

1.Introduction

The ARTEMIS code (Agitation and Refraction with TElemac on a MIld Slope) solves the Elliptic mild slope equation (1) using finite element techniques inside the TELEMAC modelling system structure. This equation is obtained from the Navier-Stokes equations according a set of hypothesis (low value of wave steepness, low value of bottom slope). The main results at each node of the computational mesh are the height, the phase and the direction of the wave. The main application of ARTEMIS concerns the wave agitation in harbours or small bays. ARTEMIS is able to take into account the following phenomena :

- Wave reflection by an obstacle,
- Wave diffraction behind an obstacle,
- Wave refraction by bottom variation,
- Regular wave,
- Monodirectional or multidirectional random wave,
- Bottom friction,
- Bathymetric breaking.
- Taking into account of the dissipation by breaking and/or bottom friction,
- Improvement of boundary conditions taking into account incidence angle on a solid boundary, and input/output angle in a liquid boundary,

However, the actual version of the software is unable to take into account the following phenomena :

- Wave refraction by current,
- Dry area inside the domain (tidal flats).

The application domains of the software are various. In particular, it offers the possibility to study wave agitation in harbour or bay, to appreciate the impact of harbour equipment (pier, dike), to evaluate the wave agitation behind a breach, the wave decay behind an island or flat, seiching in a channel, etc.

ARTEMIS was developed by the National Hydraulics and Environnement Laboratory (Laboratoire National d'Hydraulique et d'Environnement - LNHE) of the Research and Studies Directorate of the French Electricity Board (EDF-R&D). The program complies with EDF-DER's Quality Assurance procedures for scientific and technical programs (2). This sets out rules for developing and checking product quality at all stages. In particular, a program covered by Quality Assurance procedures is accompanied by a Validation File that describes a set of test cases (see reference (3)). This document can be used to determine the performance and limitations of the software and define its field of application. The test cases are also used for developing the software and are checked each time new versions are produced. Description sheets for each test case have been updated for the version 6.2 of the code.

1.1 Situation of ARTEMIS within the TELEMAC modelling system

The ARTEMIS software is part of a complete set of computational software and their associated preand post-processors, the TELEMAC system. This offers all the modules required for constructing a model and using it to simulate 2D and 3D hydrodynamic phenomena (wave and current), water

quality and sedimentology.

The TELEMAC system comprises the following modules :

- The MATISSE software designed, using the bathymetric and/or topographic data, to generate a mesh consisting of triangular elements,
- The STBTTEL software designed to retrieve the file from a mesh generator, possibly interpolate a bathymetry, and generate a geometry file in the SELAFIN format that can be read by the simulation modules as well as the RUBENS software. STBTTEL also conducts a number of mesh coherence checks,
- The TELEMAC-2D software designed to perform the hydrodynamic simulation in two horizontal space dimensions. In addition, TELEMAC-2D can simulate the transport of dissolved tracers,
- The TELEMAC-3D software itself, designed to carry out the hydrodynamic simulations of flows in three space dimensions. Besides, TELEMAC-3D can simulate the transport of tracers. The SEDI-3D library contains of the relevant subroutines for the simulation of non-cohesive sediment transport. The implementation of the TELEMAC-3D software is the subject matter of this document,
- The SISYPHE software designed to carry out the simulation the transport of sediment through bed load traction and suspension.
- The ARTEMIS software designed to simulate the changes in the features of wave agitation either in a coastal water body or a harbour,
- The TOMAWAC software designed to simulate, through a spectral method, the sea state in permanent or transitory conditions,
- The ESTEL-2D software designed to simulate the underground flows in two vertical space dimensions,
- The ESTEL-3D software designed to simulate the underground flows in three dimensions,
- The POSTEL-3D software designed to prepare the 2D cross sections in the 3D result file, for a processing by the RUBENS graphics software,
- The RUBENS software designed to graphically process the results of the various simulation modes,
- The SPARTACUS-2D software designed to simulate the two-dimensional laminar and turbulent flows through the SPH method.

As a complement to the TELEMAC chain, the FUDAA-PREPRO software (as developed from the FUDAA platform by the CETMEF's, Informatics and Modelling Research Department) covers all the preprocessing tasks involved by the achievement of a digital hydraulic study.

1.2 User programming

When he uses a simulation module from the TELEMAC SYSTEM, the user may have to program specific functions which are not provided in the code's standard release. In particular, that is made through a number of so-called « user » subroutines the sources of which are supplied within the distribution. The procedure to be carried out in that case comprises the steps of :

- Recovering the standard version of the user subroutine(s) as supplied in the distribution and copying it into the current directory.
- Amending the subroutine(s) according to the model to be constructed.
- Concatenating the whole set of subroutines into a single Fortran file which will be compiled during the ARTEMIS launching process.

During that programming stage, the user can gain access to the various variables of the software through the Fortran 90 structures.

All the data structures are gathered within Fortran files, which are known as modules. For ARTEMIS,

the file name is DECLARATION_ARTEMIS. To gain access to the ARTEMIS data, just insert the command USE DECLARATIONS_ARTEMIS into the beginning of the subroutine. Adding the command USE BIEF may also be necessary in order to reach the structure in the BIEF library.

2. Theoretical Aspects

This report is the theoretical note of the scientific software ARTEMIS, version 3.0. ARTEMIS deals with the modelling of wave propagation towards the coast and wave agitation in harbours. This software is integrated in the TELEMAC system, and is developed following the Quality Assurance procedures in effect in the Research Division of Electricité de France.

After a few words about the statistical and spectral approaches of the wave description, we present the theoretical model of monochromatic waves as used by ARTEMIS, which is based on the Berkhoff's or mild slope equation verified by the velocity potential. Then, we describe how non-linear dissipative processes have been included in the initial linear model. The various formulations implemented in the software to quantify the wave energy dissipation through surf-breaking and bottom friction are explained. The treatment of liquid and solid boundaries is realized through Neumann conditions, and enables to model incident waves (radiation condition), free wave-output, full or partial reflections. A specific methodology, presented in the report, has been developed in ARTEMIS to take into account directional spreading and frequency distribution of the wave energy (random waves). Finally, we describe the numerical algorithm implemented to solve the mathematical model, as expressed through the finite element formulation, using functions developed in the BIEF library of the TELEMAC system.

2.1 Generals about waves

2.1.1 Definitions

Waves deserve their name! Defining a "wave" does call for special care when its specific physical variables are to be determined. A wave is defined by the deformation of the free surface between two successive occurrences either of a zero up-crossing or a zero down-crossing including a crest and a trough. Depending on which one of the above conventions is adopted, the distribution of wave heights, as defined by the differences between the elevations of the wave crests and troughs, may be different. The period of a wave can be defined using the symbol T .

The characteristics of the sea state (height, period...) can be known from the free surface deformation signal $z(x,y,t)$ as a function of horizontal position (x,y) and time t , following two kinds of approaches, namely statistical and spectral approaches.

2.1.2 Statistical approach

Once symbols are chosen for defining the wave height and period, all the recorded heights H_i and periods of N waves can be computed from signal $z(x,y,t)$. The following parameters are then defined :

\overline{H} : wave height average as defined by (rarely used variable):

$$\overline{H} = \frac{1}{N} \sum_{i=1}^N H_i$$

$H_{1/n}$: height average in the upper (1/n)th percentile of the wave height population

$H_{1/3}$: the significant height is particularly defined for $n = 3$

H_{rms} : root mean square value of wave heights H_i as defined by

$$H_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N (H_i)^2}$$

H_{max} : maximum recorded heigh

$T_{H1/n}$: average of periods T_i in the upper third of the wave height population

$T_{1/3}$: average of periods T_i in the upper third of the wave period population

\overline{T} : average of all the wave periods T_i

It was found in deep water that the likeliness of a wave height H^0 exceeding a height H was properly approximated by such a rule as:

$$\text{Prob}(H^0 \geq H) = -\left(\frac{H}{b}\right)^2 \quad (1)$$

The related probability density is given by the following relation:

$$p(H) = \frac{2H}{b^2} e^{-\left(\frac{H}{b}\right)^2} \quad (2)$$

It defines a so-called Rayleigh's statistical distribution of waves. Parameter b characterizes the population. It can be related, for example, to the root mean square value of wave height H_{rms} . This is because:

$$(H_{rms})^2 = \int_0^{\infty} H^2 p(H) dH \quad (3)$$

That integral can easily be computed: its value is b2. The Rayleigh's law is then finally written as follows:

$$p(H) = \frac{2H}{H_{rms}^2} e^{-\left(\frac{H}{H_{rms}}\right)^2} \quad (4)$$

The following relations are given by that wave height distribution law:

$$H_{1/10} = 1.27 H_{1/3}$$

$$H_{1/3} = 1.60 \overline{H} = 1.416 H_{rms}$$

Note

These results are obtained by computing the $H_{1/n}$ heights as follows: let H^* be such that $\text{Prob}(H^0 > H^*) = 1/n$. Immediately, we have:

$$H^* = \sqrt{\ln(n)} H_{rms}$$

$H_{1/n}$ is subsequently defined by:

$$\frac{1}{n} H_{1/n} = \int_{H^*}^{\infty} H p(H) dH$$

This integral is expressed as a function of Hrms and leads to the above mentioned relations. That distribution can be highly modified in shallow water under the influence of dissipation processes (primarily breaking). A truncated model of probability density incorporating the concept of maximum boundary height can be adopted. Other sea state parameters can also be set either through the statistical analysis or from each individual wave.

2.1.3 Spectral Approach

The initial step is the signal $\zeta(x, y, t)$ of free surface deformation in time over the horizontal domain being investigated. After submitting the signal $\zeta(x, y, t)$ to a generalized Fourier transform combining time and space dependence of the free surface elevation, we can write, after Massel (5) :

$$\zeta(x, y, t) = \zeta_0(x, y) + \lim_{F \rightarrow \infty} \int_{f=-F}^{+F} \int_{\theta=0}^{2\pi} \tilde{A}_{x,y}(f, \theta) e^{\tilde{\psi}_{x,y}(f, \theta)} e^{i(2\pi - k(x \cos \theta + y \sin \theta))} df d\theta \quad (5)$$

k is a function of f through a so-called "dispersion" relation that will be identified hereinafter

θ denotes a propagation direction with a positive counterclockwise orientation with respect to an axis Ox

$\tilde{A}_{x,y}(f, \theta)$ et $\tilde{\psi}_{x,y}(f, \theta)$ are obtained through a generalized Fourier transform applied to the three-dimension case (5), which is actually reduced to two dimensions because of the dispersion relation between two dimensions. In addition, $\tilde{A}_{x,y}(-f, \theta)$ is the conjugate complex of $\tilde{A}_{x,y}(f, \theta)$. \tilde{A} is expressed in meters.

$\zeta_0(x, y)$ is the average level of the free surface, with an assumed zero value all over the domain being investigated.

The directional spectrum $E(f, \theta)$ of wave energy at every point (x,y) in the domain is related to $\tilde{A}_{x,y}(f, \theta)$ by the following formal relation :

$$E(f, \theta) df.d\theta = \frac{1}{2} \rho g \tilde{A}_{x,y}'^2(f, \theta) \quad (6)$$

wherein $\tilde{A}_{x,y}'^2(f, \theta) = 2\tilde{A}_{x,y}'^2(f, \theta)$

The directional spectrum $S(f, \theta)$ of wave action, as expressed in $m^2.Hz^{-1}$, is usually defined by

$$S(f, \theta) = \frac{E(f, \theta)}{\rho g} \quad (7)$$

The following spectral wave parameters can be set from $S(f, \theta)$:

m_n : n-order moment of directional spectrum as defined by :

$$m_n = \int_{f=0}^{+\infty} \int_{\theta=0}^{2\pi} f^n S(f, \theta) df.d\theta$$

m_0 : especially for n = 0, the variance m_0 of the free surface difference of level is defined

H_{m_0} : significant spectral height, as defined by :

$$H_{m_0} = 4\sqrt{m_0}$$

H_e : Spectral energy height as defined by the relation :

$$m_0 = \frac{1}{8} H_e^2$$

Thus, H_e is related to H_{m_0} by :

$$m_0 = \frac{1}{8} H_e^2$$

T_p : peak period as determined by the maximum value of spectrum $S(f, \theta)$

θ_m : mean wave direction, as defined by the expression :

$$v = B \cos(\omega t - \varphi_2)$$

$T_{0,1}$: First mean spectral wave periods defined by :

$$T_{0,1} = \frac{m_0}{m_1} = \frac{\int_{f=0}^{+\infty} \int_{\theta=0}^{2\pi} S(f, \theta) df . d\theta}{\int_0^{2\pi} \int_{f=0}^{+\infty} f S(f, \theta) df . d\theta}$$

$T_{0,2}$: Second mean spectral wave periods defined by:

$$T_{0,2} = \sqrt{\frac{m_0}{m_2}} = \sqrt{\frac{\int_{f=0}^{+\infty} \int_{\theta=0}^{2\pi} S(f, \theta) df . d\theta}{\int_0^{2\pi} \int_{f=0}^{+\infty} f^2 S(f, \theta) df . d\theta}}$$

T_M : Third mean spectral wave TM defined by:

$$T_M = \frac{\int_0^{2\pi} \int_{f=0}^{+\infty} (1/f) S(f, \theta) df . d\theta}{\int_0^{2\pi} \int_{f=0}^{+\infty} S(f, \theta) df . d\theta}$$

\bar{k} : Mean wave number, computed from the dispersion relationship using the mean pulsation $\bar{\omega} = \frac{2\pi}{T_{0,1}}$

