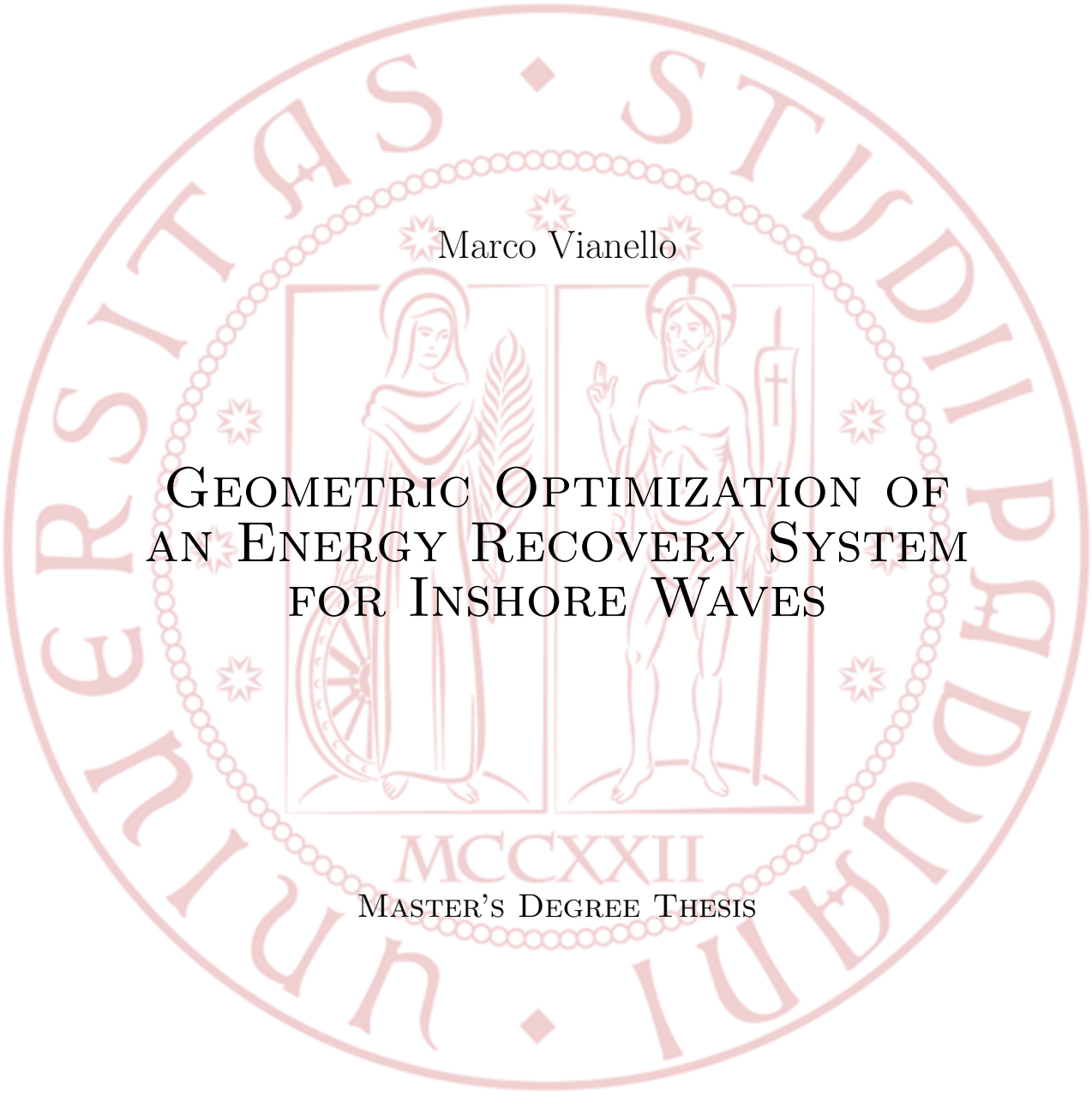


Marco Vianello

Geometric Optimization of an Energy Recovery System for Inshore Waves

Master's Degree Thesis

"If you are out to describe the truth,
leave elegance to the tailor"

Albert Einstein

...A Giada ed
alla mia famiglia...

Contents

Nomenclature

Symbol	Quantity	Unit
a	wave amplitude	m
C	wave celerity	m/s
E_k	kinetic energy	J/m
E_p	potential energy	J/m
F	specific energy flux	W/m
g	gravitational acceleration	m/s^2
h	water depth in quiet conditions	m
h	height of the pipe flow	m
H	wave height	m
H_0	unperturbed wave height	m
k	wave number	m^{-1}
l	depth of the lower end the conversion system	m
L	wave length	m
L_0	unperturbed wave length	m
L_p	inclined plane chord length	m
M_i	number of cells included in the pipe-flow at time i	-
N	number of time-steps	-
P_0	theoretical unperturbed wave power	W/m
$\mathbb{S}$	space of the configuration variables	-
T	wave period	s
u	horizontal component of velocity	m/s
$\mathbf{v}$	velocity vector	m/s
$\mathbf{X}$	design vector	-
w	vertical component of velocity	m/s
δ	adimensional depth of the upper edge of the plane	-
η	profile function of the wave	m
η_b	breaking efficiency of a given configuration	-
ϕ	velocity potential	m^2/s
γ	Iribarren-Battjes empirical parameter	-
Γ_b	breaking index	-
ϑ	average inclination of the plane	rad
ρ	local density	kg/m^3
ω	wave angular frequency	rad/s
Δz_j	height of cell j	m
Δt_i	duration of a time-step	s
$\nabla\cdot$	divergence operator	-
$\nabla\times$	rotor operator	-

Part I

Introduction and theoretical background

Chapter 1

Energy from the sea

1.1 Introduction

Recent instabilities of world economy, due to the increasing price of carbon-derivative fuels along with the connected socio-political turbulences, have aroused the interest in the production of renewable energy among the most industrialised western nations. In this scenario, the sea represents a huge reserve of energy that occurs in a number of different forms.

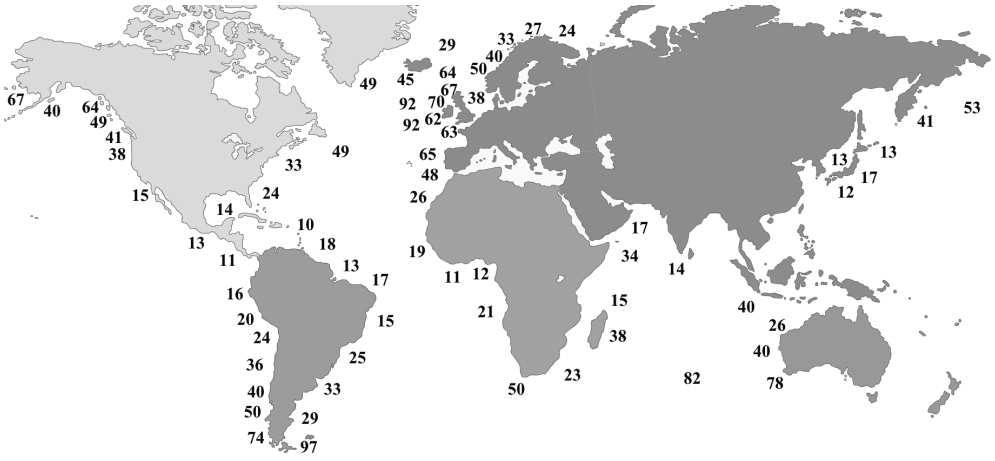


Figure 1.1: Distribution of marine energy over the globe (in kW/m)

Among these, the best known and most freely available are wave energy and tidal energy; some research was also performed by constructing a heat cycle based on geothermal vents, without achieving significant results [1]. National governments as well as private industries are increasing their efforts in order to develop reliable technologies to exploit this great resource. With few exceptions, the actual techniques to extract energy from waves and tides are quite different, reflecting the very different characteristics of these two sources of energy.
Ocean waves arise from the transfer of energy from the sun to wind and, finally, water. Solar energy generates wind that blows over the sea surface, converting wind energy to wave energy, which can travel thousands of miles with little energy loss. Most importantly, waves are a regular source of power, whose intensity can be accurately predicted even several days before their arrival [2].

1.2 Generalities about sea waves

Wave energy is due to the movement of water particles close to the sea surface, being the development of a wave determined by the wind action on water surface. Beneath the free surface, each single water particle makes a circular motion, while energy transmission occurs in the direction of the wave propagation. In the absence of current there is no net water movement associated to energy transportation. This is in sharp contrast with tide energy, were water and energy move together.
When the wave motion occurs offshore, most of the movement of water occurs on site and consists of small oscillations of the individual particles that make only negligible shifts forward. The offshore waves therefore imply a displacement of forms rather than the substance or, in other words, a shift of energy rather than of substance. Only when the wave breaks a significant mass flow is realized.

Sea waves consist on a movement of surface water, mainly due to the action of wind; wind effects can be felt up to a maximum depth of 150 meters. In this case the wave form because the wind pushes the water layer surface, giving some of its energy; the friction between the wind and the water surface moves the surface particles in a circulatory motion.
Size of waves directly depends on the water basin in which they develope and from the source that has generated them. Under normal conditions, offshore waves may reach 6 meters in height in the Mediterranean Sea and 18 meters in the Atlantic Ocean and the Pacific Ocean. However,

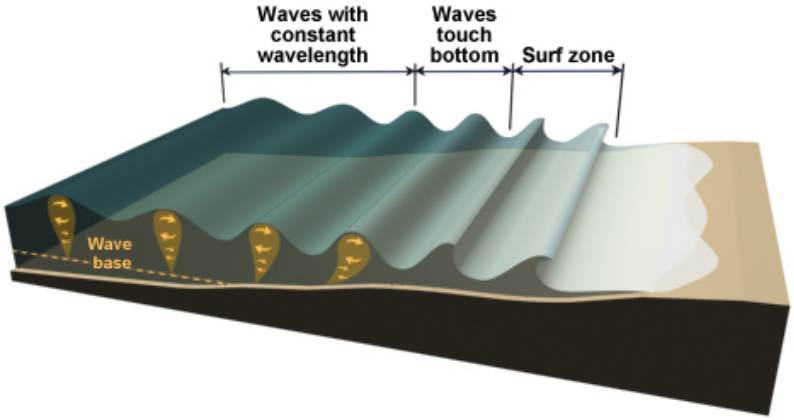


Figure 1.2: Waves and breaking

when a wave hits an obstacle, it may rise and reach greater heights: the highest waves formed in this way can be seen in the storms at north-west of the Cape of Good Hope. The waves motion is classified according to a scale ranging from 1 (corresponding to the calm sea) up to 10 (which indicates a severe storm with waves over 13 meters)

Effects on the coast Waves can have two effects on the coast: their violent action erodes the coast on which they fall or ,in some places where their action is softer, sediments which they lead may cause the advancing of the coast towards the sea.

The wave energy so comes from the movement of water close to the sea surface, on which the action of the wind determines the formation and development of them; as the wind comes from the action of the sun on the atmosphere, waves represent a reserve of solar energy.
As can be seen from Fig 1.2, below the surface in deep water the individual particles of water make circular movements, while the transmission of energy takes place in the direction of propagation of the wave: in the absence of current there is no net movement of water related to the transport of energy.
The phenomenon of the wave breaking, due to the static action of the seabed, water particles acquire a horizontal velocity component, which will be object of study in this thesis, and from which it is intended to obtain mechanical energy.

1.3 Current recovery technologies

The realistically usable worldwide resource has been estimated to be greater than 2 TW. Locations with the most potential for wave power include the western seaboard of Europe, the northern coast of the UK, and the Pacific coastlines of North and South America, Southern Africa, Australia, and New Zealand. The most common categories are shortly described below.

Overtopping converters

These devices [3] are based on the action of waves pushing water up a ramp from which it spills into a basin. In some systems using a flat ramp width constant operating conditions the water may be raised up to 3 m. The water collected in the basin is drained then back into the sea through a turbine (usually a Kaplan type), using a conventional low-dropout hydraulic technology adapted to marine conditions.

The first important development of this category was the Tapchan (Tapered Channel) configuration, designed for coastal use, using a basin on land. The technology was later adapted open sea, using floating docks and ramps.

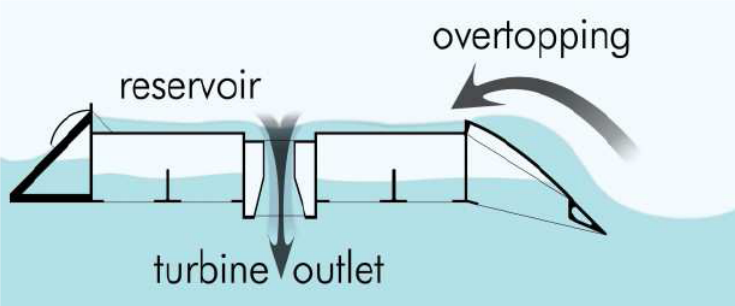


Figure 1.3: Overtopping converter

Articulated rafts.

These devices are based on the relative motion of articulated floating segments [4][5]. A hydraulic system between each pair of segments supplies a hydraulic accumulator, from which the pressurized fluid drives a generator. Articulated rafts are placed perpendicular to the wave front.

Figure 1.4: Articulated rafts

Oscillating water column devices

This is one of the most common device in order to produce energy from waves. An oscillating water column (OWC) includes a partially submerged structure (collector), open to the sea surface. Under the action of waves, the water flows into and out of the structure, compressing the air inside a collector open to the atmosphere, which in turn flows through a turbine that extracts energy and drives a generator.

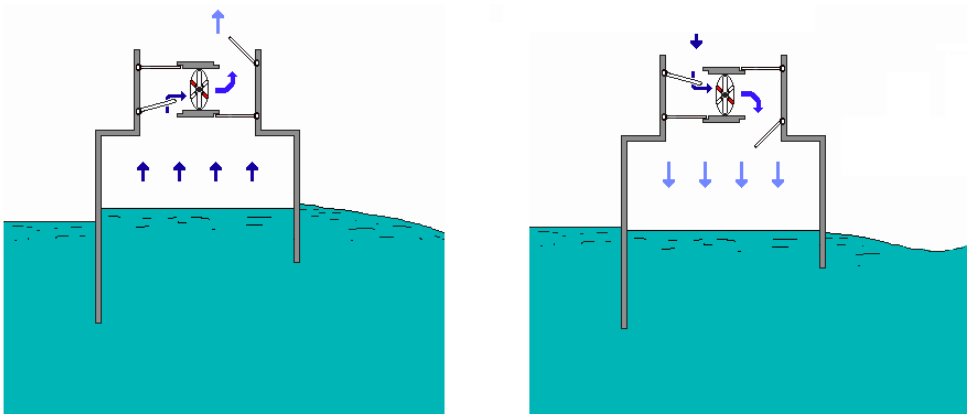


Figure 1.5: Oscillating water column application

The *Wells turbine* is the most used for this application, mainly because of its simplicity and performance. The OWC design is the most mature wave energy collector in terms of number and duration of *in-sea* prototypes tested to date. Research on OWCs started in the 1980s, in conjunction with their installation in countries such as Japan [6]. Ideally, the air chamber dimensions should be designed to maximize energy capture in the local wave climate. On the other hand, it was proved that the generator design is almost completely independent of wave climate, such that only areas of extreme wave energy can marginally benefit from larger generators[7].

Buoys and floaters

The rise and fall of the waves moves a rack and pinion within the buoy and spins a generator. The electricity is transmitted to shore over a submerged transmission line. A 150 kW buoy has a diameter of 36 feet (11 m) and is 145 feet (44 m) tall, with approximately 30 feet of the unit rising above the ocean surface.
Using a three-point mooring system, they are designed to be installed one to five miles (8 km) offshore in water 100 to 200 feet (60 m) deep.

Figure 1.6: PowerBuoy with peak-rated power output of 150 kW.

Chapter 2

Mathematical models for sea waves

2.1 Classification

Ocean waves are a surface movement mainly due to the action of wind, which can be felt in the open sea up to a maximum depth of 150 meters. In this case waves form because the wind pushes the surface water layer, giving some of its energy. The friction between the wind and the water surface moves the surface particles of a circulatory motion.

In view of these considerations, we can report a classification of ocean waves *(Kinsman 1983)*, based on the period T of propagation:

- capillar waves (0-0.1 s)
- ultragravity waves (0.1-1 s)
- gravity waves (1-30 s)
- infragravity waves (30 s - 5 min)
- long period waves (30 min - 24 h)
- tidal waves (24 h)

The first four can be considered caused by the action of wind, while the other from movements of larger scale such as seismic or lunar; waves of wind can be divided into *living waves* and *long waves* depending on the distance of damping.

