Omega_S \text{ and } x < d_w \\ H_0^w(x, t)K_S(x, t) & \text{for } x \in \Omega_S \text{ and } x \geq d_w \\ \gamma h(x, t) & \text{for } x \in \Omega_B \end{cases} \quad (27)$$

for all $(x, t) \in \Omega \times [0, T]$.

This is the model used in the subsequent applications of Section IV.

Illustration

An illustration of the the wave height provided by this model is given in Figure 12 on different types of seabeds.

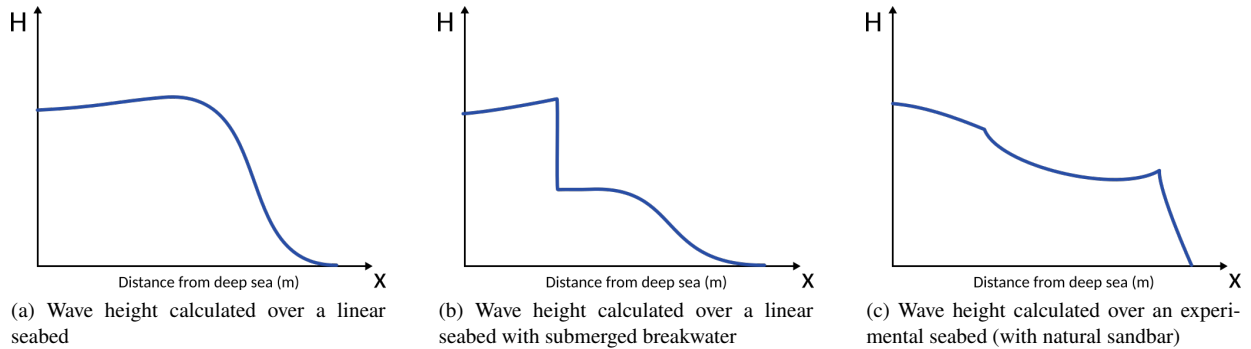


Figure 12: Illustration of Hydrodynamic model no.7 on different seabeds

With the proper choice of a_{AD} and b_{AD} , we can now calibrate the hydrodynamic model to fit the required profile. This is especially useful in the case of an experimental flume setting where wave height has been collected as part of the experiment.

❶ This model is recommended on regular seabeds, with and without submerged breakwaters, as well as in experimental settings.

II.3 Morphodynamic model by wave energy minimization

II.3.1 Introduction

This section is devoted to the presentation of the Opti-Morph model. This one-of-a-kind morphodynamic model is based on optimization theory. The fundamental assumption governing Opti-Morph states that the seabed evolves over time so as to minimize a certain quantity, named cost function. The choice of cost function depends on what is considered the driving force behind the morphodynamic response to the seabed. Several cost functions have been considered, but all revolve around wave energy. In other words, the shape of the seabed varies in an effort to minimize the energy of the surface waves at that given time. At each time, the model indicates the direction to a local minimum of the cost function with regard to the parameterization of the seabed. Two physical parameters limit or encourage seabed mobility depending on the properties of the sediment and the depth of the water. Furthermore, constraints are added to this optimization problem as a means to incorporate additional physics to the model. Constraints are regarded as secondary processes in regards to the minimization of the cost function, which is deemed the primary force behind the morphodynamic response to the seabed. Three constraints have been included in this model. The first concerns the maximal slope of the seabed, the second manages the sandstock of the profile in the case of an experimental flume, and the third concerns the presence of bedrock.

The optimization problem that Opti-Morph seeks to solve is:

For each $t \in [0, T]$, find the shape of the seabed $\psi(t) \in \Psi$ such that the cost function $\mathcal{J}(t)$ is minimal, while subjected to constraints.

This morphodynamic model is associated with one of the hydrodynamic models from section II.2, which provides the morphodynamic model with the necessary hydrodynamic quantities.

II.3.2 Governing seabed dynamics

In order to describe the evolution of the seabed, we assume that at each time $t \in [0, T]$, the seabed ψ , in its effort to minimize a certain energy related quantity \mathcal{J} , verifies the following dynamics over Ω :

$$\begin{cases} \psi_t(\cdot, t) = \Upsilon \Lambda d(\cdot, t) \\ \psi(\cdot, 0) = \psi_0(\cdot) \end{cases} \quad (28)$$

where ψ_t is the evolution of the seabed over time, Υ is the mobility of the sand ($m.s.kg^{-1}$), Λ is the excitation of the seabed by the water waves ($-$), and d is the direction of descent (J).

II.3.2.a. Parameter Υ

The first parameter Υ takes into account the physical characteristics of the sand and represents the mobility of the sediment. For Υ great, as is the case with finer particles, the seabed may be submitted to significant change. For Υ close to zero, little mobility is observed, as is the case of a seabed composed of larger rocks. This parameter, expressed in $m.s.kg^{-1}$, may vary over the cross-shore profile. Further interpretation of the nature of this parameter will be provided in future works.

II.3.2.b. Parameter Λ

The second parameter Λ represents the influence of the water depth on the seabed and is defined using the orbital velocity damping function φ (cf. [Soulsby, 1987]):

$$\begin{aligned} \varphi : \Omega \times [0, h_0] &\longrightarrow \mathbb{R}^+ \\ (x, z) &\longmapsto \frac{\cosh(k(x)(h(x) - (h_0 - z)))}{\cosh(k(x)h(x))} \end{aligned} \quad (29)$$

An illustration of the orbital velocity of the wave particles is given in Figure 13. This function describes the excitation of the water particles for a given location along the cross-shore profile and a given water depth. However, our interest lies in the excitation of the seabed by the surface waves. Therefore, it is natural to consider the orbital damping function at $z = \psi(x)$. The parameter Λ of Equation (28) is therefore defined by:

$$\Lambda(x) = \varphi(x, \psi(x)) = \frac{1}{\cosh(k(x)h(x))} \quad (30)$$

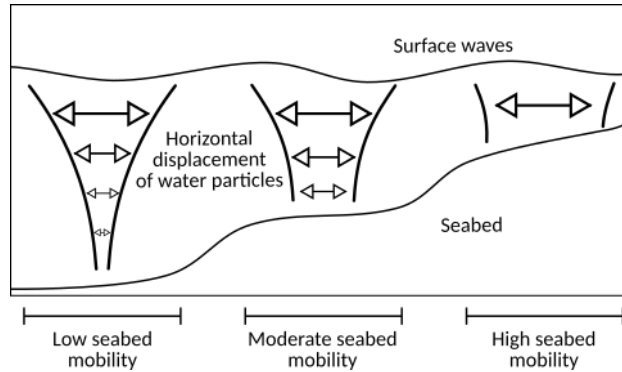


Figure 13: Illustration of the orbital velocity over the cross-shore profile

This parameter governs the manner in which the waves affect the seabed. In deeper waves, the surface waves have little to no effect on the seabed below. No movement should be observed of the seabed, and thus $\Lambda \approx 0$ over this portion of the cross-shore profile. When the waves have a large impact on the seabed, e.g. at the coast, greater movement can be observed and as such we set $\Lambda \approx 1$. An illustration of Λ is given in Figure 14.

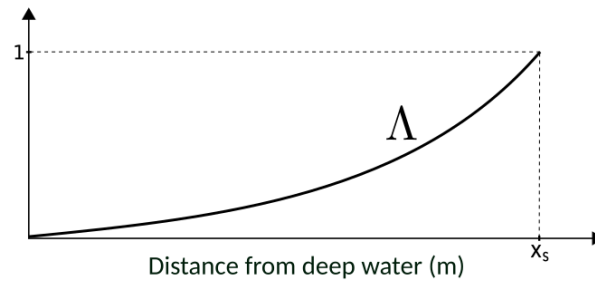


Figure 14: Variation of the parameter Λ over the cross-shore profile

II.3.2.c. Direction of descent d :

The vector d is the direction of descent. In unconstrained circumstances, we set $d = -\nabla_{\psi} \mathcal{J}$, i.e. the direction indicating the minimum of the cost function \mathcal{J} with regards to the seabed ψ . However, adding constraints changes the value of d , but increases the efficiency of the model, by incorporating more physics into the model. This results in a less optimal direction of descent but one that is capable of respecting the criteria required by the constraints, and as such produces more realistic morphodynamic results.

The following section explores the different cost functions / directions of descent implemented in Opti-Morph.

Keyword	Definition	Commentary
CF0	$d = -\nabla_{\psi} \mathcal{E}_S \chi_{\Omega_S}$	Recommended
CF1	$d = -\frac{x_B}{x_S} \nabla_{\psi} \mathcal{E}_S \chi_{\Omega_S}$	
CF2	$d = -\frac{x_B^2}{x_S} \nabla_{\psi} \mathcal{E}_S \chi_{\Omega_S}$	
CF3	$d = -x_B \nabla_{\psi} \mathcal{E}_S \chi_{\Omega_S}$	
CF4	$d = -\frac{x_B}{x_S} \int_{\Omega_S} \nabla_{\psi} \mathcal{E}_S \chi_{\Omega_S}$	
CF5	$d = (1 - \Lambda)CF2 + \Lambda CF4$	where Λ is the excitation of the seabed
CF6	$d = (1 - \Lambda)CF3 + \Lambda CF4$	where Λ is the excitation of the seabed

Table 2: Table of the different directions of descent implemented in Opti-Morph

II.3.3 Choice of direction of descent

The term "cost function" is used for the quantity to be minimized and is noted \mathcal{J} . The term "direction of descent" is used for direction indicating the manner in which the seabed varies, and is denoted d . In the more simpler cases, $d = -\nabla_{\psi} \mathcal{J}$, but exploring other directions of descents leads to more complex formulations. It is not always possible to express the cost function \mathcal{J} when exploring directions.