\bar{C} : Mean phase velocity $\bar{C} = \frac{\bar{\omega}}{\bar{k}}$

\bar{C}_g : Mean phase velocity $\bar{C}_g = \frac{d\bar{\omega}}{d\bar{k}}$

As in the statistical analysis, other parameters can be computed from the energy spectrum. If the height distribution complies with a Rayleigh's law, then the statistical approach and the spectral approach may correspond to each other:

$$H_{m0} \approx H_1/3$$

$$H_c \approx H_{rms}$$

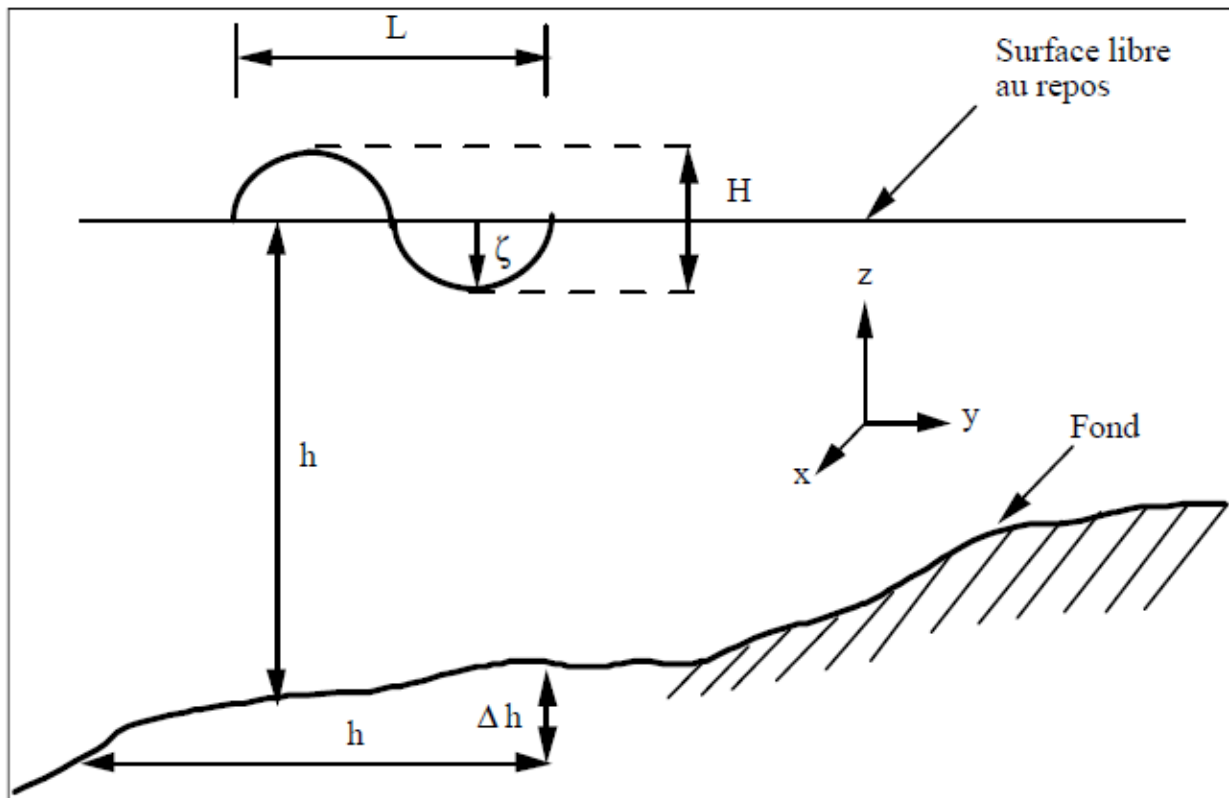
These two approaches are not necessarily equivalent to each other in shallow waters where the dissipation processes (mainly surf-breaking) and the non-linear transfers between frequencies affect the distribution of wave heights.

2.2 Theoretical modelling

2.2.1 Concept's background. Basic assumptions

All the wave propagation theories rely on the Newtonian fluid dynamics model as described by the Navier-Stokes' equations. These general equations can more or less be simplified according to the extent of complexity as kept by the modelling hypothesis. We will consider how to set the Berkhoff's model (1) or in other word the "mid slope equation", describing the monochromatic wave refraction and diffraction both by slowly spatially-changing bottoms and by obstacles. Such processes as partial or total reflection, wave inflow or outflow, are governed by boundary conditions to be specified hereinafter.

The fluid movement is located in an Oxyz Cartesian co-ordinate system whose axis z coincides with the ascending vertical. The origin of axis Oz is taken at the free surface at rest. The free surface elevation and the bottom depth are denoted z and -h, respectively (h being the positive or negative water depth at rest). H is the wave height, L is its wavelength.



The general modelling assumptions are as follows :

- the fluid (seawater) is assumed to be nonviscous
- the flow is assumed to be irrotational. Thus, one can introduce such a potential Φ as $\nabla\Phi = \vec{u}$ wherein \vec{u} is the fluid particle velocity vector
- the fluid is assumed to be incompressible
- the bottoms do not change in time

2.2.2 Development of Berkhoff's equation

2.2.2.1 Forming the equations

Once the above assumptions and definitions are available, the momentum conservation and continuity equations can be simplified as follows :

$$\begin{cases} \nabla^2\Phi = 0 \\ \nabla\left(\frac{\partial\Phi}{\partial t} + \frac{1}{2}(\nabla\Phi)^2 + \frac{p}{\rho} + gz\right) = 0 \end{cases} \quad (8)$$

using the conventional notations, namely p for pressure, ρ for water density and g for gravity. Equation (8.1) expresses the conservation of mass and Equation (8.2) expresses the conservation of momentum (as written in the form of the Bernoulli's equation). Pressure p is not hydrostatic due to the wave action.

These two equations are associated to the boundary conditions both at the free surface ($z = \zeta$) and the bottom ($z = -h$); the boundary conditions are written as follows :

$$\begin{cases} \left. \frac{\partial \Phi}{\partial z} \right|_{z=\zeta(x,y,t)} = \frac{\partial \zeta}{\partial t} + \frac{\partial \Phi}{\partial x} \cdot \frac{\partial \zeta}{\partial x} + \frac{\partial \Phi}{\partial y} \cdot \frac{\partial \zeta}{\partial y} \\ \left. \frac{\partial \Phi}{\partial z} \right|_{z=-h(x,y)} = -\frac{\partial \Phi}{\partial x} \cdot \frac{\partial h}{\partial x} - \frac{\partial \Phi}{\partial y} \cdot \frac{\partial h}{\partial y} \end{cases} \quad (9)$$

Equation (9.1) expresses the free surface imperviousness and equation (9.2) expresses the bottom imperviousness.

In spite of their seemingly simple form, these two sets of equations, however, are still difficult to solve, since they include non-linear terms :

$(\nabla \Phi)^2$ in (8.2) and $\frac{\partial \Phi}{\partial x} \cdot \frac{\partial \zeta}{\partial x} + \frac{\partial \Phi}{\partial y} \cdot \frac{\partial \zeta}{\partial y}$ in (9.1).

Through assumptions about the wave and bottom characteristics, these equations will be linearized and will then be solved more easily.

2.2.2.2 Linearization

That linearization is conducted under the following assumptions :

- Gentle steepness : $\varepsilon_1 = H/L \ll 1$
- Small wave height against depth : $\varepsilon_2 = H/h \ll 1$

The solution $\Phi_0 = 0$ and $\zeta_0 = 0$ (i.e. rest) is a particular solution to the problem. Further solutions disturbing that particular solution will be searched for. The following linear system is achieved through a series development of the potential, only keeping 1-order terms in ε_1 :

$$\begin{cases} \nabla^2 \Phi = 0 \\ \text{en } z=0 \quad \frac{\partial \Phi}{\partial t} + g\zeta = 0 \\ \text{en } z=0 \quad \frac{\partial \Phi}{\partial z} = \frac{\partial \zeta}{\partial t} \\ \text{en } z=-h(x,y) \quad \frac{\partial \Phi}{\partial z} = -\frac{\partial \Phi}{\partial x} \cdot \frac{\partial h}{\partial x} - \frac{\partial \Phi}{\partial y} \cdot \frac{\partial h}{\partial y} \end{cases} \quad (10)$$

In order to derive Equation (10.2), the Bernoulli's equation at the surface (Equation (8.2)) was written. Pressure is then the air pressure, i.e. a constant pressure. Thus, it disappears from the equation, since its gradient is null

We have at the surface (i.e. in $z = 0$, since the free surface distortions are ignored to 0-order in the series development in ε_1) :

$$\begin{aligned} \nabla \left(\frac{\partial \Phi}{\partial t} + g\zeta \right) &= 0 \\ \text{or also} \quad \frac{\partial \Phi}{\partial t} + g\zeta &= \lambda(t) \end{aligned}$$

wherein λ function is only time-dependent. That function can be introduced into the time derivative of the potential. This is because the physical quantity herein is velocity, which remains unchanged when a term that is only time-dependent is added to the potential.

By deleting z between equations (10.2) and (10.3), one finally comes to:

$$\left\{ \begin{array}{l} \nabla^2 \Phi = 0 \\ \text{en } z=0 \quad \frac{\partial^2 \Phi}{\partial^2 t} + g\zeta = 0 \\ \text{en } z=-h(x,y) \quad \frac{\partial \Phi}{\partial z} = -\frac{\partial \Phi}{\partial x} \cdot \frac{\partial h}{\partial x} - \frac{\partial \Phi}{\partial y} \cdot \frac{\partial h}{\partial y} \end{array} \right. \quad (11)$$

The linear theory is derived from equations (11), by separating t and z variables as well as x and y variables.

Considering the problem to be solved, the vertical component z probably does not play the same part as components x and y . Furthermore, since the equations are linear, every potential can be broken down into a time Fourier series. One may only study the complex potential functions in $e^{-i\omega t}$. Φ will then be searched for as follows:

$$\Phi(x, y, z, t) = f(z, h) \cdot \phi(x, y, z) \cdot e^{-i\omega t} \quad (12)$$

A complex notation is used for computing the unknown functions. The return to physical quantities takes place by taking the real part of the complex variables. It is noteworthy that if h is not constant, f , like ϕ , depends on z , as well as of x and y through h .

2.2.2.3 Solution in case of a horizontal flat bottom

In case of an infinite, horizontal flat bottom, then f only depends on z and ϕ only depends on x or y (e.g. x). The solution to the problem is in the following form:

$$\Phi(x, y, z, t) = -i \frac{gH}{2\omega} \cdot \frac{ch(k(z+h))}{ch(kh)} e^{i(kx-\omega t)} \quad (13)$$

That potential describes the propagation of a monochromatic monodirectional wave (first-order Stokes' wave). Quantities k , h , g and ω are related to each other by the dispersion function (i.e. Airy's relation) :

$$\frac{\omega^2}{gk} = th(kh) \quad (14)$$

Note

so-called evanescent modes, which can play a significant part near the obstacles, are solutions in the form :

$$\cos(p_n(z+h)) \left(A_n e^{p_n x} + B_n e^{-p_n x} \right)$$

which can be added to that generalpurpose solution. Parameters p_n are then solutions to the following transcendent equation:

$$\omega^2 = g p_n \tan(p_n h) \quad (15)$$

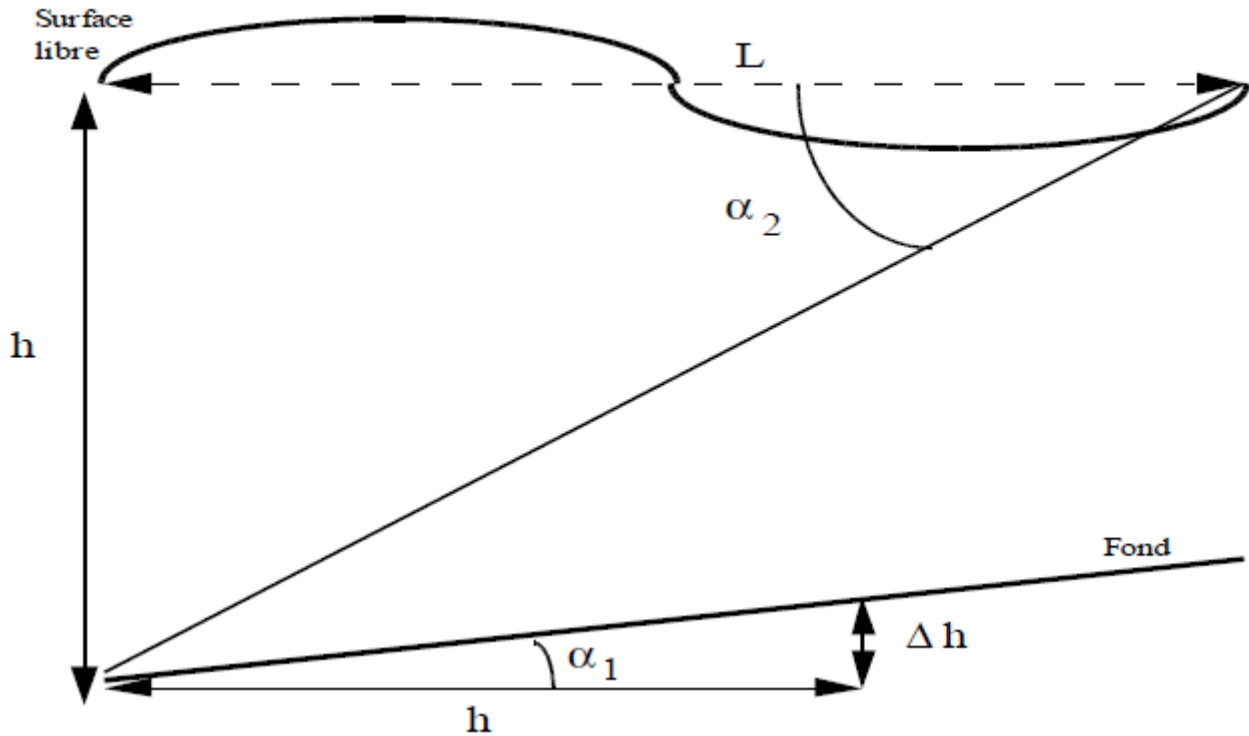
2.2.2.4 Solution in case of a low variation bottom (Mild-slope equation)

In order to solve the problem resulting from linearized equations (11) in case of a spatially-varying bottom, a further assumption will be made :

$$\epsilon_3 = \frac{\frac{\Delta h}{h}}{\frac{h}{L}} \ll 1$$

wherein Δh denotes the bottom depth variation over a horizontal distance such as h (refer to the diagram below).

It can be assumed from that inequality that the bottom slope $\tan \alpha_1$ remains lower by an order of magnitude than the slope as defined by the $h/L = \tan \alpha_2$ ratio (refer to the diagram below) :



Through that additional assumption, Berkhoff (1), develops the solution of equations by looking for such a solution as:

$$\Phi(x, y, z, t) = \frac{ch(k(z + h))}{ch(kh)} \cdot \phi(x, y) \cdot e^{-i\omega t} \quad (16)$$

This is a product of three terms. The first one is related to function f . It is the same as the solution achieved for a horizontal flat bottom, which is suitable in that case since bottom changes slowly. The second term ϕ is looked for as independent from z to the first order in ϵ_3 and is called a reduced potential. The third term means that time-periodic solutions are being searched for.