2.2 Characteristics of a water-wave

Consider a wave motion $\eta(x,t)$ around an average value of h (free surface) describable by a sinusoidal function of the coordinates x and t. A similar wave motion is called *plane wave*, and the general equation which governs the profile is the following:

$$\eta(x,t) = a \cdot \sin(kx - \omega t + \varphi) \tag{2.1}$$

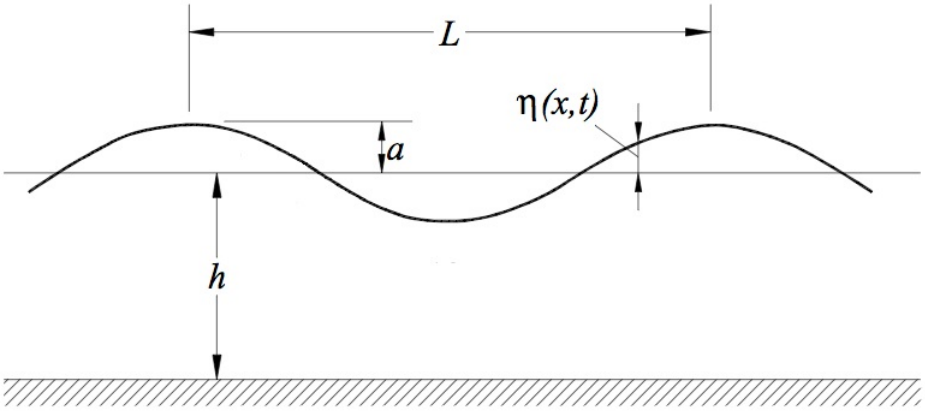


Figure 2.1: Characteristics of a plane wave

We note that a similar phenomenon can be described with a *two-dimensional* model (which assumes unitary y-coordinate) but then results can be adapt to the three-dimensional case through appropriate considerations that will be discussed.

The quantities characterizing the propagation of motion are summarized according to Figure 2.1 and are:

- L : wavelength or spatial period [m]
- a : wave amplitude [m]
- h : average level of the free surface [m]
- $\eta(x,t)$: current deviation from the average level [m]
- T : period [s]

There are also some important parameters that are derived directly from the previous ones:

- C : wave celerity [m·s^{-1}]
- $k = \frac{2\pi}{L}$: wave number [m^{-1}]
- $\omega = \frac{2\pi}{T}$: wave angular frequency [s^{-1}]

2.3 Governing equations

Equations that govern the motion of the fluid inside the test tank are *Navier-Stokes* equations, i.e. the continuity equation (conservation of mass) and the momentum equation (variation of momentum).

Conservation of mass

The velocity vector is in the form $\mathbf{v} = (u, 0, w)$ having regard to the two-dimensionality of the problem, and the balance equation of mass in a volume portion is, in general, given by:

$$\frac{\partial \rho}{\partial t} + \rho(\nabla \cdot \mathbf{v}) + \mathbf{v} \cdot \nabla \rho = 0 \tag{2.2}$$

Assuming density ρ constant in time and uniform in space, the previous equation becomes:

$$\nabla \cdot \mathbf{v} = 0 \tag{2.3}$$

or equivalently, whereas the wave is two-dimensional:

$$\boxed{\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0} \tag{2.4}$$

Variation of momentum

As regards the balance equation of momentum, it presents itself in the most general form as:

$$\rho \frac{d\mathbf{v}}{dt} = \Sigma \left(\mathbf{F}\right)_m + \Sigma \left(\mathbf{F}\right)_{sup} \tag{2.5}$$

in which the terms in the second member respectively indicate the sum of the body forces and surface forces acting on the unit of mass.
The body forces include the contribution of *weight* and *Coriolis force* (negligible if we consider little geographically extended domain).
The surface forces contain the contribution of pressure and that of surface frictions. However, assuming the fluid *non-viscous*, the friction forces *water-bottom*, *water-water* and *air-water* are neglected in the discussion.
The forces acting on a portion of the fluid volume is thus reduced to:

$$\mathbf{F}_{sup} = -\nabla p \quad , \qquad \mathbf{F}_m = (0)\mathbf{i} - (\rho g)\mathbf{k}$$

Explaining the material derivative and projecting (2.5) along the two cartesian axes, having regard to the simplifications made, we obtained

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x} \tag{2.6}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} - g \tag{2.7}$$

By the hypothesis that the vorticity vector is null (plausible hypothesis if the wave does not break) so that the motion can be considered as irrotational, you get an additional condition:

$$\nabla \times \mathbf{v} = 0 \quad \Longrightarrow \quad \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z} = 0 \tag{2.8}$$

Given the irrotational velocity field, we deduce the existence of a scalar potential ϕ, known throughout the domain that describes the kinematic field, such that:

$$\frac{\partial \phi}{\partial x} = u \quad , \quad \frac{\partial \phi}{\partial z} = w \tag{2.9}$$

Depending on the potential of the velocity is then possible to rewrite (2.4) as:

$$\boxed{\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial z^2} = 0} \tag{2.10}$$

or, in compact notation:

$$\nabla^2 \phi = 0$$

As for the equation of motion, can be expressed as a function of the velocity potential, obtaining the known Bernoulli equation:

$$\boxed{\frac{\partial \phi}{\partial t} + \frac{p}{\rho} + gz + \frac{1}{2}\left(\frac{\partial \phi}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial \phi}{\partial z}\right)^2 = const} \tag{2.11}$$

By setting the boundary conditions to the system of differential equations given by (2.10) and (2.11) you can trace the scalar potential ϕ and then know, at each point of the domain and at each time instant, the velocity vector (in magnitude and direction) and the pressure field.

Kinematic boundary conditions

The kinematic conditions pose constraints on the properties of the velocity vector in correspondence of particular geometrical discontinuities of the domain, for example the *seabed* or the *free surface*.
If we assume that the axis x is the same height as the free surface, the free surface motion is described by (2.1) and the points in question are identified by the relation:

$$z = \eta(x,t) \tag{2.12}$$

Consequently, the total derivative with respect to time is expressed by:

$$\frac{d}{dt}\eta(x,t) = \frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} \tag{2.13}$$

and remembering that $\frac{d\eta}{dt} = w = \frac{\partial \phi}{\partial z}$ and $u = \frac{\partial \phi}{\partial x}$ the previous equation, and then the kinematic boundary condition on the free surface, can be rewritten as:

$$\boxed{\frac{\partial \phi}{\partial z} = \frac{\partial \eta}{\partial t} + \frac{\partial \phi}{\partial x}\frac{\partial \eta}{\partial z}} \tag{2.14}$$

While the points in contact with the seabed are characterized by the constant equation:

$$z \equiv -h \tag{2.15}$$

The velocity of these points, due to the impermeability of the seabed, must have only component along x, then:

$$\boxed{(w)_{z=-h} = \left(\frac{\partial \phi}{\partial z}\right)_{z=-h} = 0} \tag{2.16}$$

To these must be added the *wave-inlet* boundary conditions, i.e. those that describe the periodic wave generation. This aspect is dealt with in Chapter 6.

Dynamic boundary conditions

The dynamic boundary conditions set the distribution of pressures acting on a given surface. In particular, on all the free surface points acts the same pressure, equal to atmospheric pressure:

$$(p)_{z=\eta} \equiv p_{atm} \tag{2.17}$$

In view of this, we can rewrite the equation (2.11) as:

$$\frac{\partial \phi}{\partial t} + \frac{p_{atm}}{\rho} + g\eta + \frac{1}{2}\left(\frac{\partial \phi}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial \phi}{\partial z}\right)^2 = const \tag{2.18}$$

and noting that the contribution of the static pressure is constant, it can be brought to second member:

$$\boxed{\frac{\partial \phi}{\partial t} + g\eta + \frac{1}{2}\left(\frac{\partial \phi}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial \phi}{\partial z}\right)^2 = const} \tag{2.19}$$

Find a solution in terms of $phi(x, z, t)$ and $p(x, z, t)$ to the system of differential equations given by (2.10) and (2.11) is not analytically simple because of the non-linearity introduced by the kinetic terms in the second equation: it is therefore necessary to introduce some simplification.

2.4 Linear theory of Airy

Airy has analytically found the potential function $\phi(x, z, t)$ which solves the continuity equation and the motion equation in the hypothesis of linearity, that is when the quadratic terms can be neglected compared to the other; this hypothesis is valid only with the assumption of *small amplitude waves* relative to the length and depth: $\eta(x, t) \ll L, h$.

In view of these considerations, we can simplify the equation of motion in order to make it linear:

$$\begin{cases} \nabla^2 \phi = 0 \\ \dfrac{\partial \phi}{\partial t} + \dfrac{p}{\rho} + gz = const \end{cases} \tag{2.20}$$

The system admits analytical solution, in particular the velocity potential ϕ is given by the following expression:

$$\phi(x, z, t) = -\left(\frac{a \cdot g}{\omega}\right) \frac{\cosh k(h + z)}{\cosh hk} \cos(kx - \omega t) \tag{2.21}$$

From the known potential, the expressions of components of the velocity vector can be directly derived as a function of spatial and temporal parameters:

$$u(x, z, t) = \frac{\partial \phi}{\partial x} = \frac{a \cdot g \cdot k}{\omega} \sin(kx - \omega t) \frac{\cosh k(h + z)}{\cosh hk} \tag{2.22}$$

$$w(x, z, t) = \frac{\partial \phi}{\partial z} = -\frac{a \cdot g \cdot k}{\omega} \cos(kx - \omega t) \frac{\sinh k(h + z)}{\cosh hk} \tag{2.23}$$

It can be shown that, as already assumed in the introduction phase, the wave profile in the case of linear simplification of the motion equations is a harmonic function of the parameters x and t:

$$\eta(x, t) = a \cdot \sin(kx - \omega t) \tag{2.24}$$

2.5 The second-order theory of Stokes

When a two-dimensional wave propagates on a relatively low depth , the surface profile tends to change with respect to the sinusoidal so as to give rise to an increase of the crests and a flattening of the troughs so that the profile is not symmetrical with respect to the level of quietness: in this case is no longer valid the representation by means of the linear theory with decreasing depth. So it is necessary to take into account in the equation of motion of the integrated terms of higher order:

$$\frac{\partial \phi}{\partial t} + \frac{p}{\rho} + gz + \frac{1}{2}\left(\frac{\partial \phi}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial \phi}{\partial z}\right)^2 = const \tag{2.25}$$

Without mentioning the demonstration, the velocity components that are obtained from the derivation of the potential are:

$$u = \frac{agk}{\omega}\left[\frac{\cosh k(d+z)}{\cosh dk}\sin(kx-\omega t) + \frac{3ak\cosh 2k(d+z)}{4\sinh^3 dk\cdot\cosh dk}\sin(2kx-2\omega t)\right] \tag{2.26}$$

$$w = -\frac{agk}{\omega}\left[\frac{\sinh k(d+z)}{\cosh dk}\cos(kx-\omega t) + \frac{3ak\sinh 2k(d+z)}{4\sinh^3 dk\cdot\cosh dk}\cos(2kx-2\omega t)\right] \tag{2.27}$$

Direct application to CFD simulations These last two expressions represent, for each x fixed position, the components of the velocity vector of all the points, from the bottom to the free surface: hence will have fundamental importance in CFD simulations because representing the kinematic boundary conditions that describe the arrival of the wave from the undisturbed marine system into the virtual test tank.

2.6 Wave breaking

2.6.1 Stability and shoaling

When a wave propagates through a decreasing depth seabed, the wavelength decreases and the height increases: this in order to allow the conservation of energy flow per unit of length. The parameters that characterize this phenomenon, called *shoaling*, can be found in:

- H/H_0 : ratio between the local wave height and the unperturbed wave height (for deep seabed) ;
- h/L_0 : ratio between the local depth and the unperturbed wavelength

It is possible to represent the correlation between the two parameters in the Cartesian plane, in order to understand how the amplitude varies as a function of the local depth of the bottom:

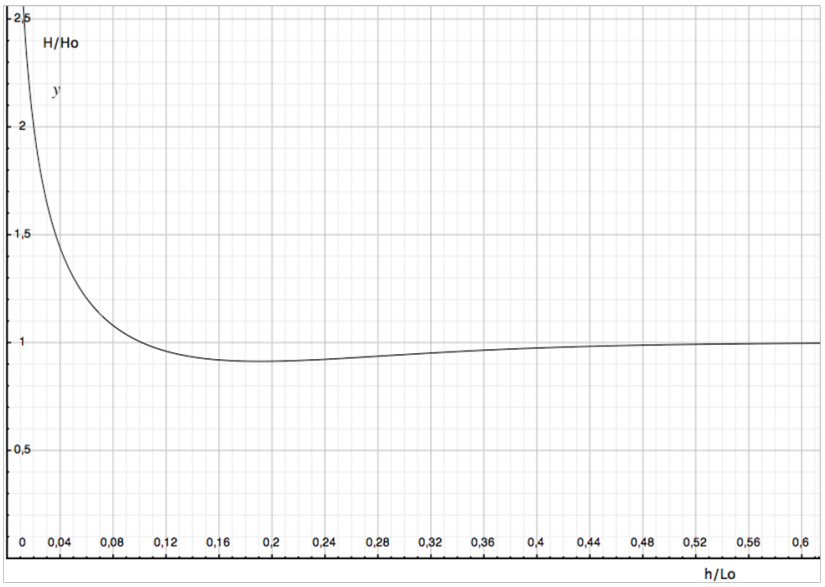


Figure 2.2: Wave height as a function of local depth

The figure shows that for very low depth, the conservation of energy requires that the amplitude increases up to an infinite value for $h \to 0$. On the other hand, if the depth is much greater than the wavelength, it falls assuming wave in deep water, therefore $H = H_0$.

Stability index It is reasonable to assume that the amplitude of a wave can not increase to very high values; intuitively, if the amplitude tends to high values while the distance between two ridges is reduced gradually, the profile of the wave becomes unstable: this context is the phenomenon known as *wave breaking*. The breaking criterion we introduce is the *McCowan* one, that fixes the stability limit condition by the index:

$$\boxed{\Gamma_b = \frac{H}{h} \leq 0,78} \tag{2.28}$$

2.6.2 Types of wave breaking

The wave breaking is a complex phenomenon that gives rise to considerable complications in the analytical description of the fluid motion, having regard to these complications, the approach that is adopted for the study of the problem is empirical.

The way in which the wave breaking occurs are varied and depend on the slope of the seabed and the wave kinematics. The wave breaking figures are therefore different to each other, and depend on the empirical parameter of *Iribarren-Battjes*:

$$\gamma = \frac{\tan \vartheta}{\sqrt{H_0/L_0}} \tag{2.29}$$

where ϑ indicates the slope of the seabed.

In relation to the value of γ parameter, we have three different types of wave breaking, called respectively:

- *spilling breaking* $(\gamma < 0,5)$: turbulence on the crest;
- *plunging breaking* $(0,5 < \gamma < 3,3)$: turbulence in the lower part of the water column;
- *surging breaking* $(\gamma > 3,3)$: turbulence generated by the friction of the bottom;

Spilling breaking

Also called *overthrow breaking*, is characterized by slight slope seabeds, with a unbalanced forward wave and subsequent localized breaking in a strip along the ridge.

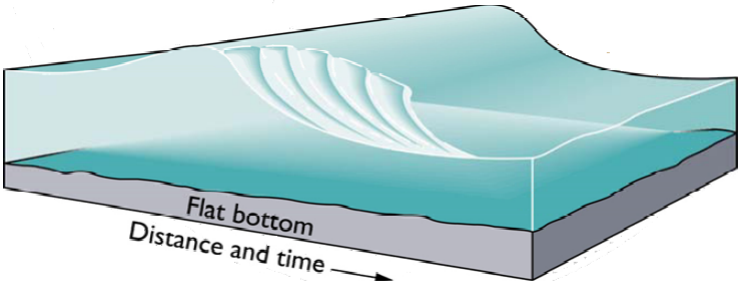


Figure 2.3: Spilling breaking

Generally this type of wave breaking presents foam on the side of the ridge on the side of the propagation of the wave motion.

Plunging breaking

This type of breaking is characterized by average slope seabeds, evident curl in wave crest, plunge forward of the water forming the crest; this gives rise to high horizontal components of velocity, and strong release of kinetic energy.

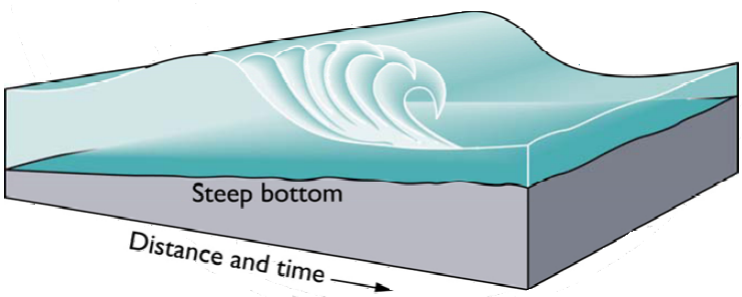


Figure 2.4: Plunging breaking

Surging breaking

Also called *lift breaking*, is characterized by steep seabed which cause a phenomenon of reflection: the wavefront rises suddenly, and collapsing immediately after.

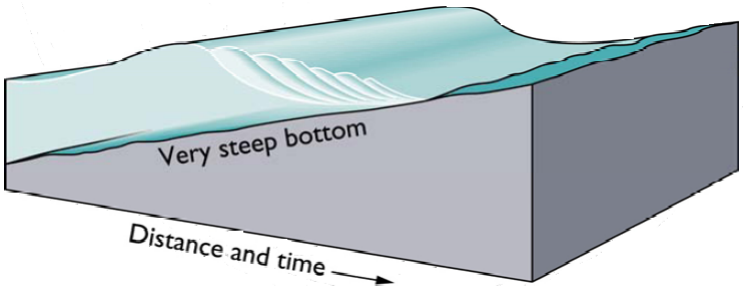


Figure 2.5: Surging breaking

2.7 Energy associated to wave

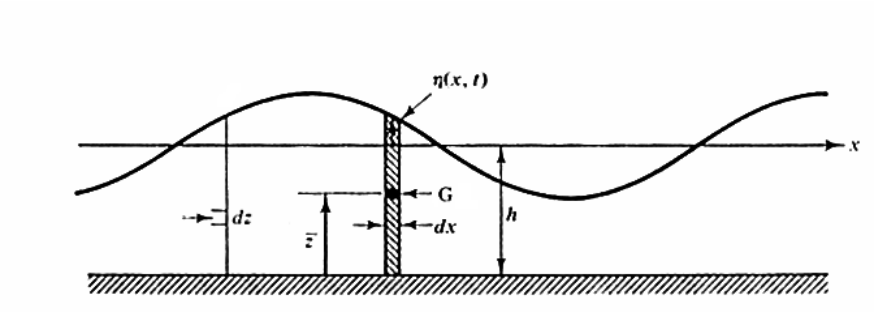
As already mentioned in the introductory part, a unperturbed wave involves the transport, in the direction of the velocity (in our case x) of a quantity of energy. The transmitted energy is composed of:

- *Potential energy*, resulting from the displacement of a mass (water) from the equilibrium position with respect to the gravitational field;
- *Kinetic energy*, due to the magnitude of the velocity vector of the different points of the domain occupied by the fluid;

The determination of the wave energy is of fundamental importance because it provides a term of comparison to estimate the performance of a hypothetical conversion device, or to estimate the power required to a wave-maker for creating a similar wave.