For the purpose of illustrating the simplicity of implementing a new cost function, seven different directions of descent have been considered. Modifying d modifies the physics behind the morphodynamic response of the seabed. These choices are shown in Table 2 and are all based on the energy of shoaling waves given by the following equation:

$$\mathcal{E}_S = \frac{1}{16} \int_{\Omega_S} \rho_w g H^2(\psi, x, t) dx \quad (31)$$

where Ω_S is the shoaling zone, ρ_w is the density of the water ($kg.m^{-3}$), g is gravitational acceleration ($m.s^{-2}$), H is the height of the wave (m) and ψ is the elevation of seabed (m). We denote χ_{Ω_S} the characteristic function of the subset Ω_S of Ω .

The introduction of directions CF1, CF2, and CF3 occurred while exploring the different possible dimensions of the cost function \mathcal{J} . Ultimately, it was decided to maintain a cost function expressed in $J.m^{-1}$, so CF0 was retained. Directions CF4, CF5, and CF6 were proposed in an attempt to combine two different physics depending on the location along the cross-shore profile. A simple well-chosen factor may be needed to ensure that d has a consistent dimension. All of the considered directions have been kept in order to demonstrate how easy it is to introduce a new cost function to the model but should be adopted with caution.

The first and simplest choice is CF0 and is the one used in the subsequent applications of Section IV. The directions CF5 and CF6 are more complex and combines two physics to simulate the seabed evolution. More details can be found for these choices in sections II.3.3.a. and II.3.3.b..

II.3.3.a. Cost function based on \mathcal{E}_S

The evolution of the seabed is assumed to be driven by the minimization of a cost function \mathcal{J} , here described as the potential energy of shoaling waves:

$$\mathcal{J}(\psi, t) = \frac{1}{16} \int_{\Omega_s} \rho_w g H^2(\psi, x, t) dx \quad (32)$$

Differentiating \mathcal{J} with respect to ψ yields the direction of descent CF0. This direction was used in [Cook et al., 2021b] and is currently the recommended choice of direction.

II.3.3.b. Cost function combining two physics

In this section, we suppose that two different physics govern the evolution of seabed based whether the waves act globally or locally on the seabed. Both physics are based on the potential energy of the waves over the cross-shore profile, given by Equation (31). However, unlike in Section II.3.3.a., an explicit formulation of \mathcal{J} is not possible.

As mentioned in Section II.3.2 with the introduction of the Λ parameter, the seabed evolves differently in deep waters and at the coast. Therefore, we define two different physics governing the seabed evolution depending on location across the cross-shore profile. We denote A_1 the gradient of the potential surface energy of the waves and A_2 the gradient of the mean value of A_1 .

$$A_1(x, t) = -\nabla_{\psi} \mathcal{E}(x, t) \quad \forall (x, t) \in \Omega \times [0, T] \quad (33)$$

$$A_2(x, t) = -\frac{1}{x_B(t)} \int_{\Omega_s} \nabla_{\psi} \mathcal{E}(x, t) dx \quad \forall (x, t) \in \Omega \times [0, T] \quad (34)$$

The mean value A_2 was chosen to represent the physics governing the seabed over the deeper waters since the seabed is affected in a global manner in this zone. The gradient of the potential surface energy A_1 will be used at the coast, where the waves have a local effect on the seabed. In order to differentiate the different zones of the cross-shore profile, the parameter Λ is used (cf. Equation (30)). We set the direction of descent as:

$$d = \Lambda A_1 + (1 - \Lambda) A_2 \quad (35)$$

The term A_1 is dominant near the coast (for a local effect) and the term A_2 is dominant in deep waters (for a global effect). The parameter Λ is used to weight these two physics.

❶ Up to the multiplication of a constant, this approach is used for the directions CF5 and CF6.

II.3.4 Constraints

As mentioned already, the driving force behind the morphodynamic response to the seabed is assumed to be the minimization of the energy of the shoaling waves. Any physical phenomenon deemed secondary to this mechanism is represented in the form of a constraint. Constraints are added to incorporate more physics into the model, and as such provide more realistic results. At the current stage of development, three constraints have been implemented, though more can be introduced if necessary. This includes a maximal slope constraint to prevent unrealistically steep seabeds, a sandstock constraint for flume configurations, and a bedrock constraint.

II.3.4.a. Maximal sand slope constraint

The slope of the seabed is bounded by a grain-dependent threshold M_{slope} ([Dean and Dalrymple, 1991]). If a slope becomes too steep and exceeds this threshold, avalanching occurs. The maximal sand slope constraint prevents the slope from exceeding this upper limitation and is conveyed by the following inequality:

$$\left| \frac{\partial \psi}{\partial x} \right| \leq M_{\text{slope}} \quad (36)$$

❶ The parameter M_{slope} may vary over the cross-shore profile.

II.3.4.b. Sandstock constraint

In the case of an experimental flume, the sediment composing the seabed cannot leave the confines of the tank over the course of the simulation. Also, sediment cannot be added during this period. Therefore, a sandstock constraint is introduced which asserts that the quantity of sand in a flume must be constant over time, contrarily to an open-sea simulation where the sediment can move freely between the limits of the domain. The sandstock constraint is therefore expressed by the following equation:

$$\int_{\Omega} \psi(x, t) dx = \int_{\Omega} \psi(t = 0, x) dx \quad \forall t \in [0, T] \quad (37)$$

❶ This constraint is essential for validating the numerical model with experimental data.

❶ The sandstock constraint is also applicable to an open-sea configuration when little to no transfer of sediment is observed between the deep sea and the nearshore area.

For a given time $t \in [0, T]$, we define $C_{\text{sand}}(t)$ as the difference between the current and initial sandstock:

$$C_{\text{sand}}(t) = \int_{\Omega} (\psi(x, t) - \psi(t = 0, x))^2 dx \quad (38)$$

❶ The exponent 2 ensures that $C_{\text{sand}} \geq 0$, while keeping C_{sand} differentiable.

The optimization problem becomes:

For each $t \in [0, T]$, find the shape of the seabed $\psi(t) \in \Psi$ such that the cost function $\mathcal{J}(t)$ is minimal, while maintaining $C_{\text{sand}}(t) = 0$.

Two methods can be adopted to ensure that the sandstock remains constant over time, the *penalization method* and/or the *feasible direction method*.

Penalization method

This method consists of adding a penalty term to the cost function \mathcal{J} . So instead of minimizing \mathcal{J} , we minimize both \mathcal{J} and C_{sand} simultaneously. The new cost function \mathcal{J}_{pen} is given by:

$$\mathcal{J}_{\text{pen}} = \mathcal{J} - \beta C_{\text{sand}} \quad (39)$$

where $\beta > 0$ is the sandstock constraint precision parameter and determines the importance of the conservation of the sand constraint. For β small, the constraint is largely ignored and the minimization of the \mathcal{J} governs the evolution of the seabed. For β great, the sandstock constraint dominates the minimization method.

▲ A simple well-chosen factor may be used to express \mathcal{J}_{pen} as the same dimension as \mathcal{J} .

Feasible directions

Another possible approach to incorporate the sand constraint into the morphodynamic model is to use the *feasible direction method*.

Since $C_{\text{sand}}(0) = 0$, we wish to minimize \mathcal{J} while keeping C_{sand} constant. This equates to following the direction $\nabla_{\psi} \mathcal{J}$ while keeping $\nabla_{\psi} C_{\text{sand}} = 0$. In order to do so, we project the direction $\nabla_{\psi} \mathcal{J}$ onto the orthogonal of $\nabla_{\psi} C_{\text{sand}}$. Hence, the direction of descent d becomes:

$$d = \nabla_{\psi} \mathcal{J} - \left\langle \nabla_{\psi} \mathcal{J}, \frac{\nabla_{\psi} C_{\text{sand}}}{\|\nabla_{\psi} C_{\text{sand}}\|} \right\rangle \frac{\nabla_{\psi} C_{\text{sand}}}{\|\nabla_{\psi} C_{\text{sand}}\|} \quad (40)$$

This new direction of descent, illustrated by Figure 15, describes a less optimal path to the minimum of \mathcal{J} , but ensures that $\nabla_{\psi} C_{\text{sand}}(t) = 0$, i.e. $C_{\text{sand}}(t) = 0$, for all $t \in [0, T]$.

We can easily show that the new direction d and $\nabla_{\psi} C_{\text{sand}}$ are now orthogonal:

$$\langle d, \nabla_{\psi} C_{\text{sand}} \rangle = \left\langle \nabla_{\psi} \mathcal{J} - \left\langle \nabla_{\psi} \mathcal{J}, \frac{\nabla_{\psi} C_{\text{sand}}}{\|\nabla_{\psi} C_{\text{sand}}\|} \right\rangle \frac{\nabla_{\psi} C_{\text{sand}}}{\|\nabla_{\psi} C_{\text{sand}}\|}, \nabla_{\psi} C_{\text{sand}} \right\rangle = 0$$

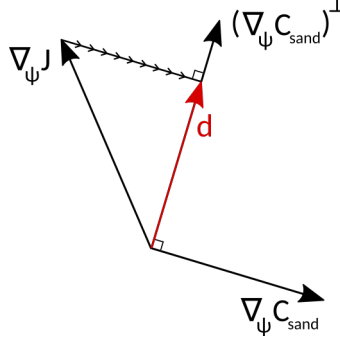


Figure 15: Illustration of the new direction of descent in \mathbb{R}^2 : the direction $\nabla_{\psi} \mathcal{J}$ is projected onto the orthogonal of $\nabla_{\psi} C_{\text{sand}}$ to yield d .

❗ The feasible direction method can also be used to guarantee that the total energy of the waves is conserved for a same forcing condition with regards to the evolution of the seabed, i.e. $\mathcal{E}(\psi_1, t) = \mathcal{E}(\psi_2, t)$ for all $\psi_1, \psi_2 \in \Psi$ and $t \in [0, T]$. All one needs to do is project $\nabla_{\psi} \mathcal{J}$ onto the common orthogonal vector of $\nabla_{\psi} C_{\text{sand}}$ and $\nabla_{\psi} \mathcal{E}$.

II.3.4.c. Bedrock constraint

The third constraint concerns the existence of bedrock in the beach configuration. We assume bedrock to be a solid invariable feature to the cross-shore profile, with a layer of sediment covering it.