On completion of computations, Berkhoff obtains a general-purpose equation as confirmed by the reduced potential ϕ :

$$\nabla \cdot (CC_g \nabla \phi) + CC_g k^2 \phi = 0 \quad (17)$$

wherein:

- C and C_g denote the phase and group velocities, respectively
- k denotes the wave number : $k = \frac{2\pi}{L}$

- ∇ and $\nabla \cdot$ denote the gradient and divergence operators, respectively, as restricted to the horizontal co-ordinates

The unknown ϕ is an a priori complex number. Velocities are expressed from the theory as achieved for the case of a horizontal bottom:

$$C = \frac{\omega}{k} \quad (18)$$

\tag{19}

$$C_g = \left(1 + \frac{2kh}{sh(2kh)} \right) \quad (1)$$

The mild-slope equation is a two-dimension model making to possible to take into account the effects of refraction by bathymetry which is assumed to exhibit slow variations in space, as well as the effects of diffraction by both above-water and under-water obstacles. The total or partial reflection processes are taken into account in the imposed boundary conditions. Energy should also be supplied to the flow domain, since there is no source term in Equation (17). The boundary conditions make it possible to satisfy that requirement, as well as to freely release the wave energy. That model represents the propagation of steady waves, what means that the whole wave energy is contained in only one frequency. The way that mild-slope model of monochromatic wave can be used for describing a random wave behaviour will be discussed.

2.2.2.5 Solution in case of rapidly varying topography

Since version 6.1, ARTEMIS can take into account effects of a rapidly varying topography in the mildslope equation. Second order terms from gradient and curvature are integrated in the mild-slope equation. The refraction-diffraction equation becomes:

$$\nabla \cdot (CC_g \nabla \phi) + CC_g (k^2(1 + f) + ik\mu) \phi = 0$$

$$\text{with } f = E_1(kh) \cdot (\nabla h)^2 + \frac{E_2(kh)}{k_0} \cdot \Delta h$$

and k_0 is the wave number for infinite depth.

The first term represents gradient effects and the second deals with curvature of the bottom. It is possible for the user to only take into account gradient effects. One can also choose to take into account only curvature effects.

Several expressions have been proposed for E_1 and E_2 functions, by different authors: Massel, Chamberlain and Porter, Chandrasekera and Cheung, Suh Lee and Park. In ARTEMIS, Chamberlain and Porter expressions have been considered.

$$E_1(kh) = \frac{x^4 + 4x^3 \sinh(x) - 9 \sinh(x) \sinh(2x) + 3x(x + 2 \sinh(x)) \cdot (\cosh^2(x) - 2 \cosh(x) + 3)}{6n(x + \sinh(x))^3}$$

$$E_2(kh) = \frac{\sinh(x) - x \cdot \cosh(x)}{4n \cdot \cosh^2(x/2) \cdot (x + \sinh(x))}$$

$$n = \frac{1}{2} \left(1 + \frac{x}{\sinh(x)} \right)$$

$$x = 2kh$$

2.2.2.6 Return to physical quantities

Once the complex reduced potential $\phi(x, y) = \phi_r + \phi_i$ is computed from Equation (17), it becomes possible to return to total potential $\Phi(x, y, z, t)$ and to derive all the wave characteristics from it :

- Wave height

Wave height is proportional to the ϕ module and is given by:

$$H = \frac{2\omega}{g} |\phi| \quad (20)$$

- Wave phase

Wave phase Ψ , define in $]-\pi; \pi]$ interval, is the argument in ϕ . It is obtained first by computing the variable Ψ^* :

$$\Psi^* = \arctan\left(\frac{\phi_i}{\phi_r}\right)$$

$$\Psi = \Psi^* - \pi \begin{cases} \text{if } \phi_i < 0 \\ \text{if } \phi_r < 0 \end{cases}$$

$$\Psi = \Psi^* + \pi \begin{cases} \text{if } \phi_i > 0 \\ \text{if } \phi_r < 0 \end{cases}$$

otherwise $\Psi = \Psi^*$

- Free surface elevation

Free surface elevation is expressed by:

$$\zeta = -\frac{1}{g} \text{Re}\left(\frac{\partial(\phi e^{-i\omega t})}{\partial t}\right) = -\frac{\omega}{g} (\phi_i \cos(\omega t) - \phi_r \sin(\omega t)) \quad (21)$$

- Surface velocity

Surface velocity is expressed as the total gradient of the potential function, i.e. :

$$\begin{aligned} \vec{V} &= \text{Re}(\nabla(\phi, f, e^{-i\omega t})) \\ u &= \frac{\partial \phi_r}{\partial x} \cos(\omega t) + \frac{\partial \phi_i}{\partial x} \sin(\omega t) \\ v &= \frac{\partial \phi_r}{\partial y} \cos(\omega t) + \frac{\partial \phi_i}{\partial y} \sin(\omega t) \\ w &= \frac{\omega^2}{g} (\phi_r \cos(\omega t) + \phi_i \sin(\omega t)) \end{aligned} \quad (22)$$

- Wave incidence

The hodograph of horizontal surface velocity at each point is in the form of a centered ellipse.

Comparing with the major axis, the minor axis is negligible in a rectilinear propagation area; the ellipse is then nearly rectilinear. On the other hand, as soon as cross waves are observed, no main propagation direction can be really distinguished any longer and the minor axis-to-major axis ratio is no longer close to zero. It was decided in ARTEMIS to define the wave incidence from the direction of the ellipse's major axis, in the direction in which the free surface elevation is positive.

A simple transformation is made in order to achieve a reduction to:

$$u = A \cos(\omega t - \varphi_1)$$

$$v = A \cos(\omega t - \varphi_1)$$

Two different cases are then considered:

- if $|\varphi_2 - \varphi_1| = 0$ or π , then polarization is linear : the ellipse is nothing but a direction segment whose tangent to axis (Ox) has a B/A value, to within the sign.
- otherwise, polarization is elliptical. Through a change in time origin made by setting $\varphi = \varphi_2 - \varphi_1$ and $t' = \omega t - \varphi_1$ the equations can be written in the form :

$$u = A \cos(t')$$

$$v = B \cos(t' - \varphi)$$

The direction of major axis is obtained by maximizing the square of velocity module. After computing the derivative for determining the extrema, it can be found that the minimum and maximum velocity module \vec{V} ($t' = t'_0$) obtained for phase t'_0 verifying:

That equation defines four possible phases t'_0 corresponding in pairs to the major axis and the minor axis, respectively. Only one solution per axis is kept with the relation:

$$t'_0 = \frac{1}{2} \arctan \left(\frac{B^2 \sin(2\varphi)}{A^2 + B^2 \cos(2\varphi)} \right) + n \frac{\pi}{2} \quad n \in 0, 1$$

and the square of velocity module is computed for these values of t'_0 to clear up the uncertainty about n by selecting the value for which there is a maximum velocity. Lastly, the incidence direction is selected by prescribing a positive free surface elevation at time t corresponding to phase t'_0 .

- Wave energy

Wave energy per unit horizontal free surface is expressed by the following formula:

$$E = \frac{1}{8} \rho g H^2 = \frac{1}{2} \cdot \frac{\rho \omega^2 |\phi|^2}{g} \quad (23)$$

One half of it consists of potential energy, whereas the other half comprises kinetic energy.

The mild-slope model is a linear model, since if ϕ_0 is a solution to Equation (17), then $A \cdot \phi_0$ is also a solution $\forall A \in \mathbb{R}$. This is an important feature that evidences the limitations of that model when the bottoms have a small depth. This is because, if the bottoms come up, the shoaling process will “swell” the wave height to such an extent that it will become inconsistent with the variable water depth. In order to improve the domain validity and to broaden the range of applications of ARTEMIS, we have amended the initial mild-slope equation in order to take into account the prevailing dissipative processes in shallow water, namely bathymetric breaking and bottom friction.

2.2.3 Amended mild-slope equation

In order to take into account the dissipative effects resulting from bathymetric breaking and bottom friction, Booij and De Girolamo et al. suggest modifying the mild-slope Equation by introducing an additional complex term:

$$\nabla(CC_g \nabla \phi) + CC_g(k^2 + ik\mu)\phi = 0 \quad (24)$$

If the reduced potential is broken down in the form $\phi = Ae^{i\beta}$ (A being the potential amplitude and β being the potential phase), the previous equation leads, through a distinction between real part and imaginary part, to an eikonal equation that is verified by the phase :

$$(\nabla\beta)^2 = k^2 + \frac{\nabla \cdot (CC_g \nabla A)}{CC_g A} \quad (25)$$

and to a balance equation between the energy flow and the dissipation, in the following form :

$$\text{div} \left(A^2 C_g \frac{\nabla \beta}{|\nabla \beta|} \right) = -\mu C_g A^2 \quad (26)$$

It is an energy flow balance equation to within the " $\rho g/2$ " term. Coefficient μ is a dissipation coefficient expressed in m^{-1} . The rate D of energy dissipation is expressed by the following relation:

$$D = \frac{1}{2} \rho A^2 \mu C_g \quad (27)$$

Several models are provided for computing the dissipation coefficient μ , depending on whether wave energy is dissipated through breaking and/or bottom friction.

2.3 Formulation of the dissipation coefficient

2.3.1 Surf-breaking

2.3.1.1 Theoretical approaches

Surf-breaking occurs when the velocity of a water particle at the free surface exceeds the wave propagation velocity. That criterion may result in the determination of a critical breaking height H_m (Miche's criterion) :

$$H_m = \frac{0.88}{k} \text{th} \left(\frac{\gamma_s}{0.88} kh \right) \quad (28)$$

A critical wave height H_m is related to each water depth h . Coefficient γ_s takes the bottom slope m into account. The following dependence can be suggested :

$$\gamma_s = \frac{b}{1 + a \frac{b}{gT^2}} \quad \text{avec} \quad \begin{cases} a = 43.75(1 - e^{-19m}) \\ b = 1.56(1 + e^{-19.5m}) - 1 \end{cases} \quad (29)$$

Coefficient γ_s is assumed to be constant in ARTEMIS; it is given by the user (typical value: $\gamma_s = 0.8$)

Two theoretical approaches are proposed and incorporated into ARTEMIS for assessing the waveinduced dissipation of energy

- The former, as proposed by Dally et al., is obtained from surf-breaking along a sloping

beach and subsequent wave propagation over a flat, horizontal bottom. It consists in assuming that the divergence of wave energy flow can be assessed through :

$$\frac{\partial P}{\partial x} = \frac{\partial(EC_g)}{\partial x} = \frac{(EC_g)_s - (EC_g)}{l_s} \quad (30)$$

wherein x denotes the energy propagation direction. Index "s" corresponds to a stable area at the end of a length l_s beyond the surf-breaking area. That length is a priori not known. The earlier authors proposed to write:

$$l_s = \frac{h}{K}$$

For a given water depth h, the more sloping is the bottom, the higher is K, and thus the lower is l_s . Then the authors suggest the following approximations:

$$(C_g)_s \approx C_g$$

$$(H)_s \approx \Gamma h$$

In that case, dissipation D can be assessed:

$$D = \frac{1}{8} \rho g C_g H^2 \frac{K}{h} \left(1 - \left(\frac{\Gamma h}{H} \right)^2 \right) \quad (31)$$

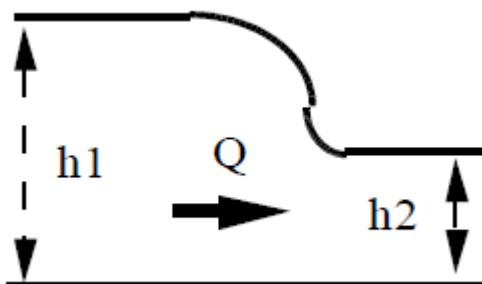
and, lastly, the dissipation coefficient μ is given by :

$$\mu = \frac{K}{h} \left(1 - \left(\frac{\Gamma h}{H} \right)^2 \right) \quad (32)$$

Dally et al. (15) recommend the following values for K and Γ according to the bottom slope

pente des fonds	K	Γ
1/80	0.100	0.350
1/65	0.115	0.355
1/30	0.275	0.475

- The latter approach, as proposed by Battjes & Janssen, is based upon the analogy between the surf-breaking area and the hydraulic jump.



The head loss related to the opposite schematic jump is :

$$\Delta h = \frac{(h_1 - h_2)^3}{4h_1 h_2} \quad (33)$$

The power dissipation is then :

$$\varepsilon = \rho g (\Delta h) Q$$

Wherein $\Delta h = h_1 - h_2$

Q is the flow rate through the jump per unit width of channel. From the jump conjugation relation, we have:

$$Q = \sqrt{\frac{gh_1 h_2 (h_1 + h_2)}{2}}$$

Assuming that $h_1 \approx h_2 \approx h$, and writing $H = h_1 - h_2$, we finally have:

$$\varepsilon \approx \frac{1}{4} \rho g H^3 \sqrt{\frac{g}{h}} \quad (34)$$

When these results are applied to waves with a height H, a frequency f and a length L, the power dissipation per unit area is :

$$D = \frac{\varepsilon}{L} \approx \frac{1}{4} \rho g f \frac{H^3}{h} \quad (35)$$

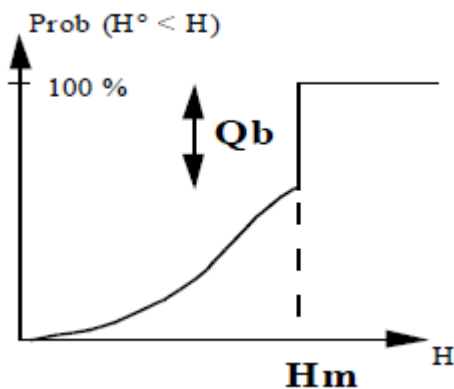
$$\mu = 2f \frac{H}{h C_g} \quad (36)$$

Both models as described in this section are incorporated into ARTEMIS software.

2.3.1.2 Steady wave / Random wave

When steady waves are processed using ARTEMIS, either abovementioned formulation for expressing the dissipation coefficient can be selected. That coefficient has to be incorporated into the mild-slope equation.

In the case of random waves, the Battjes & Janssen's formulation alone can be applied. Due to surfbreaking, the distribution of wave heights does no longer follow such a Rayleigh's probability distribution as that described in §2.1.2. Battjes & Janssen (16) suggest that the Rayleigh's distribution should be truncated from critical height H_m . The probability Q_b that the wave height should be equal to critical height H_m is given by the following implicit relation:



$$\frac{1 - Q_b}{\ln Q_b} = - \left(\frac{H_{rms}}{H_m} \right)^2$$

The rate of breaking Q_b is implicitly computed in ARTEMIS through an iterative procedure. Peak frequency f_p is chosen as the spectrum representative frequency. In addition, it is assumed that $H_m \approx h$. Dissipation is then expressed as:

$$D = \frac{\alpha}{4} Q_b f_p \rho g H_m^2 \quad (37)$$

and the dissipation coefficient as :

$$\mu = \frac{2\alpha\alpha_p Q_b}{C_g} \left(\frac{H_m}{H_{rms}} \right)^2 \quad (38)$$

The value of coefficient α to be selected is one or so. It is set in ARTEMIS.