2.7.1 Potential energy

Following the analytical determination of the average potential energy for a wave of type (2.1) is proposed. Consider an infinitesimal element of the fluid column as in figure:



The potential energy relative to that element can be written as:

$$dE_p = dm \cdot g \cdot z_G = \rho g \cdot dx \cdot (h + \eta) \cdot \left(\frac{h + \eta}{2} \right) \tag{2.30}$$

Imposing that the wave profile is described by (2.1) ie a harmonic sinusoidal, we can estimate the potential energy averaged over a period between two crests:

$$\begin{aligned}\overline{E}_p &= \frac{1}{LT}\int d(E_p) = \frac{1}{LT}\int_0^T\int_0^L \rho g\frac{(h+\eta)^2}{2}dxdt = \\ &= \frac{\rho g}{2LT}\int_0^T\int_0^L \left[h + a\sin(kx-\omega t)\right]^2 dxdt = \\ &= \frac{\rho g}{2LT}\int_0^T\int_0^L \left[h^2 + 2ah\sin(kx-\omega t) + a^2\sin^2(kx-\omega t)\right] dxdt\end{aligned}$$

using the subtraction formulas of sine and cosine we can develop the harmonic terms as:

$$\sin(kx-\omega t) = \sin kx\cos\omega t - \cos kx\sin\omega t$$

$$\cos(kx-\omega t) = \cos kx\cos\omega t + \sin kx\sin\omega t$$

and remembering $\sin^2\alpha = \frac{1-cos2\alpha}{2}$ you can rewrite the quadratic term as:

$$\sin^2(kx-\omega t) = \frac{1-\cos(2kx-2\omega t)}{2} = \frac{1-\cos 2kx\cos 2\omega t - \sin 2kx\sin 2\omega t}{2}$$

All products of harmonic functions that result, give zero contribution since the integral is calculated over the spacial period L and the time period T:

$$\iint(\sin x\cos t)\,dxdt = \int_0^L(\sin x)\,dx\cdot\int_0^T(\cos t)\,dt = 0\cdot 0 = 0$$

The final expression that results is therefore:

$$\begin{aligned}\overline{E}_p &= \frac{\rho g}{2LT}\int_0^T\int_0^L\left(h^2+\frac{a^2}{2}\right)dxdt = \\ &= \frac{\rho g}{2LT}\int_0^T\int_0^L\left(h^2+\frac{(H/2)^2}{2}\right)dxdt = \frac{\rho g h^2}{2}+\frac{\rho g H^2}{16}\end{aligned}$$

The latter expression makes evident how, in calculating total potential energy, there are two contributions: the energy due to the wave profile and the potential energy that would have the same mass of water in conditions of quiet, the latter amounting to $\frac{\rho g h^2}{2}$.
If you want to get the part of energy due to wave only, simply subtract the cited term, then:

$$\left(\overline{E}_p\right)_{wave} = \frac{\rho g H^2}{16} \tag{2.31}$$

2.7.2 Kinetic energy

The kinetic energy of the wave motion is due to the sum of the energies possessed individually by each infinitesimal element of the domain, the latter equal to:

$$dE_k = \frac{1}{2} \cdot dm \cdot |\mathbf{v}|^2 = \frac{1}{2} \left(\rho \cdot dx \cdot dz\right) \left(u^2 + w^2\right) \tag{2.32}$$

It should be noted that the two-dimensionality of the problem demands that variables we are examining are *specific quantities*, so refer to $dy = 1$. Imposing that the wave profile in this case also have sinusoidal harmonic character, we can estimate the average kinetic energy between two crests over a period:

$$\overline{E}_k = \frac{1}{LT} \int d(E_c) = \frac{1}{LT} \int_0^T \int_0^L \int_{-h}^{\eta} \rho \frac{u^2 + w^2}{2} dz dx dt$$

Replacing the components u and w with respective expressions as a function of t, z and x and following similar steps to those of the potential energy calculation, we obtain:

$$\left(\overline{E}_k\right)_{wave} = \frac{\rho g H^2}{16} \tag{2.33}$$

As mentioned in the introductory chapter, *specific average total energy* of a wave that propagates in the x direction, is obtained by adding the components of potential energy and kinetic energy specific just obtained:

$$\overline{E} = \overline{E}_p + \overline{E}_k = \frac{\rho g H^2}{8} \tag{2.34}$$

And so the *total energy* held in a progressive wave along a wave front of length l is obtained as:

$$E = \overline{E} \cdot l = \frac{\rho g H^2}{8} \cdot l \tag{2.35}$$

From the energy conversion point of view , the interesting parameter is the power potentially developable by an installation. For this reason it is necessary to know the power of an incident wave; the theoretical *unperturbed wave power* P_O is defined as the wave energy per unit time transmitted in the direction of wave propagation, and can be shown to be:

$$P_0 = \frac{\rho g H^2 L}{16T} \tag{2.36}$$

For finite amplitude waves, or shallow water, the above formula can be adapted as follows:

$$\boxed{P_0 = \frac{\rho g H^2 L}{16T}\left(1 + \frac{2kh}{\sinh(2kh)}\right)} \tag{2.37}$$

This last result is of fundamental importance in this work, as it will be a reference for the energy considerations for the device to be configured.

Chapter 3

Modern methods of optimization

Optimization is the act of obtaining the best result under given circumstances. In design, construction, and maintenance of any engineering system, engineers have to take many technological and managerial decisions at several stages. The ultimate goal of all such decisions is either to minimize the effort required or to maximize the desired benefit.

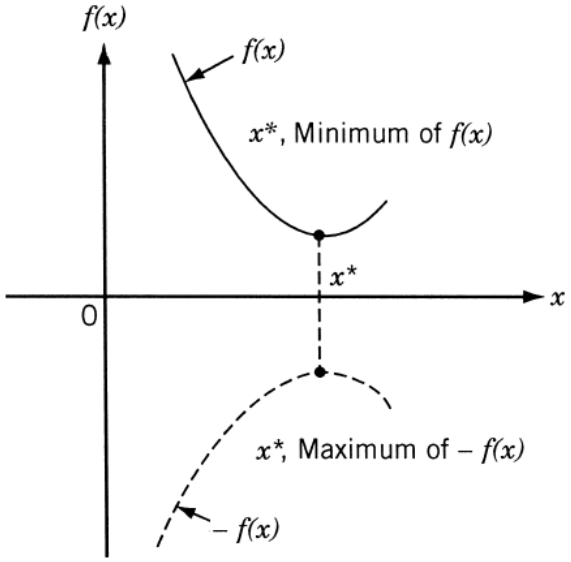


Figure 3.1: Minimum of *f(x)* is same as maximum of *-f(x)*.

Since the effort required or the benefit desired in any practical situation can be expressed as a function of certain decision variables, optimization can be defined as the process of finding the conditions that give the maximum or minimum value of a function.
It can be seen from Fig. 3.1 that if a point $x*$ corresponds to the minimum value of function $f(x)$, the same point also corresponds to the maximum value of the negative of the function, $-f(x)$. Thus without loss of generality, optimization can be taken to mean minimization since the maximum of a function can be found by seeking the minimum of the negative of the same function.

In recent years, some optimization methods that are conceptually different from the traditional mathematical programming techniques have been developed. These methods are labeled as *modern* or *nontraditional* methods of optimization. Most of these methods are based on certain characteristics and behavior of biological, molecular, swarm of insects, and neurobiological systems. The most important methods are:

- Genetic algorithms
- Simulated annealing
- Particle swarm optimization
- Ant colony optimization
- Fuzzy optimization
- Neural-network-based methods

Most of these methods have been developed only in recent years and are emerging as popular methods for the solution of complex engineering problems. Most require only the function values (and not the derivatives).
The genetic algorithms are based on the principles of natural genetics and natural selection. Simulated annealing is based on the simulation of thermal annealing of critically heated solids.

Both genetic algorithms and simulated annealing are stochastic methods that can find the global minimum with a high probability and are naturally applicable for the solution of discrete optimization problems. The particle swarm optimization is based on the behavior of a colony of living things, such as a swarm of insects, a flock of birds, or a school of fish. Ant colony optimization is based on the cooperative behavior of real ant colonies, which are able to find the shortest path from their nest to a food source.

In many practical systems, the objective function, constraints, and the design data are known only in vague and linguistic terms. Fuzzy optimization methods have been developed for solving such problems.
In neural-network-based methods, the problem is modeled as a network consisting of several neurons, and the network is trained suitably to solve the optimization problem efficiently.

3.1 Genetic Algorithms

Many practical optimum design problems are characterized by mixed continuous-discrete variables, and discontinuous and nonconvex design spaces. If standard nonlinear programming techniques are used for this type of problem they will be inefficient, computationally expensive, and, in most cases, find a relative optimum that is closest to the starting point. Genetic algorithms (GAs) are well suited for solving such problems, and in most cases they can find the global optimum solution with a high probability. Although GAs were first presented systematically by Holland, the basic ideas of analysis and design based on the concepts of biological evolution can be found in the work of Rechenberg. Philosophically, GAs are based on Darwin's theory of survival of the fittest. Genetic algorithms are based on the principles of natural genetics and natural selection. The basic elements of natural genetics reproduction, crossover, and mutation are used in the genetic search procedure.

GAs differ from the traditional methods of optimization in the following respects:

- A *population* of points (trial design vectors) is used for starting the procedure instead of a single design point. If the number of design variables is n, usually the size of the population is taken as 2n to 4n. Since several points are used as candidate solutions, GAs are less likely to get trapped at a local optimum.

- GAs use only the values of the objective function. The *derivatives* are not used in the search procedure.

- In GAs the design variables are represented as strings of *binary variables* that correspond to the chromosomes in natural genetics. Thus the search method is naturally applicable for solving discrete and integer programming problems. For continuous design variables, the string length can be varied to achieve any desired resolution.

- The objective function value corresponding to a design vector plays the role of *fitness* in natural genetics.

- In every new generation, a new set of strings is produced by using randomized parents selection and crossover from the old generation (old set of strings). Although randomized, GAs are not simple random search techniques. They efficiently explore the new combinations with the available knowledge to find a new generation with better fitness or objective function value.

3.1.1 Design variables

In GAs, the design variables are represented as strings of binary numbers, 0 and 1. If each design variable $x_i, i = 1, 2, ..., n$ is coded in a string of length q, a design vector is represented using a string of total length nq.

For example, if a string of length 5 is used to represent each variable, a total string of length 20 describes a design vector with $n = 4$. The following string of 20 binary digits denote the vector ($x_1 = 18$, $x_2 = 3$,$x_3 = 1$, $x_4 = 4$):

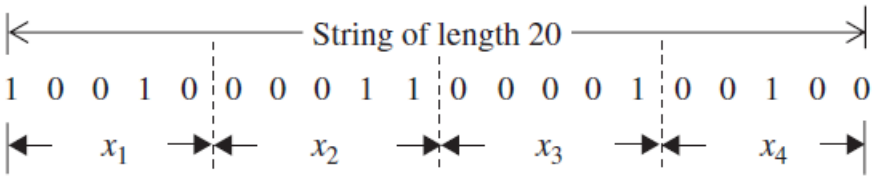


Figure 3.2: Example of codification of four design variables

In general, if a binary number is given by $b_q b_{q-1}...b_2 b_1 b_0$, where $b_k = 0$ or 1, $k = 0, 1, 2, ..., q$, then its equivalent decimal number y (integer) is given by:

$$y = \sum_{k=0}^{q} 2^k b_k \tag{3.1}$$

This indicates that a continuous design variable x can only be represented by a set of discrete values if binary representation is used. If a variable x (whose bounds are given by $x^{(l)}$ and $x^{(u)}$) is represented by a string of q binary numbers, its decimal value can be computed as

$$x = x^{(l)} + \frac{x^{(u)} - x^{(l)}}{2^q - 1} \sum_{k=0}^{q} 2^k b_k \tag{3.2}$$

3.1.2 Objective Function and Constraints

Because genetic algorithms are based on the *survival of the fittest* principle of nature, they try to maximize a function called the fitness function. Thus GAs are naturally suitable for solving unconstrained maximization problems. The fitness function, $F(X)$, can be taken to be same as the objective function $f(X)$ of an unconstrained maximization problem so that $F(X) = f(X)$.

A minimization problem can be transformed into a maximization problem before applying the GAs. Usually the fitness function is chosen to be nonnegative. The commonly used transformation to convert an unconstrained minimization problem to a fitness function is given by:

$$F(\mathbf{X}) = \frac{1}{1 + f(\mathbf{X})} \tag{3.3}$$

It can be seen that the conversion does not alter the location of the minimum of $f(X)$ but converts the minimization problem into an equivalent maximization problem. A general constrained minimization problem can be stated as:

Minimize $f(\boldsymbol{X})$

subject to:

$$g_i(\mathbf{X}) \leq 0 \quad , \quad i = 1, 2, ..., m \tag{3.4}$$

and

$$h_j(\mathbf{X}) = 0 \quad , \quad j = 1, 2, ..., p \tag{3.5}$$

This problem can be converted into an equivalent unconstrained minimization problem by using the concept of penalty function. The fitness function is modified as following:

$$F(\mathbf{X}) = f(\mathbf{X}) + \sum_{i=1}^{m} r_i \cdot \tilde{g}_i(\mathbf{X})^2 + \sum_{j=1}^{p} R_j \cdot h_j(\mathbf{X})^2 \tag{3.6}$$

being

$$\tilde{g}_i(\mathbf{X}) = \begin{cases} g_i(\mathbf{X}), & \text{if } g_i(\mathbf{X}) > 0 \\ 0, & \text{if } g_i(\mathbf{X}) \leq 0 \end{cases} \tag{3.7}$$

In most cases the penalty parameters r_i and R_j associated with the constraints values are usually kept constant throughout the solution process as:

$$r_i = r, \quad i = 1, 2, ..., m \quad \text{and} \quad R_j = R, \quad j = 1, 2, ..., p \tag{3.8}$$

3.1.3 Genetic Operators

The solution of an optimization problem by GAs starts with a population of random strings denoting several design vectors. The population size in GAs (n) is usually fixed. Each string (or design vector) is evaluated to find its fitness value.
The population is operated by three operators (reproduction, crossover, and mutation) to produce a new population of points. The new population is further evaluated to find the fitness values and tested for the convergence of the process.
One cycle of reproduction, crossover, and mutation and the evaluation of the fitness values is known as a generation in GAs. If the convergence criterion is not satisfied, the population is iteratively operated by the three operators and the resulting new population is evaluated for the fitness values. The procedure is continued through several generations until the convergence criterion is satisfied and the process is terminated. The details of the three operations of GAs are given below.

Reproduction

Reproduction is the first operation applied to the population to select good strings of the population to form a mating pool. The *reproduction operator* is also called the *selection operator* because it selects good strings of the population. The reproduction operator is used to pick the fittest strings (above-average) from the current population and insert their multiple copies in the mating pool based on a probabilistic procedure. In a commonly used reproduction operator, a string is selected from the mating pool with a probability proportional to its fitness. Thus if F_i denotes the fitness of the i_{th} string in the population of size n, the probability for selecting the i_{th} string for the mating pool p_i is given by:

$$p_i = \frac{F_i}{\sum_{j=1}^{n} F_j} \; ; \qquad j = 1, 2, ..., n \tag{3.9}$$

The implementation of the selection process given by (3.9) can be understood by imagining a roulette wheel with its circumference divided into segments, one for each string of the population, with the segment lengths proportional to the fitness of the strings as shown in Fig. 3.1. By spinning the roulette wheel n times (n being the population size) and selecting, each time, the string chosen by the roulette-wheel pointer, we obtain a mating pool of size n.

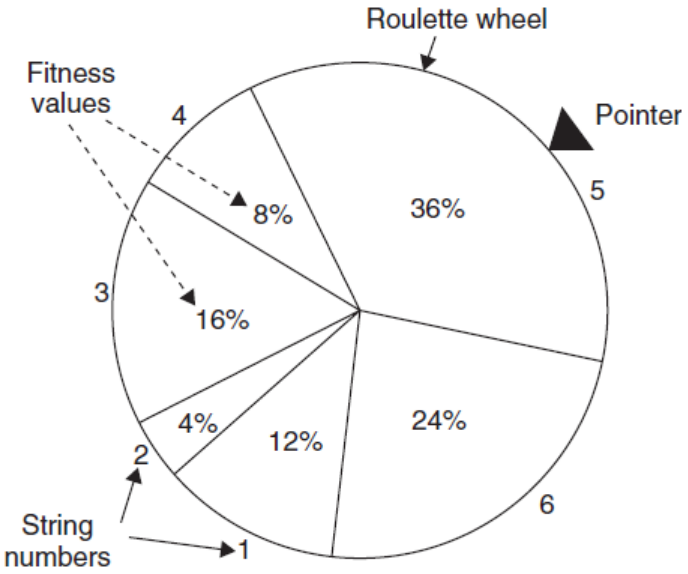


Figure 3.3: Roulette-wheel selection scheme

By this process, the string with a higher (lower) fitness value will be selected more (less) frequently to the mating pool because it has a larger (smaller) range of cumulative probability. Thus strings with high fitness values in the population, probabilistically, get more copies in the mating pool. It is to be noted that no new strings are formed in the reproduction stage; only the existing strings in the population get copied to the mating pool. The reproduction stage ensures that highly fit individuals (strings) live and reproduce, and less fit individuals (strings) die. Thus the GAs simulate the principle of *survival-of-the-fittest* of nature.