Let B be the elevation of the bedrock over Ω as in Figure 1. By definition, B remains constant over time and the seabed ψ cannot appear lower than the bedrock:

$$\psi(x, t) \geq B(x) \quad \forall x \in \Omega, t \in [0, T] \quad (41)$$

In other words, the bedrock acts as a lower bound of the seabed elevation. Equality of Equation (41), for a given $x \in \Omega$, implies exposure of the bedrock.

III Numerical model

III.1 Presentation

In this section, we present the numerical model Opti-Morph and how to use it.

III.1.1 Workflow

Figure 16 illustrates the workflow of the Opti-Morph model, with the associated hydrodynamic model. Before launching the model, the user must first define the initial setting of the simulation. This includes the forcing data, the choice of hydrodynamic model, the seabed elevation data, the choice of cost function, and the constraints.

For each time step, the forcing data is provided to the hydrodynamic model. This model then calculates the wave height over the cross-shore profile and thus provides the cost function \mathcal{J} (or direction of decent d) used by Opti-Morph's morphodynamic module. Using the imported sand characteristics, the new shape of the seabed is determined by minimizing the cost function \mathcal{J} (or following the direction of descent d). Constraints are applied to the seabed either before or after the minimization takes place, and the new seabed is retained. At the next time step, the hydrodynamic model is fed a new forcing condition as well as the new seabed. This cycle continues over the course of the simulation, and illustrates the intricate interaction between the hydrodynamic and morphodynamic processes.

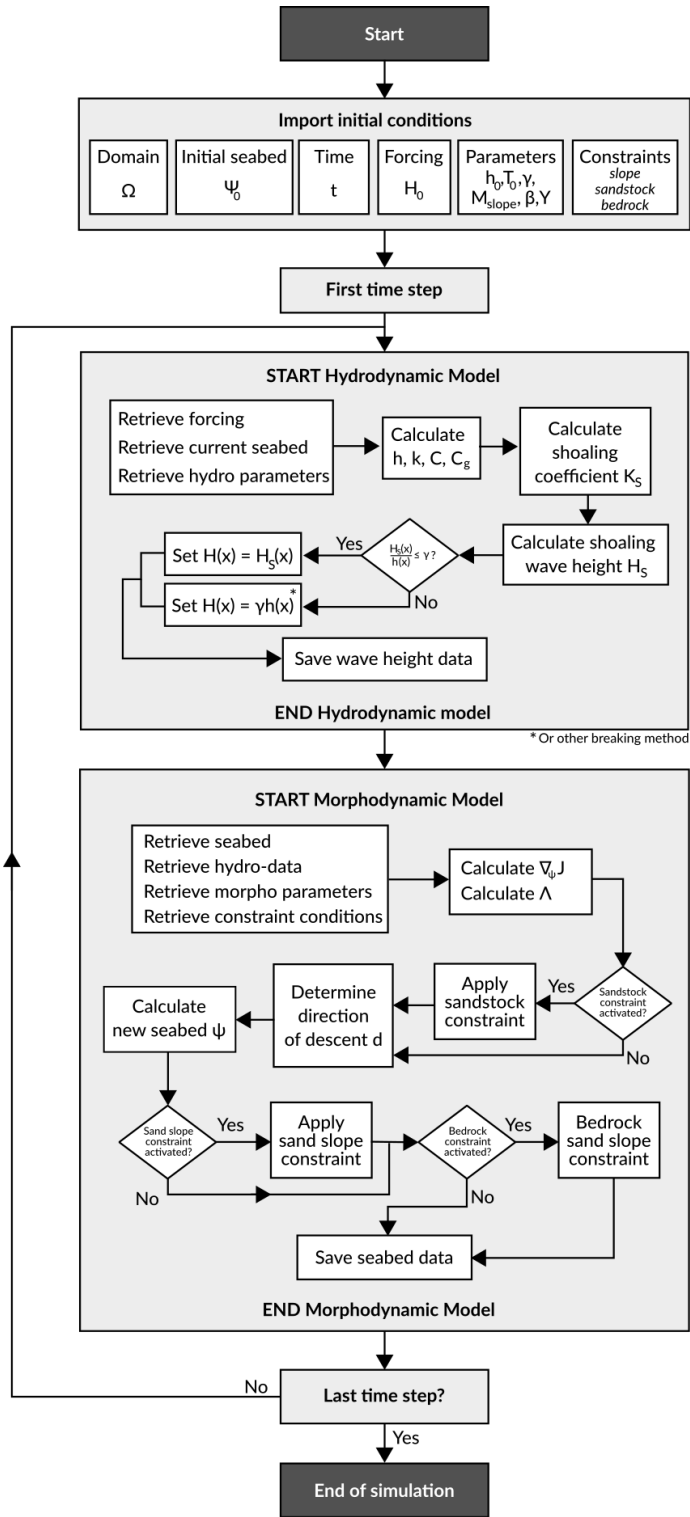


Figure 16: Diagram of the workflow of the Opti-Morph model

III.1.2 Algorithm summary

A detailed summary of the algorithm performed by Opti-Morph is provided below.

For each time step t_n , for $n \in [1, N_T]$, the following steps are applied:

→ **Apply Hydrodynamic model**

- Step 1: Import forcing data $H_0(t_n)$
- Step 2: Import current morphodynamic data ψ_{n-1}
- Step 3: Calculate wave height H_n
- Step 4: Save H_n

→ **Apply Morphodynamic model**

- Step 1: Import current wave height H_n
- Step 2: Import Υ and Λ parameters
- Step 3: Calculate cost function \mathcal{J}_n
- Step 4: Calculate $\nabla_{\psi} \mathcal{J}_n$
- Step 5: Obtain the direction of descent d_n
- Step 6: Apply local a priori constraints (if necessary)
- Step 7: Apply global a priori constraints (if necessary)
- Step 8: Determine new seabed: $\psi_n = \psi_{n-1} + \Upsilon \Lambda d_n$
- Step 9: Apply local a posteriori constraints (if necessary)
- Step 10: Apply global a posteriori constraints (if necessary)
- Step 11: Save ψ_n

❗ The sandstock constraint is considered a global a priori constraint whereas the sand slope and the bedrock constraints are considered local a posteriori constraint.

III.1.3 Class organisation

The numerical model Opti-Morph has been implemented using an oriented object structure. Figure 17 illustrates the structure of the model using a UML diagram. Structuring the model using classes was chosen in order to allow flexibility and creativity within the model. Each object can be regarded as a building block which can easily be modified or replaced depending on the user's intentions. For instance, a different hydrodynamic model can be implemented and adapted to the Opti-Morph model with ease. The same applies for the choice of cost function; to introduce a different cost function \mathcal{J} , the user simply has to implement the new function as well as its gradient and leave the rest of the Opti-Morph model unchanged.

An abstract class `Model` is used to define the general characteristics of a model. We have two different types of model: a hydrodynamic model and a morphodynamic model. The abstract class `Model` is used because both models share a common structure and methods. For instance, in this model we find the accessors relative to the parameters and variables of the model in question. A model contains both a set of parameters and a set of variables. The parameters are considered constant over the entirety of the simulation and determine the characteristics of the model. As its name suggests, variables vary over the execution of the simulation. For instance, the wave-breaking index γ is considered a parameter of hydrodynamic model whereas wave height H is considered a variable. The accessors relative to this data are respectively `getP`, `setP` and `getV`, `setV`.

The `Domain` class defines the mesh of the domain as well as certain physical characteristics of the configurations such as the mean water level h_0 .

No.	Description	Keyword
1	Shoaling model with decreasing exponential breaking	shoaling_1run
2	Shoaling model with decreasing exponential breaking and energy conservation	shoaling_2run
3	Global shoaling model with Munk's breaking condition	shoaling_throughout
4	Local shoaling model with Munk's breaking condition	shoaling_incremental
5	Weighted window local shoaling model	shoaling_window
6	Weighted window local shoaling model and improved boundary conditions	shoaling_window_LC
7	Weighted window shoaling model with anti-dissipative effect	shoaling_window_LC_ADT

Table 3: Table of the different hydrodynamic models and their keywords

The hydro-morphodynamic class `Hydro_morpho_model` is the central component of the Opti-Morph model. This class links the hydrodynamic class `Hydro_model` with the morphodynamic class `Morpho_model`. It is also here where the run method is located. Because of the close relation between the hydrodynamic model `Hydro_model` and the morphodynamic model `Morpho_model`, both classes are associated with each other. Therefore, a hydrodynamic object can be found in the morphodynamic model and vice versa.

The hydrodynamic class `Hydro_model` which inherits from the model class `Model` is used to determine the wave height over the cross-shore profile. To do so, we use the morphodynamic data (such as the seabed) as well as forcing data. The forcing data is stored in a separate class named `Forcing`. Applying the `run` method calculates the wave height and saves the result in a table under the keyword `H`. Each of the hydrodynamic models presented in Section II.2 are sub-classes of `Hydro_model`. A summary of the different keywords is features in table 3. Additional tools used by the `Hydro_model` class can be found in the imported file `hydrotools.py`.

The morphodynamic class `Morpho_model` which also inherits from the model class `Model` is used to determine the evolution of the seabed using optimization methods. Applying the `run` method calculates the seabed elevation at that given time step, using the data calculated by the hydrodynamic model `Hydro_model`. The physics is determined by the choice of cost function, whose keywords are summarized in Table 2. The constraints associated to this optimization problem are stored in a separate class named `Constraints`. The results of the morphodynamic simulation are saved in a table with the seabed stored under the keyword `psi`. Additional tools and methods relative to the morphodynamic model can be found in the imported file `morphotools.py`.

A final class named `plot_data` is called by the `Hydro_morpho_model` class and provides the visual representations of the results. Several methods have been implemented for different graphs depending on the users needs. Numerical data can also be exported if the user wishes to plot the data manually using external tools.