2.3.2 Bottom friction

2.3.2.1 Theoretical approach

Waves exert on the bottom a shear stress $\overline{\tau_b}$ that can be expressed in accordance with the following quadratic law (Jonsson) :

$$\overline{\tau_b} = \frac{1}{2} f_w \rho |\overline{U_b}| \overline{U_b} \quad (39)$$

wherein f_w is a friction coefficient which depends on the nature of bottoms, is the instantaneous orbital velocity just above the bottom boundary layer. The mean energy dissipation over a wave period related to that stress is:

$$D = \frac{1}{T} \int_0^T \overline{\tau_b} \overline{U_b} dt \quad (40)$$

- For computing that integral, Kostense et al. linearised the expression of bottom friction

by introducing the dissipation coefficient μ , the water depth h and the surface orbital velocity \overline{U} :

$$\overline{\tau_b} = \rho \mu C_g h \overline{U_e} \quad (41)$$

From these two expressions of $\overline{\tau_b}$, they derived a formulation of the dissipation coefficient for friction :

$$\mu = \frac{1}{2} \cdot \frac{f_w}{C_g h \cosh^2(kh)} U_e \quad (42)$$

wherein U_e is an effective velocity representing the flow and being given by the following ration :

$$U_e = \frac{\int_0^T |\overline{U}|^3 dt}{\int_0^T |\overline{U}|^2 dt} \quad (43)$$

Accurately computing the value of U_e is a heavy task. We have incorporated an approximate, quick

computation of that variable into ARTEMIS :

$$U_e \approx 1.2 \sqrt{\frac{1}{2} \left(\left(\frac{\partial \phi_r}{\partial x} \right)^2 + \left(\frac{\partial \phi_i}{\partial x} \right)^2 + \left(\frac{\partial \phi_r}{\partial y} \right)^2 + \left(\frac{\partial \phi_i}{\partial y} \right)^2 \right)} \quad (44)$$

The error made through that approximation does not exceed 15 %.

- Putnam & Johnson simplified the computation of dissipation by assuming that the nearbottom

orbital velocity U_b was suitably described by the results of Stokes' waves on a flat bottom (see in § 2.2.2.3). An expression is easily derived for dissipation coefficient m :

$$\mu = \frac{2}{3\pi} \cdot \frac{f_w H \omega^3}{g C_g h \sinh^3(kh)} \quad (45)$$

Both above formulations are incorporated into ARTEMIS; the user chooses either formulation. The friction coefficient f_w is still to be determined. In the case of sandy bottoms, a number of authors proposed assessment methods for the computation of f_w . We have adopted for ARTEMIS the Van Rijn's approach which consists in determining f_w as a function of a Reynolds number which is computed both from the excursion of a fluid particle over the bottom and a relative roughness, which takes into account the grain (or skin) roughness, the shape roughness if ripples appear, as well as from the excursion of fluid particles over the bottom. When the bottoms are not sandy or if the user does not want to apply the above procedure, the value of f_w over the computational domain can be specified, whether a space-uniform or space-variable value is chosen.

The typical values of f_w are : 10^{-3} or so for a smooth concrete bottom ; 10^{-2} for a gravel bottom ; 10^{-1} for stones, rocks.

Note

- Dissipation through bottom friction is low as compared with breaking-induced dissipation and is exerted over long distances. Breaking-related dissipation is greater, but spatially localized.
- Both for breaking and bottom friction, coefficient m depends on height H , and then on potential ϕ which is a priori unknown. That dependence cannot be implicitly solved through a matrix system. We have therefore incorporated a new algorithm of such a kind as the "firing procedure" algorithm provided for solving the potential when some dissipation occurs. That algorithm is discussed in § 2.7.

2.4 Boundary conditions

In order to be mathematically and physically well formulated, the problem to be solved should take into account the prescribed conditions at the boundaries of the domain being investigated. Physically, there can be two kinds of boundaries :

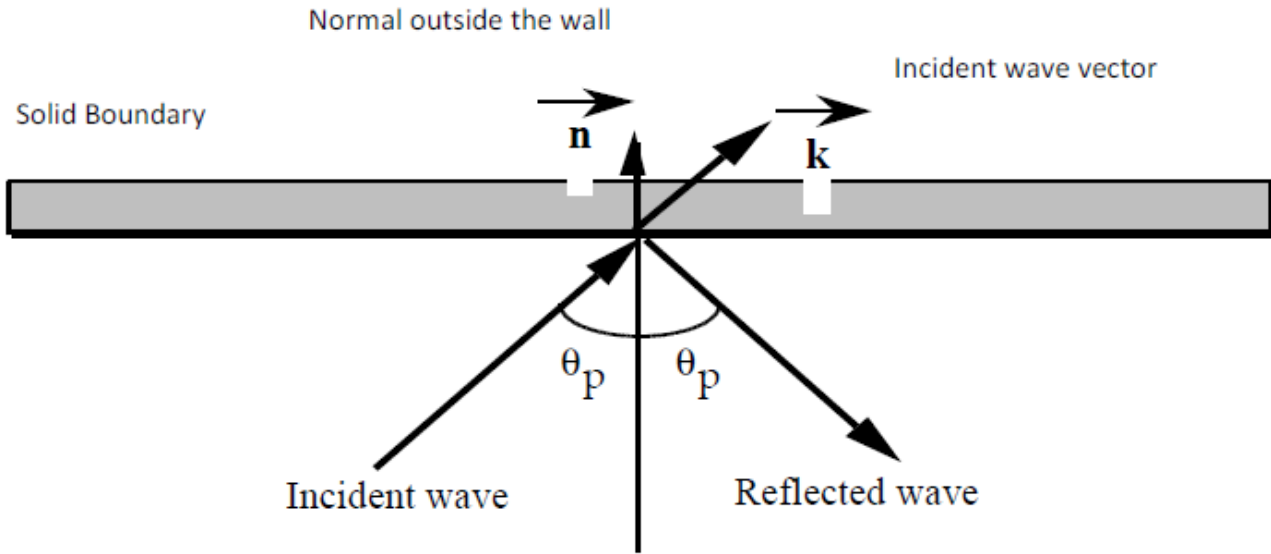
- either solid boundaries fully or partly reflecting the waves (breakwaters, quays, beaches, ...) ;
- or openings facing the open sea or an inner dock (liquid boundaries) through which waves flow in or out.

These kinds of boundary conditions should mathematically be expressed in the computation unknown, i.e. the reduced potential ϕ .

2.4.1 Solid boundaries

The instance when waves come against a wall and have a main direction of propagation which makes

an angle θ_p to the normal outside the wall will now be contemplated :



In front of that wall, incident waves and reflected waves, corresponding to potentials ϕ_{inc} and ϕ_{ref} are superposed on each other. The reflection process is modelled through a relationship between potentials ϕ_{inc} and ϕ_{ref} . Let the wall reflection coefficient, which is a priori complex, be called $Re^{i\alpha}$. Then it expresses not merely a loss of amplitude upon the reflection from the wall (coefficient R), but also a wall-induced time lag (coefficient α).

At the wall, velocities and elevations of free surface incident and reflected waves are related

$$\begin{cases} \vec{u}_{ref} \cdot \vec{n} = -Re^{i\alpha} \vec{u}_{inc} \cdot \vec{n} \\ \zeta_{ref} = Re^{i\alpha} \zeta_{inc} \end{cases} \quad (46)$$

or also, taking as the incident wave a traveling monodirectional wave in the form : $e^{i(k'x' - \omega t)}$

$$\begin{cases} \frac{\partial \phi_{ref}}{\partial n} = -Re^{i\alpha} e^{i(\vec{k}_{ref} - \vec{k}_{inc})} \frac{\partial \phi_{inc}}{\partial n} \\ \phi_{ref} = Re^{i\alpha} e^{i(\vec{k}_{ref} - \vec{k}_{inc})} \phi_{inc} \end{cases} \quad (47)$$

and, by replacing ϕ_{ref} with $\phi - \phi_{inc}$, we finally get :

$$\frac{\partial \phi_{ref}}{\partial n} - i \frac{1 - Re^{i\alpha}}{1 + Re^{i\alpha}} k \cos(\theta_p) = 0 \quad (48)$$

Thus, three data, namely R , α and θ_p are required for applying condition (48), which couples the real and imaginary components, respectively ϕ_r and ϕ_i and prevents the equation from being differently solved in ϕ_r and ϕ_i . If R is equal to 1 and α is equal to $2n\pi$ (n being an integer), it can be found that θ_p has no action. The values of the coefficient R and - more hardly - of α can be specified according to the kind of wall being investigated from references .

The angle θ_p is a priori unknown. In some simple cases, however, it may be specified by the user. An iterative procedure based upon a "prediction-correction" method could provide a more accurate assessment of θ_p . That option, however, was not adopted in ARTEMIS because of the heavy computational cost involved by that technique. Thus, the user has either to specify that angle of

incidence “in the best possible way” in the available suitable sub-routine, or to make iterations himself in order to define the values of θ_p to be specified

2.4.2 Liquid boundaries

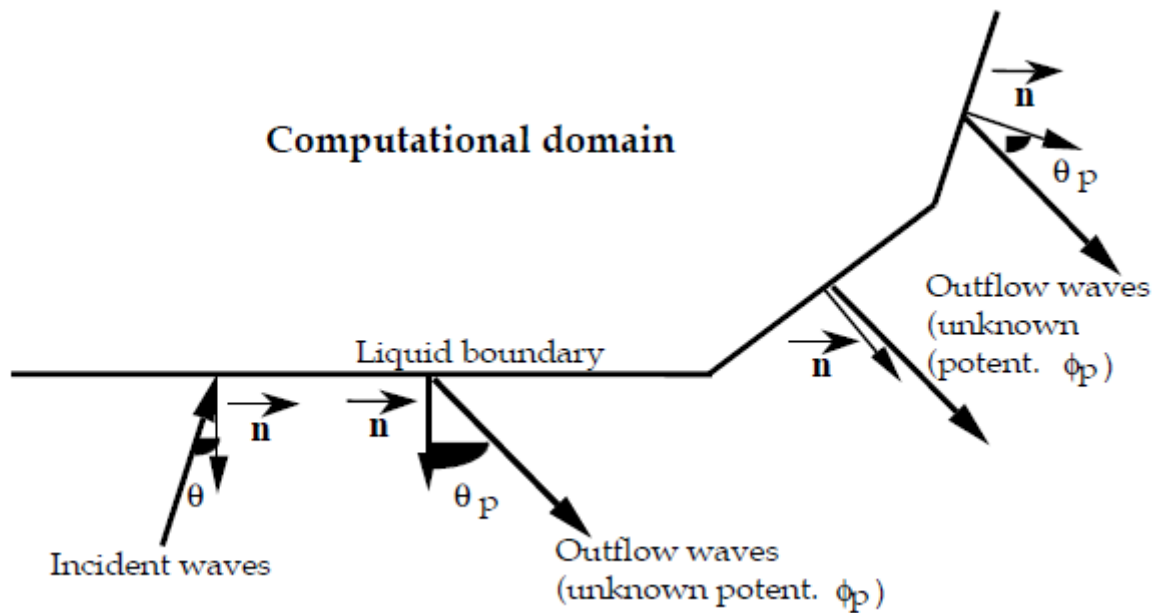
2.4.2.1 Incident wave conditions or incident potential conditions

At the seaward-facing boundaries, the boundary condition should provide for the free outflow of waves from the harbour and the inflow of waves from the open sea. At such a boundary, the potential ϕ is obtained by superposing a known potential γ (resulting from the incident waves and the waves reflected by the shore out of the computational domain) on an unknown potential ϕ_p (caused by the waves coming from the harbour): $\phi = \gamma + \phi_p$.

Writing, at such a boundary:

$$\frac{\partial \phi}{\partial n} - ik \cos(\theta_p) \phi = \frac{\partial \gamma}{\partial n} - ik \cos(\theta_p) \gamma \quad (49)$$

does express that $\phi = \gamma + \phi_p$ as long as $\frac{\partial \phi}{\partial n} - ik \cos(\theta_p) \phi$ i.e. as long as those waves making up ϕ_p are waves which travel out of the domain while making an angle θ_p with the normal to the boundary.



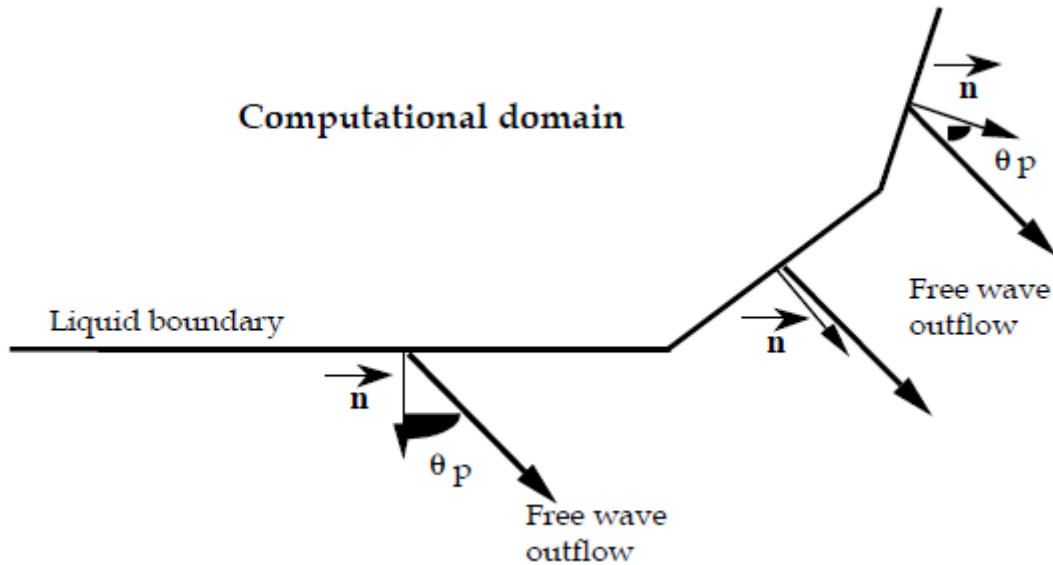
The potential γ is user-specified by means of keywords and programming in the BORH subroutine through the incident wave height and direction of propagation. In case of a random wave computation, the incident wave height to be specified is the significant height. On the other hand, the monochromatic wave height values obtained using ARTEMIS are to be compared, as required, with such data as energy height (H_e) or rms height (H_{rms}) from random wave experiments. Angle θ_p is to be specified as much as possible. In most cases θ_p is a priori unknown. It is then recommended either to set $\theta_p = 0$, or to make iterations for assessing its value, as suggested in the previous paragraph. It would also be possible to compute the difference $\phi - \gamma$ after an initial computation giving a solution for ϕ , and to derive the wave outflow angle corresponding to that difference $\phi - \gamma$.

