Appendix

A A Genetic Algorithm to Solve the Optimization Problem (I.1)

In this section, we describe in more detail how the dual genetic algorithm presented in the introduction works.

A.1 Population Creation

First, a population $\psi_{i=0,\dots,N_{pop}}$ of N_{pop} individuals is created (Figure I.2.1)). Initially, all these individuals are identical to the mother individual ψ_M. For each of these individuals, a random number of "bumps" (represented by figure A.1) is added to ψ_M. The parameters of these "bumps" (number, amplitude A, position, width L) are determined randomly: this ensures diversity in the initial population, as illustrated in figure I.2.1).

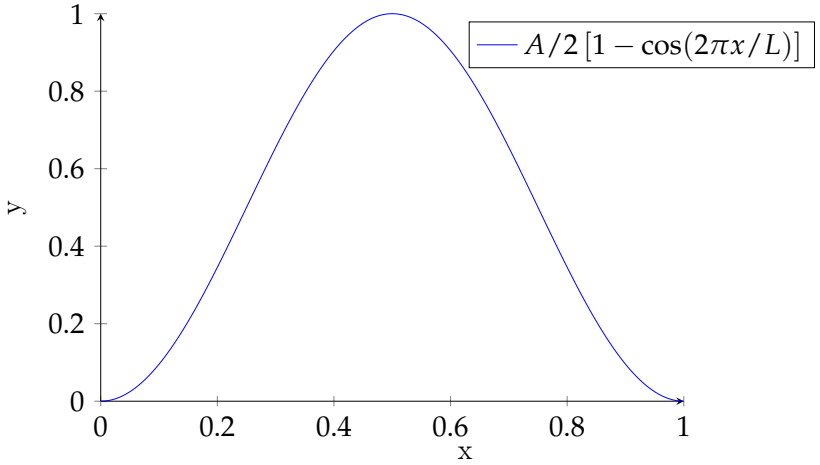

Figure A.1 – Illustration of a sinusoidal half-wrap of amplitude $A = 1$ and length $L = 1$.

The choice of representing our population with sinusoidal evolutions comes from

Fourier who assures that it's possible to represent any function with an infinite number of sinusoids.

A.2 Selection

For each generation (and therefore iteration), 2 individuals ψ_q, ψ_p (with q, $p \in [0, ..., N_{pop}]$) are randomly selected. These two individuals from the population N_{pop} are then compared in the form of a "duel". In the case of minimization, the individual with the highest $\mathcal{J}$ cost function is mutated. The other individual is retained. Here, the cost function $\mathcal{J} = \frac{1}{16}\rho_w g \int_\Omega H^2 d\Omega$ is the wave energy integrated over the whole domain. This selection is represented by Figure I.2.2) and is algorithmically translated by the following pseudo-code 2.

Algorithm 2 Selection and mutation

Input: N_{pop} the number of individuals in the population, $\psi_{i=0,...,N_{pop}}0$ the population, `randint` the function returning a random integer, `calc_J` the calculation of the $\mathcal{J}$ cost function, `mutation` the function performing a mutation.

Output: The new ψ_i population with a mutation.

$p \leftarrow$ `randint`$(0, N_{pop})$ ▷ Random selection of an individual from the population.
$q \leftarrow$ `randint`$(0, N_{pop})$
$\mathcal{J}_p \leftarrow$ `calc_J`(ψ_p) ▷ Calculation of cost function
$\mathcal{J}_q \leftarrow$ `calc_J`(ψ_q)
if $\mathcal{J}_p > \mathcal{J}_q$ **then**
    $\mathcal{J}_p \leftarrow$ `mutation`$(\mathcal{J}_q)$ ▷ Cloning + Mutation
else
    $\mathcal{J}_q \leftarrow$ `mutation`$(\mathcal{J}_p)$
end if

A.3 Mutation

As previously indicated in the 2 algorithm, the individual undergoing mutation becomes an evolution of the individual with the lowest $\mathcal{J}$ cost function. Mutation acts in a very similar way to population creation. One or more of the individual's bumps will be selected. They will then be mutated by changing their parameters (number, amplitude A, position, width L). They can thus be made to move slightly, have a larger (or smaller) amplitude and a slightly different length. Unlike the creation of the population, the parameters are not entirely regenerated; the old parameters evolve by a small amount ε in order to ensure the convergence of the population. This small ε shift can be seen on the I.2.3) where there is not much variability between the two individuals (the old and the

mutated). Moreover, the mutated individual has a slightly lower cost function than the original individual.

A.4 Final Population

By cleverly choosing a convergence criterion, we decide to stop mutations within the population. If the criterion is well chosen, we should obtain a population that has entirely converged at a single point, as shown in figure I.2.4). In our case, the algorithm stops when $\mathcal{J}$ hardly evolves over a certain number of iterations. This gives us the convergence curve figure A.2.

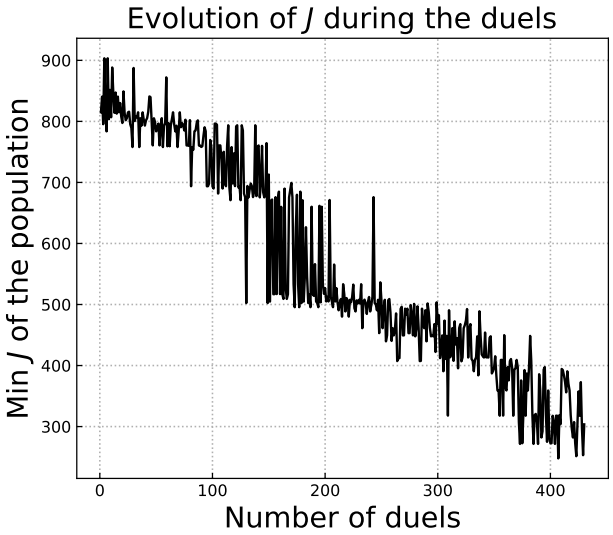

Figure A.2 – Evolution of $\mathcal{J}$ during duels.