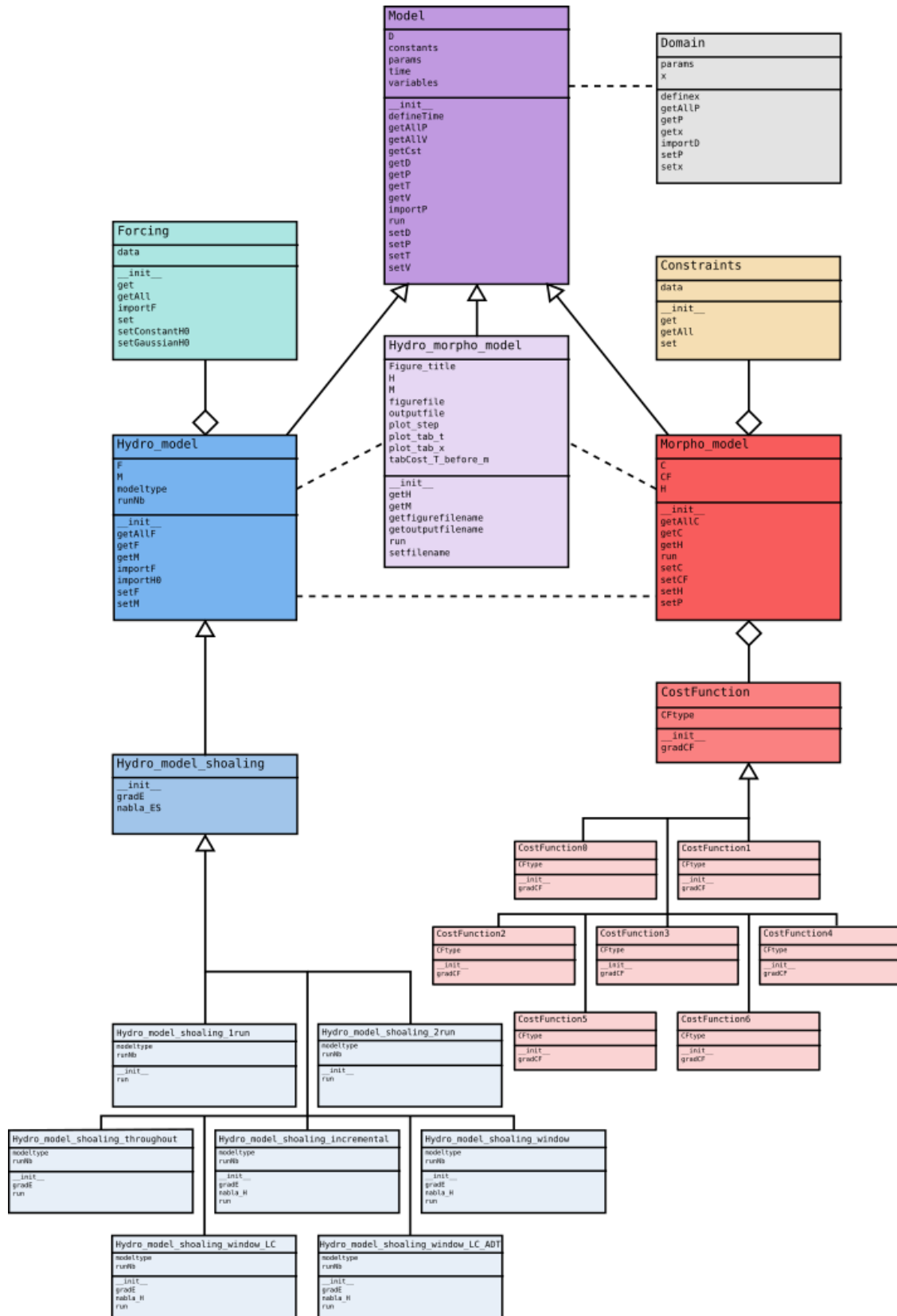


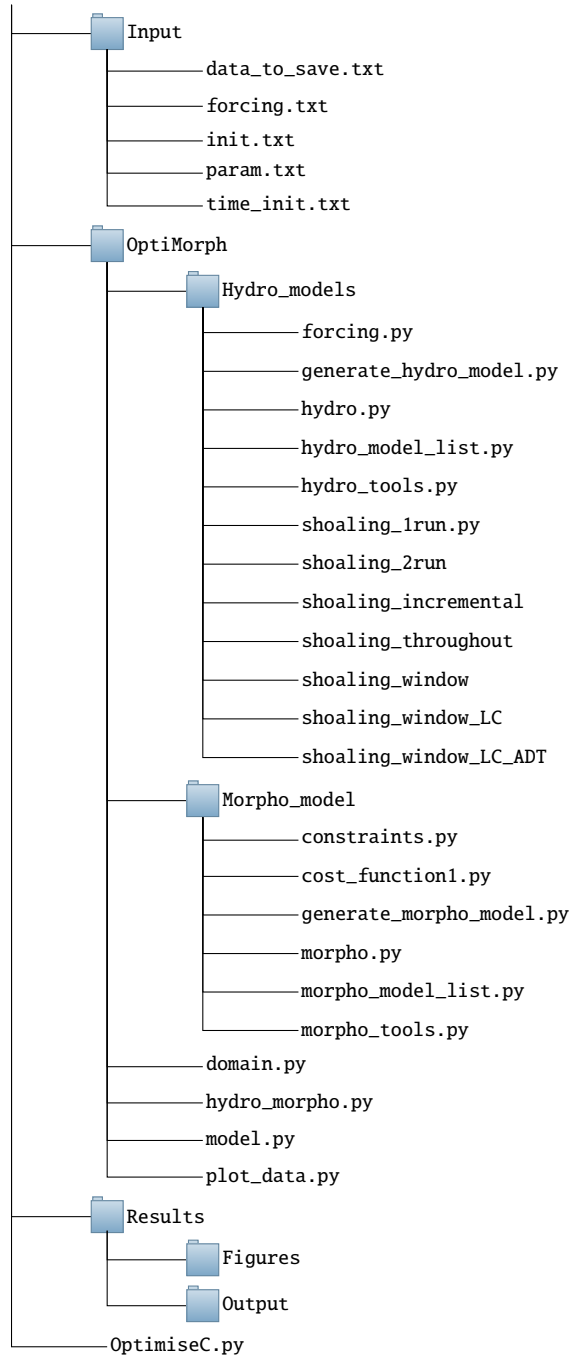
Figure 17: UML diagram of the Opti-Morph model

III.2 Running Opti-Morph

III.2.1 Getting started

This model was developed in Python 3.5.2. Therefore, in order to execute the Opti-Morph model, the user will need a version of Python compatible with Python 3.5.2 to be installed on their workstation as well as the following packages: `math`, `numpy`, `scipy`, `abc`, `matplotlib`, `netCDF4`.

The files should be organised as follows:



III.2.2 Main file

The main file of Opti-Morph that should be run either from the terminal or the chosen development environment is named `OptimiseC.py`.

The `OptimiseC.py` file is organised as followed:

- Import packages
- Import data from Input folder
- Figure preparation
- Definition of domain
- Definition of Hydrodynamic model
- Definition of Morphodynamic model
- Definition of Hydro-Morphodynamic model
- Definition of Output
- Run model

Below is an example of `OptimiseC.py` is provided below. The sections highlighted in yellow are modifiable by the user. In particular, the choice of hydrodynamic model (line 52) and cost function (line 67). This will no longer be the case once a graphical user interface (GUI) is built.

```
1 from OptiMorpho.domain import *
2 from OptiMorpho.model import *
3 from OptiMorpho.Hydro_models.hydro import *
4 from OptiMorpho.Morpho_model.morpho import *
5 from OptiMorpho.hydro_morpho import *
6 from OptiMorpho.enum_lists import *
7 from OptiMorpho.enum_generators import *
8
9 from math import pi
10
11 # Import data
12 params = genfromtxt('Input/param.txt')
13 h0 = params[0]
14 T0 = params[1]
15 gamma = params[2]
16 beta = params[3]
17 sandflag = params[4]
18 smoothflag = params[5]
19
20 init = genfromtxt('Input/init.txt')
21 x = init[:,0]
22 psi0 = init[:,1]
23 bedrock = init[:,2]
24 mobility = init[:,3]
25 slopemax = init[:,4]
26
27 init_time = genfromtxt("Input/time_init.txt")
28 time = init_time[:,0]
29 H0 = init_time[:,1]
30
31 dts = genfromtxt('Input/data_to_save.txt', dtype = None)
32 dtcpt = int(dts[0])
33 outargs_x = []
34 outargs_t = []
```

```

35 inX = True
36 if int(dts[1]) == 1:
37     outargs_x.append('default')
38 for i in range(2, len(dts)):
39     if (dts[i].decode('utf-8') == "END"):
40         inX = False
41     if inX == True:
42         outargs_x.append(dts[i].decode('utf-8'))
43     elif (dts[i].decode('utf-8') != 'END'):
44         outargs_t.append(dts[i].decode('utf-8'))
45
46 # Domain
47 D1 = Domain()
48 D1.setx(x)
49 D1.setP("h0", h0)
50
51 # Hydrodynamic model
52 H1 = generate_hydro_model(Enum_H.shoaling_window_LC_ADT)
53 H1.setF("T0", T0)
54 H1.setF("C0", 9.81/(2*pi)*T0)
55 H1.setF("sigma0", 2*pi/T0)
56 H1.setF("theta0", 0)
57 H1.setP("gamma", gamma)
58 H1.setP("dwin", 5)
59 H1.setP("Nwin", floor(dwin/xstep))
60 H1.setP("aAD", 1)
61 H1.setP("bAD", 1.2)
62 H1.setT(time)
63 H1.setF("H0", H0)
64
65 # Morphodynamic model
66 M1 = Morpho_model()
67 M1.setCF(Enum_CF.CF0)
68 M1.setC("sand", sandflag)
69 M1.setC("Mslope", slopemax)
70 M1.setP("rho0", mobility)
71 M1.setP("bedrock", bedrock)
72 M1.setV("beta", beta)
73 M1.setP("psi0", psi0)
74 M1.setP("smoothing", smoothflag)
75
76 # Hydro-morphodynamic model
77 HMI = Hydro_morpho_model(D1, H1, M1)
78 HMI.Figure_title = "Example of simulation with linear seabed"
79 filetag = "Test_1"
80 HMI.setfilename(filetag)
81
82 # Set time-dependent and space-dependent variables to plot
83 to_plot_t(HMI, outargs_t)
84 to_plot_x_during_run(HMI, dtcpt, outargs_x)
85
86 # Run Opti-Morph
87 HMI.run()

```

Listing 1: Example of OptimiseC.py file

III.2.3 Input data

The input data is found in a folder named `Input`. Four files should appear in this folder:

- `init.txt`: initial cross-shore data
- `time_init.txt`: forcing data
- `param.txt`: model parameters
- `data_to_save.txt`: requested output data

III.2.3.a. The `init.txt` file

Let $(x_p)_{p \in [0, N_\Omega]}$ be the discretization of Ω , with N_Ω the total number of points.