2.4.2.2 Free outflow

In that case, $\gamma = 0$ and the boundary conditions is written as:

$$\frac{\partial \phi}{\partial n} - ik \cos(\theta_p) \phi = 0 \quad (50)$$

Once again, the angle θ_p is to be specified as best as possible, if the wave outflow direction is known a priori (refer to the following Figure 5) If no proper angle θ_p is specified, stray reflections contaminating the computational domain from the boundary will predictably be noticed.



2.5 Treatment of random mono/multi directional waves

An actual wave exhibits a random feature when its energy is determined neither by a single wave period or frequency nor by a single direction of propagation (see in § 2.1). It may be interpreted as the superposition of several monodirectional waves with different periods and random lags in relation to each other. The energy of an actual wave is the sum of monodirectional wave energies making it up.

The waves whose frequencies range from f to $f+df$ and whose directions of propagation range from θ et $\theta + d\theta$, contribute to the total energy (to within the term ρg) up to $S(f, \theta).df.d\theta$ wherein $S(f, \theta)$ denotes the spectral density of variance as already introduced in § 2.1.3).

There are various formulations for expressing the spectral density of variance according to the wave features. The most commonly used formulation was developed from the JONSWAP campaign in the North sea. It assumes that the energy spectrum can be written in the form of the following product:

$$S(f, \theta) = J(f).G(\theta) \quad (51)$$

The frequency dependency $J(f)$ is expressed by :

$$J(f) = \delta H_s^2 f_p^4 f^{-5} . \exp\left(-1.25\left(\frac{f_p}{f}\right)^4\right) \gamma^{\exp\left(-0.5\left(\frac{f/f_p-1}{\sigma}\right)^2\right)} \quad (52)$$

- H_s is the significant wave height
- f_p is the peak frequency corresponding to the spectral maximum
- σ is a dimensionless parameter which is determined as follows :

LaTeX render failed

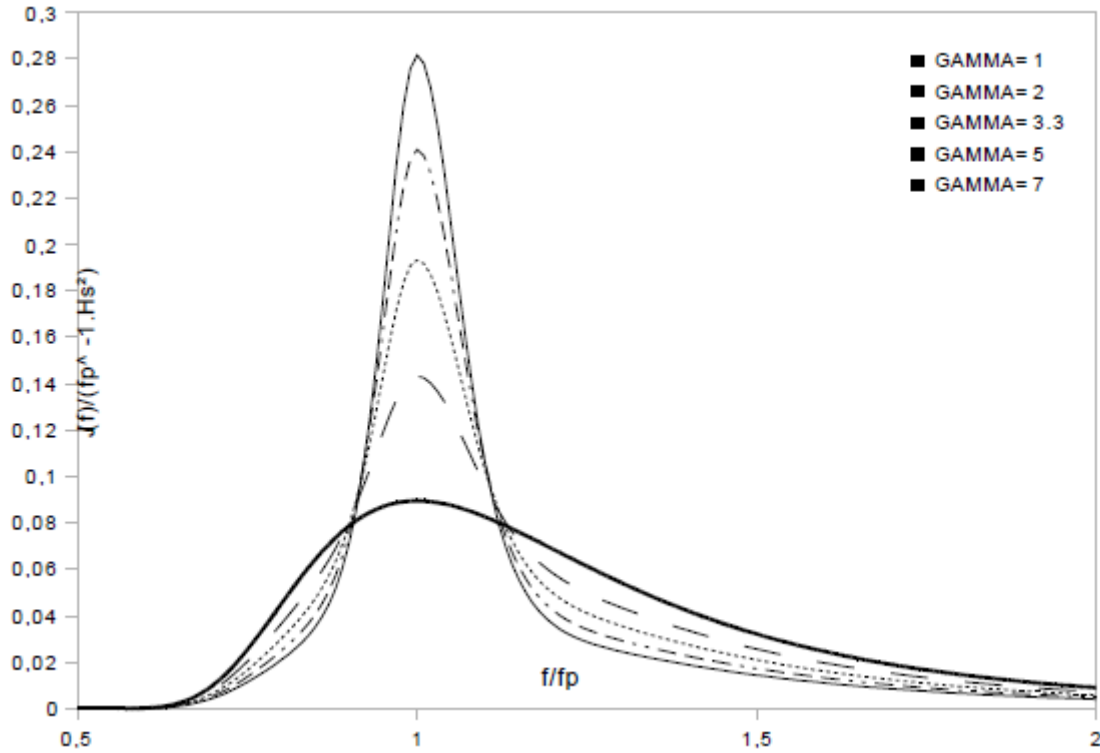
- γ is a real number usually ranging from 1 to 7. The spectrum is called a Pierson- Moskowitz

spectrum when the value is 1. Otherwise, it is called a JONSWAP spectrum. This parameter determines the spectral frequency spreading.

- δ is a weight coefficient that depends on γ :

$$\delta = \frac{0.0624}{0.230 + 0.0336\gamma - \frac{0.185}{1.9+\gamma}}$$

The trend of $J(f)$ versus f/f_p is given in for various values of γ .

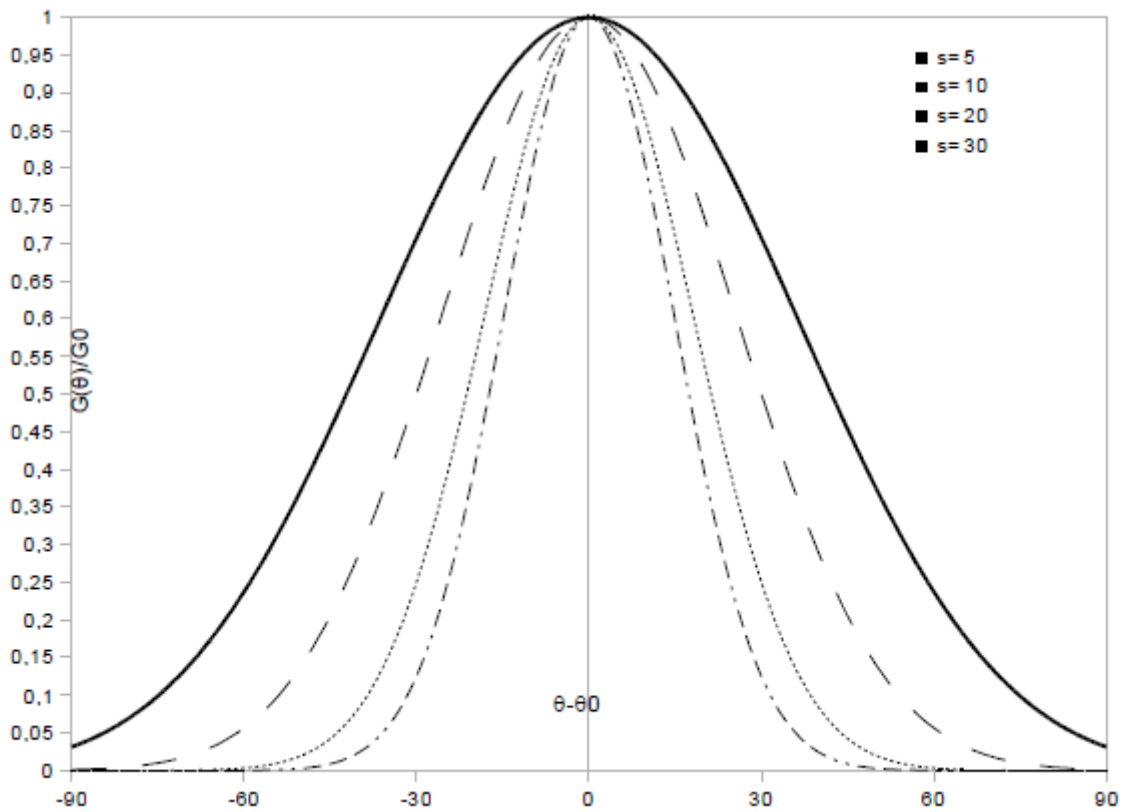


The directional distribution $G(\theta)$ is often chosen in the following for :

$$G = G_0 \cos^{2s} \left(\frac{\theta - \theta_0}{2} \right) \quad (53)$$

- θ is the angle indicating the wave propagation direction
- θ_0 denotes the main direction of wave propagation
- s is a positive real exponent of some tens (the higher s is, the less spread the directional spectrum is).
- G_0 is a constant as determined by normalizing function G : $\int_{\theta_{min}}^{\theta_{max}} G d\theta = 1$

θ_{min} and θ_{max} indicate the boundary propagation directions of the waves being studied.



For ARTEMIS, a random mono/multi-directional wave computation will consist in discretizing the axes of the incident wave energy spectrum into a sequence of values f_i for frequencies and θ_j for directions ($1 \leq i \leq \text{imax}$ and $1 \leq j \leq \text{jmax}$), then making a steady wave computation for each of these pairs by solving the mild-slope equation associated with the prescribed boundary conditions, and lastly recombining the results for providing the random wave features.

We have adopted a frequency and direction discretization of spectra which divides each of the spectra into equal energy bands. Thus, $J(f_i) \cdot \Delta f_i$ and $G(\theta_j) \cdot \Delta \theta_j$ are constant for any i and j . The maximum values of i and j , i.e. imax and jmax , are set by the user through keywords. The various steady wave computations are then made with the same incident energy, the amount of which is $J(f_i) \cdot \Delta f_i \cdot G(\theta_j) \cdot \Delta \theta_j$. It is noteworthy that the constants being used in the expressions of J and G (G_0, H_s, \dots) are purposeless for computing f_i and θ_j .

Frequency f_i denotes the i band in the $J(f)$ spectrum. It is computed so that f_i divides the i band into a couple of bands also of the same area. Practically, when it is desired to cut the spectrum into imax bands of the same area, it is actually cut into 2imax bands while systematically recording the frequencies at the limits of these bands. The f_i values are obtained taking the 2nd, 4th... recorded frequencies. The same procedure is applied to the directional spectrum $G(\theta)$.

For each pair (f_i, θ_j) , the model provides a significant wave height $H_{m0,k}$ ($1 \leq k \leq \text{imax jmax}$) at each node of the grid. Wave defined by (f_i, θ_j) at a node of the grid contributes to the total energy up to

$$\frac{1}{8} \rho g H_{e,k}^2 = \frac{1}{4} \rho g H_{m0,k}^2$$

wherein H_e , k and $H_{m0,k}$ denote the spectral energy height and significant height, respectively, as computed for waves defined by (f_i, θ_j) .

The total incident energy is $\frac{1}{8} \rho g H_{e,inc}^2$, wherein $H_{e,inc}$ is the energy height of incident waves. Since each computation is made with the same energy (the bands cutting the spectrum axes define

the same volume), $H_{e,inc}$ is identical for all the computations. The total energy at a point in the domain is:

$$\frac{1}{8}\rho g \frac{1}{i_{max} \cdot j_{max}} \sum_k H_{e,k}^2 \quad (54)$$

The resulting energy height H_e is derived there from at each node of the grid:

$$H_e = \sqrt{\frac{1}{i_{max} \cdot j_{max}} \sum_k H_{e,k}^2} \quad (55)$$

The significant wave height H_{m0} can be derived from the energy height H_e through the relation mentioned in § 2.1.3:

$$H_{m0} = \sqrt{2} H_e$$

That quantity is retrieved by ARTEMIS in the case of random waves. Assuming that the distribution of wave heights follows a Rayleigh's law, the spectral and statistical coefficients, i.e. $(H_{m0}; H_e)$ on the one hand and $(H_{1/3}; H_{rms})$ on the other hand, respectively, are equivalent to one another.

2.6 Radiation stresses; Wave-driven forcing terms

Both in monochromatic and random modes, ARTEMIS enables to compute the radiation stresses S_{xx} , S_{xy} and S_{yy}

$$S_{ij} = \frac{E}{2} \left(\frac{k_i k_j}{k^2} \frac{2C_g}{C} + \left(\frac{2C_g}{C} - 1 \right) \delta_{ij} \right)$$

- $E = \frac{1}{8}\rho g H_e^2$
- indexes i and j correspond to x or y

In regular mode (monochromatic wave), the energetic wave height H_e is the wave height directly given by ARTEMIS. In random mode, this wave height H_e is roughly proportional to the significant wave height: $H_e \approx H_s/\sqrt{2}$. Other parameters involved in the previous general equation correspond to the mean values defined in the table given above: mean wave number, mean celerities, ...

We can deduce from these radiation stresses the associated forcing F_x and F_y to be introduced in a current model (for instance TELEMAC-2D), which will be able to compute the wave-driven currents. These forcing terms are expressed as follows:

$$F_x = -\frac{1}{h} \left(\frac{\partial S_{xx}}{\partial x} + \frac{\partial S_{xy}}{\partial y} \right)$$

$$F_y = -\frac{1}{h} \left(\frac{\partial S_{xy}}{\partial x} + \frac{\partial S_{yy}}{\partial y} \right)$$

with: h the water depth at rest

Remark 1: Simulation of the wave-driven currents with the TELEMAC-2D software is performed through an external coupling between ARTEMIS and TELEMAC-2D.

Remark 2: in the monochromatic mode, oscillations are often observed on the wave heights results. They may be generated by (i) real reflections against solid obstacles which propagate within the

computational domain, or by (ii) undesired reflections due to boundary conditions problems. These undesired reflections may greatly alter the computation of the wave forcing terms. To avoid this problem, it is possible in release 6, for the monochromatic mode, to impose an automatic smoothing of the wave heights field, in order to filter these oscillations, whose characteristic « wave » length is half the « wave » length of the waves. To achieve this smoothing, one has to activate the new keyword:

WAVE HEIGHTS SMOOTHING

whose mnemonic is LISHOU (default value : NO).

However, it is preferable to use ARTEMIS in random mode to get satisfactory estimation of the wave forcing terms

2.7 Digital solution through the finite element method

2.7.1 System of equations to be solved

The differential system to be solved, the unknown of which is the reduced potential ϕ , comprises the amended mild-slope equation as well as equations expressing the boundary conditions at each boundary of the computational domain.

$$\phi(x, y) = \phi_r(x, y) + i\phi_i(x, y) \quad (56)$$

ϕ_r and ϕ_i being real functions depending on x and y, which are the real and imaginary components of ϕ , respectively. That function f will hereinafter be termed the potential by misuse of language, though multiplying it by $f(z)e^{-i\omega t}$ is required for getting the total potential Φ .