Crossover

After reproduction, the crossover operator is implemented. The purpose of crossover is to create new strings by exchanging information among strings of the mating pool.
Many crossover operators have been used in the literature of GAs. In most crossover operators, two individual strings (designs) are picked (or selected) at random from the mating pool generated by the reproduction operator and some portions of the strings are exchanged between the strings. In the commonly used process, known as a *single-point* crossover operator, a crossover site is selected at random along the string length, and the binary digits (alleles) lying on the right side of the crossover site are swapped (exchanged) between the two strings. The two strings selected for participation in the crossover operators are known as *parent strings* and the strings generated by the crossover operator are known as *child strings*. For example, if two design vectors (parents), each with a string length of 10, are given by:

$$\text{(PARENT 1)} \qquad \mathbf{X_1} = [\ 0\ 1\ 0\ |\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ ]$$

$$\text{(PARENT 2)} \qquad \mathbf{X_2} = [\ 1\ 0\ 0\ |\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ ]$$

the result of crossover, when the crossover site is 3, is given by:

$$\text{(OFFSPRING 1)} \qquad \mathbf{X_3} = [\ 0\ 1\ 0\ |\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ ]$$

$$\text{(OFFSPRING 2)} \qquad \mathbf{X_4} = [\ 1\ 0\ 0\ |\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ ]$$

Since the crossover operator combines substrings from parent strings (which have good fitness values), the resulting child strings created are expected to have better fitness values provided an appropriate (suitable) crossover site is selected. However, the suitable or appropriate crossover site is not known before hand. Hence the crossover site is usually chosen randomly.

The child strings generated using a random crossover site may or may not be as good or better than their parent strings in terms of their fitness values. If they are good or better than their parents, they will contribute to a faster improvement of the average fitness value of the new population. On the other hand, if the child strings created are worse than their parent strings, it should not be of much concern to the success of the GAs because the bad child strings will not survive very long as they are less likely to be selected in the next reproduction stage (because of the survival-of-the-fittest strategy used).

As indicated above, the effect of crossover may be useful or detrimental. Hence it is desirable not to use all the strings of the mating pool in crossover but to preserve some of the good strings of the mating pool as part of the population in the next generation. In practice, a crossover probability, pc, is used in selecting the parents for crossover. Thus only 100 p_c percent of the strings in the mating pool will be used in the crossover operator while 100(1 - p_c) percent of the strings will be retained as they are in the new generation (of population).

Mutation

The crossover is the main operator by which new strings with better fitness values are created for the new generations. The *mutation operator* is applied to the new strings with a specific small mutation probability, p_m. The mutation operator changes the binary digit (*allele*'s value) 1 to 0 and vice versa. Several methods can be used for implementing the mutation operator. In the single-point mutation, a mutation site is selected at random along the string length and the binary digit at that site is then changed from 1 to 0 or 0 to 1 with a probability of p_m. Numerically, the process can be implemented as follows: a random number between 0 and 1 is generated; If the random number is smaller than p_m, then the binary digit is changed. Otherwise, the binary digit is not changed.

The purposes of mutation are:

- generating a string (design point) in the neighborhood of the current string, thereby accomplishing a local search around the current solution
- mantaining diversity in the population.

3.1.4 Algorithm

The computational procedure involved in maximizing the fitness function $F(x_1, x_2, x_3, ..., x_n)$ in the genetic algorithm can be described by the following steps.

1. Choose a suitable string length $l = nq$ to represent the n design variables of the design vector $\boldsymbol{X}$. Assume suitable values for the following parameters: population size m, crossover probability p_c, mutation probability p_m, permissible value of standard deviation of fitness values of the population $(s_f)_{max}$ to use as a convergence criterion, and maximum number of generations i_{max} to be used an a second convergence criterion;

2. Generate a random population of size m, each consisting of a string of length $l = nq$. Evaluate the fitness values F_i , $i = 1, 2, ..., m$ of the m strings;

3. Carry out the *reproduction* process;

4. Carry out the *crossover* operation using the crossover probability p_c;

5. Carry out the *mutation* operation using the mutation probability p_m to find the new generation of m strings;

6. Evaluate the fitness values F_i , $i = 1, 2, ..., m$ of the m strings of the new population. Find the standard deviation of the m fitness values;

7. Test for the convergence of the algorithm or process. If $s_f \leq (s_f)_{max}$, the convergence criterion is satisfied and hence the process may be stopped. Otherwise, go to step 8.

8. Test for the generation number. If $i \geq i_{max}$, the computations have been performed for the maximum permissible number of generations and hence the process may be stopped. Otherwise, set the generation number as $i = i + 1$ and go to step 3.

The reproduction operator selects good strings for the mating pool, the crossover operator recombines the substrings of good strings of the mating pool to create strings (next generation of population), and the mutation operator alters the string locally.

The use of these three operators successively yields new generations with improved values of average fitness of the population. Although, the improvement of the fitness of the strings in successive generations cannot be proved mathematically, the process has been found to converge to the optimum fitness value of the objective function. Note that if any bad strings are created at any stage in the process, they will be eliminated by the reproduction operator in the next generation. The GAs have been successfully used to solve a variety of optimization problems in the literature.

3.2 Fuzzy-logic theory

In traditional designs, the optimization problem is stated in precise mathematical terms. However, in many real-world problems, the design data, objective function, and constraints are stated in vague and linguistic terms. For example, the statement, *"This beam carries a load of 1000 lb with a probability of 0.8"* is imprecise because of randomness in the material properties of the beam. On the other hand, the statement, *"This beam carries a large load"* is imprecise because of the fuzzy meaning of *"large load"*.
Similarly, in the optimum design of a machine component, the induced stress (σ) is constrained by an upper bound value (σ_{max}) as ($\sigma < \sigma_{max}$). If (σ_{max}) = $30psi$, it implies that a design with (σ) = $30.000psi$ is acceptable whereas a design with (σ) = $30.001psi$ is not acceptable. However, there is no substantive difference between thw two designs. It appears that it is more reasonable to have a transition stage from absolute permission to absolute impermission. This implies that the constraint is to be stated in fuzzy terms. *Fuzzy theories* can be used to model and design systems involving vague and imprecise information.

Representation of fuzzy sets

Let X be a classical crisp set of objects, called the *universe*, whose generic elements are denoted by x. Membership in a classical subset A of X can be viewed as a characteristic function μ_A from X to [0,1] such that:

$$\mu_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases} \tag{3.10}$$

The set [0, 1] is called a valuation set. A set A is called a *fuzzy set* if the valuation set is allowed to be the whole interval [0, 1].

The fuzzy set A is characterized by the set of all pairs of points denoted as:

$$A = x, \mu_A(x) \tag{3.11}$$

where $\mu_A(x)$ is called the *membership function* of x in A. The closer the value of $\mu_A(x)$ is to 1, the more x belongs to A. The basic crisp set operations of union, intersection, and complement can be represented on Venn diagrams as shown in Fig. 3.4 Similar operations can be defined

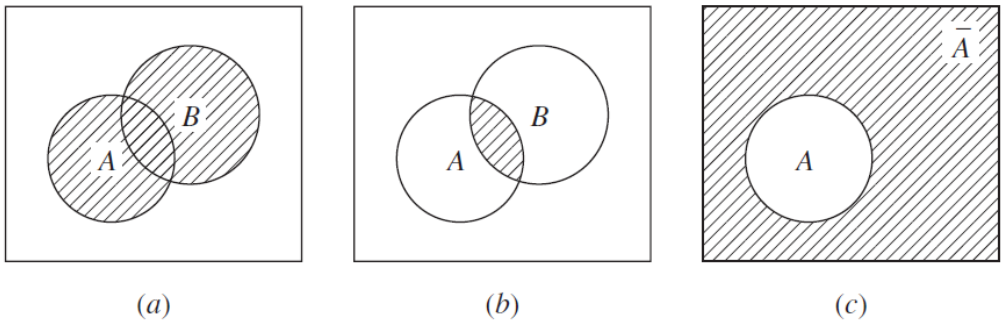


Figure 3.4: Basic set operations in *crisp set* theory: (a) A or B or both: $A \cup B$; (b) A and B: $A \cap B$; (c) not A: $\overline{A}$

for fuzzy sets, noting that the sets A and B do not have clear boundaries in this case. The graphs of μ_B and μ_B can be used to define the set-theoretic operations of fuzzy sets.
The union of the fuzzy sets A and B is defined as:

$$\mu_{A\cup B}(y) = \max[\mu_A(y); \mu_B(y)] = \begin{cases} \mu_A(y), & \text{if } \mu_A(y) > \mu_B(y) \\ \mu_B(y), & \text{if } \mu_A(y) < \mu_B(y) \end{cases} \tag{3.12}$$

The intersection of the fuzzy sets A and B is similarly defined as:

$$\mu_{A\cap B}(y) = \min[\mu_A(y); \mu_B(y)] = \begin{cases} \mu_A(y), & \text{if } \mu_A(y) < \mu_B(y) \\ \mu_B(y), & \text{if } \mu_A(y) > \mu_B(y) \end{cases} \tag{3.13}$$

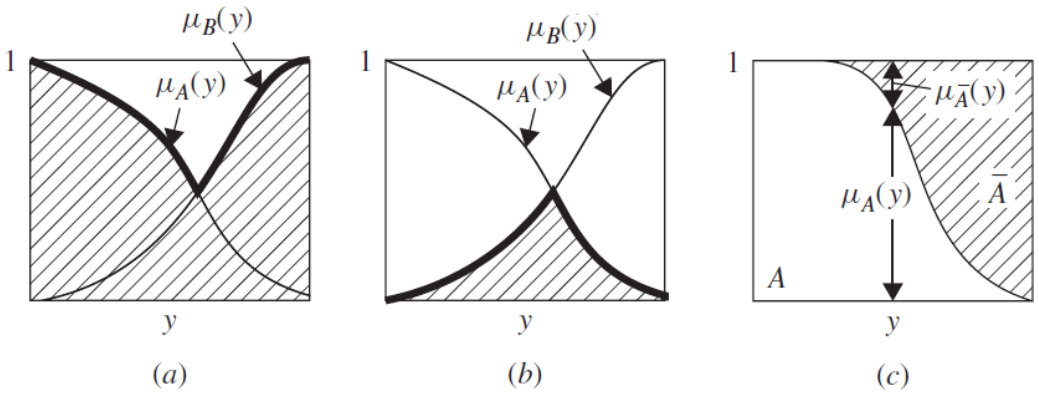


Figure 3.5: Basic set operations in *fuzzy set* theory: (a) union (b) intersection (c) complement

Optimization of fuzzy systems

The conventional optimization methods deal with selection of the design variables that optimizes an objective function subject to the satisfaction of the stated constraints. For a fuzzy system, this notion of optimization has to be revised. Since the objective and constraint functions are characterized by the membership functions in a fuzzy system, a design (decision) can be viewed as the intersection of the fuzzy objective and constraint functions. The conventional optimization problem was stated as follows:

$$\text{Find } \mathbf{X} \text{ which minimizes } f(\mathbf{X})$$

subject to

$$g_j^{(l)} \leq g_j(\mathbf{X}) \leq g_j^{(u)}, \qquad j = 1, 2, ..., m \tag{3.14}$$

While the optimization problem of a fuzzy system is stated as follows:

$$\text{Find } \mathbf{X} \text{ which minimizes } f(\mathbf{X})$$

subject to

$$g_j(\mathbf{X}) \in G_j, \qquad j = 1, 2, ..., m \tag{3.15}$$

where G_j denotes the fuzzy interval to which the function $g_j(\mathbf{X})$ should belong. Thus the fuzzy feasible region S, which denotes the intersection of all G_j , is defined by the membership function:

$$\mu_S(\mathbf{X}) = \min[\mu_{G_j}(\mathbf{X})] \tag{3.16}$$

Since a design vector $\boldsymbol{X}$ is considered feasible when $\mu_S(\mathbf{X}) > 0$, the optimum design is characterized by the maximum value of the intersection of the objective function and the feasible domain:

$$\mu_D(\mathbf{X}^*) = \max\left[\mu_D(\mathbf{X}^*)\right] \mathbf{X}* \in D \tag{3.17}$$

being

$$\mu_D(\mathbf{X}^*) = \min\ \left[\mu_f(\mathbf{X}); \min\ {}_j\mu_{G_j}(\mathbf{X})\right] \tag{3.18}$$

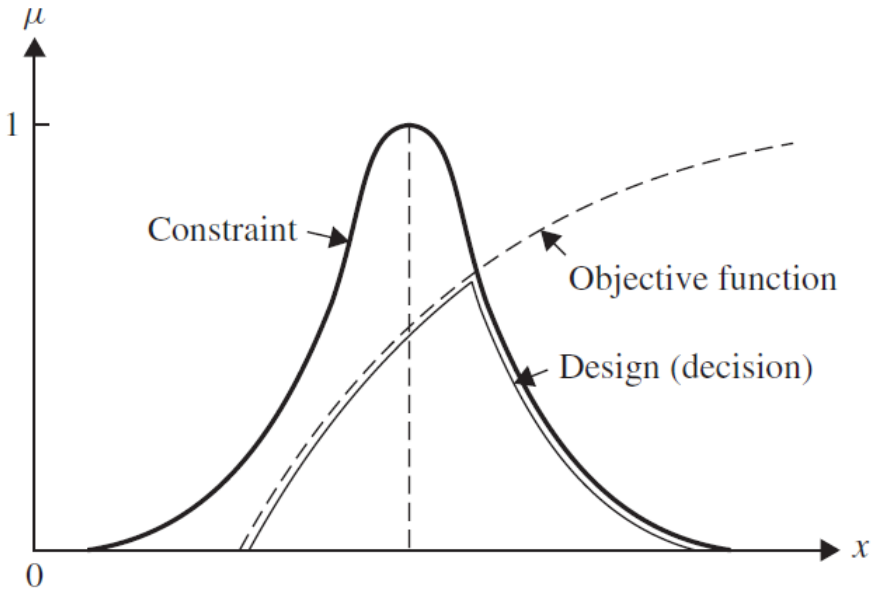


Figure 3.6: Concept of fuzzy decision

Part II

Optimization of breaking performances

Chapter 4

Numerical model

4.1 Representation of the test-tank

As shown in Fig 4.1, a numerical wave tank is modelled to represent the test facility of the Department of Naval Architecture, Ocean and Environmental Engineering of the University of Trieste (Italy), where the proposed numerical analyses are planned to be experimentally reproduced in the prosecution of the hereby presented investigation.

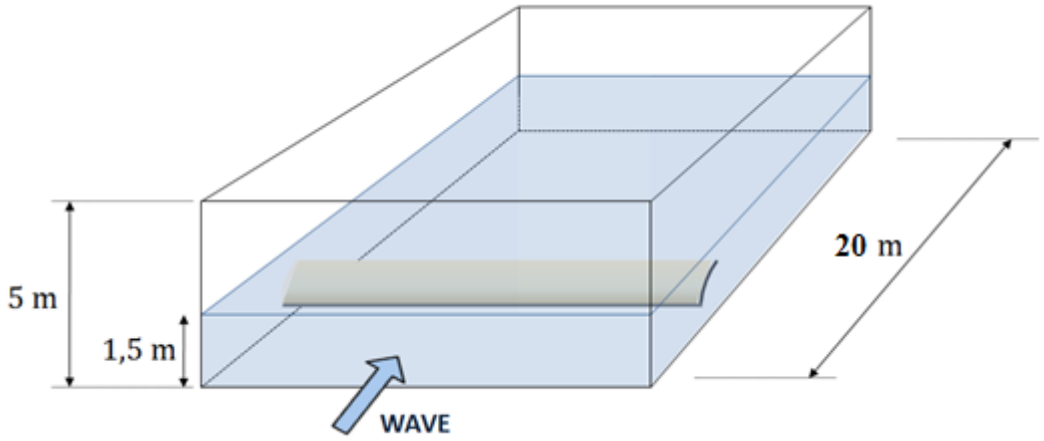


Figure 4.1: Schematic representation of the numerical wave tank

4.2 Meshing of the geometry

The calculation grid has been modeled as a compromise between good accuracy in the description of the physical phenomenon of the breaking wave and reasonable computation-time of the CFD simulation, so that it can be inserted in the optimization loop for the calculation of the objective function.
In this regard, after previous experiments and validations [1] , [11] the mesh has thickened in correspondence of the following points:

- *velocity inlet* : this edge is the critical area in which conditions on velocity are imposed and allow the waves to propagate;
- *free surface*: the multiphase model used in this work (VOF) loses accuracy at the interface between the two phases if the grid is not sufficiently thick in the border zone. Therefore, it is necessary to thicken the mesh at the free surface;
- *inclined plane*: this area requires a greater accuracy in the discretization because the wave motion becomes irregular and potentially turbulent.