The `init.txt` file is composed of 5 columns of data:

- the discretization of the domain: $(x_p)_{p \in [0, N_\Omega]}$
- the initial seabed: $(\psi(x_p))_{p \in [0, N_\Omega]}$
- the bedrock: $(B(x_p))_{p \in [0, N_\Omega]}$
- the sand mobility parameter: $(Y(x_p))_{p \in [0, N_\Omega]}$
- the slope parameter: $(M_{\text{slope}}(x_p))_{p \in [0, N_\Omega]}$

▲ Ideally, the bathymetric data should ensure that at $x = 0$, we are in deep water. That is, the water depth h is greater than half the wavelength L (cf. [Dean and Dalrymple, 1991]).

▲ For a seabed with the same type of sediment over the cross-shore profile, the mobility parameter and slope parameter will be constant over Ω .

▲ Setting the bedrock as $B(x_p) = 0$ for all $p \in [0, N_\Omega]$ deactivates the bedrock constraint.

▲ The user must verify that the initial seabed is consistent with the slope constraint parameter, i.e. that the slope of ψ_0 doesn't exceed M_{slope} . Otherwise, the model will automatically rectify the seabed so as to comply with the slope constraint after the first time step. This may result in a significant and instantaneous change to the seabed.

Examples of the `init.txt` file can be found in Section IV, Figures 21a and 24a.

III.2.3.b. The `time_init.txt` file

Let $(t_p)_{p \in [0, N_T]}$ be the discretization of the time interval $[0, T]$, with N_T the total number of points.

The `time_init.txt` file is composed on two columns of data and contains the time-dependent data:

- the discretization of the simulated time interval: $(t_p)_{p \in [0, N_T]}$
- the forcing wave height data: $(H_0(t_p))_{p \in [0, N_T]}$

Examples of the `time_init.txt` file can be found in Section IV, Figures 21b and 24b.

III.2.3.c. The `param.txt` file

The physical and numerical parameters of the model have been grouped together in one file named `param.txt` and organised by type to allow the user to easily modify the different parameters governing the Opti-Morph model. This file should contain the following parameters, in the given order:

```

# Domain parameters
h0

# Hydro parameters
T0
gamma

# Morpho parameters
beta

# Constraint flag
sandc

# Smoothing flag
smoothf

```

Here, h_0 is the still water level (m), T_0 is the wave period (s), here assumed constant over the cross-shore profile and γ is the wave-breaking index defined in Section II.2.2. The parameter β is the precision parameter required for the sandstock constraint (cf. section II.3.4) and $sandc$ is the flag associated with the sandstock constraint. The final parameter is the $smoothf$ is a smoothing function, that the user can apply to the seabed to remove noise due to numerical inaccuracies.

▲ To activate the sandstock constraint, set the flag to 1 and set the associated parameter β to its desired value. To deactivate the constraint, set the flag to 0. The same applied to the $smoothf$ flag.

▲ The slope constraint is permanently activated. In order to locally/temporarily deactivate it, the user can simply increase M_{slope} so that the constraint cannot be triggered (see Section IV.2).

Examples of the `param.txt` file can be found in Section IV, Figures 21c and 24c.

III.2.3.d. The `data_to_save.txt` file

The Opti-Morph model offers the possibility of creating a graphic representation of the different types of data, as well as exporting this data. The user provides the keywords of the data they wish to retrieve at the end of the simulation. The list of the required output figures/data is given in the `data_to_save.txt` file, situated in the Input folder.

The `data_to_save.txt` file is separated into 3 sections. The first concerns the number of time steps between each outputs of the space-dependent variables. Then the list of the space-dependent variables should be given, followed by the list of time-dependent variables.

An example of `data_to_save.txt` file is given below:

```

# Output data
T_out
default_out

# List of space-dependent variables to export/plot
*List*
END

# List of time-dependent variables to export/plot
*List*
END

```

where `T_out` is number of time steps between each output for the space-dependent variables. If the user desires a snapshot of the seabed, wave height, or any other space-dependent variable every 50 time incrementations, then they should indicate "50" in the first row of `data_to_save.txt` file.

❗ This parameter has no impact on the time-dependent figures/exports.

The keyword `default_out` is the flag indicating whether the user requires the default figure to be produced by Opti-Morph. An example of this figure is given in Figure 18. The list of space-dependent variables (defined over the cross-shore profile) is then given, with each keyword situated on a different line. The list of keywords can be found in Table 4 (left). The keyword `END` is required to mark the end of this first list. The list of time-dependent variables (defined over the time-series) follows, with one keyword per line. The list of keywords can be found in Table 4 (right). Once again, the keyword `END` is needed to mark the end of this second list.

Space dependent keywords	time-dependent keywords
<ul style="list-style-type: none"> • <code>ADT²</code>: anti-dissipative term • <code>C</code>: phase velocity • <code>Cg</code>: group velocity • <code>H</code>: wave height • <code>HS</code>: shoaling wave height • <code>Lambda</code>: wave excitation parameter • <code>h</code>: water depth • <code>k</code>: wave number • <code>psi</code>: seabed elevation • <code>slope</code>: slope of the seabed 	<ul style="list-style-type: none"> • <code>E_after</code>: total wave energy after the seabed response • <code>E</code>: total wave energy • <code>EBS</code>: energy over Ω_B • <code>EOB</code>: energy over Ω_S • <code>H0</code>: forcing wave height • <code>alpha¹</code>: breaking parameter • <code>nbB</code>: position of x_B in the discretization of the domain Ω • <code>nbS</code>: position of x_S in the discretization of the domain Ω • <code>sandstock</code>: sandstock • <code>xB</code>: location of (first) breaking • <code>xS</code>: position of the shoreline

Table 4: Table of keywords associated with the spatial (*left*) and temporal (*right*) variables available to plot/export

¹: only available for `shoaling_1run` and `shoaling_2run`

²: only available for `shoaling_window_LC_ADT`

III.2.4 Output data

At the end of the simulation, the requested figures/exports are found in the `Results` folder, with the figures in `Figures` and the exported data in `Output`.

III.2.4.a. Figure folder

At the end of the simulation, PNG images of the requested variables can be found in the `Figure` file.

If requested using the `default_out` flag set to 1, Opti-Morph produces a 6-panel figure detailing the main contributing variables of the simulation at a given time. An example of the default figure is given in Figure 18.

FLUME CONFIGURATION
 $t = 6.01e + 02$ (iteration 600/1799)

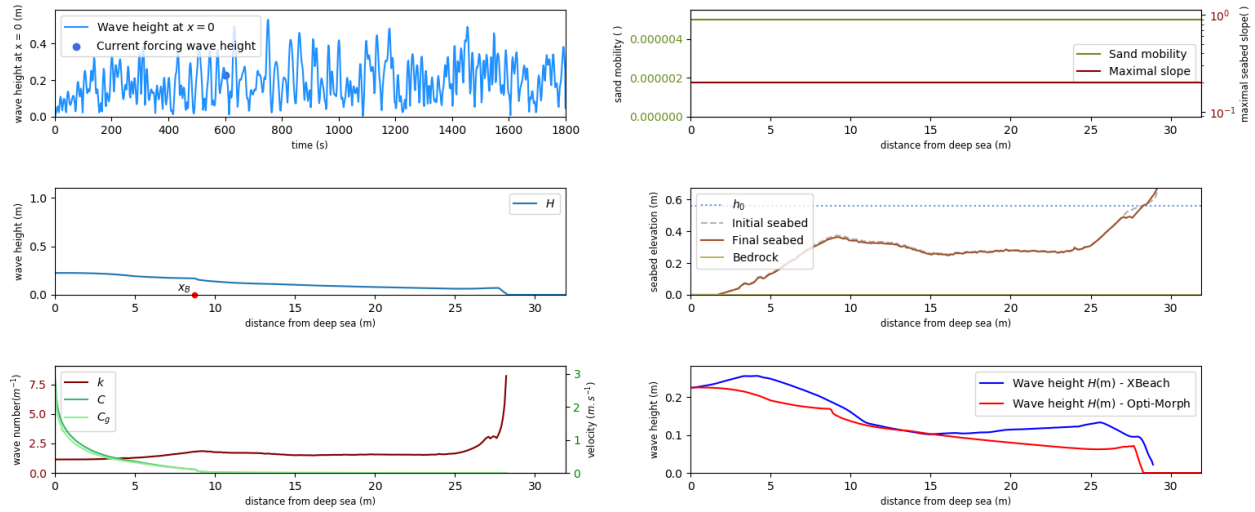
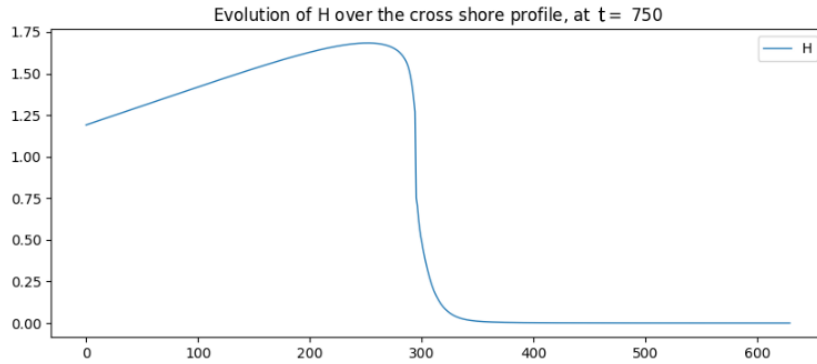