Note that the population tends towards a min value. Then, to select the best candidate, we simply select the candidate with the lowest cost function.

B A few Attempts to Improve the Model

Throughout the thesis, we have considered a single $\mathcal{J}$ functional, the wave energy integral. This choice was made deliberately, as we feel that this functional is the most realistic and produces the most realistic results. However, a multitude of different functionals and approaches have been tested in order to improve our model. In this section, we present some of the non-concurrent tests that have been carried out to improve our model. We have essentially focused on two cases: a case with linear bathymetry and an experimental case.

B.1 Reference Cases

In order to test our new functional as well as possible, we will carry out numerous simulations based essentially on the two following test cases. These two cases were presented in Chapter 1, so we know where their limits lie and what results we can expect.

B.1.1 Case 1: Simulation of a One Week Storm on a Linear Beach

This first benchmark simulation is presented in Chapter 3 of Cook (2021). This simulation is described as highly morphogenic in that it simulates a storm over a few days with the following parameters:

Physics	Simulation parameters			Hydrodynamic		Morphodynamic		Domain		
Parameters	Δx	Δt	T_f	H_{max}	T_0	Y	M_{slope}	L	h_0	slope α
Values	1 m	400 s	1 week	2 m	2 s	4.25e-5 $m.s.kg^{-1}$	20%	600 m	7 m	11%

Table B.1 – Parameter of the storm simulation.

which are, the spatial step Δx , the time step Δt, the duration of the simulation T_f, the maximum water height in forcing H_{max}, the wave period T_0, the sand abrasion Y, the maximum sand slope M_{slope}, the length of the simulation domain L, the closing depth h_0. This is represented by the following forcing and domain figure B.1.

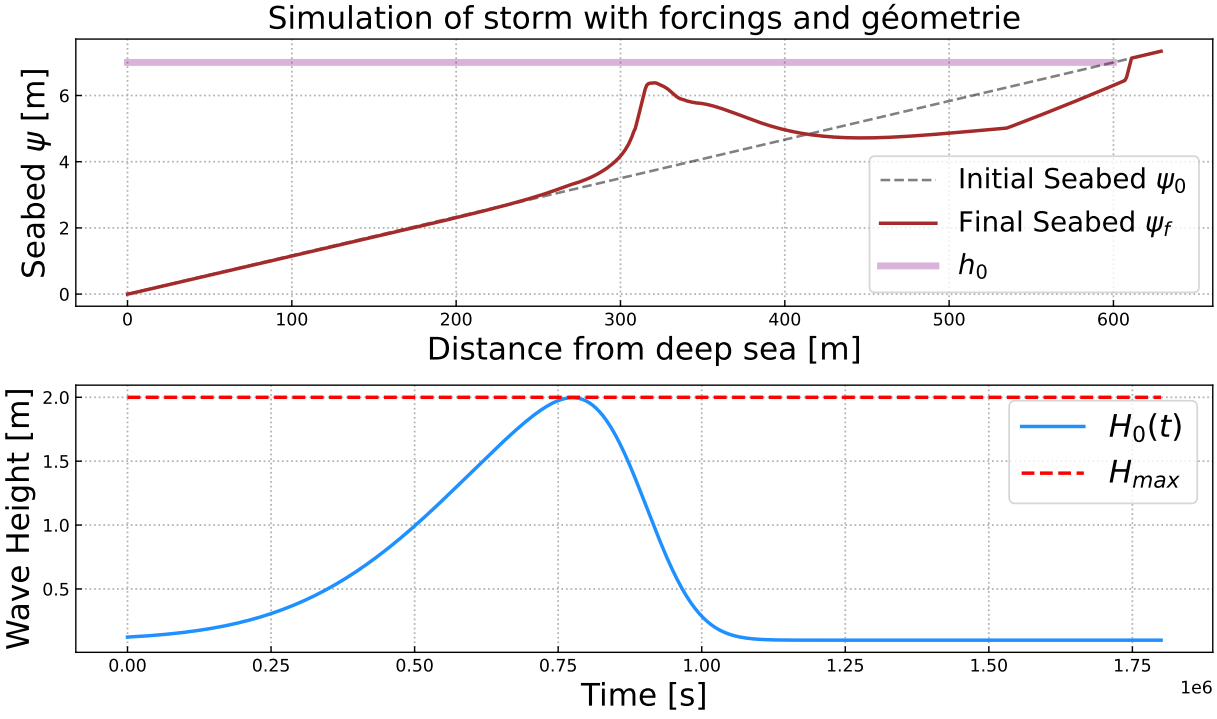

Figure B.1 – Forcing for the one week storm simulation.

The result of this simulation shows the formation of a realistic sand bar. We use this simulation to observe how the model behaves when the physics are changed.

B.1.2 Case 2: Simulation of a Flume Experiment: COPTER

The second reference simulation is present in the article by Cook (2021). This reference case, called COPTER compares simulations with real data from basin tests. Hydro-morphodynamic simulations are carried out using our OptiMorph model and the well know XBeach model. The results of these simulations are shown in Figure B.2.