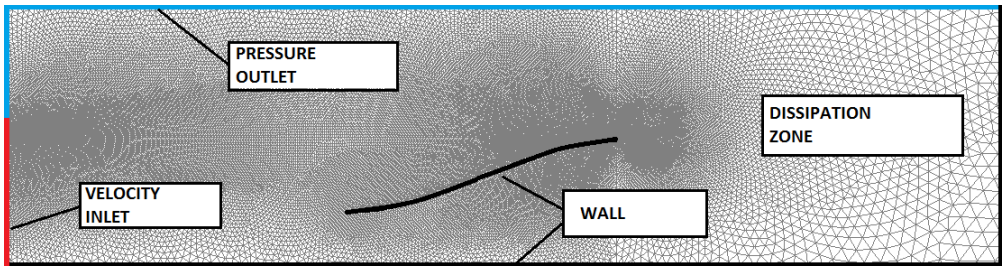


Figure 4.2: Numerical model of the phisical geometry

Downstream of the plane the grid was thinned out so to allow the waves to dissipate [12] and avoid phenomena of wave reflection that would distort the results.

4.3 Settings of CFD simulations

4.3.1 Multi-phase model

In computational fluid dynamics, the *volume of fluid method* (or in short *VOF* method) is a numerical technique for tracking and locating the free surface (or fluid-fluid interface). It belongs to the class of Eulerian methods which are characterized by a mesh that is either stationary or is moving in a certain prescribed manner to accommodate the evolving shape of the interface.
The method is based on the idea of so-called *volume fraction* function Φ that indicates the phase distribution in a volume control (cell).
So *VOF* allows to use static grids, but requires a fine discretization at the interface for well-describing the phase border zone.

4.3.2 Governing equations

A *laminar* fluid model is adopted for the numerical solution of the problem. In this regard, Lal and Elangovan [6] compared the characteristic values of a sea wave using different viscous models (laminar, $k - \epsilon$ and SST), without registering any significative difference.
This, combined with the agility and speed of calculation of the laminar model, has led to its adoption for the present numerical computations. As already proposed in [4] and [7], the solver solution controls adopted in the simulations are the body force weighted pressure discretization with the *PISO* pressure velocity coupling. The momentum transport equation discretization is implemented with the *MUSCL third order* and the modified high resolution interface capturing *(HRIC)* options. Both the MUSCL third order and the HRIC options provide increased solution power for breaking waves and other complex multiphase problems. The time step of the simulation is set at $5 \cdot 10^{-3}$ s. Finer time discretizations did not provide significant variations in the numerical results. Residuals convergence criterion for each physical time step is set to 10^{-4}.

4.4 Generation of wave motion

With respect to Figure 3, wave trains are generated at the left boundary (were a *velocity inlet* boundary condition is settled), by imposing a time-dependent variation to the two components (horizontal and vertical) of the water velocity (set initially quiet inside the whole computational domain), according to (2.26) and (2.27).

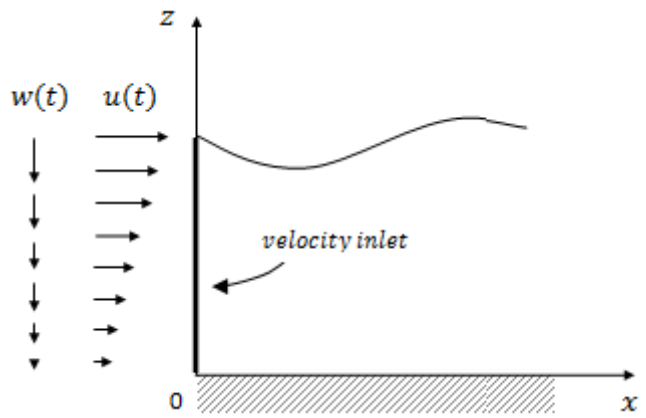


Figure 4.3: Kinematic boundary condition on velocity components to generate the wave

The following table summarizes the main features of the computed wave trains.

Symbol	Value	Unit
L (*)	3	m
a (*)	7.5	cm
T (*)	1.386	s
h (*)	1.5	m
k (**)	2.09	1/m
ω (**)	4.53	rad/s

Table 4.1: Wave train characteristics adopted for the proposed numerical simulations; (*) independent parameters, (**) derived parameters

4.5 Model validation

In order to validate the numerical model, after generating the wave train by imposing $u(t)$ and $w(t)$ at the left boundary of the computational domain, the horizontal component of water velocity is evaluated at a fixed reference section ($x = 4$ m) from the inlet, after a sufficient time span ($t = 10$ s) in order to reach a periodic-state of the flow field, as summarized in the following figure.

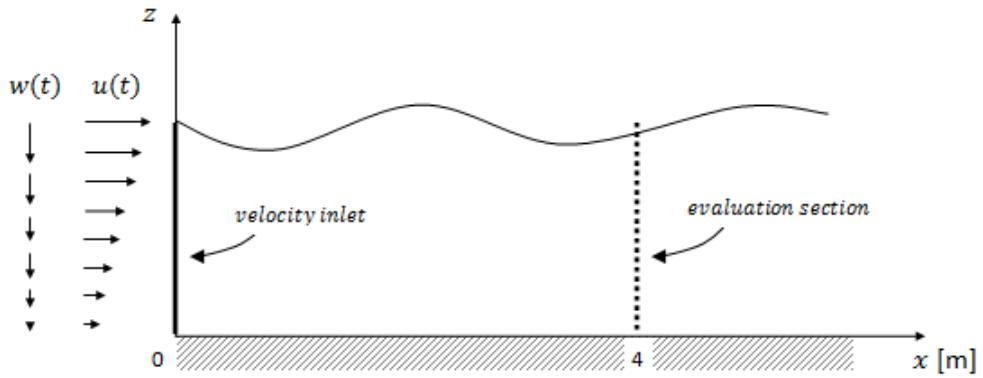


Figure 4.4: Reference schema for the validation of the numerical model

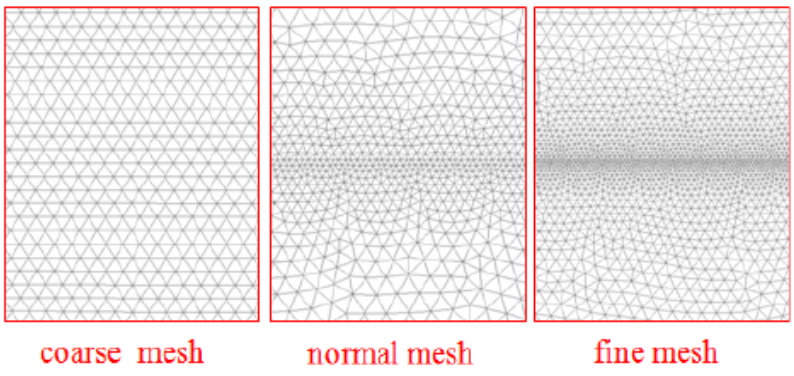


Figure 4.5: Close-up on the free surface for the three adopted spatial domain discretizations

Numerical predictions of *u(t)* are compared with the theoretical values given by (2.26) for three tested spatial domain distributions, whose main geometrical features are summarized in Table 4.2.

Case	Type	Number of Elements
Case 1	Coarse	448116
Case 2	Normal	545174
Case 3	Fine	765212

Table 4.2: Characteristic sizes of the three grid configurations

A triangular mesh with a growth rate equal to 1.15 linking a minimum cell size of 10 mm (close to the free surface) to a maximum dimension of 50 mm is chosen, since finer grid discretizations do not provide significant variations in the numerical results, as can be seen from the figure:

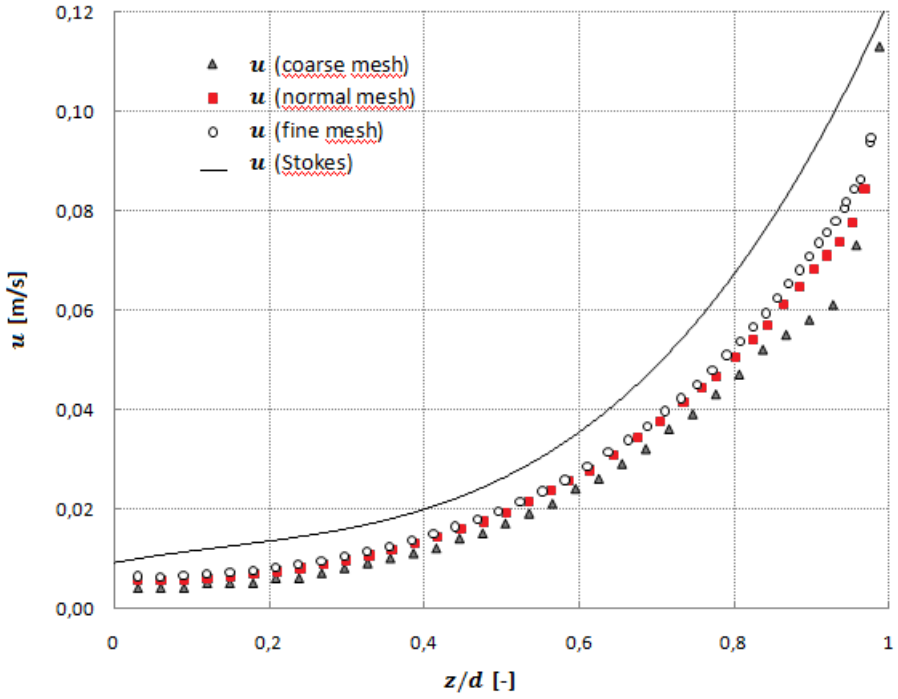


Figure 4.6: Validation of the numerical model based on x-velocities

4.6 Evaluation of the available energy flux

For every analyzed individual j, the breaking efficiency [1] was defined as:

$$(\eta_b)_j = \frac{\overline{F}_j}{P_0} \tag{4.1}$$

where P_0 represents the *theoretical unperturbed wave power* given by eq. (2.37) and $\overline{F}_j$ represents the specific flux of kinetic energy averaged over a period. The computation of $\overline{F}_j$ is described below. The elementary

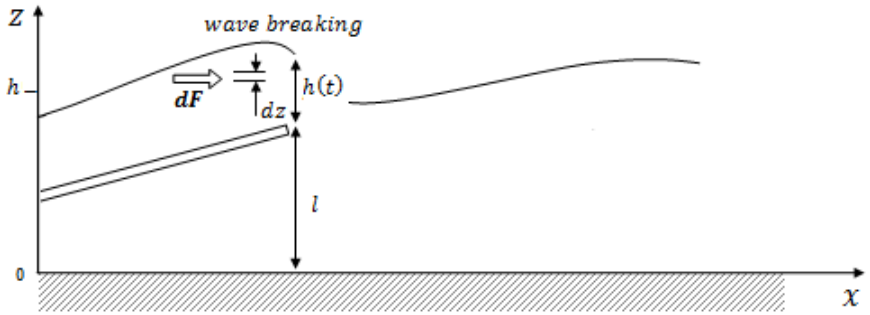


Figure 4.7: Evaluation of the energy flux F across a vertical section

contribute dF of kinetic energy flux can be written as:

$$dF = d\dot{E}_k = \frac{1}{2}\dot{m} \cdot u^2 = \frac{1}{2}\rho \cdot u^3 \cdot dz \tag{4.2}$$

The *total specific energy flux* which interests the general vertical section at time t can be evaluated by integrating the elementary contribute dF over $h(t)$, in formulas:

$$F(t) = \int_l^{l+h(t)} \frac{\rho \cdot u^3}{2} dz \tag{4.3}$$

[1] This definition of efficiency ignores the content of kinetic energy due to the velocity component along the y-axis, which is considered unlikely to be converted into mechanical energy, because too fluctuating.

The *total average specific energy flux* which interests the general vertical section can be evaluated as:

$$\overline{F} = \frac{1}{T}\int_0^T \int_l^{l+h(t)} \frac{\rho \cdot u^3}{2} dz\, dt \tag{4.4}$$

Considering the spatial and temporal discretization adopted in the numerical simulations, (4.4) can be approximated by the following expression, in which integrals are replaced by sums and differentials dt and dz are replaced by finite intervals:

$$\overline{F} \simeq \frac{1}{T}\sum_{i=1}^{N}\sum_{j=1}^{M_i}\left(\frac{\rho \cdot u^3}{2}\right)\Delta z_j \Delta t_i \tag{4.5}$$

The adopted time-discretization is $\Delta t_i = 0.125$ s, so that the wave period is composed by $N = 11$ time-steps. M_i indicates the number of cells making up the pipe flow $h(t)$ at time i. Because of the difficulties of

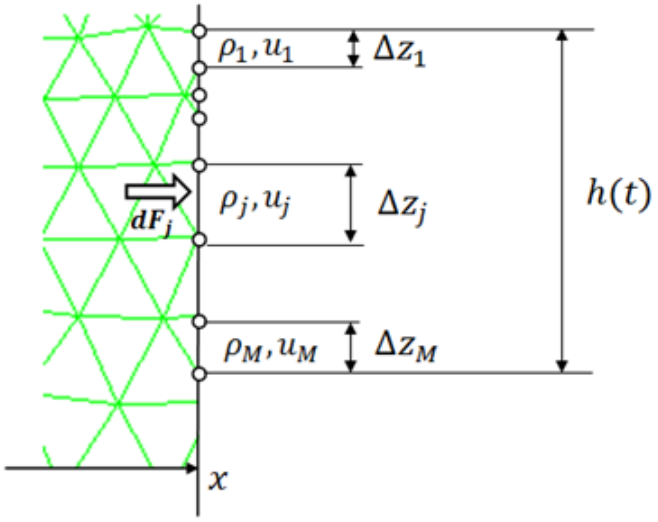


Figure 4.8: Schema of the numerical calculation of $F(t)$

accurately identifying the instantaneous height $h(t)$ of the pipe flow (see section 4.7) a constant value $\tilde{h}$ was fixed for the calculation, increased with respect to $h(t)$ so that $\tilde{h} \geq h(t)\ \ \forall\, t \in T$.

4.7 Correction of the interface gradient

4.7.1 Limits of VOF model

The model used to describe the free surface is the VOF *(Volume Of Fluid)* which unlike the front-tracking methods, considers the interface zone as composed of a single fluid with discontinuous properties, introducing an additional unknown factor to the problem that expresses the volume fraction occupied by one of the fluid in the control volume.
Assigning the value 0 to air and the value 1 to water, a typical distribution of phase (and so of density) is represented in fig.9 as a function of the vertical coordinate z:

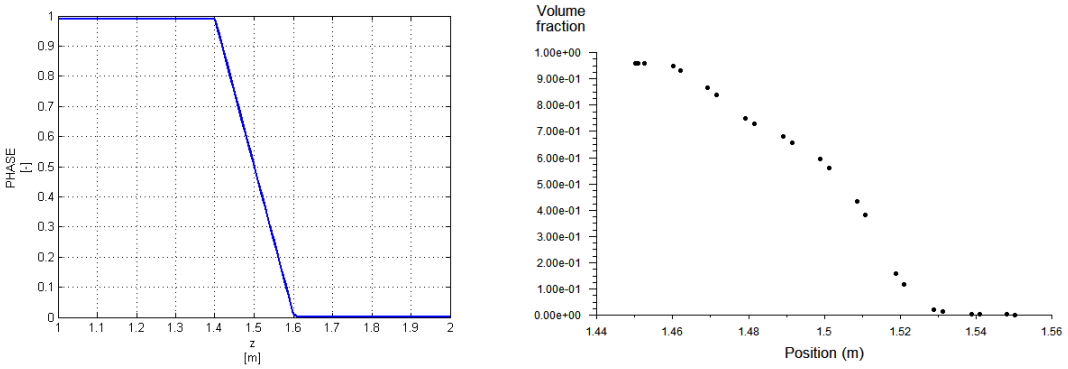


Figure 4.9: Distribution of the volume fraction at the interface

The slope of the inclined line shown in figure is greater the greater is the accuracy in the discretization of the domain at the interface between the two phases. From simulation results it was found that the elements which greatly contribute to the calculation of F are not characterized by a definite value of volume fraction (next to 0 or 1) but an intermediate value, with questionable physical significance.

