EU Fifth Framework Programme 1998-2002
Energy, Environment and Sustainable Development

Environmental Design of Low Crested Coastal Defence Structures

D19

Calibrated 2DV near field flow model (Cobras- Cornell Breaking Wave and Structures)

Contract n°EVK3-CT-2000-00041

Dear Partners:

As you might know COBRAS is the acronym for COrnell BReaking wave and Structure interaction. The model is a modified version of the original code named RIPPLE developed by NASA. This modified version has been developed by Prof. Philip Liu from the University of Cornell and his collaborators at Cornell and at other institutions.

At the University of Cantabria we have been working several years with the model trying to improve its capabilities.

According to Prof. Liu you are welcome to use COBRAS for your own research. Please make sure that you give proper acknowledgement to COBRAS and the relevant references in your future publications. A list of references may be found at the end of the Users Manual developed by the University of Cantabria. If you are going to make significant changes in the model, please let us know or send a note directly to Prof. Liu.

For those of you interested in using the model we may organize an introductory course. The code and further references will be provided during the course.

COBRAS

Cornell Breaking Wave and Structures

User's manual

Version 1.01

INDEX

1. INTRODUCTION

The COBRAS model (Cornell Breaking Wave and Structures) is a bidimensional numerical model that solves the Reynolds Averaged Navier Stokes (RANS) 2DV equations, with a three-dimensional nonlinear k-ε turbulence model. The original model, developed by NASA, was called RIPPLE and has been modified at Cornell University to deal specifically with breaking waves and flow within porous media.

COBRAS model features :

- has been codified in FORTRAN.

- uses a VOF (Volume of Fluid) method for tracking the free surface . This method allows an arbitrary free surface definition and the presence of unconnected zones, i.e. drops, plunging jets, etc. Free surface tracking can also be determined in the COBRAS model using the Young method or the MAC (Marker and Cell) method. Young method is more accurate to define free surface, but it is nonconservative. MAC method requires free surface information all along the computational domain (like BIEM methods).

- solves the equations of motion using two finite difference methods, improving the accuracy of the results. The mesh is formed by rectangular submesh regions, and the grid size can be nonuniform : a finer grid can be defined for specific study zones representation.

- allows the definition of obstacles and porous media, as many as wanted, which are defined through continuous conic functions. For the porous media, a nonlinear energy dissipation model is included.

- includes a conservative pollutant dispersion model. Constant flow and arbitrary gravitational system can be defined as initial conditions. Fluid density is constant but has a user-defined value. Surface tension model can be included. Different types of waves can be defined : linear, Stokes II, solitary, cnoidal and Sokes V. Moreover the effect of a jet at any of the boundaries and at any position can be simulated. The turbulence model can consider linear or nonlinear eddy viscosity or Reynolds stress terms.

- provides complete information of pressure, vorticity, turbulent viscosity, kinetic energy, turbulent dissipation, horizontal and vertical velocities, free surface configuration, mass conservation, overtopping volume...

2. GENERAL EQUATIONS

Continuity :

$$\frac{\partial < u_i >}{\partial x_i} = 0$$

Momentum :

$$\frac{\partial < u_i >}{\partial t} + < u_j > \frac{\partial < u_i >}{\partial x_j} = -\frac{1}{r}\frac{\partial < p >}{\partial x_i} + g_i + \frac{1}{r}\frac{\partial < t_{ij} >}{\partial x_j}$$

RANS Equations :
Reynolds Averaged
Navier-Stokes

where *i,j=1,2,3* for each of the fluid three dimensions. u_i represents the *ith* component of velocity vector, *r* is the density, *p the* pressure, g_i the *ith* component of the gravity acceleration and t_{ij} the viscous stresses. The <> symbol represents time- averaging. Likewise :

$$< t_{ij} >= 2mS_{ij} + r < u_i ' u_j ' >$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial < u_i >}{\partial x_j} + \frac{\partial < u_j >}{\partial x_i} \right)$$

$m = n\ r$ is the dynamic viscosity and n is the kinematic viscosity (XNU).