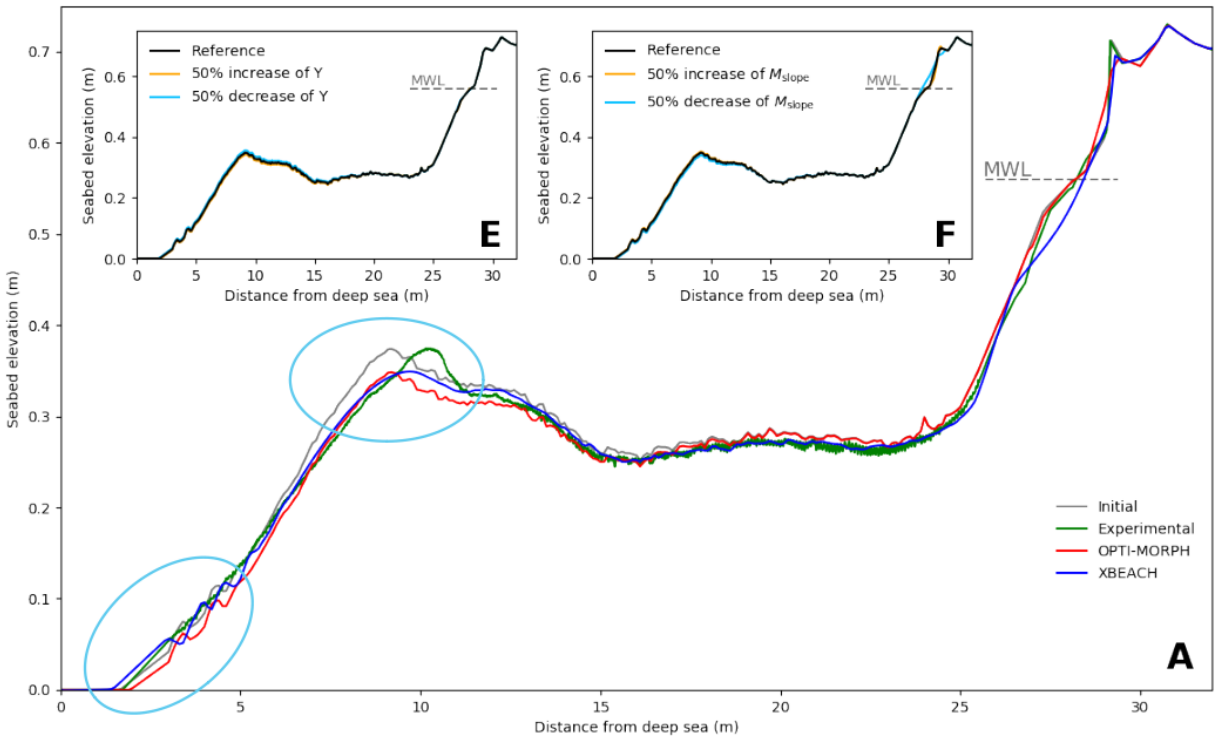

Figure B.2 – Weaknesses (circled in turquoise) of the Copter simulation at the morphodynamic level.

Analysis of these simulations revealed a number of major areas for improvement. These points have been circled in turquoise on the Figure B.2. 1) The movement of sand upstream of the simulation. The sandy movements at this point are too great with the OptiMorph model. 2) The sedimentary bar is supposed to be moving to the coast. However, it is sagging.

For the first point, it is possible that the experimental surveys present too many uncertainties at this level. Indeed, it is very difficult to perceive precisely the morphodynamic displacements at the level of the beater. The second point leads us to believe that some-

thing is missing in our model, such as a current that would allow us to move this bar that we have. The purpose of our functional approach will be to try to physically represent this current or other quantities.

B.2 Improvement by Functional Approach

The first idea that comes to mind improving the model in a rather naive way would be to change what governs the morphodynamics, namely the cost function $\mathcal{J}$. A very large number of functional $\mathcal{J}$ was tested in order to take into account more physics. In the work of Cook et al. (2021c), a number of functional has been tested, as shown in the table B.2.

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	$d = (1-\Lambda)\text{CF2} + \Lambda\text{CF4}$	where Λ is the excitation of the seabed
CF6	$d = (1-\Lambda)\text{CF3} + \Lambda\text{CF4}$	where Λ is the excitation of the seabed

Table B.2 – Old cost functions $\mathcal{J}$.

The results of these tests showed that the most physical and relevant functional was the wave energy functional $\mathcal{J} = \mathcal{E}_\mathcal{H}$. This one showed the most relevant results present in (Cook 2021).

In a similar way, we tried out a large number of functional. Some of the functional we tested are shown in the table B.3.

Keyword	Définition	Commentaire
CF8	$d = -\nabla_\psi \left(\mathcal{E}_S - \varepsilon\rho U_{orb}^2\right) \chi_{\Omega_S}$	Kinetic energy removal
CF9	$d = -\nabla_\psi \left(\rho U_{orb}^2\right) \chi_{\Omega_S}$	Kinetic energy
CF10	$d = -\nabla_\psi \left(\varepsilon C_g H^2\right) \chi_{\Omega_S}$	
CF11	$d = -\nabla_\psi \left(C_g H\right) \chi_{\Omega_S}$	
CF12	$d = -\nabla_\psi \left(\varepsilon (C_g H)^2\right) \chi_{\Omega_S}$	
CF13	$d = -\nabla_\psi \left(S_{xx}\right) \chi_{\Omega_S}$	Radiation stress
CF14	$d = -\nabla_\psi \left(\nabla S_{xx}\right) \chi_{\Omega_S}$	Gradient of adiation stress
CF15	$J = \frac{1}{8}\rho_w g \int_{\Omega_S} H^2 dx + \rho_s g \int_{\Omega_S} (\psi(t) - \psi_0(\tau - t))^2 dx$	Sand displacement memory

Table B.3 – Nouvelles fonctions de coût $\mathcal{J}$.

these functions have been found by relying on basic physics: calculation of forces, work, kinetic energy, ... but also by relying on the balance of moments (Sous et al. 2020):

$$\frac{\partial}{\partial x}\left(\rho U^2\right) = -g\rho\frac{\partial \bar{\eta}}{\partial x} - \frac{1}{(\bar{\eta}+h)}\frac{\partial S_{xx}}{\partial x} - \frac{1}{(\bar{\eta}+h)}\bar{\tau}_b \quad \left[Pa\cdot m^{-1}\right], \tag{B.1}$$

where U is the depth-averaged velocity, g the gravitational acceleration, ρ the water density, $\bar{\eta}$ the wave setup, h the still water depth, S_{xx} the radiation stress, and τ_b the bed shear stress.