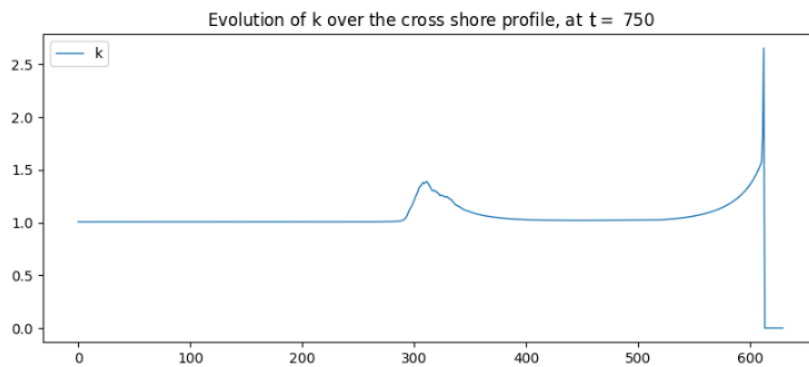
Figure 18: Example of the default figure, mid simulation. *Upper left:* Time series of the forcing wave height H_0 with the current forcing indicated with a circular point. *Middle left:* Wave height calculated over the cross-shore profile using the hydrodynamic model no.7. *Bottom left:* Variation of the wavenumber k (red), phase velocity C (green) and group velocity C_g (light green) over the cross-shore profile. *Upper right:* Variation over the cross-shore profile of the two hyper-parameters required by Opti-Morph: sand mobility parameter Υ (green) and the maximal slope parameter M_{slope} . Here constant, these parameters vary when the seabed features different types of sediment over Ω , or in the presence of man-made structures (such as submerged breakwaters, see Section IV.2). *Middle right:* Seabed elevation over the cross-shore profile. The initial seabed (grey dashed line), the bedrock (grey solid line) and the mean water height (dotted blue line) have also been specified. *Bottom right:* Comparison of wave height with another hydro-morphodynamic model, if available. Here, a comparison with Xbeach’s hydrodynamic module is conducted.

These figures are saved in the folder Figures under the title Figure_hyd_mor_beta_gamma_time.png where beta, Mslope, gamma are the value of the parameters chosen for the simulation and time is the time step.

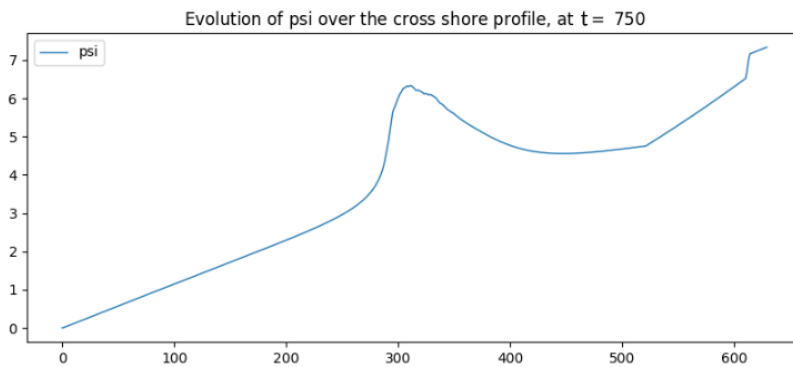
Opti-Morph can also provide simple plots of the quantities involved in the model simulation. Examples of these plots can be found in Figure 19, where the wave height, wavenumber and seabed elevation are given mid-simulation.



(a) Wave height over the cross-shore profile, mid-simulation



(b) Wavenumber over the cross-shore profile, mid-simulation



(c) Seabed elevation over the cross-shore profile, mid-simulation

Figure 19: Examples of simple plots generated by Opti-Morph during the simulation

III.2.4.b. Output folder

At the end of the simulation, text files of the requested data can be found in the Output folder. The user can then use this data to perform a more thorough post-processing analysis of the morphodynamic and hydrodynamic processes that took place during the simulation.

IV Applications

In this section, the seabed is described as a simple linear function over the cross-shore profile. First, we simulate the results over a homogeneous sandy seabed, then we look at introducing submerged structures designed to limit wave activity at the coast.

IV.1 Linear Seabed Beach Configuration

IV.1.1 Setting

The initial cross-shore configuration is given in Figure 20: the domain measures 600 m, the mean water level is set at 7 m and we apply a storm profile to the seabed, given by the top left graph of Figure 20. Here we consider a homogeneous sandy seabed, and therefore the mobility of the seabed Υ and the maximal slope parameter M_{slope} are assumed constant over the cross-shore profile Ω .

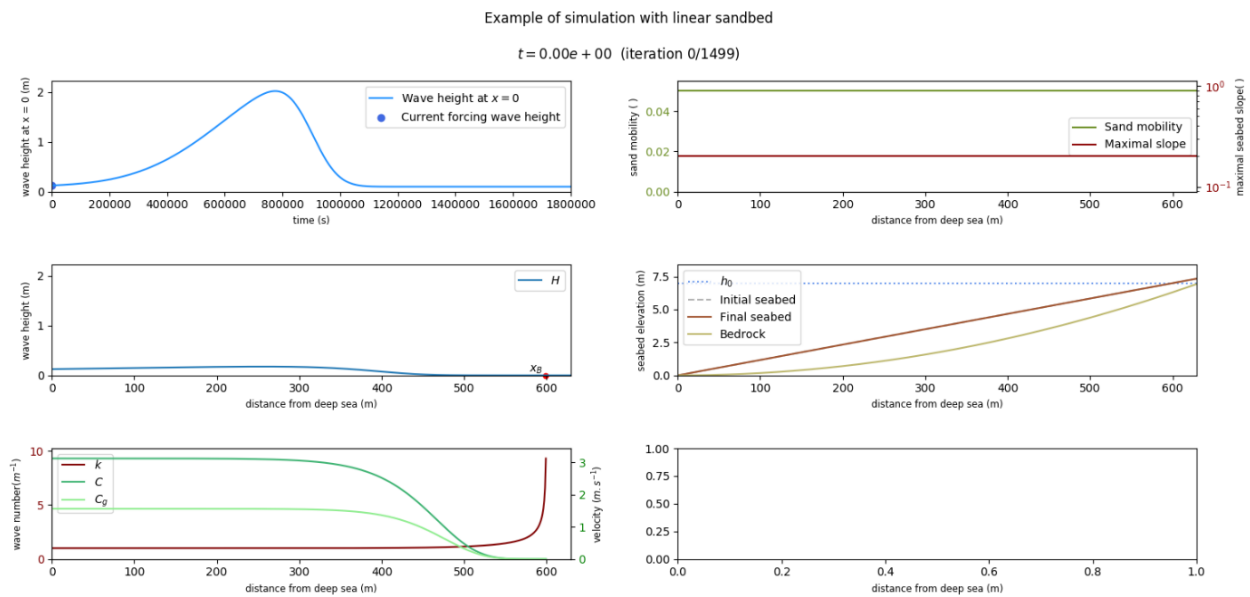


Figure 20: Initial sandy beach configuration

IV.1.2 Input files

The input files have been constructed accordingly, with Figure 21 providing an overview of the different files.

0.0000e+00	0.0000e+00	0.0000e+00	5.0000e-02	2.0000e-01
1.0000e+00	1.1667e-02	1.7516e-05	5.0000e-02	2.0000e-01
2.0000e+00	2.3333e-02	7.0064e-05	5.0000e-02	2.0000e-01
3.0000e+00	3.5000e-02	1.5764e-04	5.0000e-02	2.0000e-01
4.0000e+00	4.6667e-02	2.8025e-04	5.0000e-02	2.0000e-01
5.0000e+00	5.8333e-02	4.3790e-04	5.0000e-02	2.0000e-01
6.0000e+00	7.0000e-02	6.3057e-04	5.0000e-02	2.0000e-01
7.0000e+00	8.1667e-02	8.5828e-04	5.0000e-02	2.0000e-01
8.0000e+00	9.3333e-02	1.1210e-03	5.0000e-02	2.0000e-01
9.0000e+00	1.0500e-01	1.4189e-03	5.0000e-02	2.0000e-01
1.0000e+01	1.1667e-01	1.7516e-03	5.0000e-02	2.0000e-01
1.1000e+01	1.2833e-01	2.1194e-03	5.0000e-02	2.0000e-01
1.2000e+01	1.4000e-01	2.5223e-03	5.0000e-02	2.0000e-01
1.3000e+01	1.5167e-01	2.9602e-03	5.0000e-02	2.0000e-01
1.4000e+01	1.6333e-01	3.4331e-03	5.0000e-02	2.0000e-01
1.5000e+01	1.7500e-01	3.9411e-03	5.0000e-02	2.0000e-01
1.6000e+01	1.8667e-01	4.4841e-03	5.0000e-02	2.0000e-01
1.7000e+01	1.9833e-01	5.0621e-03	5.0000e-02	2.0000e-01
1.8000e+01	2.1000e-01	5.6751e-03	5.0000e-02	2.0000e-01
1.9000e+01	2.2167e-01	6.3232e-03	5.0000e-02	2.0000e-01
2.0000e+01	2.3333e-01	7.0064e-03	5.0000e-02	2.0000e-01
2.1000e+01	2.4500e-01	7.7245e-03	5.0000e-02	2.0000e-01
2.2000e+01	2.5667e-01	8.4777e-03	5.0000e-02	2.0000e-01
2.3000e+01	2.6833e-01	9.2659e-03	5.0000e-02	2.0000e-01
2.4000e+01	2.8000e-01	1.0089e-02	5.0000e-02	2.0000e-01
2.5000e+01	2.9167e-01	1.0947e-02	5.0000e-02	2.0000e-01
2.6000e+01	3.0333e-01	1.1841e-02	5.0000e-02	2.0000e-01
2.7000e+01	3.1500e-01	1.2769e-02	5.0000e-02	2.0000e-01
2.8000e+01	3.2667e-01	1.3732e-02	5.0000e-02	2.0000e-01
2.9000e+01	3.3833e-01	1.4731e-02	5.0000e-02	2.0000e-01

domain discretization initial seabed bedrock mobility parameter maximal slope parameter