The differential system to be solved is then as follows:

$$\begin{cases} \nabla \cdot (CC_g \nabla \phi) + CC_g(k^2 + ik\mu)\phi = 0 \\ \text{and according to the boundary conditions} \\ \frac{\partial \phi}{\partial n} - i \frac{1-Re^{i\alpha}}{1+Re^{i\alpha}} k \cos(\theta_p) \phi = 0 & \text{(solid wall)} \\ \frac{\partial \phi}{\partial n} - ik \cos(\theta_p) \phi = -ik \cos(\theta_p) \gamma & \text{(incident wave)} \\ \frac{\partial \phi}{\partial n} - ik \cos(\theta_p) \phi = 0 & \text{(free outflow)} \end{cases} \quad (57)$$

2.7.2 Variational formulation of the problem

The mild-slope equation is elliptical and can only be solved through the finite element method. The differential system (57) to be solved is then transformed through a variational formulation of the problem.

if Ω is the computational domain, Γ its boundary and if Ψ denotes a basic function of class C1 on W, the variational formulation of the mild-slope equation is expressed as :

$$\forall \Psi \in C_1(\Omega), \int_{\Omega} \Psi \left(\nabla \cdot (CC_g \nabla \phi) + CC_g(k^2 + ik\mu)\phi \right) d\Omega = 0 \quad (58)$$

The normal derivative of the potential at the boundary Γ of the domain will then come out. It can be replaced by its expression as given by the boundary conditions, which is in the following form:

$$\frac{\partial \phi}{\partial n} = (A\phi_i + B\phi_r) + i(-A\phi_i + B\phi_r) - E - iF \quad (59)$$

wherein A, B, E and F are real numbers determined by the kinds of boundary conditions as specified in all the parts of Γ .

$$\begin{cases} \int_{\Gamma} \Psi CC_g (A\phi_i + B\phi_r) d\Gamma - \int_{\Omega} CC_g \nabla \Psi \nabla \phi_r d\Omega + \int_{\Omega} CC_g \Psi (k^2 \phi_r - k\mu \phi_i) \Omega = \int_{\Gamma} \Psi CC_g E d\Gamma \\ \int_{\Gamma} \Psi CC_g (A\phi_r + B\phi_i) d\Gamma - \int_{\Omega} CC_g \nabla \Psi \nabla \phi_i d\Omega + \int_{\Omega} CC_g \Psi (k^2 \phi_i - k\mu \phi_r) \Omega = \int_{\Gamma} \Psi CC_g F d\Gamma \end{cases} \quad (60)$$

2.7.3 Digital solution algorithm

2.7.3.1 Discretization. Setting into a matrix form

Linear basic functions are chosen on triangular elements. These functions are really Y functions of class C_1 . ϕ_r and ϕ_i are resolved on these basic functions Ψ .

$$\begin{cases} \phi_r = \sum_{j=1}^{NPOIN} \phi_r^j \Psi_j^e \\ \phi_i = \sum_{j=1}^{NPOIN} \phi_i^j \Psi_j^e \end{cases} \quad (61)$$

NPOIN is the total number of points in the grid. The Ψ_j^e functions are linear at each element with an e index and are null everywhere else.

By incorporating these expressions into the previously described set of equations, a matrix system can be obtained in the following form:

$$\begin{bmatrix} AM & BM \\ -BM & AM \end{bmatrix} \cdot \begin{bmatrix} \phi_r \\ \phi_i \end{bmatrix} = \begin{bmatrix} CV1 \\ CV2 \end{bmatrix} \quad (62)$$

In order to obtain a positive diagonal along AM (the reason for it will be explained later), the couple of previous equations have multiplied by -1. AM is a matrix provided with NPOIN x NPOIN dimensions and having as a general term:

$$AM_{jk} = - \int_{\Gamma} \Psi_j \Psi_k CC_g B d\Gamma + \int_{\Omega} CC_g \nabla \Psi_j \nabla \Psi_k d\Omega - \int_{\Omega} CC_g k^2 \Psi_j \Psi_k \Omega \quad (63)$$

BM is also a NPOIN x NPOIN-dimensional matrix having as a general term:

$$BM_{jk} = - \int_{\Gamma} \Psi_j \Psi_k CC_g A d\Gamma + \int_{\Omega} CC_g k\mu \Psi_j \Psi_k \Omega \quad (64)$$

ϕ_r and ϕ_i are a couple of NPOIN-dimensional vectors comprising ϕ_r^j and ϕ_i^j components. CV1 and CV2 are a couple of vectors that are null everywhere but at the edge of the grid. The NPTFRdimensioned edge vectors have the following general terms:

$$CV1_j = - \int_{\Gamma} \Psi_k CC_g E d\Gamma \quad (65)$$

$$CV2_j = - \int_{\Gamma} \Psi_k CC_g F d\Gamma \quad (66)$$

If the diagonal term in the AM matrix is positive, a suitable preconditioning will become applicable to the system, what will make the resolution much faster. Regardless of the edge term, AM_{jk} has approximately the sign of $1/(\Delta x)^2 - k^2$ (Δx denoting the mean size of the meshes). As long as $\Delta x < L/2\pi$, i.e. as long as $\Delta x < L/7$, the diagonal term remains positive. Due to that condition, the selected mesh size shall be smaller than L/7. It seems that at least 7 grid nodes per wavelength

are required.

In simple examples, however, we have tested coarser grids having 4-5 nodes per wavelength. The results for these computational conditions remain satisfactory even though, in such a case, no preconditioning is applied to the matrix system. With less than 4 nodes per wavelength, the spatial resolution is too low and the information is deteriorated.

2.7.3.2 Computing the matrices

The AM and BM matrices as well as the CV1 and CV2 members are computed by the subroutines of the finite element library BIEF. These matrices are computed for one element at a time. Only their diagonals are put together. For assisting computations, we have taken the product CCg with P1 discretization on each of the triangles. The coefficients A, B, E and F are taken as constants on each of the edge segments (P0 discretization).

C and Cg are computed from the wave number k as obtained through the Airy's dispersion relation:

$$\frac{\omega^2}{gk} = th(kh) \quad (67)$$

Taking $x = kh$ and $y = \frac{\omega^2 h}{gk}$, the dispersion relation becomes $y = xth(x)$. A precise, explicit form of it is used at 0.510^{-3} :

$$x = \sqrt{y} \left(y + \frac{1}{1 + \sum_{j=1}^5 p_j y^j} \right)^{1/2} \text{ with } \begin{cases} p_1 = 0.6522 \\ p_2 = 0.4622 \\ p_3 = 0.0000 \\ p_4 = 0.0864 \\ p_5 = 0.0675 \end{cases} \quad (68)$$

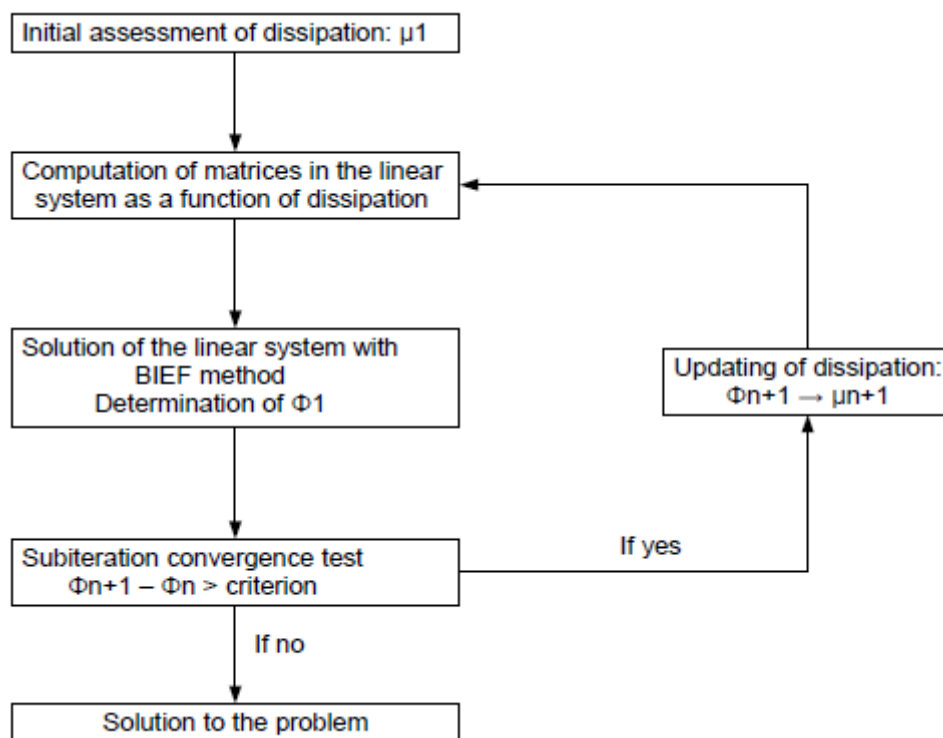
2.7.3.3 Solution of linear system

The dissipation process is taken into account by a new term in the BM matrix. That term is related to a dissipation coefficient μ which is explained in § 2.3 depending on whether either or both of breaking and bottom friction are considered. The difficulty lies in that μ always depends on H, i.e. on the unknown quantity ϕ in the problem. Thus, the linear system cannot be solved directly. The solution that was adopted is a prediction-correction scheme. A first assessment of the coefficient m is made throughout the computational domain. The linear system is then solved though one of the methods as proposed in the BIEF library of the TELEMAC system

- Either using an iterative method : conjugate gradient, conjugate gradient on normal equation, GMRES solver, ... ;
- Or using the direct solver.

A first solution ϕ_1 is obtained. It makes it possible to compute a second assessment of dissipation μ_2 . That second assessment is used for solving again the linear system and obtaining a solution μ_2 . The process is interrupted when the user thinks that the difference between the two successive solutions is sufficient or when the number of such repeated subiterations has reached a boundary value which is also set by the user. A sub-relaxation was prescribed for that process in order to assist the convergence of the method. That relaxation coefficient is specified by the user..

The overall solution algorithm is summarised in the diagram below :



It seems that the most suitable iterative resolver for the linear system being considered is the direct solveur. That resolver is taken by default by the model. The matrix may previously be diagonally preconditioned. The user, however, may freely choose any other resolver and any other preconditioning as proposed through keywords in BIEF (see in § 8).

As regards random waves, the previously described algorithm will be followed as many times as necessary because of the energy spectrum discretization (i.e. $i_{max} \times j_{max}$ times). In order to improve the wave computation time corresponding to the pairs (f_{i+1}, θ_j) , (f_i, θ_{j+1}) , and (f_{i+1}, θ_{j+1}) , however, those values which were previously computed for the pair (f_i, θ_j) are used for initializing the unknowns and the dissipation coefficient.

Practically, it appears that the number of iterations implied by the resolver convergence is substantially the same as the number of nodes in the NPOIN grid. Since the required time for an iteration (consisting in a product of a matrix by a vector) is proportional to the number of nodes, the time for a computation by ARTEMIS is globally proportional to the square of the number of nodes. The proportionality coefficient depends, among others, on the kind of computational domain boundaries, on bathymetry and on the computing machine. The more numerous the liquid boundaries are, the smaller it is. The greater the depths are, the higher it is. This is because, as it is well known, a diffusion matrix can more hardly be inverted than a mass matrix. Now, the greater the depth is, the higher the phase and group velocities are. The diffusion matrix then becomes increasingly prevailing in relation to the mass matrix. The required number of sub-iterations for providing the convergence of the computation of the dissipation terms is usually 4 or 5.

2.7.4 Matrix in the case of rapidly varying topography

2.7.4.1 Variational formulation of the problem

With the same idea, the variational formulation becomes for a basic function Ψ (C1):

$$\int_{\Omega} \Psi \left(\nabla \cdot (CC_g \nabla \phi) + CC_g (k^2(1+f) + ik\mu) \phi \right) d\Omega = 0$$

An integration in parts gives :

$$\int_{\Gamma} \Psi CC_g \frac{\partial \phi}{\partial n} d\Gamma - \int_{\Omega} CC_g \nabla \Psi \nabla \phi d\Omega + \int_{\Omega} CC_g (k^2(1+f) + ik\mu) \phi \Psi d\Omega$$

The boundary conditions can be expressed as :

$$\frac{\partial \phi}{\partial n} = (A\phi_i + B\phi_r) + i(-A\phi_i + B\phi_r) - E - iF$$

Separating imaginary and real parts (f is real) :

$$\begin{cases} \int_{\Gamma} \Psi CC_g (A\phi_i + B\phi_r) d\Gamma - \int_{\Omega} CC_g \nabla \Psi \nabla \phi_r d\Omega + \int_{\Omega} CC_g \Psi (k^2(1+f)\phi_r - k\mu\phi_i) \Omega = \int_{\Gamma} \Psi CC_g E d\Gamma \\ \int_{\Gamma} \Psi CC_g (A\phi_r + B\phi_i) d\Gamma - \int_{\Omega} CC_g \nabla \Psi \nabla \phi_i d\Omega + \int_{\Omega} CC_g \Psi (k^2(1+f)\phi_i - k\mu\phi_r) \Omega = \int_{\Gamma} \Psi CC_g F d\Gamma \end{cases}$$

2.7.4.2 Matrix

Matrix AM, expressed in 2.7.3.1 is modified. BM doesn't change and the second member (CV1 and CV2) is unchanged too.

$$AM_{jk} = - \int_{\Gamma} \Psi_j \Psi_k CC_g B d\Gamma + \int_{\Omega} CC_g \nabla \Psi_j \nabla \Psi_k d\Omega - \int_{\Omega} CC_g k^2(1+f) \Psi_j \Psi_k \Omega$$

2.7.4.3 Computation and implementation

The user include the keyword «RAPIDLY VARYING TOPOGRAPHY » in the case file. This keyword can take the value 0, 1, 2 or 3. Default value is 0.

- 0 : correspond to classical mild-slope equation.
- 1 : gradient effects are taken into account.
- 2 : curvature effects are taken into account.
- 3 : both gradient and curvature effects are taken into account.

This value is red in the BERKHO routine. Is the value is different from 0 routine PENTCO is called. In PENTCO we compute the term « f » in the whole domain. Depending on the user's choice, Matrix AM is modified in BERKHO.