B.2.1 Functional with Kinetic Energy Dissipation (CF8)

The first functional aims at introducing a term taking into account the kinetics of a wave. Indeed, if we take into account the kinetics of a wave, it could be that it could artificially simulate a "current" which would allow us to obtain a displacement of the bar.

$$\mathcal{J} = \frac{1}{16}\int_{\Omega_s}(\rho g H^2 - \varepsilon\rho U_{orb}^2)dx \tag{B.2}$$

with ε in m^{-1} that we will fix for the moment arbitrarily. In order to implement this functional, it is necessary to differentiate it according to ψ (calculation of $\nabla_\psi\mathcal{J}$). The term $\mathcal{J}_\mathcal{H}$ has already been differentiated 2.1, it only remains to differentiate the term $\int_{\Omega_s}\varepsilon\rho U_{orb}^2 dx$. This is easily done via the following expression for the orbital velocity:

$$\begin{aligned} U_{orb} : \Omega\times[0,h_0] &\longrightarrow \mathbb{R}^+ \\ (x,\psi) &\longmapsto \frac{\cosh\left(k(x)\left(h(x)-(h_0-\psi)\right)\right)}{\cosh(k(x)h(x))}. \end{aligned} \tag{B.3}$$

We start again from the calculation of the gradient which allows us to obtain the new expression:

$$\nabla_\psi\mathcal{J} = \frac{1}{4}\rho\left(gH\partial_\psi H + \varepsilon\partial_\psi U_{orb}\right) \tag{B.4}$$

with the only term here that we don't know: $\partial_\psi U_{orb}$. With the equation (B.3), we obtain :

$$\boxed{\partial_\psi U_{orb} = \left(\partial_\psi u\, v - u\,\partial_\psi v\right)v^{-2}}$$

with:

$$
\begin{aligned}
u &= \cosh\left(k\left(h-(h_0-\psi)\right)\right) \\
v &= \cosh(kh) \\
\partial_\psi u &= \sinh(k\left(h-(h_0-\psi)\right)\left(h+\psi-h_0\right)k_\psi \\
\partial_\psi v &= \sinh(kh)(k_\psi h-k)
\end{aligned}
$$

which is easy to calculate because we know the values of h_ψ and k_ψ which were specified in the part 2.
We now try to run simulations on Case 1 (B.1.1) and Case 2 (B.1.2) with different values of ε from 0.001 to 0.009. The results are shown in figures B.3 and B.4.

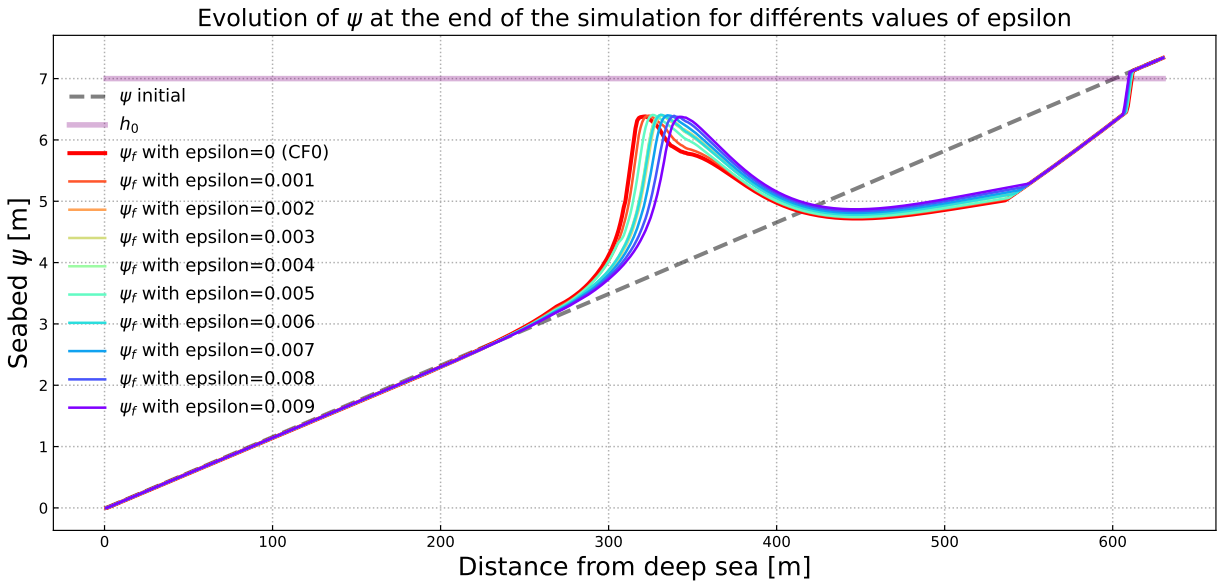

Figure B.3 – Results for different ε with the dissipation functional for simulation B.1.1.