(a) init.txt file

0.0000e+00	1.2440e-01
1.2000e+03	1.2469e-01
2.4016e+03	1.2499e-01
3.6024e+03	1.2529e-01
4.8032e+03	1.2560e-01
6.0040e+03	1.2590e-01
7.2048e+03	1.2621e-01
8.4056e+03	1.2653e-01
9.6064e+03	1.2684e-01
1.0807e+04	1.2717e-01
1.2000e+04	1.2749e-01
1.3209e+04	1.2782e-01
1.4410e+04	1.2815e-01
1.5610e+04	1.2848e-01
1.6811e+04	1.2882e-01
1.8012e+04	1.2916e-01
1.9213e+04	1.2951e-01
2.0414e+04	1.2985e-01
2.1615e+04	1.3021e-01
2.2815e+04	1.3056e-01
2.4016e+04	1.3092e-01
2.5217e+04	1.3129e-01
2.6418e+04	1.3165e-01
2.7618e+04	1.3202e-01
2.8819e+04	1.3240e-01
3.0020e+04	1.3278e-01
3.1221e+04	1.3316e-01
3.2422e+04	1.3355e-01
3.3622e+04	1.3394e-01
3.4823e+04	1.3433e-01

time discretization forcing wave height

(b) time_init.txt file

```
# Domain parameters
7 # still water level [m]

# Hydro parameters
2 # wave period [s]
0.55 # wave breaking index []

# Morpha parameters
0.1 # sandstock constraint precision parameter []

# Constraint flags
1 # 0: OPEN SEA CONFIG. / 1: BASIN CONFIG.

# Smoothing flag
0
```

(c) param.txt file

```
# Output data
50 # number of time incrementations between each figure/output
1 # default figures (1: ACTIVATED ; 0 DEACTIVATED)

# List of functions to save/plot defined over the cross-shore
psi
H
END

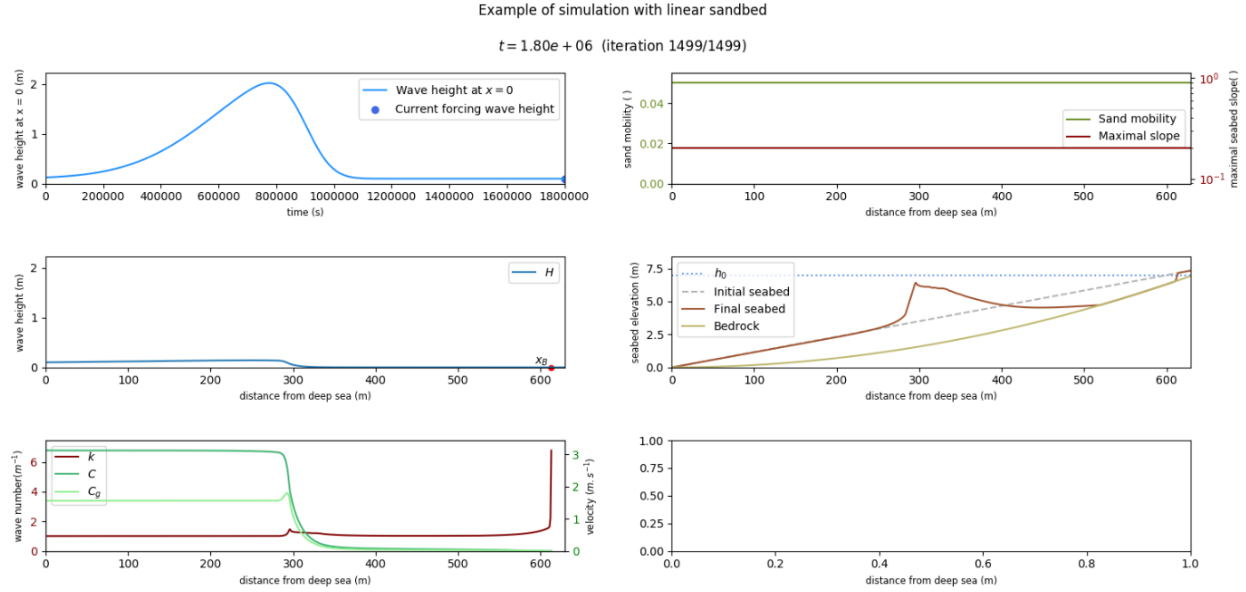
# List of functions to save/plot defined over the time series
sandstock
xB
xS
END
```

(d) data_to_save.txt file

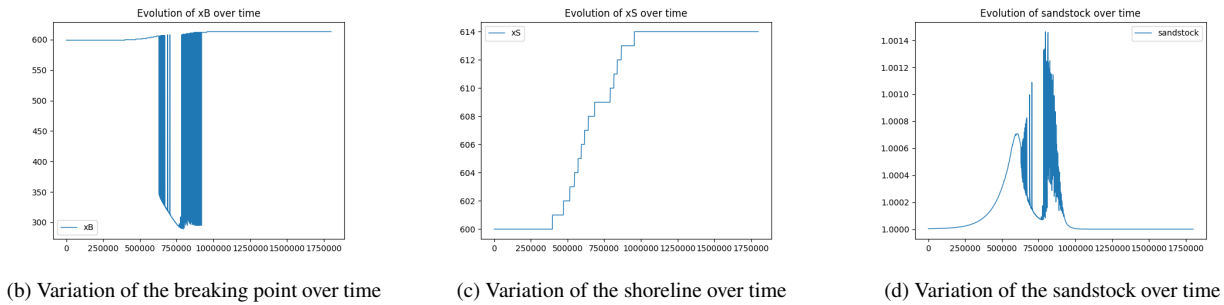
Figure 21: Input files used in the simulation

IV.1.3 Results

At the end of the simulation, we get the following results of Figure 22. Here, the default figure has been requested, as well as xB, xS, and sandstock.



(a) Result of the beach configuration at the end of simulation



(b) Variation of the breaking point over time

(c) Variation of the shoreline over time

(d) Variation of the sandstock over time

Figure 22: Various plots provided by Opti-Morph

A thorough analysis of the results of Opti-Morph can be found in [Cook et al., 2021b] for an experimental flume configuration and [Cook et al., 2021a] for a linear seabed with submerged breakwaters.

IV.2 Beach Configuration with submerged breakwaters

IV.2.1 Setting

In this simulation, we introduce a solid submerged structure, in the same manner as [Cook et al., 2021a]. To do this, we modify the seabed profile, as well as the sand mobility parameter Υ and the maximal slope parameter M_{slope} , which are no longer constant over the cross-shore profile. In the case of the mobility parameter, no movement can occur at the location of the structures, i.e. $\Upsilon = 0$ where the breakwater is positioned. Similarly, the maximal slope parameter has also been modified to locally deactivate the slope constraint over the structure. Figure 23 shows the new initial configuration incorporating a submerged breakwater located at $x = 180\text{ m}$.

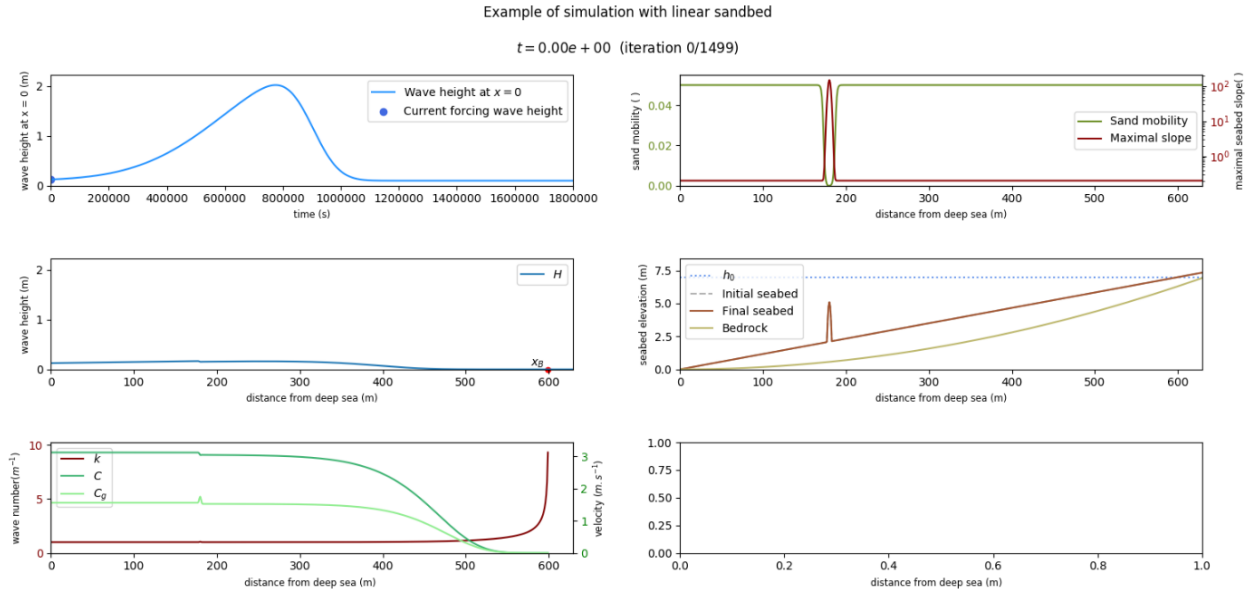


Figure 23: Initial sandy beach configuration with a submerged breakwater located at $x = 180\text{ m}$

IV.2.2 Input files

Similar to before, the input files have been constructed, with Figure 21 providing an overview of the different files. Now, the mobility parameter Υ and the slope parameter M_{slope} vary over the cross-shore profile.