Kinematic boundary condition :

$$\frac{\partial < r >}{\partial t} + < u_i > \frac{\partial < r >}{\partial x_i} = 0$$

_k_-*e* **_t_urbulence transport model** :

$$\frac{\partial k}{\partial t} + < u_j > \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\frac{n_t}{s_k} + n \right) \frac{\partial k}{\partial x_j} \right] - < u_i ' u_j ' > \frac{\partial < u_i ' >}{\partial x_j} - e$$

$$\frac{\partial e}{\partial t} + < u_j > \frac{\partial e}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\frac{n_t}{s_e} + n \right) \frac{\partial e}{\partial x_j} \right] + 2C_{1e}\,\frac{e}{k} n_t S_{ij} \frac{\partial < u_i >}{\partial x_j} - C_{2e}\,\frac{e^2}{k}$$

where :

$k = \frac{1}{2} < u_i ' u_i ' >$ is the turbulent kinetic energy,

$e = n \left\langle \left(\frac{\partial u_i '}{\partial x_k} \right)^2 \right\rangle$ is the turbulent dissipation rate,

$n_t = C_d \frac{k^2}{e}$ is the eddy viscosity,

$s_k = 1.0$,
$s_e = 1.3$,
$C_{1e} = 1.44$
$C_{2e} = 1.92$

$$\mathrm{C_d} = \frac{2}{3} \left(\frac{1}{7.4 + \mathrm{S_{max}}} \right)$$

$$S_{max} = \frac{k}{e} max \left(\left| \frac{\partial < u_i >}{\partial x_i} \right| \right)$$

Linear closure model for Reynolds stresses (isotropic eddy viscosity), KEMODEL=1 :

$$< \mathrm{u_i} ' \mathrm{u_j} ' > = -2 \nu_t \mathrm{S_{ij}} + \frac{2}{3} \mathrm{k} \delta_{ij}$$

Nonlinear closure model for Reynolds stresses (anisotropic eddy viscosity), KEMODEL=4 :

$$\begin{aligned} < u_i ' u_j ' > = \; & \frac{2}{3} k d_{ij} - C_d \frac{k^2}{e} \left(\frac{\partial < u_i >}{\partial x_j} + \frac{\partial < u_j >}{\partial x_i} \right) \\ & - \frac{k^3}{e^2} \left[C_1 \left(\frac{\partial < u_i >}{\partial x_l} \frac{\partial < u_l >}{\partial x_j} + \frac{\partial < u_j >}{\partial x_l} \frac{\partial < u_l >}{\partial x_i} - \frac{2}{3} \frac{\partial < u_l >}{\partial x_k} \frac{\partial < u_k >}{\partial x_l} d_{ij} \right) \right. \\ & + C_2 \left(\frac{\partial < u_i >}{\partial x_k} \frac{\partial < u_j >}{\partial x_k} - \frac{1}{3} \frac{\partial < u_l >}{\partial x_k} \frac{\partial < u_l >}{\partial x_k} d_{ij} \right) \\ & \left. + C_3 \left(\frac{\partial < u_k >}{\partial x_i} \frac{\partial < u_k >}{\partial x_j} - \frac{1}{3} \frac{\partial < u_l >}{\partial x_k} \frac{\partial < u_l >}{\partial x_k} d_{ij} \right) \right] \end{aligned}$$

where :

d_{ij} is the Kronecker delta,

$$C_1 = \frac{1}{185.2 + D_{\max}^2}$$

$$C_2 = -\frac{1}{58.5 + D_{\max}^2}$$

$$C_3 = \frac{1}{370.4 + D_{\max}^2}$$

$$D_{\max} = \frac{k}{\varepsilon}\max\left(\left| \frac{\partial < u_i >}{\partial x_j} \right| \right)$$

If $C_1=C_2=C_3=0.0$, the nonlinear model reduces to the linear model.