4.7.2 Fuzzy-logic approach

In this regard, it was decided to weigh the contributions in the integration of energy flow based on the value of volume fraction, by adopting a distinction between air and water phases by means of fuzzy logic, which responds to the following proposition:

"Cells that contain water only are those characterized by a phase rather close to one"

so that the objective function:

$$F(t) = \int_0^{\tilde{h}} \frac{\rho(z) \cdot u^3(z)}{2} dz \tag{4.6}$$

is replaced by:

$$F(t) = \int_0^{\tilde{h}} \frac{(\rho)_{water} \cdot u^3(z)}{2} \cdot \tilde{f}(\phi) dz \tag{4.7}$$

in which $\tilde{f}(\phi)$ is a filter-function on the phase defined by:

$$\tilde{f}(\phi) = \frac{1}{1 + e^{-20(\phi - 0.5)}} \tag{4.8}$$

Operatively, the filter is applied simply by changing the definition of the integrand function from (4.6) in (4.7) within the calculation routine in the CFD solver.
In fact, the entire procedure of evaluation of the energy flow is automatically handled by the CFD solver, which provides output the value, integrated in space. The integration over time is performed manually at the conclusion of the simulation.

```
define
custom-field-functions
define
"energy-flux"
0.5 * 998.2 * x_velocity^3 * 1/(1+e^(-20*(phase - 0.5)))
```

The filter function shown in fig. has been shaped by trial, requiring that an element with $\phi = 0.6$ will be disadvantaged by 10% compared to an element that has fully unitary phase: this implies that an element with $\phi = 0.8$ does not differ from one with $\phi = 1$ (both certainly containing water). The need to introduce the filter function arises after finding

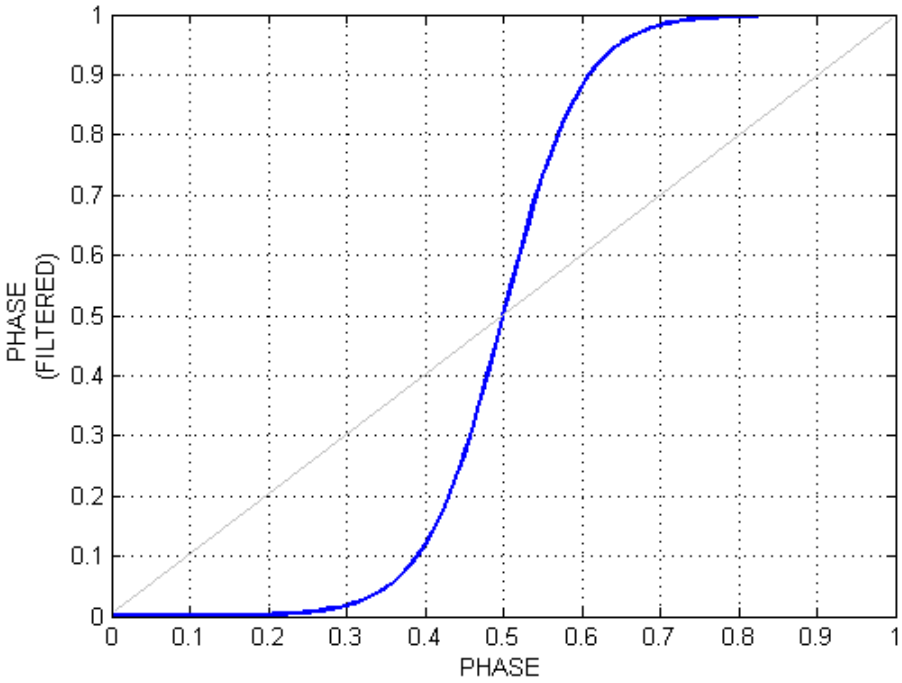


Figure 4.10: Filter $\tilde{f}(\phi)$ function on the phase values

excessive instability of the results by varying meshes: small changes in the discretization caused large variations in energy flow. This fact indicatives a physically unreliable calculation.
Of course, by varying the value of the phase in this way, also the density varies approximately proportional and therefore distort the results obtaining a density distribution that does not satisfy the Navier-Stokes equations. Nevertheless, the results are more stable and therefore the calculation is considered more reliable.

4.7.3 Differences on results

To evaluate the influence of the uncertainty of the VOF model at the interface, the values of $F(t)$ obtained in the optimum configuration and filtered by the algorithm (13) were compared to the unfiltered values (fig.16): The instantaneous flow of energy in the time gap corresponding

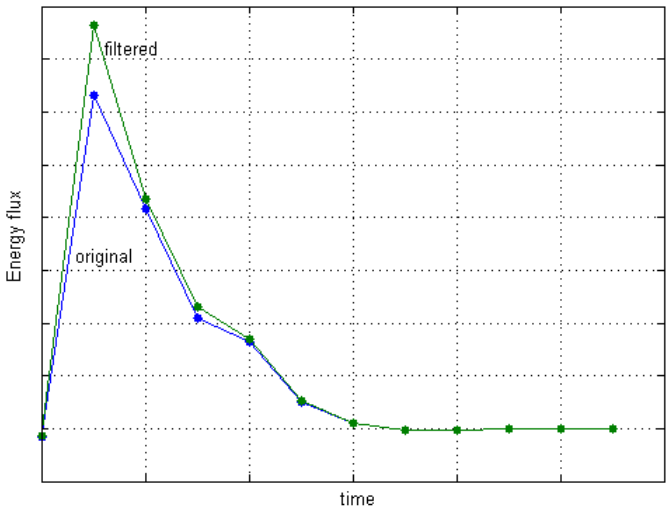


Figure 4.11: Comparison between filtered values of $F(t)$ and unfiltered

to wave breaking results slightly amplified compared to the unfiltered case. Shown below a comparison of the contours of kinetic-energy flux in the original and filtered case.

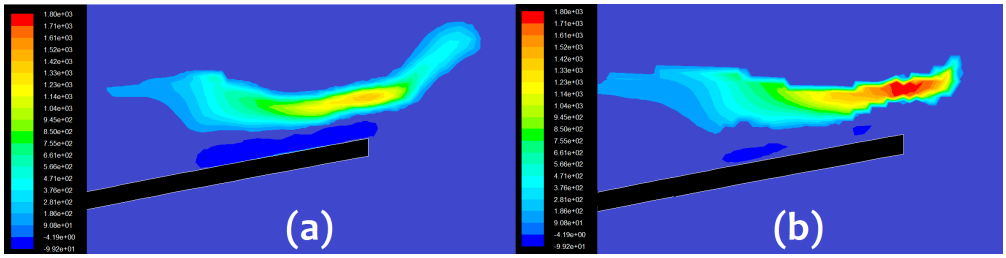


Figure 4.12: Contours of energy flux in original (a) and filtered case (b)

Chapter 5

Optimization procedure

5.1 Preliminary choises

In order to reduce the number of variables processed by the algorithm genetic it was necessary carry out some preliminary choices. The length of the segment joining the extremes of the inclined plane was set equal to the wavelength:

$$L' = L_{wave} \tag{5.1}$$

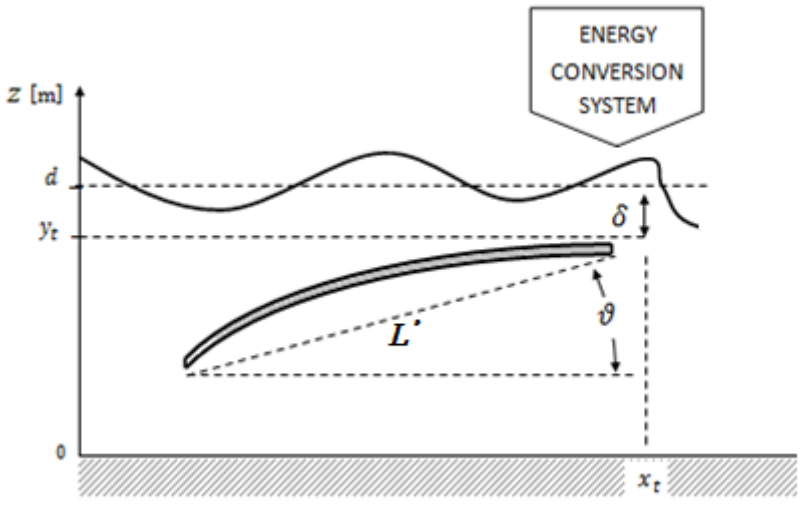


Figure 5.1: Variables to be fixed before the genetic optimization

To establish plausible values of ϑ, some more consideration are required, particularly with regard to the type of breaking to be obtained through the numerical simulations.
As explained in chapter (2.6), the behavior of a wave during its breaking is well-described by the Iribarren-Battjes empirical parameter:

$$\gamma = \frac{\tan \vartheta}{\sqrt{H_0/L_0}} \tag{5.2}$$

where ϑ is the local slope of the seabed.
The most interesting type of breaking, in order to induce a horizontal net flow downstream of the plane, is *plunging wave*, which can be obtained for values of γ varying between 0.5 and 3. So, reversing the previous equation with respect to ϑ, the following relationship can be obtained:

$$\gamma = 1.7 \quad \Longrightarrow \quad \vartheta = 15^\circ \tag{5.3}$$

A preliminary optimization cycle has also allowed to determine the optimal depth at which place the upper end of the breaking plane.
To make results compatible with other types of wave, the depth of the plane was non-dimensionalized with respect to the wave heigh:

$$\delta = \frac{d - y_t}{H} \tag{5.4}$$

The method used in the pre-optimization is the *grid method* [14] whereby the space of variables is explored through a uniform rectangular discretization, and the objective function is evaluated at all grid points. The pre-optimization lead to an optimal depth equal to:

$$\boxed{\delta_{opt} \approx 1/4} \tag{5.5}$$

so that the top of the inclined plane was placed as:

$$y_t = d - H/4 \tag{5.6}$$

5.2 Introduction to the optimization loop

Given the long time required for completion of a simulation, the procedure has been implemented so as to automatically perform the various steps of the optimization loop, as shown in the following figure:

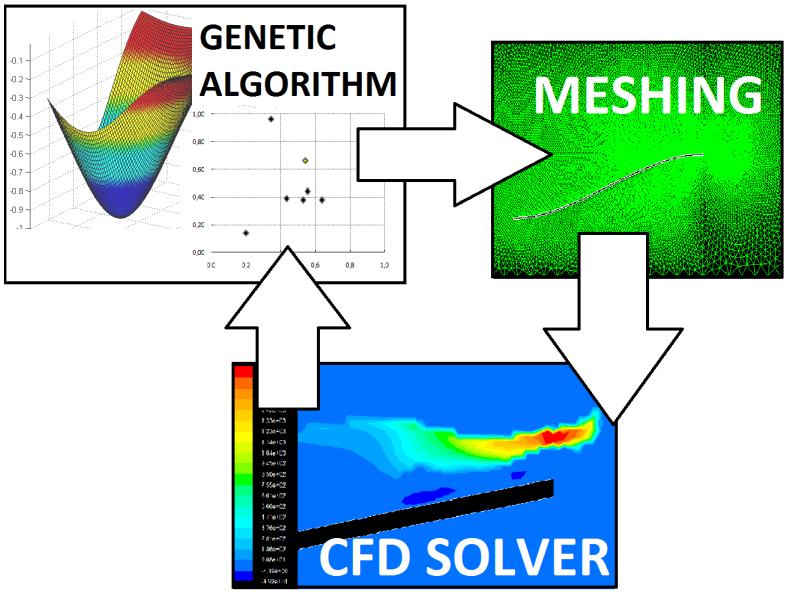


Figure 5.2: Representation of the optimization loop

The whole optimization procedure is automatically managed by using three different commercial software , each having respectively the function of:

- managing the genetic algorithm and processing the decision variables (MATLAB)
- defining the profile of the seabed and meshing the geometry of the model, on the basis of the decision variables received in input (Ansys GAMBIT)
- evaluating the objective function through an unsteady CFD simulation (Ansys FLUENT)

The setup of the genetic algorithm described in section 5.5 is implemented in the file *options-ga.m*, while the entire optimization loop is managed by the main program, *master.m*, which combines individuals from generation to generation in accordance with in options.

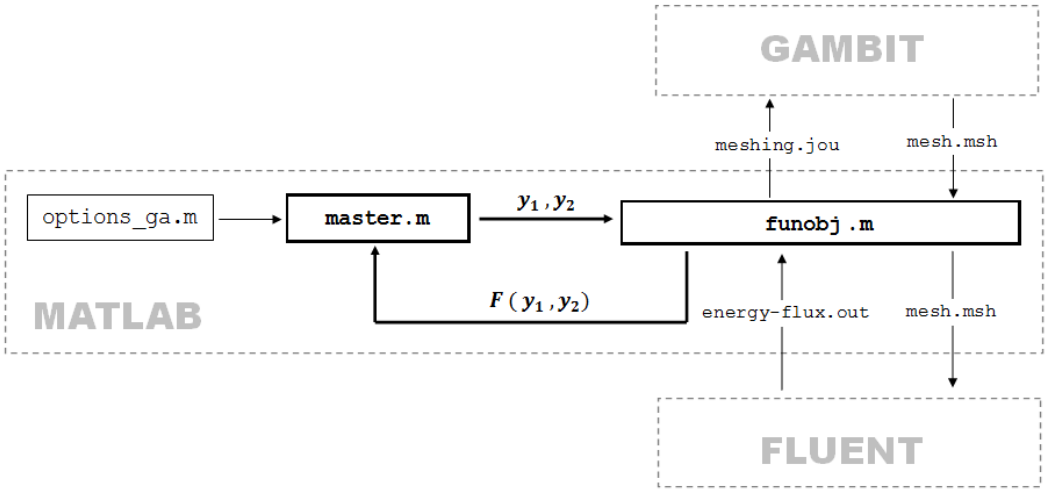


Figure 5.3: Detailed description of the optimization loop

The main routine is that which calculates the objective function (*funobj.m*) starting from the two decision variables received from *master.m*, and its functions are in the order:

- managing the data storage;
- deriving the Bezier curve starting from the decision variables;
- producing in output the journal file for the meshing software;
- starting GAMBIT and (afterwards) FLUENT, syncing them properly;
- from the output of FLUENT, calculating the objective function and returning it to *master.m*

5.3 Design variables

As explained in paragraph 5.1, it was found as the average slope of the seabed does not clearly influence the energetic performances of the breaking wave: it was therefore decided to fix the two ends of the inclined plane and vary the profile of the plane by a Bezier curve linking the two extremes, defined by two decicion variables y_1 and y_2 both varying between y_b and y_t.
A Bezier curve is a function defined by a set of control points P_0 through P_n, where n is called its *order* ($n = 1$ for linear, 2 for quadratic, etc.). The first and last control points are always the end points of the curve; however, the intermediate control points (if any) generally do not lie on the curve. The mathematical definition of a Bezier curve is:

$$[x(t); y(t)] = \sum_{i=0}^{n} C_{n,i} \cdot t_i (1 - t_i)^{n-i} [X_i; Y_i] \tag{5.7}$$

being:

$$C_{n,i} = \frac{n!}{i!(n-i)!} \tag{5.8}$$

and $[X_i; Y_i]$ the coordinate of the Bezier points A,B,C,D: obviously A and D are fixed, B and C are driven by the decision variables.

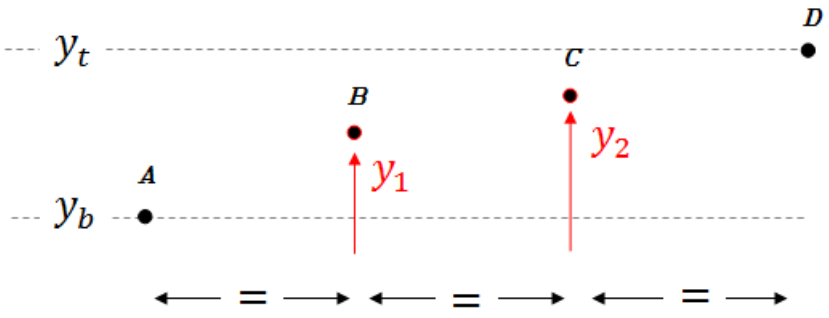


Figure 5.4: Bezier polygon dependent on decision variables y_1 and y_2

The MATLAB routine that pratically calculates the profile by using the Bezier Curves is shown below:

```
function [xy_o]=bezier(x_i,y_i,npoints) % x_i and y_i are row vectors
n=length(x_i)-1;
c=coefficients(n+1);
t=[0:1/(npoints-1):1];
for k=1:n+1
    c_t(k,:)=c(k)*t.^(k-1).*(1-t).^(n-(k-1));
end
x_o=c_t'*x_i';  y_o=c_t'*y_i';  xy_o=[x_o y_o];
end
```

In order to generalize the obtain results, the decision variables have been non-dimensionalized with respect to the positions of the ends of the seabed:

$$\xi_1 = \frac{y_1 - y_b}{y_t - y_b} \qquad ; \qquad \xi_2 = \frac{y_2 - y_b}{y_t - y_b} \tag{5.9}$$

Being $0 \leq \xi_1, \xi_2 \leq 1$, the limit configurations that can be processed by the optimization algorithm are illustrated in fig.3. Obviously among all possible configurations, there is also the straight plane, which corresponds to the design point:

$$\xi_1 = 1/3 \qquad ; \qquad \xi_2 = 2/3 \tag{5.10}$$

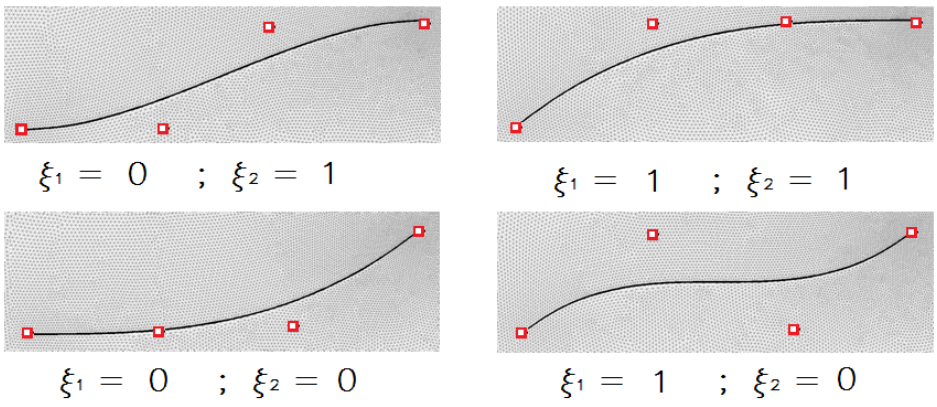


Figure 5.5: Limit configurations of the decision variables

5.4 Objective function

The optimization problem in question is mono-objective and consists in maximizing the energy performance of a sea-wave in correspondence of its breaking point (fig.5.6) by varying the geometry (2D) of the seabed. The objective function, evaluated over a period, was illustrated in section