Figure 24: Input files used in the simulation

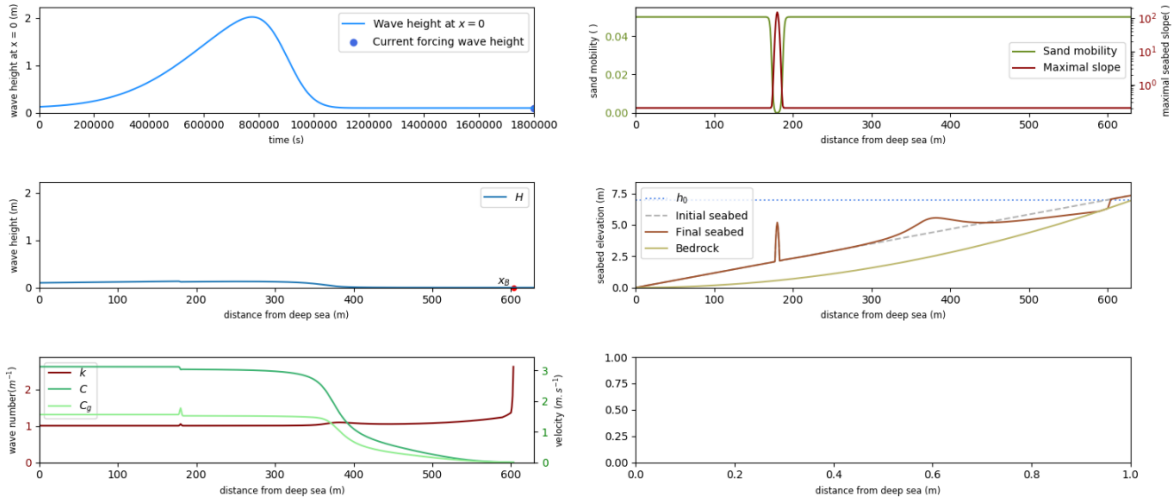
A thorough analysis of the behavior of Opti-Morph in regards to the introduction of geotubes can be found in [Cook et al., 2021a].

IV.2.3 Results

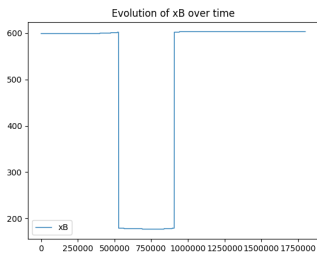
At the end of the simulation, we get the following results of Figure 25. As before, the default figure has been activated (`default_out = 1`), and the graphs of `xB`, `xS`, and `sandstock` were requested.

Example of simulation with linear sandbed

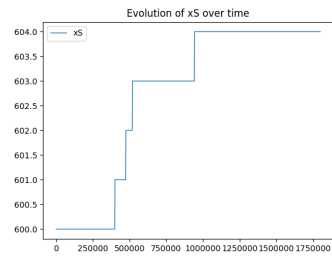
$t = 1.80e + 06$ (iteration 1499/1499)



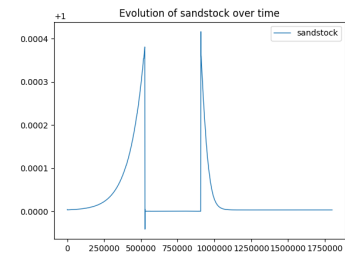
(a) Result of the beach configuration with submerged breakwaters at the end of simulation



(b) Variation of the breaking point over time



(c) Variation of the shoreline over time



(d) Variation of the sandstock over time

Figure 25: Various plots provided by Opti-Morph

A List of parameters

A summary of the notations, variables and parameters used throughout this user guide is provided in the following table.

Name	Keyword	Description	Unit
Ω	Omega	domain of the study of the cross-shore profile	m
Ω_S	Omega_S	subset of Ω over which the waves shoal	m
Ω_B	Omega_B	subset of Ω over which the waves break	m
x_{\max}	x_max	upper-bound of the domain, situated beyond the shore	m
x_S	x_S	shoreline position	m
x_B	x_B	location of (first) breaking of the waves	m
Ψ	Psi	set of parameters describing the seabed	m
ψ	psi	elevation of the seabed	m
ψ_0	psi_0	initial seabed (at $t = 0$)	m
ψ_t	-	evolution of the seabed over time	$m.s^{-1}$
B	bedrock	bedrock	m
T	T	upper-bound of the time series	s
h_0	h_0	mean water level	m
h	h	water depth	m
T_0	T_0	wave period	m
H_0	H_0	wave height at $x = 0$	m
H	H	wave height	m
C	C	phase velocity of the waves	$m.s^{-1}$
C_0	C0	initial phase velocity of the waves	$m.s^{-1}$
C_g	C_g	group velocity of the waves	$m.s^{-1}$
n	n	ratio of phase velocity and group velocity	-
k	k	wave number	m^{-1}
L	-	wavelength	m
σ	sigma	wave pulsation	s^{-1}
K_S	K_S	shoaling coefficient	-
γ	gamma	wave-breaking index	-
α	alpha	breaking type parameter	-
d_{win}	dwin	maximal distance of local spatial dependency of a wave	m
w	-	weighting function	-
H_0^w	-	weighted average of the seaward waves	m
α_w	-	boundary condition transitional parameter	-
a_{AD}	aAD	first parameter in the anti-dissipative term	-
b_{AD}	bAD	second parameter in the anti-dissipative term	-
χ_{AD}	ADT	anti-dissipative term	-
g	g	gravitational acceleration	$m.s^{-2}$
ρ_w	rho_w	water density	$kg.m^{-3}$
\mathcal{J}	J	cost function to be minimized	$J.m^{-1}$
\mathcal{E}	E	Potential wave energy	$J.m^{-1}$
-	E_after	Potential wave energy associated with the updated seabed	$J.m^{-1}$
\mathcal{E}_S	E0B	Potential energy of the shoaling waves	$J.m^{-1}$
\mathcal{E}_B	EBS	Potential energy of the breaking waves	$J.m^{-1}$
Υ	Upsilon	mobility of the sediment	$m.s.kg^{-1}$
φ	-	orbital velocity damping function	-
d	d	direction of descent	J
M_{slope}	M_slope	angle of repose of the sediment	-
C_{sand}	-	difference in sandstock between the initial and current seabed	-
β	beta	sandstock constraint precision parameter	-
\mathcal{J}_{pen}	-	cost function incorporating the sandstock penalty term	$J.m^{-1}$
N_{Ω}	N	number of points in the discretization of the domain	-

N_T	-	number of points in the discretization of the time series	-
$(x_p)_{p \in [0, N_\Omega]}$	\mathbf{x}	discretization of the domain Ω	-
$(t_p)_{p \in [0, N_T]}$	\mathbf{t}	discretization of time interval $[0, T]$ of the simulation	-
-	nbB	position of x_B in the discretization of the domain Ω	-
-	nbS	position of x_S in the discretization of the domain Ω	-
-	sandc	flag activating the sandstock constraint	-
-	smoothf	flag activating a smoothing effect on the seabed	-
-	T_out	number of time steps between output	-
-	default_t_out	flag activating the creation of the default figure output	-
-	CF0	choice of direction of descent	J

References

- [Bouharguane et al., 2010] Bouharguane, A., Azerad, P., Bouchette, F., Marche, F., and Mohammadi, B. (2010). Low complexity shape optimization and a posteriori high fidelity validation. *Discrete and Continuous Dynamical Systems-series B - DISCRETE CONTIN DYN SYS-SER B*, 13.
- [Cook et al., 2021a] Cook, M., Bouchette, F., Mohammadi, B., and Fraysse, N. (2021a). Application of Opti-Morph: Optimized beach protection by submerged geotextile tubes. Preprint.
- [Cook et al., 2021b] Cook, M., Bouchette, F., Mohammadi, B., Meulé, S., and Fraysse, N. (2021b). Opti-Morph, a new platform for sandy beach dynamics by constrained wave energy minimization. preprint.
- [Cook et al., 2021c] Cook, M., Bouchette, F., Mohammadi, B., Sprunck, L., and Fraysse, N. (2021c). Optimal port design minimizing standing waves with a posteriori long term shoreline sustainability analysis. preprint.
- [Dean and Dalrymple, 1991] Dean, R. G. and Dalrymple, R. A. (1991). *Water Wave Mechanics for Engineers and Scientists*. WORLD SCIENTIFIC.
- [Griewank and Walther, 2008] Griewank, A. and Walther, A. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, second edition.
- [Hascoët and Pascual, 2004] Hascoët, L. and Pascual, V. (2004). TAPENADE 2.1 user's guide. Technical Report RT-0300, INRIA.
- [Isèbe et al., 2008a] Isèbe, D., Azerad, P., Bouchette, F., Ivorra, B., and Mohammadi, B. (2008a). Shape optimization of geotextile tubes for sandy beach protection. *International Journal for Numerical Methods in Engineering*, 74(8):1262–1277.
- [Isèbe et al., 2008b] Isèbe, D., Azerad, P., Mohammadi, B., and Bouchette, F. (2008b). Optimal shape design of defense structures for minimizing short wave impact. *Coastal Engineering*, 55(1):35–46.
- [Mohammadi and Bouchette, 2014] Mohammadi, B. and Bouchette, F. (2014). Extreme scenarios for the evolution of a soft bed interacting with a fluid using the value at risk of the bed characteristics. *Computers and Fluids*, 89:78–87.
- [Mohammadi and Bouharguane, 2011] Mohammadi, B. and Bouharguane, A. (2011). Optimal dynamics of soft shapes in shallow waters. *Computers and Fluids*, 40:291–298.
- [Munk, 1949] Munk, W. (1949). The solitary wave theory and its application to surf problems. *Annals of the New York Academy of Sciences*, 51:376 – 424.
- [Soulsby, 1987] Soulsby, R. (1987). Calculating bottom orbital velocity beneath waves. *Coastal Engineering - COAST ENG*, 11:371–380.