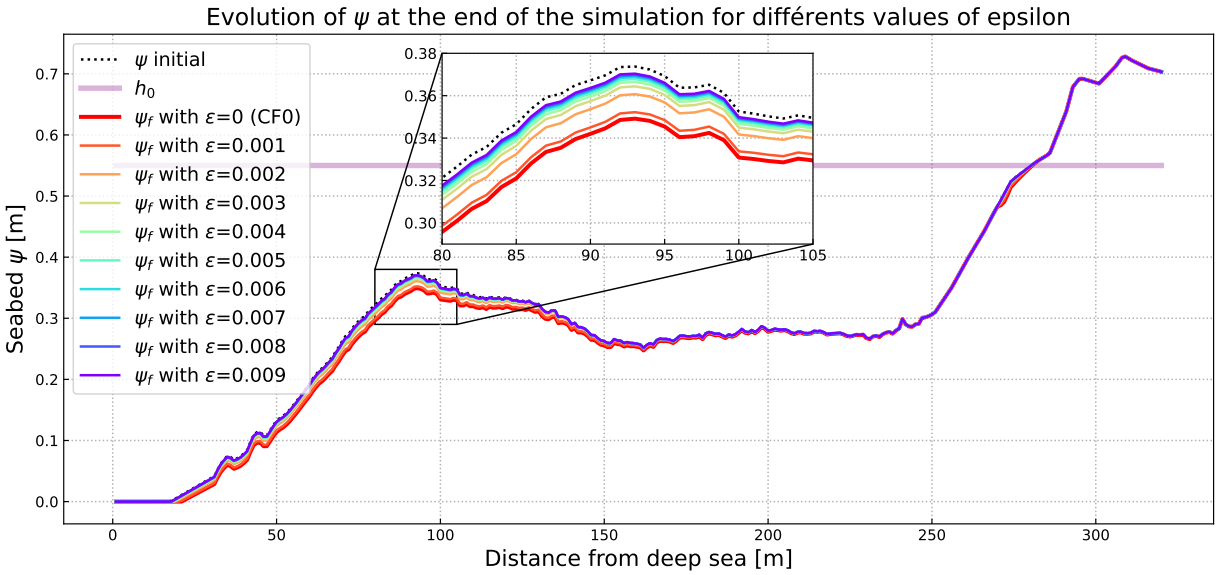

Figure B.4 – Results for different ε with the dissipation functional for simulation B.1.2.

In these cases, the CF0 functional is represented in red. Results from figure B.3 suggest that it is possible to move the bar with this functional. However, looking at the figure B.4 we notice that there is no displacement of the sedimentary bar. In fact, the effect of attenuation is merely to attenuate sediment mobility. The conclusion on this functional from a numerical point of view is that it simply attenuates the descent in the gradient descent method. The results with ε of the first case are very similar to other results of ψ at 90-100% of the simulation time.

To understand what the model does with this functional, it may be interesting to look at the distribution of each term on the domain. That is to say, where does the term $\mathcal{J}_{cinetique}$ acts and where does the term $\mathcal{J}_{\mathcal{H}}$ act. We display for the first case on figure B.5, the rapports $\frac{J_c}{J_{tot}}$ and $\frac{J_{\mathcal{H}}}{J_{tot}}$ at different moments of the simulation: at the beginning, at the peak and at the end.

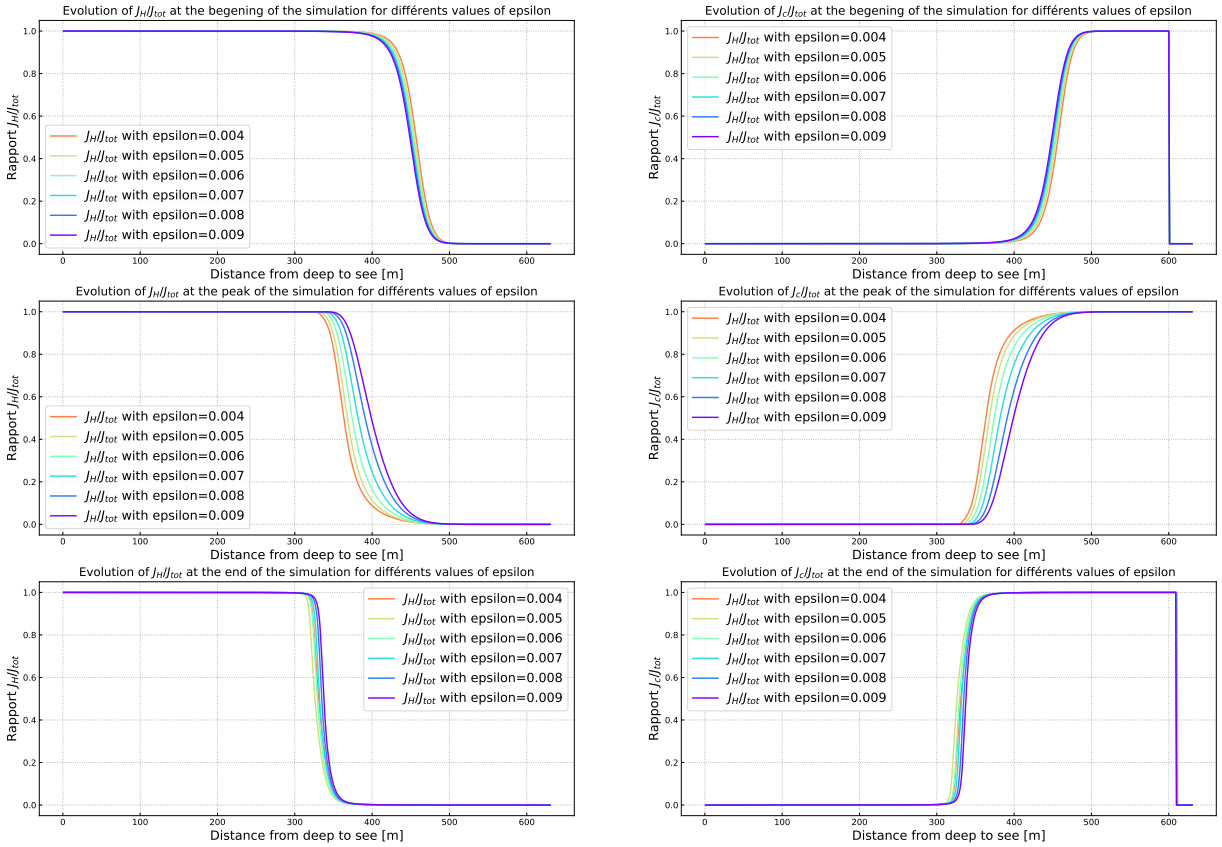

Figure B.5 – Rapports $\frac{J_c}{J_{tot}}$ and $\frac{J_{\mathcal{H}}}{J_{tot}}$ at different times of the simulation: at the beginning, at the peak and at the end: according to different values of ε.