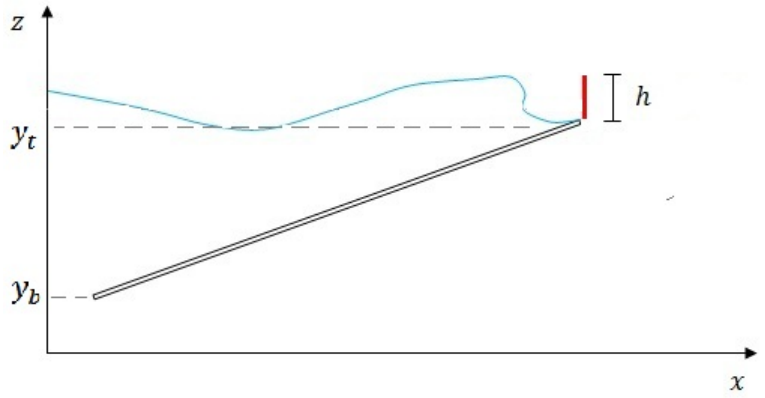


Figure 5.6: Evaluation section of the objective function

4.6 with the corrections made in accordance with section 4.7:

$$\overline{F} = -\frac{1}{T}\sum_{i=1}^{N}\sum_{j=1}^{M_i}\left(\frac{\rho_{water}\cdot u^3}{2}\cdot\frac{1}{1+e^{-20(\phi-0.5)}}\right)\Delta z_j \Delta t_i \tag{5.11}$$

The minus sign is due to the default configuration of the GA (minimization) while in the case under examination is required maximization of the objective function. The time integration is carried out in MATLAB environment, while the spacial integration is automatically-provided by the CFD solver in which a user-defined function was define:

```
define
custom-field-functions
define
"energy-flux"
0.5 * 998.2 * x_velocity^3 * 1/(1+e^(-20*(phase - 0.5)))
```

5.5 Settings of the Genetic Algorithm

5.5.1 Test function

To determine the best setup of the genetic algorithm that can ensure a good chance of finding the minimum of $F(y1; y2)$ have made various tests, by using an objective function $\tilde{F}$ in the form of Gaussian bell, suitably shaped so that it presents a minimum within the domain considered:

$$\tilde{F}(y_1; y_2) = -e^{-\left[3(y_1-1)^2+3(y_2-0.9)^2\right]} \tag{5.12}$$

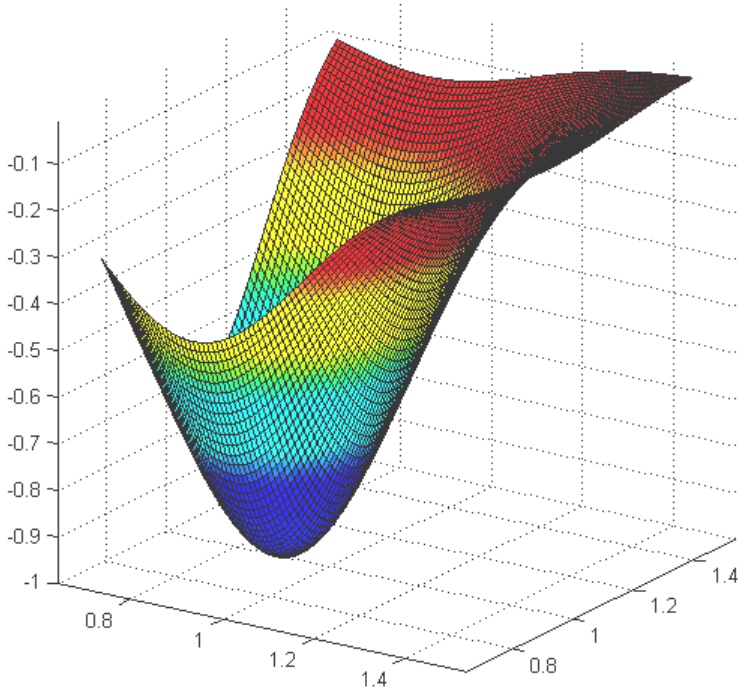


Figure 5.7: Test function of the Genetic Algorithm

That, as it was built, has a minimum at the point:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}_{opt} = \begin{pmatrix} 1 \\ 0.9 \end{pmatrix} \tag{5.13}$$

With 5 individuals the minimum is reached on average after 7-8 generations, launching the algorithm several times, however, it sometimes happens that the minimum is reached with sufficient accuracy only after 11-12 generations.
From here, the need to increase the number of individuals with the aim of increasing the capacity of exploration of the algorithm, and can have a greater probability of reaching the desired minimum.

5.5.2 Option settings

Having to do with an objective function which is very onerous to be evaluated (as it requires an unsteady CFD simulation) it was necessary to reach a compromise between the security to achieve the optimum point, and limiting the total number of simulations. The chosen parameters are:

- *Number of individuals* = **8**
- *Number of generations* = **8**
- *Crossover function* = **crossover intermediate**
- *Elite count* = **1**
- *Crossover fraction* = **6/7**

Elite Count parameter indicates the number of individuals with the best fitness values in the current generation that are guaranteed to survive to the next generation. These individuals are called *elite children*. When Elite count is at least 1, the best fitness value can only decrease from one generation to the next.

Setting Elite count to a high value causes the fittest individuals to dominate the population, which can make the search less effective.
Using these parameters, in each generation there is an individual that survives from the previous generation, one that is created by a genetic mutation, and the remaining are created by crossover between two parents.
Individuals that make up the initial population were uniformly distributed in the domain of the decision variables, so that no macro-area results unexplored by the code:

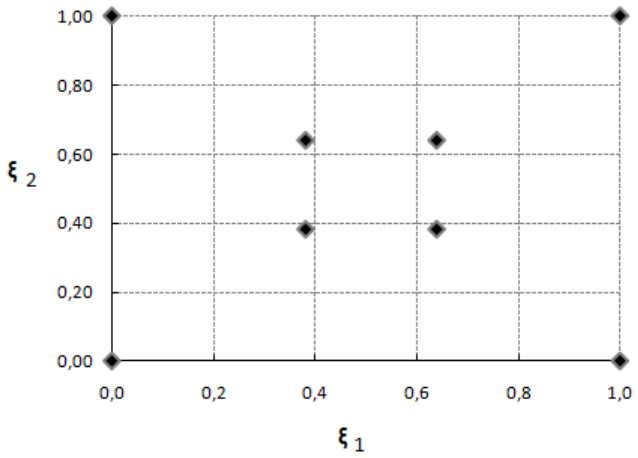


Figure 5.8: Initial population on the space of the decision variables $\mathbb{S}$

Chapter 6

Results and developments

6.1 Optimization results

Fig.7 shows the evolution of the population to varying generations, highlighting the fitness of the best individual and worst one.

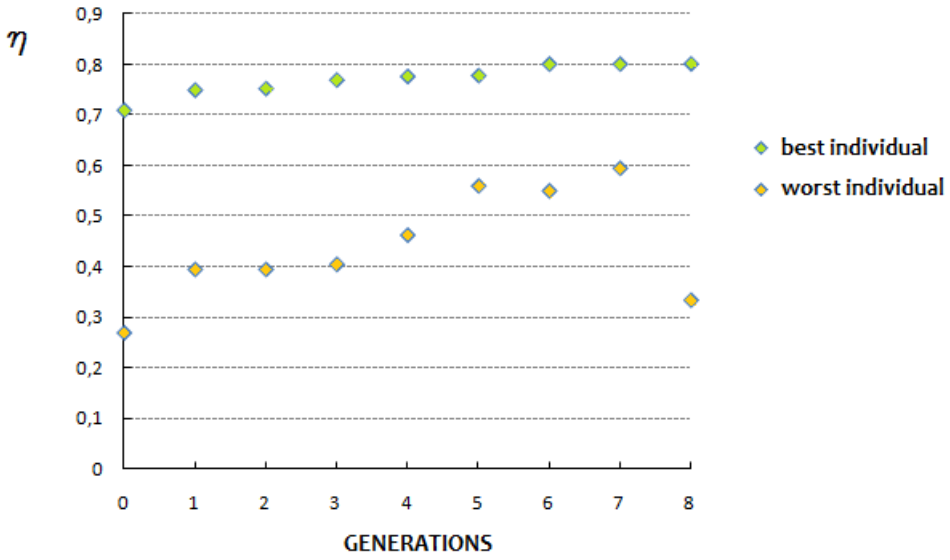


Figure 6.1: Time-evolution of the fitness of the population

As shown in the figure, the efficiency presents great variability within the domain of the decision variables, ranging from 20% to 70% in the initial generation, and from 30% to 80% in the initial generation.

The genetic algorithm leads to an improvement of yield of 10%, that from the point of view of energy conversion is a significant achievement. Below is illustrated in detail the evolution of the different individuals of the population in subsequent generations.

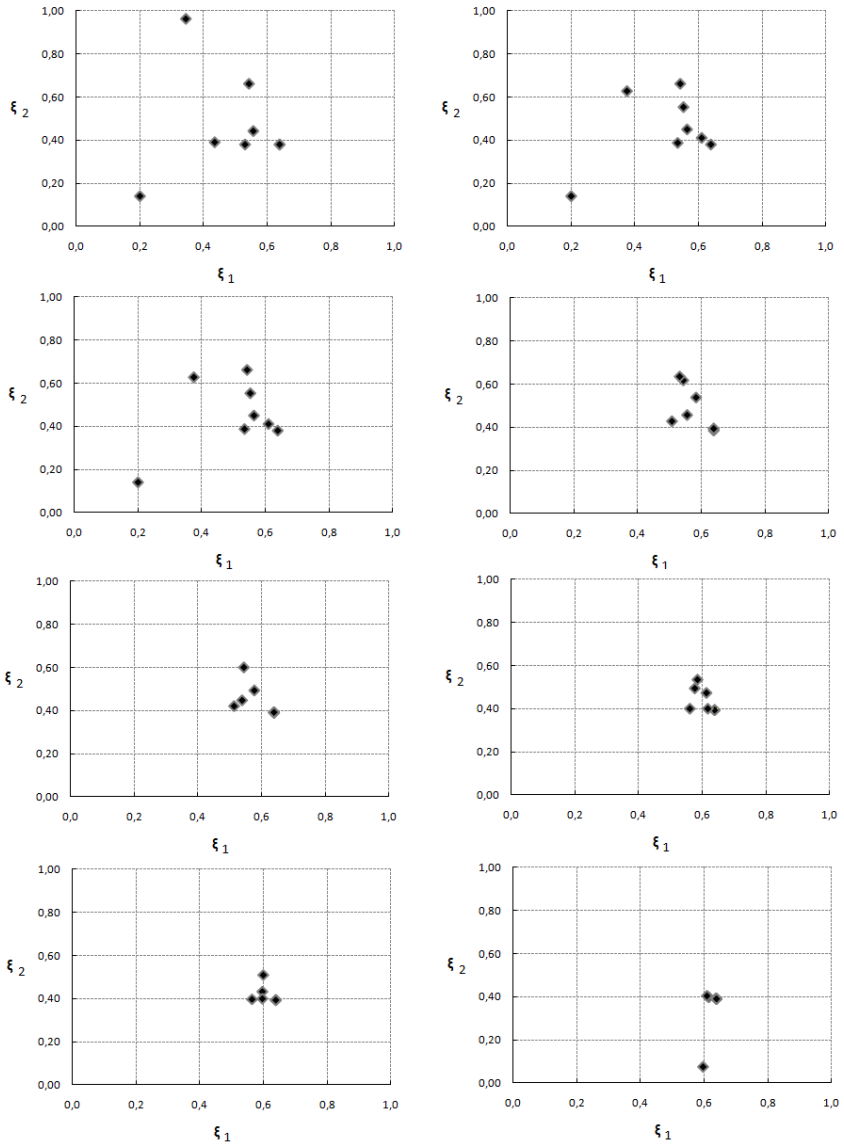


Figure 6.2: Time-evolution of individuals

After eight generations, the individual characterized by the best fitness was found to have the following *genotype*:

$$(\xi_1; \xi_2) = (0.64; 0.39) \tag{6.1}$$

which corresponds the *phenotype*:

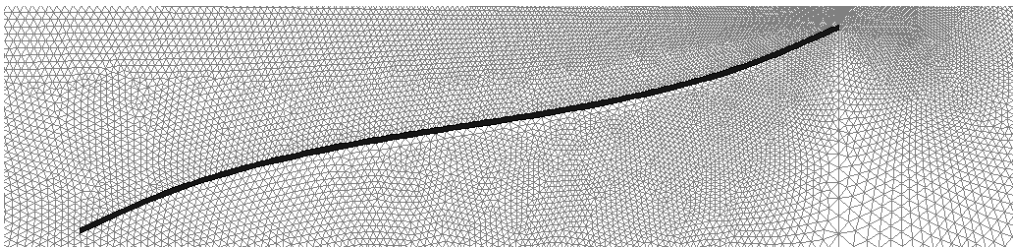


Figure 6.3: Phenotype of the optimal individual

As shown in fig.6.2, all individuals in the latest generation are crowded next to the optimal individual decribed in (6.1): this is an indication of the significance of the results.

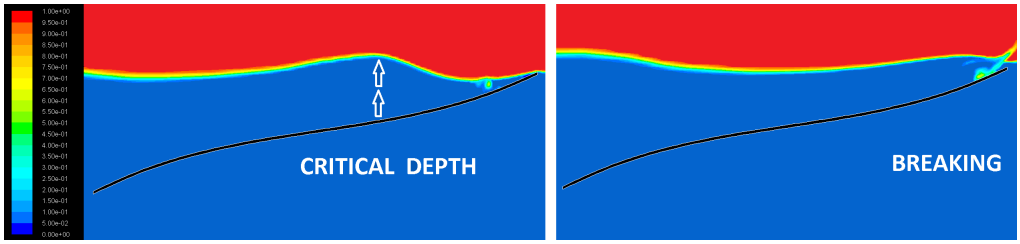


Figure 6.4: Contours of phases: (a) $t = T, ..., nT$; (b) $t = \frac{T}{3}, ..., \frac{3n+1}{3}T$

The particular shape of the seabed profile found by the genetic algorithm allows incident wave to reach the critical depth (and consequently to initiate the wave breaking as described in (13)) in correspondence of the central zone of the plane, so that the maximum flow of kinetic energy takes place at the top of the seabed (fig.6.4).

Contours of the specific flux of kinetic energy are illustrated in Fig. 6.5.

$$f(t) = \frac{\rho \cdot u^3}{2} \tag{6.2}$$

The value of $f(t)$ is almost zero everywhere before the wave breaking because of the reduced horizontal speeds, and reaches its maximum at the instants of wave breaking in correspondence of the evaluation section of the objective function. Figures 6.6 and 6.7 illustrate how solutions that

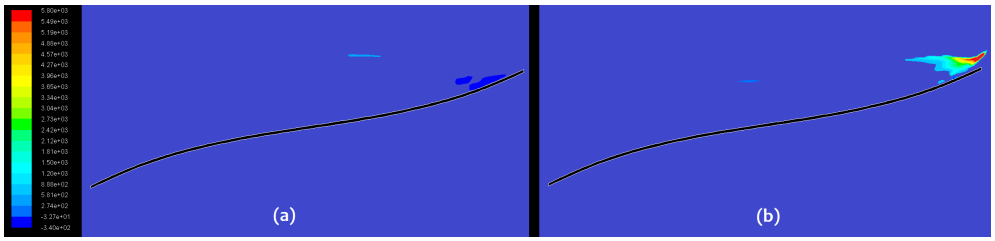


Figure 6.5: Contours of $f(t)$: (a) $t = T, ..., nT$; (b) $t = \frac{T}{3}, ..., \frac{3n+1}{3}T$

are very different from the optimal are not suitable to induce the required type of wave breaking. Particularly, a profile similar to fig.6.6 causes a premature wave breaking and the consequent dissipation of energy before investing the evaluation section. The profile of the plane corresponds to the couple:

$$(\xi_1; \xi_2) = (1; 1) \tag{6.3}$$

Figure 6.6: Premature breaking: contours of specific energy flux

On the contrary, a profile similar to fig.6.7 does not induce a proper wave breaking: most of the energy is reflected back (in the opposite direction of the wave propagation) and the wave breaking takes place downstream of the inclined plane, dissipating the remaining energy in hydrodynamic frictions.The profile of the plane corresponds to the couple:

$$(\xi_1; \xi_2) = (0; 0) \tag{6.4}$$

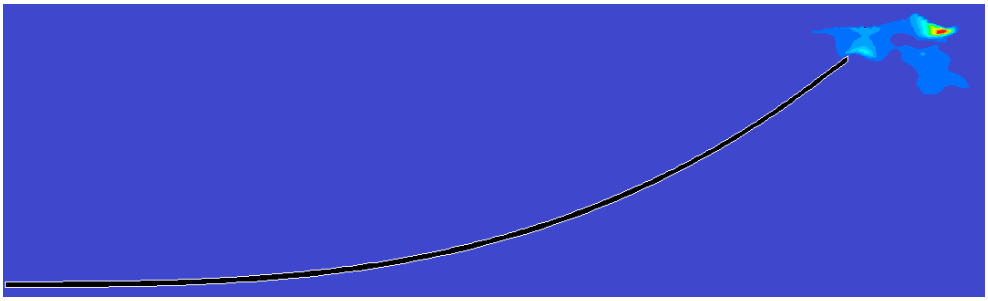


Figure 6.7: Wave reflection and late breaking: contours of specific energy flux

6.2 Installation solutions

A first elementary system solution consists in placing the rotor at the point of maximum average flow of kinetic energy, so that the breaking wave invests the lower blades of the wheel, exchanging energy.