k-l **mixing length model** (KEMODEL=2) :

$$l = k\ y\left[1 - \exp\left(-\frac{y\sqrt{t_w / r}}{26.0 * n}\right)\right]$$

$$\frac{\partial k}{\partial t} + < u_j > \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\frac{n_t}{s_k}\frac{\partial k}{\partial x_j}\right] + 2S_{ij}n_t\frac{\partial < u_i >}{\partial x_j} - C_d\frac{k^{3/2}}{l}$$

where :

l is the mixing length,

$k=0.41$ is von Kármán constant,

y is the vertical coordinate,

$t_w = r u_*^2$ is the shear stress at the boundary, which can be obtained through a logarithmic law or : $k = \frac{u_*^2}{\sqrt{C_d}}$

Reynolds tensor model (KEMODEL=3):

$$\begin{aligned}\frac{\partial < u'_i u'_j >}{\partial t} + < u_l > \frac{\partial < u'_i u'_j >}{\partial x_l} = \quad & -\frac{\partial}{\partial x_l} < u'_l u'_i u'_j > - \frac{1}{r}\left(\frac{\partial < u_j p >}{\partial x_i} + \frac{\partial < u_i p >}{\partial x_j}\right) \\ & - < u'_i u'_l > \frac{\partial < u_j >}{\partial x_l} - < u'_j u'_l > \frac{\partial < u_i >}{\partial x_l} + < \frac{p}{r} S_{ij} > \\ & - 2n < \frac{\partial u_i}{\partial x_l}\frac{\partial u_j}{\partial x_l} >\end{aligned}$$

Volume-Averaged Reynolds Averaged Navier-Stokes Equations (VARANS) - fluid-porous structure interaction - :

Continuity :

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0$$

Momentum :

$$\frac{1+C_A}{n}\frac{\partial \overline{u_i}}{\partial t} + \frac{\overline{u_j}}{n^2}\frac{\partial \overline{u_i}}{\partial x_j} = -\frac{1}{r}\frac{\partial \overline{p}}{\partial x_i} + \frac{n}{n}\frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_j} - \frac{1}{n^2}\frac{\partial \overline{u_i"u_j"}}{\partial x_j}$$

where :

$C_A = g_p \frac{1-n}{n}$ is the added mass coefficient,

$g_p = 0.34$,

n is the effective porosity (XPOROSITY),

u_i" is the ith velocity component spacial fluctuation and the overbar " ‾ " denotes the space-averaging.

Closure model :

$$-\frac{1}{n^2}\frac{\partial \overline{u_i"u_j"}}{\partial x_j} = -g a_p \overline{u_i} - g b_p \overline{u_c}\,\overline{u_i}$$

where :

$a_p = a_p \frac{(1-n)^2}{n^3}\frac{n}{gD_{50}^2}$,

$a_p \approx 200$ (XALPHA),

D_{50} is the averaged sediment diameter (D50),

$b_p = b_p\left(1+\frac{7.5}{KC}\right)\frac{1-n}{n^3}\frac{1}{gD_{50}}$,

$b_p \approx 1.1$ (XBETA),

$KC = \frac{\overline{u_c}T}{nD_{50}}$ is the Keulegan-Carpenter number,

$\overline{u_c}$ is the flow characteristic velocity and can be estimated as :

$\overline{u_c} \approx \sqrt{\overline{u_i}\,\overline{u_i}}$ and

T is the process characteristic period.

Pollutant transport (conservative):

$$\frac{\partial <c>}{\partial t}+<u_j>\frac{\partial <c>}{\partial x_j}=\frac{\partial}{\partial x_j}\left[n_c\frac{\partial <c>}{\partial x_j}\right]-\frac{\partial}{\partial x_j}<u_j'c'>$$

where :

c is the pollutant or substance concentration,
n_c is the substance kinematic velocity,
u_j' is the j^{th} velocity component fluctuation and
c' is the substance concentration fluctuation.

Closure model :

$$-\frac{\partial}{\partial x_j}<u_j'c'>=\frac{\partial}{\partial x_j}\left[n_{tc}\frac{\partial <c>}{\partial x_j}\right]$$

where n_{tc} is the turbulent diffusivity and can be linked to the eddy viscosity through :

$n_{tc} = c_{tc}\, n_t$, where $c_{tc} \approx 1.0$.