To get the value of « f », PENTCO calls to functions: FCTE1.f, for gradient terms:

$$E_1(kh) = \frac{x^4 + 4x^3 \sinh(x) - 9 \sinh(x) \sinh(2x) + 3x(x + 2 \sinh(x)) \cdot (\cosh^2(x) - 2 \cosh(x) + 3)}{6n(x + \sinh(x))^3}$$

$$\text{with } x = 2kh$$

FCTE2.f, for curvature terms.

$$\tilde{E}_2(kh) = \frac{(\tanh(x) - x) \cdot \cosh(x)}{x(x + \sinh(x))^2}$$

If x is too small (lower à 10^{-3}), the value of the function is simply:

$$\tilde{E}_2(kh) = \frac{1}{12}$$

We note that

$$\frac{E_2(kh)}{k_0} = 2h.\tilde{E}_2(kh)$$

At the end, PENTCO computes:

- RAPIDLY VARYING TOPOGRAPHY = 1

$$f = E_1(kh).(\nabla h)^2$$

- RAPIDLY VARYING TOPOGRAPHY = 2

$$f = 2h\tilde{E}_2(kh).\Delta h$$

- RAPIDLY VARYING TOPOGRAPHY = 3

$$f = E_1(kh).(\nabla h)^2 + 2h\tilde{E}_2(kh).\Delta h$$

In any other case, the value of f is 0.

3. Inputs and outputs

3.1 Preliminary remarks

During a computation, the ARTEMIS software uses a number of input and output files, some of which are optional.

The input files are the following:

- The geometry file
- The steering file
- The boundary conditions file
- The bottom topography file
- The FORTRAN file
- The binary data files
- The formatted data files

The output files are the following:

- The results file
- The printout listing
- The formatted data file
- The binary data file

3.2 The files

3.2.1 The steering file

This is a text file created by FUDAA-PREPRO or directly by the text editor. In a way, it represents the control panel of the computation. It contains a number of keywords to which values are assigned. If a keyword is not contained in this file, ARTEMIS will assign it the default value defined in the dictionary file. If such a default value is not defined in the dictionary file, the computation will stop with an error message. For example, the command `WAVE PERIOD = 10.` enables the user to specify that the wave period is 10 seconds.

ARTEMIS reads the steering file at the beginning of the computation.

The dictionary file and steering file are read by a utility called DAMOCLES, which is included in ARTEMIS. Because of this, when the steering file is being created, it is necessary to comply with the rules of syntax used in DAMOCLES (this is done automatically if the file is created with FUDAAPREPRO). They are briefly described below and an example is given in Appendix 8. The rules of syntax are the following:

- The keywords may be of Integer, Real, Logical or Character type.
- The order of keywords in the steering file is of no importance.
- Each line is limited to 72 characters. However, it is possible to pass from one line to the next as often as required, provided that the name of the keyword is not split between two lines.
- The signs ":" or "=" can be used indiscriminately as separator for the name of a keyword and its value. They may be preceded or followed by any number of spaces. The value itself may appear on the next line. For example: `WAVE PERIOD = 10.orWAVE PERIOD: 10.or againWAVE PERIOD =10.`
- Characters between two "/" on a line are considered as comments. Similarly, characters between a "/" and the end of line are also considered as comments. For example: `SOLVER = 3 / Conjugated gradient on normal equation`
- A line beginning with "/" is considered to be all comment, even if there is another "/" in the line. For example: `/ The geometry file is ./mesh/geo`
- A line beginning with "/" is considered to be all comment, even if there is another "/" in the line. For example: `/ The geometry file is ./mesh/geo`
- When writing integers, do not exceed the maximum size permitted by the computer (for a computer with 32-bit architecture, the extreme values are -2 147 483 647 to + 2 147 483 648. Do not leave any space between the sign (optional for the +) and number. A full stop is allowed at the end of a number.
- When writing real numbers, the full stop and comma are accepted as decimal points, as are E and D formats of FORTRAN. (1.E-3 0.001 0,001 1.D-3 represent the same value).

- When writing logical values, the following are acceptable: 1 OUI YES .TRUE. TRUE VRAI and 0 NON NO .FALSE. FALSE FAUX.
- Character strings including spaces or reserved symbols (" / " , " : " , " = " , " & ") must be placed between apostrophes ('). The value of a character keyword can contain up to 144 characters. As in FORTRAN, apostrophes in a string must be doubled. A string cannot begin or end with a space. For example :
TITLE = 'CAS DE L"EPI'

In addition to keywords, a number of instructions or meta-commands interpreted during sequential reading of the steering file can also be used :

- Command &FIN indicates the end of the file (even if the file is not finished). This means that certain keywords can be deactivated simply by placing them behind this command in order to reactivate them easily later on. However, the computation continues.
- Command &ETA prints the list of keywords and the value that is assigned to them when DAMOCLES encounters the command. This will be displayed at the beginning of the printout listing (see § 3.2.5).
- Command &LIS prints the list of keywords. This will be displayed at the beginning of the printout listing (see § 3.2.5).
- Command &IND prints a detailed list of keywords. This will be displayed at the beginning of the printout listing (see § 3.2.5).
- Command &STO stops the program and the computation is interrupted.
- The name of this file is given by using the keyword: STEERING FILE.

3.2.2

It is a SELAFIN-formatted binary file which can then be read by RUBENS or FUDAA-PREPRO and is generated either by the MATISSE software or by the STBTTEL module (from the file(s) originating from of mesh generator). The SELAFIN format structure is described in Appendix 11.

This file contains all the information concerning the mesh, i.e. the number of mesh points (NPOIN variable), the number of elements (NELEM variable), the number of nodes per element (NDP variable), arrays X and Y containing the coordinates of all the nodes and array IKLE containing the table of connectivities.

This file can also contain bottom topography information at each mesh point, if such bottom topography has been interpolated during the running of module STBTTEL (for more information, see the User's Manual for this software).

ARTEMIS stores information on the geometry at the start of the results file. Because of this, the computation results file can be used as a geometry file if a new simulation is to be run on the same mesh.

The name of this file is given with the keyword: GEOMETRY FILE In order to reach a good accuracy, it is necessary to build a mesh using at least 3 points by wave length. In addition, during the numerical treatments, a mesh with 7 points by wave length leads to matrices with non zero diagonals that allows the use of preconditioning in order to improve the computation speed.

When using random waves, a good method is to build a mesh using 7 points by wave length for the peak period. When considering the lowest period, the preconditioning will be probably not usable, but the refinement of the mesh will not be under the criteria of 3 points by wave length.

3.2.3 The boundary conditions file

This is a formatted file generated automatically by STBTTEL. It can be modified with a standard text editor. Each line of the file is dedicated to one point on the mesh boundary. The numbering used for points on the boundary is that of the file lines. First of all, it describes the contour of the domain trigonometrically, starting from the bottom left-hand corner (X + Y minimum) and then the islands in a clockwise direction.

See § 5.2 for a fuller description of this file.

The file name is given with the keyword: BOUNDARY CONDITIONS FILE.

3.2.4 The result file

This is the file in which ARTEMIS stores information during the computation. It is normally in Selafin format. It contains first of all information on the mesh geometry, then the names of the stored variables. It then contains the period for each computed period and the values of the different variables for all mesh points.

Its content is dependent on the value of the following keywords:

- GRAPHIC PRINTOUT PERIOD: fixes the period for outputs when using computation over multiple periods (for example, case of period scanning to search resonance period).
- VARIABLES FOR GRAPHIC PRINTOUTS : this is used to specify the list of variables to be stored in the results file. Each variable is identified by an identifier (string with up to 8 characters); these are listed in Appendix 4 with the description of this keyword.

The name of this file is given with the keyword: RESULTS FILE

3.2.5

This is a formatted file created by ARTEMIS during the computation. It contains a report on the execution of ARTEMIS. Its content varies in accordance with the values of the following keywords:

- LISTING PRINTOUT PERIOD : this fixes the period between two periods output when calculating several wave periods (period scanning). The given value is the number of periods. For example, the following sequence:
LISTING PRINTOUT PERIOD = 2

will produce a listing printout every 2 wave periods. Moreover, irrespective of the period indicated by the user, the last period is systematically printed.

- LISTING PRINTOUT: this cancels the listing printout if the value is NO (the

listing printout then only contains the program heading and normal end indication). However, this is not advisable in any circumstances.

- **VARIABLES TO BE PRINTED** : this is used to specify the list of variables for which all values will be printed at each mesh point. This is a debugging option offered by ARTEMIS that should be handled with caution so as to avoid creating an excessively large listing printout.
- **INFORMATION ABOUT SOLVER**: if this is required, at each printed period the user will have the number of iterations necessary to achieve the accuracy required during the mild slope equation calculation, or by default that reached at the end of the maximum number of iterations authorised. In addition, the user will obtain the number of sub-iterations necessary to take into account the dissipation phenomena.

The name of this file is managed directly by the ARTEMIS start-up procedure. In general, it has the name of the steering file associated with the suffix .sortie.

3.2.6 The Fortran user file

The FORTRAN file contains all the ARTEMIS subroutines modified by the user and those that have been specially developed for the computation.

This file is compiled and linked so as to generate the executable program for the simulation.

The name of this file is given with the keyword: FORTRAN FILE.

3.2.7 The ancillary files

Other files may be used by ARTEMIS.

- One or two binary data files, specified by the keywords BINARY DATA FILE 1 and BINARY DATA FILE 2 . These files can be used to provide data to the program, data reading being of course managed by the user within the Fortran program. The logical units affected to this two files are respectively 24 and 25.
- One or two formatted data files, specified by the keywords FORMATTED DATA FILE 1 and FORMATTED DATA FILE 2. These files can be used to provide data to the program, data reading being of course managed by the user within the Fortran program. The logical units affected to this two files are respectively 26 and 27.
- A binary results file specified by the keyword BINARY RESULTS FILE. This file can be used to store additional results. Write operations on the file are managed by the user in the Fortran program. The logical units affected to this file is 28.
- A formatted results file specified by the keyword FORMATTED RESULTS FILE. This file can be used to store additional results. Write operations on the file are managed by the user in the Fortran program. The logical units affected to this file is 29.

Read and write operations on these files must be managed completely by the user. Management can be done from any point accessible to the user.

For example, using a file to provide the initial conditions will mean managing it with the subroutine CONDIH. Similarly, using a file to introduce boundary conditions can be done in the BORH subroutine.

3.2.8 The dictionary file

This file contains all information on the keywords (name in French, name in English, default values, type, documentation on keyword, information required by FUDAA-PREPRO). This file can be consulted by the user but must under no circumstances be modified.

3.2.9 The libraries

When a computation is started, the main program written by the user is compiled and then linked to standard libraries in order to generate the program that is then run.

During the link phase, the following libraries are used:

- Library artemis: this library contains the subroutines that are specific to the ARTEMIS computation code .
- Library bief: this library contains all the computation modules concerning operations of the finite-element type (operations on matrices and vectors). It is common to all the simulation codes developed by the Laboratoire National d'Hydraulique within the TELEMAT structure ("bief" stands for "Bibliothèque d'Eléments Finis" - Finite Elements Library).
- Library damo: this library contains the subroutines that manage the reading of key words stored in the steering file.
- Library special : this library contains the subroutines that manage the clean interrupt of the computation by the user during a simulation.
- Library parallel : this library contains the subroutines that manage exchanges between sub-models during parallel computation. All the calls to MPI functions are located in this library
- Library paravoid : this library contains the same subroutines as library parallel but these subroutines are empty in order to allow a scalar computation

3.3 File binaries

The binary of a file is the method used by the computer for storing the information physically on the disk (in contrast to storage in ASCII form, which is used by the formatted files). The file binary parameters may be defined using a number of keywords. ARTEMIS recognises three types of binary: the standard binary of the computer on which the user is working, the IBM binary enabling a file created on an IBM computer to be re-read, and the IEEE binary that can be used, for example, to generate a file on a Cray computer that can be read by a workstation (provided that the appropriate subroutines are included when ARTEMIS is installed on the computer). The following keywords can be used:

- GEOMETRY FILE BINARY, for the geometry file,
- RESULTS FILE BINARY for the results file.

In either case, the default value specified in the dictionary file is 'STD' (default binary of the computer that is being used).

3.4 Bathymetric data

Bathymetric data may be supplied to ARTEMIS at three levels:

- Either directly in the geometry file by a bathymetry value associated with each mesh node. In this case, the bathymetric data are processed when the STBTTEL module is run before ARTEMIS is started. STBTTEL reads the information in one or more bottom topography files (5 at most) and interpolates at each point in the domain.
- Or in the form of a cluster of points with elevations that have no relation with the mesh nodes, during the ARTEMIS computation. ARTEMIS then makes the interpolation directly with the same algorithm as STBTTEL. The file name is provided by the keyword BOTTOM TOPOGRAPHY FILE. In contrast to STBTTEL, ARTEMIS only manages one bottom topography file. This may be in SINUSX format or more simply a file consisting of three columns X,Y,Z.
- Or directly in the Fortran file using the CORFON subroutine. This way is in particular used when building schematic test cases

In all cases, ARTEMIS offers the possibility of smoothing the bottom topography in order to obtain a more regular geometry. The smoothing algorithm can be iterated several times depending on the degree of smoothing required. The keyword BOTTOM SMOOTHING then defines the number of iterations carried out in the subroutine CORFON. The default value of this keyword is 0 (see also programming of the user subroutine CORFON in § 9.1).

4. Initial conditions

The purpose of the initial conditions is to define the state of the model at the time the simulation begins. The initial state must be defined by the user. This is done by using key words in simple cases, or by programming in more complex ones.

4.1 Setting using key words

In all cases, the nature of the initial conditions is fixed by the key word INITIAL CONDITIONS. This may have any of the following four values:

- 'ZERO ELEVATION': Initialises the elevation of the free surface at 0. The initial depths of water are therefore calculated from the bottom elevation.
- 'CONSTANT ELEVATION': Initialises the elevation of the free surface at the value supplied by the key word INITIAL ELEVATION. The initial depths of water are therefore calculated by taking the difference between the free surface elevation and the bottom elevation.
- 'CONSTANT DEPTH': Initialises depths of water at the value supplied by the key word INITIAL DEPTH.

- 'PARTICULAR': The initial conditions are defined by the subroutine CONDIH (see § 4.2). This solution must be used whenever the initial conditions of the model do not correspond to one of the three cases above.

It is important to stress that the initial state must not contain any dry area in the computational domain. If this is the case, ARTEMIS displays an error message on the control printout.

4.2 Setting using the subroutine CONDIH

The subroutine CONDIH must be programmed whenever the key word INITIAL CONDITIONS has the value 'PARTICULAR'.

The subroutine CONDIH initialises successively the depth of water, the wave number, the phase velocity and the group velocity. The part of the subroutine concerning depth initialisation is divided into two zones, the first corresponding to "simple" initial conditions (defined by key word) and the second to "particular" conditions.