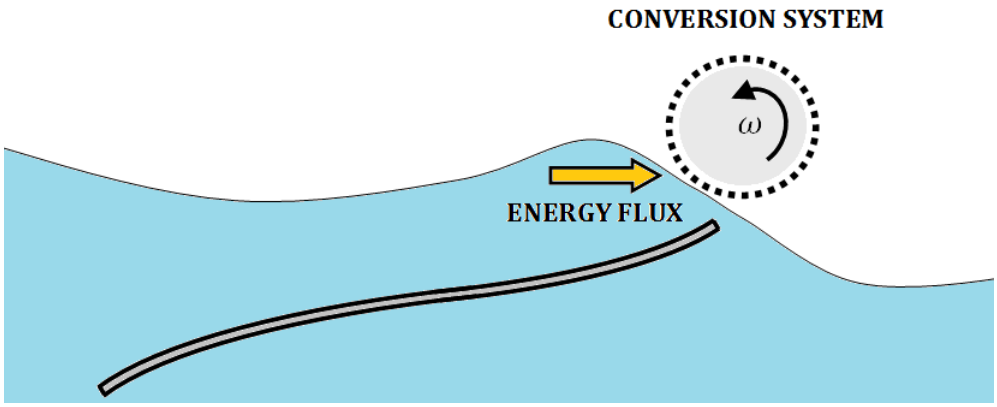


Figure 6.8: First plant solution for breaking waves

An other solution is showed in the following figure: first waves invest the upper blades of the rotor, yielding part of kinetic energy, and the return flow invests in the lower part, favoring the rotation in the same direction.

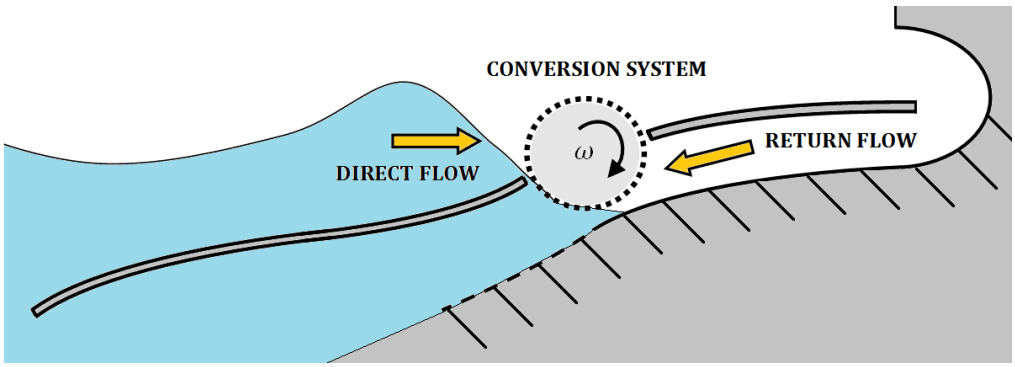


Figure 6.9: Second plant solution for breaking waves

6.3 Energetic considerations

From the previous chapter resulted that, through appropriate arrangements on the geometry of the inclined plane, it is possible to induce an "optimal" breaking to the incident wave, in order to maximize the release of kinetic energy from it.
Through the definition of *breaking efficiency*, it was estimated the maximum percentage of extractable energy compared to the nominal unperturbed wave power:

$$\overline{F}_{max} = 0.8 \cdot P_0 = 80\% \text{ of the wave power} \tag{6.5}$$

Energy availability in the Mediterranean Sea amounts to the following average values:

- 8-12 kW/m at the coast;
- 50 kW/m offshore;

While for the oceans, the energy availability are significantly higher thanks to the presence of ocean waves:

- 20-25 kW/m at the coast;
- even more than 100 kW/m offshore;

Assuming use of a water wheel for the energy conversion (University of Southampton recent studies have shown that such devices have a conversion efficiency of about η_c=0.6), it is estimated that the energy per linear meter obtainable from a breaking-wave conversion plant installed at a Mediterranean coast is:

$$P_{net} = P_0 \cdot \eta_b \cdot \eta_c = 10 \cdot 0.8 \cdot 0.6 \approx \boxed{5 \text{ kW/m}} \tag{6.6}$$

Then with a few meters of plant would be possible, in principle, to equal the performance of a common mini wind turbine plant.

Appendix

6.4 MATLAB code of the optimization loop

6.4.1 Main program

```
clear
clc

mkdir('data');

global count;
count=0;
global history;

csvwrite('.\n_generation.txt',0);

csvwrite('.\n_individual.txt',0);

fitnessfcn=@funobj_waves;
nvars=2;

A = [];
b = [];
Aeq = [];
beq = [];
LB=[0.704,0.704];
UB=[1.48,1.48];
nonlcon=[];
options=options_ga;

[x,fval,exitflag,output,population] =
ga(fitnessfcn,nvars,A,b,Aeq,beq,LB,UB,nonlcon,options);
```

6.4.2 Objective function

```
function fo=funobj_waves(x)
global count;
global history;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n_ind=csvread('.\n_individual.txt');
gen=csvread('.\n_generation.txt');
n_ind=n_ind+1;
if n_ind==9
    n_ind=1;
    csvwrite('.\n_individual.txt',1);
    gen=gen+1;
    csvwrite('.\n_generation.txt',gen);
    f_name_gen=strcat('.\data\gen_',num2str(gen));
    mkdir(f_name_gen);
else
    csvwrite('.\n_individual.txt',n_ind);
end
folder_individual=strcat('.\data\gen_',num2str(gen),
'\ind_',num2str(n_ind));
mkdir(folder_individual);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x_i=[4 4.966 5.932 6.898];
y_i=[0.704 x(1) x(2) 1.48];
nurbs=bezier(x_i,y_i,10);
f_j_g=fopen('.\meshing.jou','wt');
fprintf(f_j_g,'%s\n','vertex create coordinates 0 0 0 '); %1
fprintf(f_j_g,'%s\n','vertex create coordinates 0 1.538 0 '); %2
fprintf(f_j_g,'%s\n','vertex create coordinates 0 3.5 0 '); %3
fprintf(f_j_g,'%s\n','vertex create coordinates 6.898 3.5 0 '); %4
fprintf(f_j_g,'%s\n','vertex create coordinates 20 3.5 0 '); %5
fprintf(f_j_g,'%s\n','vertex create coordinates 20 0 0 '); %6
fprintf(f_j_g,'%s\n','vertex create coordinates 6.898 0 0 '); %7
fprintf(f_j_g,'%s\n','vertex create coordinates 6.898 1.55 0 '); %8

for i_=1:10
    vertex=['vertex create coordinates ',num2str(nurbs(i_,1)),'
',num2str(nurbs(i_,2)),' 0'];
    fprintf(f_j_g,'%s\n',vertex);
end

for j_=1:10
    vertex=['vertex create coordinates ',num2str(nurbs(j_,1)),'
',num2str(nurbs(j_,2)-0.02),' 0'];
    fprintf(f_j_g,'%s\n',vertex);
end

fprintf(f_j_g,'%s\n','vertex create coordinates 0 1.5 0 '); %29
fprintf(f_j_g,'%s\n','vertex create coordinates 6.898 1.5 0 '); %30
fprintf(f_j_g,'%s\n','vertex create coordinates 7.2 1.55 0 '); %31
fprintf(f_j_g,'%s\n','vertex create coordinates 7.2 1.46 0 '); %32
```

```
fprintf(f_j_g,'%s\n','edge create straight "vertex.1" "vertex.29" '); %1
fprintf(f_j_g,'%s\n','edge create straight "vertex.2" "vertex.3" '); %2
fprintf(f_j_g,'%s\n','edge create straight "vertex.3" "vertex.4" '); %3
fprintf(f_j_g,'%s\n','edge create straight "vertex.4" "vertex.5" '); %4
fprintf(f_j_g,'%s\n','edge create straight "vertex.5" "vertex.6" '); %5
fprintf(f_j_g,'%s\n','edge create straight "vertex.6" "vertex.7" '); %6
fprintf(f_j_g,'%s\n','edge create straight "vertex.7" "vertex.1" '); %7
fprintf(f_j_g,'%s\n','edge create straight "vertex.4" "vertex.8" '); %8
fprintf(f_j_g,'%s\n','edge create straight "vertex.8" "vertex.30" '); %9
fprintf(f_j_g,'%s\n','edge create straight "vertex.28" "vertex.7" '); %10
fprintf(f_j_g,'%s\n','edge create straight "vertex.19" "vertex.9" '); %11
fprintf(f_j_g,'%s\n','edge create nurbs "vertex.9" "vertex.10" "vertex.11"
"vertex.12" "vertex.13" "vertex.14" "vertex.15" "vertex.16" "vertex.17"
"vertex.18" interpolate '); %12
fprintf(f_j_g,'%s\n','edge create straight "vertex.18" "vertex.28" '); %13
fprintf(f_j_g,'%s\n','edge create nurbs "vertex.19" "vertex.20" "vertex.21"
"vertex.22" "vertex.23" "vertex.24" "vertex.25" "vertex.26" "vertex.27"
"vertex.28" interpolate  '); %14
fprintf(f_j_g,'%s\n','edge create straight "vertex.29" "vertex.2" '); %15
fprintf(f_j_g,'%s\n','edge create straight "vertex.30" "vertex.18" '); %16
fprintf(f_j_g,'%s\n','edge create straight "vertex.29" "vertex.30" '); %17
fprintf(f_j_g,'%s\n','edge create straight "vertex.8" "vertex.31" '); %18
fprintf(f_j_g,'%s\n','edge create straight "vertex.31" "vertex.32" '); %19
fprintf(f_j_g,'%s\n','edge create straight "vertex.32" "vertex.28" '); %20
fprintf(f_j_g,'%s\n','face create wireframe "edge.1" "edge.17" "edge.16" "edge.12"
"edge.11" "edge.14" "edge.10" "edge.7" real');
fprintf(f_j_g,'%s\n','face create wireframe "edge.4" "edge.5" "edge.6" "edge.10"
"edge.20" "edge.19" "edge.18"  "edge.8" real');
fprintf(f_j_g,'%s\n','face create wireframe "edge.2" "edge.3" "edge.8" "edge.9"
"edge.17" "edge.15" real');
fprintf(f_j_g,'%s\n','face create wireframe "edge.16" "edge.9" "edge.18" "edge.19"
"edge.20" "edge.13" real');
fprintf(f_j_g,'%s\n','edge mesh "edge.1" successive ratio1 0.975 size 0.02');
fprintf(f_j_g,'%s\n','edge mesh "edge.2" successive ratio1 1.1 size 0.06');
fprintf(f_j_g,'%s\n','edge mesh "edge.3" size 0.2');
fprintf(f_j_g,'%s\n','edge mesh "edge.4" size 0.2');
fprintf(f_j_g,'%s\n','edge mesh "edge.5" size 0.2');
fprintf(f_j_g,'%s\n','edge mesh "edge.6" size 0.2');
fprintf(f_j_g,'%s\n','edge mesh "edge.7" size 0.2');
fprintf(f_j_g,'%s\n','edge mesh "edge.8" successive ratio1 0.95 size 0.04');
fprintf(f_j_g,'%s\n','edge mesh "edge.9" size 0.007');
fprintf(f_j_g,'%s\n','edge mesh "edge.10" successive ratio1 1.08 size 0.05');
fprintf(f_j_g,'%s\n','edge mesh "edge.11" size 0.02');
fprintf(f_j_g,'%s\n','edge mesh "edge.12" successive ratio1 0.995 size 0.015 ');
fprintf(f_j_g,'%s\n','edge mesh "edge.13" size 0.01');
fprintf(f_j_g,'%s\n','edge mesh "edge.14" successive ratio1 1.007 ratio2 1.025 size 0.03');
fprintf(f_j_g,'%s\n','edge mesh "edge.15" successive ratio1 1 size 0.01');
fprintf(f_j_g,'%s\n','edge mesh "edge.16" size 0.007');
fprintf(f_j_g,'%s\n','edge mesh "edge.17" successive ratio1 1.006 ratio2 1.01 size 0.02');
fprintf(f_j_g,'%s\n','edge mesh "edge.18" size 0.01');
fprintf(f_j_g,'%s\n','edge mesh "edge.19" size 0.01');
fprintf(f_j_g,'%s\n','edge mesh "edge.20" size 0.01');

fprintf(f_j_g,'%s\n','face mesh "face.1" triangle  ');
fprintf(f_j_g,'%s\n','face mesh "face.2" triangle  ');
fprintf(f_j_g,'%s\n','face mesh "face.3" triangle  ');
fprintf(f_j_g,'%s\n','face mesh "face.4" triangle  ');

fprintf(f_j_g,'%s\n','physics create "muri" btype "WALL" edge "edge.7"
"edge.6" "edge.5" "edge.11" "edge.12" "edge.13" "edge.14" ');
fprintf(f_j_g,'%s\n','physics create "cielo" btype "PRESSURE_OUTLET" edge
```

```
"edge.2" "edge.3" "edge.4" ');
fprintf(f_j_g,'%s\n','physics create "in" btype "VELOCITY_INLET" edge "edge.1" "edge.15" ');
fprintf(f_j_g,'%s\n','export fluent5 ".\\mesh.msh" nozval  ');
fprintf(f_j_g,'%s\n','abort ');

fclose(f_j_g);

!.\gambit_launcher.bat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

check=0;
while check~=1
   if exist('.\mesh.msh','file')==2
        check=1;
        pause(10);
   else
        pause(1);
   end
end
movefile('.\meshing.jou',folder_individual);

!.\fluent_launcher.bat
check=0;
while check~=1
    if exist('.\mesh-1975.dat','file')==2
     check=1;
     pause(10);
    else
        pause(10);
    end
end
results=textread('.\energy-flux.out','%s');
i=149;
j=1;
sum=0;
while j<12
    sum = sum +  (str2double(results(i)));
    i=i+2;
    j=j+1;
end
fo = -(sum/11);

file_ind=strcat(folder_individual,'\variabili_decisione.txt');
f_=fopen(file_ind,'wt');
fprintf(f_,'%f%s',x(1),',');
fprintf(f_,'%f%s',x(2),',');
fprintf(f_,'%f%s',fo  ,',');
fclose(f_);
movefile('.\mesh.msh',folder_individual);
movefile('.\energy-flux.out',folder_individual);
movefile('.\mesh-1975.cas',folder_individual);
movefile('.\mesh-1975.dat',folder_individual);
count=count+1;
history(count,1)=x(1);
history(count,2)=x(2);
history(count,3)=fo;
end
```

Bibliography

[1] M. Raciti Castelli, M. Vianello, A. Maccarini, E. Benini, *Numerical generation of breaking waves in a water tank*, submitted for publication to Energy Conversion and Management on February 2, 2012.

[2] T. Stathopoulos, *Wind Effects on People*, International Conference on Urban Wind Engineering and Building Aerodynamics, COST Action C14, Rode-Saint-Genèse (Belgium), 2004.

[3] A. G. Jensen, J. Franke , C. Hirsch, M. Schatzmann, T. Stathopoulos, J. Wisse, N.G. Wright, *CFD Techniques - Computational Wind Engineering*, International Conference on Urban Wind Engineering and Building Aerodynamics, COST Action C14, Rode-Saint-Genèse (Belgium), 2004;

[4] N. Repalle, K. Thiagarajan, M. Morris-Thomas, *CFD Simulation of wave run-up on a spar cylinder* , 16th Australasian Fluid Mechanics Conference, 2007.

[5] S. Rajendran, N. Fonseca, G. F. Clauss, C. G. Soares, M. Klein, *Time Domain Comparison With Experiments For Ship Motions And Structural Loads On A Container Ship In Abnormal Waves* , 30th International Conference on Ocean, Offshore and Arctic Engineering, Rotterdam, The Netherlands June 19-24, 2011.

[6] Anant Lal, M. Elangovan, *CFD Simulation and validation of flap type wave-maker* , World Academy of Science Engineering and Technology, 2008.

[7] N. Repalle, K. Thiagarajan, F. Paterson, *Wave run-up investigation on a square cylinder* , 17th Australasian Fluid Mechanics Conference, 2010.

[8] R. M. Sorensen, *Basic Coastal Engineering* , Springer, 2006.

[9] R. Dean, R. Darlrymple, *Water wave mechanics for engineers and scientists* , World Scientific, 1992.

[10] C.W. Hirt, B.D. Nichols, *Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries*, Journal of Computational Physics 39, 201-225, 1981.

[11] M. Raciti Castelli, M. Vianello, E. Benini, *Influence of seabed curvature on the energy performance of breaking waves*, under review.

[12] Hafsia, Z., Ben Haj, M., Lamloumi, H., Maalel, K. and Zgolli, R., *Linear and nonlinear numerical wave generation in viscous fluid* ENIT. Laboratoire d'Hydraulique B.P. 37. Le Belvédère, 1002 Tunis, Tunisia.

[13] Battjes, J. A.,*Surf Similarity*, Proceedings of the 14th ICCE. Copenhagen, 1974. pp: 466-480.

[14] S.S. Rao, *Engineering Optimization - Theory and Practise* , John Wiley and Sons, 2009.

[15] Gerald Muller, Sally Denchfield, Reinhard Marth, Bob Shelmerdine: *Stream wheels for applications in shallow and deep water* Southampton University.

[16] M. Vianello, M. Raciti Castelli, E. Benini, *Genetic Optimization of the Seabed for Maximizing the Breaking Efficiency of Water-Waves* , in preparation.