Not surprisingly, we notice that in all cases, the $\mathcal{J}_{\mathcal{H}}$ cost function is predominant in far from the coast and then the $\mathcal{J}_c$ cost function takes over for the break-up. This can be explained because there are changes in velocities when the wave breaks and therefore the gradient is more likely to evolve. Although this work is interesting in understanding the physics behind the model despite this functional does not help us to address the weaknesses of our model. We therefore reject this functional in order to move towards other functions.

B.2.2 Functional in Terms of Representing Work

Other functions were tested in the same way at the B.2.1 part. The idea here is to consider a functional that would represent the notion of work. This approach aims to "artificially" adds the notion of current thank to the group velocity. The functionals tested are the following:

$$\mathcal{J}_{CF10} = \varepsilon C_g H^2 \tag{B.5a}$$
$$\mathcal{J}_{CF11} = C_g H \tag{B.5b}$$
$$\mathcal{J}_{CF12} = \varepsilon (C_g H)^2. \tag{B.5c}$$

The differentiation of these functional has been done in a similar way to the previous part B.2.1. The results coming from these functionals are very similar to those produced by the $\mathcal{J}_{\mathcal{H}}$ functional. These functions don't add significant value to the model, so we've decided not to keep them.

B.2.3 Functional with Radiation Stress S_{xx}

In the momentum balance on a wave (equation (B.1)), there is a radiation stress term. This one represents an excess of flow is present as a result of the orbital movement of a wave. This can be observed on the figure B.6. This excess dissipates mainly on the bottom friction.

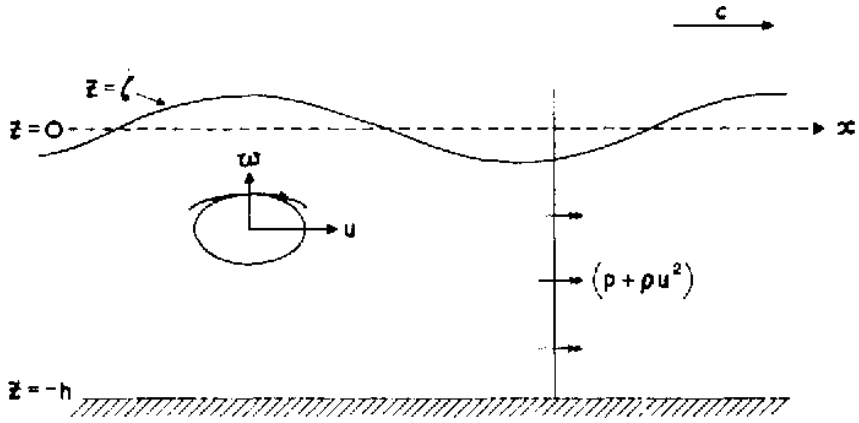

Figure B.6 – Diagram of momentum balance on a wave.

Longuet-Higgins et al. (1962) have sought to quantify this excess flow name the radiation stress S_{xx}. To quantify this variable, the following expression (equation (B.6)) was established.

$$S_{xx} = \overline{\int_{-h}^{\eta} (p + \rho \tilde{u}^2)\ dz} - \int_{-h}^{0} p_0 dz \tag{B.6}$$

The first term of this expression is the total flux of momentum of a wave averaged. It is then subtracted from it the average flow in absence of a wave. This quantity S_{xx} is thus to be seen as the difference between the time-averaged flux of momentum and the average

flux in absence of wave. The work of Longuet-Higgins et al. (1962) was to simplify the expression of S_{xx} by a simpler expression namely:

$$S_{xx} = \left(2\frac{C_g}{C} - \frac{1}{2}\right) E = \frac{1}{8}\rho g \left(\frac{2kh}{\sinh 2kh} + \frac{1}{2}\right) H^2 \quad [J.m^{-1}] \tag{B.7}$$

with C_g the group velocity $[m]$, ρ the water density $[kg.m^{-3}]$, g the gravitational constant $[m.s{-}2]$, H the significant height and E the wave energy. The new approach would be to postulate that *the system will try to minimize its energy in the sense of momentum: minimize the slopes and currents of it.* The idea transmitted through this is to suppose that the system tries to *minimize the mechanisms by which this energy is transmitted to it.* This would mean in our case to reduce the spatial gradient of the radiation stress. To be clearer, we would try to minimize the following functional:

$$\mathcal{J} = \varepsilon \nabla_x (S_{xx}) \quad [J.m^{-2}] \tag{B.8}$$

with ε in $[m]$ chosen arbitrarily. The differentiation of these functions will be done in a similar way to the previous part **B.2.1**. Comparing the results obtained with this functional with the others, we notice strong oscillations on the morphodynamics. This can be explained by the fact that the gradient is calculated numerically: this can induce many numerical biases. Moreover, calculating the gradient will necessarily lead to oscillations since the values of S_{xx} (equation (**B.8**)) are terms in sinh which oscillate a lot. We can try for example to calculate a functional based only the radiation stress of the following form (equation (**B.9**)):

$$\mathcal{J} = S_{xx} \quad [J.m^{-2}]. \tag{B.9}$$

Like the previous part, this functional gives results very close to those obtained by the functional $\mathcal{J}_{\mathcal{H}}$. This is because the expression equation (**B.8**) shows that this is simply a slightly more complex form of energy function. It is therefore normal to obtain results very close to those obtained by the $\mathcal{J}_{\mathcal{H}}$. We also reject these functional. All previous work on functional amounts to performing calculations from the same linear theory of physics with variables often very close to the calculation of $\mathcal{J}_{\mathcal{H}}$.