By default, the standard version of the subroutine CONDIH causes the calculation to stop if the key word INITIAL CONDITIONS is set at "PARTICULAR" without the subroutine actually being modified.

The user is free to fill in this subroutine as he wishes. For example, he may reread information from a formatted or binary file, using the key words FORMATTED DATA FILE or BINARY DATA FILE offered by ARTEMIS.

5. Boundary conditions

Boundary conditions are given for each of the boundary points.

Physically speaking, the boundaries of a domain may be of two types: either solid walls (dikes, piers, beaches) that reflect the waves to varying degrees, or else seaward or landward openings through which the waves enter or leave.

Like all modules of the TELEMAC system, ARTEMIS uses a local numbering system for boundary points. This is done as follows: starting from the lower left boundary point of the model ($X + Y$ minimum), the entire outer boundary is numbered trigonometrically, and then the islands are numbered clockwise.

The boundary conditions (wall, liquid boundary, etc.) apply to segments, but the user assigns the type of condition to boundary points. To apply a condition to a segment, it is necessary to configure the point at the start of the segment. It should be noted that this convention is valid for all the simulation modules of the TELEMAC system (see also the example given in § 6.2).

The type of boundary condition (solid or liquid boundary) is read from the boundary conditions file. In contrast, the type of waves that are to be processed (for example, mono-directional or multidirectional) and the

possible assigned values (if there is one, e.g. the height of the incident waves) can be given in the file of parameters or in the Fortran file (programming of the subroutine BORH for example).

In simple cases, boundary conditions may be imposed using key words. However, if the values are complex ones, it is necessary to carry out programming with the subroutine BORH.

5.1 Description of different types of condition

The type of boundary condition at a given point is specified in the boundary conditions file. The structure of this file is identical to that used by the other codes of the TELEMAC system. However, in the case of ARTEMIS, only three columns are used, the first one and the last two. The type of boundary condition is defined by the variable LIHBOR situated in the first column of the file. This variable may take the following values:

- Liquid boundary of incident wave type: LIHBOR=1 (KINC)
- Liquid boundary of incident potential type: LIHBOR=7 (KPOT)
- Liquid boundary of free exit: LIHBOR=4 (KSORT)
- Solid boundary (wall): LIHBOR=2 (KLOG)

The prescribed wave type boundary is in fact of very limited use, such as for example in the case of a wave generator that receives no reflected waves. Thus, this option is no longer available in ARTEMIS.

In most cases, it is possible to fix the type of boundary in the mesh generator, by setting a colour code. Each colour code corresponds to a particular type of boundary (wall, liquid boundary with prescribed waves, etc.).

5.2 The boundary conditions file

This file is normally supplied by MATISSE, STBTTEL or FUDAA-PREPRO, but it can be created and modified with a text editor. Each line of the file is devoted to one point of the mesh boundary. The edge points are numbered in the same way as the lines of the file. The contour of the domain is first numbered trigonometrically and then the islands in the opposite direction.

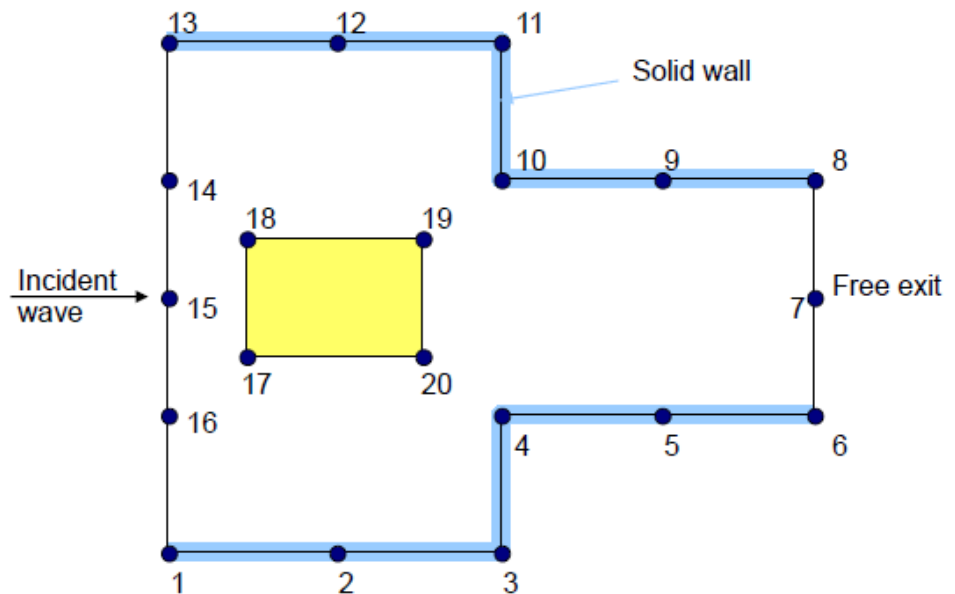
The following values are given for each point:

LIHBOR, LIUBOR, LIVBOR, HBOR, UBOR, VBOR, AUBOR, LITBOR, TBOR, ATBOR, BTOR, N, K

- LIHBOR is the code for the type of boundary. It is described in § 5.1.
- N represents the general number of the edge point.
- K represents the point number in the edge point numbering system.

The other values are not taken into account in ARTEMIS, and are ignored during the calculation.

The example below shows a file of boundary conditions for a schematic case.



1 st column Type of boundary condition		Penultimate column General number	Last column Boundary node number
1	1
2	2
2	3
2	4
2	5
4	6
4	7
4	8
2	9
2	10
2	11
2	12
1	13
1	14
1	15
1	16
2	17
2	18
2	19

5.3 Processing of solid boundaries (LIHBOR=KLOG)

A solid boundary may reflect waves to varying degrees. It is therefore necessary to characterise the wall reflection coefficient and also the possible phase shift of the reflected waves in relation to the incident waves, produced by the wall.

It must be stressed that in the case of a solid, the user must estimate beforehand and as well as possible the angle of incident wave (or of main incident wave constituent), even if this value is calculated by ARTEMIS.

All the characteristics of the wall are determined in the subroutine BORH. The values of the following parameters are specified for each solid boundary (identified by the number of the boundary nodes):

- RP : Reflection coefficient (between 0 and 1, with 1 corresponding to a

- perfectly reflecting wall, and 0 to a completely absorbing one)
- TETAP : Angle of incident wave attack in relation to the outward normal, expressed in degrees in the clockwise direction. It should be noted that the sign of this value is of no importance, since it only appears in the calculation in cosine form.
 - ALFAP : Value of the phase shift produced by the wall (positive if the reflected wave is delayed in relation to the incident wave), expressed in degrees. If ALFAP=0, the coefficient RP corresponds exactly to the definition of the reflection coefficient, i.e. $H_{totale} = H_{incidente} (1 + RP)$.

In the absence of any special programming in BORH, solid walls are all considered as being perfectly reflecting and do not produce any phase shift (RP=1, TETAP=ALFAP=0).

The following example shows a definition of two solid boundaries. The first is a wall with pure reflection, and the second one that is partially reflecting, generating a delay of 45° when the incident wave has an angle of 30°.

```
DO I = 1,20
```

```
    RP(I) = 1.D0
```

```
ENDDO
```

```
DO I = 32,40
```

```
    RP(I) = 0.5D0
```

```
    TETAP(I) = 30.D0
```

```
    ALFAP(I) = 45.D0
```

```
ENDDO
```

5.4 Processing of liquid

Liquid boundaries can be of two types:

- Incident wave type
- incident potential type
- free exit type.

5.4.1 Incident wave (LIHBOR=KINC)

An incident wave type boundary allows waves arriving from offshore to enter and waves from inside the domain to exit freely. From the theoretical standpoint, the potential at such a boundary is a known potential (incident waves) superimposed on an unknown potential (exiting waves).

In ARTEMIS, incident wave type boundaries enable waves arriving with an angle TETAP to leave the domain (N.B. this should not be confused with the angle of the incident waves TETAB).

From the point of view of the user, it is therefore necessary to provide the characteristics of the incident wave and the value of the angle TETAP.

The characteristics of the incident wave are performed through the variables

HB, TETAB and TETAP in the BORH subroutine:

- HB: Wave height (monochromatic wave, see § 6.1) or significant wave height (random wave, see § 6.2)
- TETAB: Propagation direction expressed in degrees in the trigonometric direction from the X-axis. This value can be imposed as space constant overall the domain through the keyword DIRECTION OF WAVE PROPAGATION in the steering file.
- ALFAP: Phase of the wave at the considered node, in degrees. The user has to choose a reference point where the phase is zero. If the full boundary is perpendicular to the propagation direction, then ALFAP will be zero for the whole boundary nodes. A way to specify ALFAP can be found in some test cases (BTWI, CREOCEAN, ILE_PARA) or in section 5.4.4.
- TETAP: Output direction of waves reflected inside the domain (a priori unknown). TETAP defines the non-oriented angle between the forecast output direction of the wave and the boundary normal. This value is between 0 and $\pi/2$ and must be given in degrees.

5.4.2 Incident potential (LIHBOR=KPOT)

An incident potential type boundary allows specifying a more general form of the reduced potential, waves from inside the domain can still exit freely. From the theoretical standpoint, the potential at such a boundary is a known potential superimposed on an unknown potential (exiting waves). In ARTEMIS, incident potential type boundaries enable waves arriving with an angle TETAP to leave the domain.

From the point of view of the user, it is therefore necessary to provide the characteristics of the incident wave and the value of the angle TETAP.

The characteristics of the incident potential are performed through the variables (given in the BORH subroutine):

- PRB: Real part of the incident potential
- PIB: Imaginary part of the incident potential
- DDXPRB: Real part of the derivative of incident potential with respect to X
- DDYPRB: Real part of the derivative of incident potential with respect to Y
- DDXPIB: Imaginary part of the derivative of incident potential with respect to X
- DDYPIB: Imaginary part of the derivative of incident potential with respect to Y
- TETAP : Output direction of waves reflected inside the domain (a priori unknown). TETAP defines the non-oriented angle between the forecast output direction of the wave and the boundary normal. This value is between 0 and $\pi/2$ and must be given in degrees.

Note that this option has been validated in regular waves only. Considering the actual configuration of ARTEMIS, It should work in non-regular waves also, but no test has been performed yet.

5.4.3 Free exit (LIHBOR=KSORT)

This type of boundary allows waves to leave freely. However, this type of exit is only free in ARTEMIS in the case of waves attacking the boundary at an angle TETAP. It is therefore necessary for the user to estimate the angle of attack of exiting waves beforehand.

TETAP is programmed in the subroutine BORH. The default value (TETAP=0) enables waves normal to the boundary to exit.

5.4.4 Modifications of « incident wave » boundary condition from V6P2

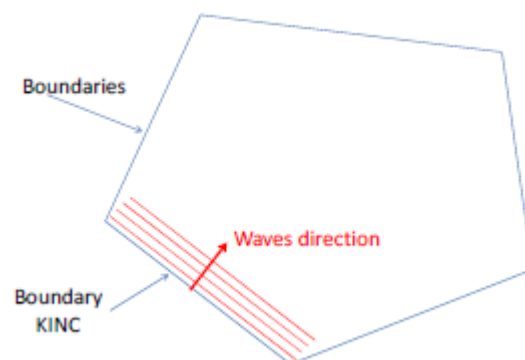
From version V6P2 of ARTEMIS, the user needs to define the phase of the incident wave by himself. This operation was done automatically by the code until V6P1. The reasons for changing are the following:

- From a physical point of view, the automatic phase calculation is not satisfying. In particular when the wave number and the water depth are variable in space.
- From a computing point of view, the automatic calculation is not really compatible with parallel computation.

Thus, the user has to give the phase in the ALFAP table, for each point of « incident wave » type (LIHBOR%I=KINC). Users will find below some examples of how to calculate the phase.

Example 1: Incident wave normal to the boundary

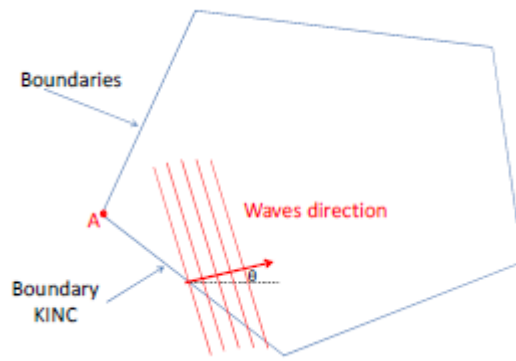
In that case, the phase can be set to 0 on the whole boundary.



Example 2: Incident wave not normal to the boundary, constant bathymetry

In that case, the phase at point I can be calculated using the following formula (users need to choose a reference point A):

$$\varphi_I = k \cdot \cos(\theta)(x_I - x_A) + k \cdot \sin(\theta)(y_I - y_A)$$



Example 3: Incident wave not normal to the boundary, variable bathymetry

The automatic calculation done by Artemis until version V6P1 used an incremental formulation on the boundary (see test cases BTWI or CREOCEAN) from a given point A:

$$\varphi_1 = \sum_{P=A}^{I-1} k(P) \cdot \cos(\theta)(x_{P+1} - x_P) + k_{ref} \cdot \sin(\theta)(y_{P+1} - y_P)$$

Another option is to choose a characteristic water depth of the boundary, then to calculate a characteristic wave number k_{ref} and to use the formula (see parallel version of test BTWI):

$$\varphi_1 = k_{ref} \cdot \cos(\theta)(x_1 - x_A) + k_{ref} \cdot \sin(\theta)(y_1 - y_A)$$

Example 4: Variation in wave direction.

This case is not really recommended, or should at least need a specific attention. The best option is probably to perform several computations with random phases and to consider mean values.

As a conclusion, it is recommended to define an “incident wave” boundary as representative as possible of a far field wave field, so to respect the following recommendations for each boundary:

- Reasonable topography variations on the boundary.
- No variation in the waves direction
- Incident wave condition applied near a solid boundary in the interest area, or in a corner, can generate local effects on the results. The interest area has to be far from those local effects.

6. Physical Parameters

Physical parameters are introduced essentially to characterise the type of calculation that is to be performed (regular waves, random waves, period scanning) and the dissipation phenomena taken into account (breaking, bottom friction).

The detailed list of keywords used by ARTEMIS, and classified by alphabetical order, is supplied in Appendix 4.

6.1 Monochromatic waves

From:

<http://wiki.opentelemac.org/> - open TELEMAC-MASCARET

Permanent link:

http://wiki.opentelemac.org/doku.php?id=user_manual_artemis&rev=1353341143

Last update: **2014/10/10 16:01**