B.2.4 Functional with Memory Term

A new approach, quite different from the previous ones, would be to incorporate "memory" into the functional. This approach was inspired by (Mohammadi et al. 2014) where we add a constraint on the movement of the sand requiring a minimum of bathymetric changes over the time interval $[t-\tau, t]$ with τ chosen so that $\tau >> T_0$ and T_0 is the wave

period. This gives us the following functional:

$$\mathcal{J} = \frac{1}{8}\rho_w g \int_{\Omega_S} H^2 dx + \rho_s g \int_{\Omega_S} (\psi(t) - \psi_0(\tau - t))^2 dx, \tag{B.10}$$

with ρ_s the density of sand $[kg.m^{-3}]$, ρ_w water density $[kg.m^{-3}]$, H significant water height $[m]$, g the gravitational constant $[m.s^{-2}]$, ψ bathymetry $[m]$ and ψ_0 initial bathymetry $[m]$.
By differentiating the equation (B.10), we obtain:

$$\nabla_\psi J = \nabla_\psi J_H + 2\rho_s g(\psi^n - \psi^{n-1}) \tag{B.11}$$

and so we can expand using the descent equation (5.16) to arrive at:

$$\psi^{n+1} = \psi^n - dt\Lambda\Upsilon\frac{\nabla_\psi \mathcal{J}_\mathcal{H}}{1 - dt\Lambda\Upsilon\rho_s g}, \tag{B.12}$$

in an unconstrained configuration. This equation is very similar to the equation (B.10). The results obtained with this functional are still very similar to those obtained with $\mathcal{J}_\mathcal{H}$. They are almost identical when we use $\tau = dt$ presented in B.1.1. The idea of acting on the numerical scheme can be an interesting approach. We could add more physics, add this notion of transport by adding a transport term in the numerical descent scheme as we will see in the next part.

B.3 Adding Transport in the Descent Scheme

By discretizing the descent equation (5.16), we obtain the following basic descent equation without constraints:

$$\psi_i^{n+1} = \psi_i^n - \Upsilon\Lambda\nabla_\psi \mathcal{J}. \tag{B.13}$$

We can easily add a term representing a horizontal transport according to a speed V $[m.s^{-1}]$ which would transform the equation (B.13) into a new equation:

$$\psi^{n+1} = \psi^n + \Upsilon\Lambda\nabla_\psi \mathcal{J}(\psi^n) - \rho V^n \nabla_x \psi^n. \tag{B.14}$$

This equation can be easily implemented by calculating $\nabla_x \psi^n$ by finite differences. For our tests, we take a constant speed $V = 0.001 m.s^{-1}$ and the orbital velocity $V = U_{orb}$. For the case 1 (B.1.1), we obtain the following final bathymetry results ψ_f on figure B.7.

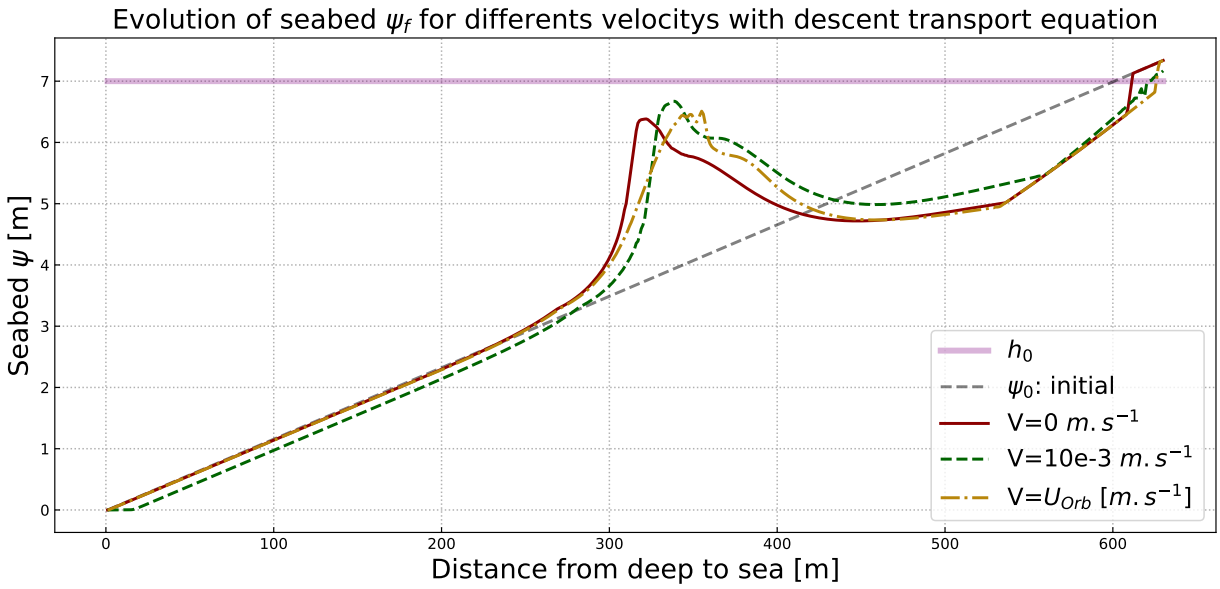

Figure B.7 – Evolution of ψ_f bathymetry for different speeds with the descent transport equation.

The case where $V = 0.001 m.s^{-1}$ (green) shows us that transport works well. However, it makes no physical sense. This is not realistic because we should not have any sand displacement in deep-water. The case where $V = U_{orb}$ seems quite realistic. Moreover, it shows a displacement of the bar without velocity towards the side. This could possibly lead to better results on case 2 B.1.2. Performing analogous simulations, we obtain the following results figure B.8.

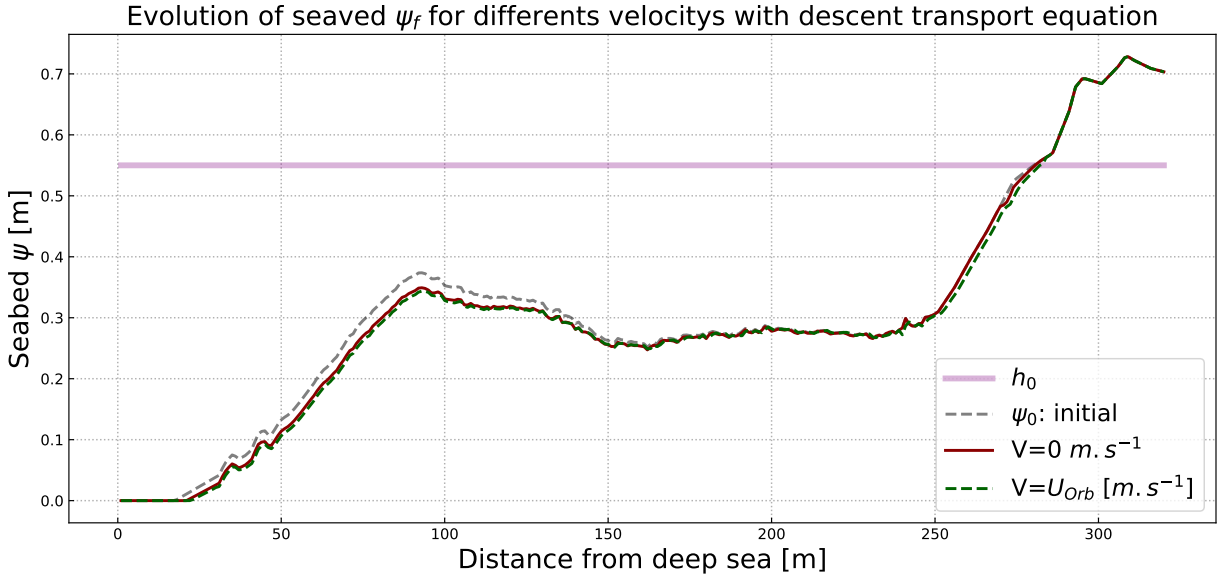

Figure B.8 – Evolution of ψ_f bathymetries for different speeds with the descent transport equation.

In this case, we don't see the displacement of the sedimentary bar we'd hoped for. Although the results are not promising, this approach is still very interesting for defining a current in our model. Assuming we have a wave-to-wave resolution model, we could obtain the actual current u. This would be much more relevant than orbital velocity.

B.4 Conclusion

This part focused on a functional approach to account for a better physics by trying to solve the limitations stated by case 1 (**B.1.1**) and 2 (**B.1.2**). Many other functional have been tested, such as some with currents (from SWAN / XBeach models), bottom stress, ... Some were interesting but none of them was conclusive enough to lift the limits with this functional approach. However, as indicated in the chapter **3** in the section **5.2**, the velocity remains to be defined cleanly in order to be robust on this transport.

C Configuration File of XBeach

params.txt

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% XBeach parameter settings input file                                      %%%
%%%                                                                             %%%
%%%      case DISCOVER1                                                       %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Bed composition parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

rhos        = 2650
D90         = 0.0002

%%% Grid parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

depfile = bathy_psi.dep
posdwn      = -1
nx = 179
ny          = 0
alfa        = 0
vardx       = 1
xfile = x.grd
yfile = y.grd
xori        = 0
yori        = 0
thetamin    = -180
thetamax    = 180
```

```
dtheta      = 360

%%% Model time %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tstop       = 151
CFL         = 0.900000

%%% Morphology parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

morstart    = 0
morfac = 0
%ne_layer = bathy_D1_b.dep

%%% Waves %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

instat = 0
Hrms = 0.42
Trep = 8.0
dir0 = 270

%%% Output %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


tint        = 1
tstart      = 150

nglobalvar = 6
zs
zb
H
k
u
taubx
```

D Configuration File of SWAN

params.txt

```
PROJECT 'maupiti1D' 'HOE'

$ Setting the configuration
$#########################
```

```
SET LEVEL 0.0 NOR 90 INRHOG 1
MODE STATIONARY ONEDIMENSIONAL
COORDINATES CARTESIAN


$ Definition of the grid/ bottom condition
$#########################################

$     REGULAR [xpc] [ypc] [alpc] [xlenc] [ylenc] [mxc] [myc]          [mdc]  [flow]  [
CGRID REGULAR 0 0 0 180 0 179 0 CIRCLE 12 0.02 0.4 36

$ INPGRID BOTTOM REGULAR  [xpinp] [ypinp] [alpinp] [mxinp] [myinp] [dxinp] [dyinp]
INPGRID BOTTOM REGULAR 0 0 0 179 0 1 0


READINP BOTTOM -1 'psi.dat' 3 0 FREE


$ Setting physical quantities for the simulation
$################################################
$ DIFFRACtion 1 0.2
OFF QUAD
GEN3
BREAKING BKD 1 0.73 7.59 -8.06 8.09
TRIAD

$ Definition of forcing conditions
$#################################

BOUND SHAPESPEC JONSWAP 3.30 PEAK DSPR DEGREES

$                              PAR [hs]  [per] [dir] [dd]
BOUNDSPEC SIDE West CON PAR 0.42 8.0 0 20

$ Si on met 20, on réduit le nombre de basse fréquence qui est à Nan -999.

$ Definition of output features
$##############################


CURVE 'profil' 0 0 179 179 0
TABLE 'profil' HEADER 'swan_output_HSIG.dat' HSIGN
TABLE 'profil' HEADER 'swan_output_T0.dat' TM01
```

```
$ Calculating waves
$####################

COMPUTE
STOP
$
$ Fin du calcul
$############################
STOP
```