VOF function :

$$\frac{\partial F}{\partial t}+\frac{\partial}{\partial x}(uF)+\frac{\partial}{\partial y}(vF)=0$$

where *r(x,y,t)=F(x,y,t)* r_f , *r(x,y,t)* is the cell density, r_f is the fluid density (RHOF).

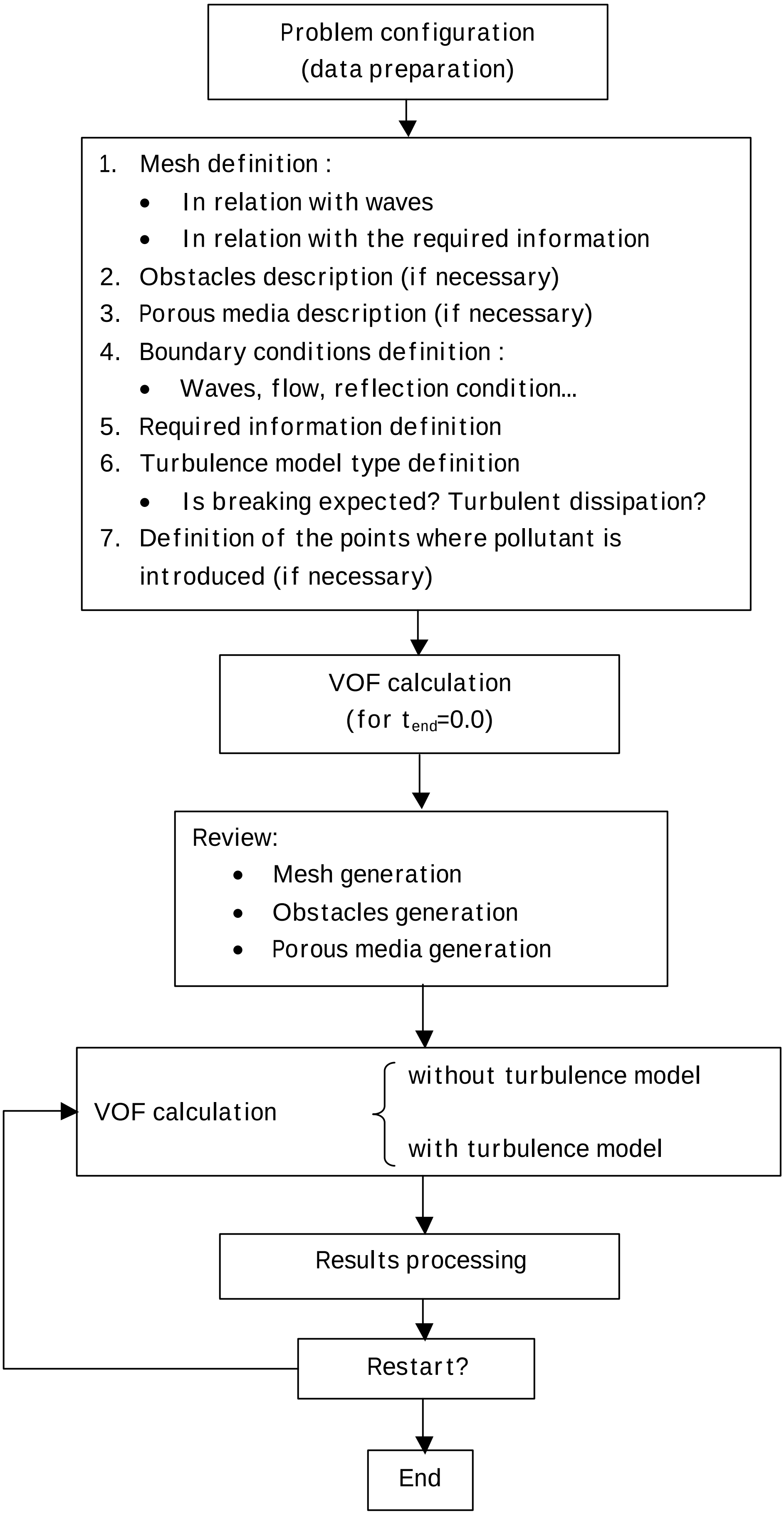
Partial cells treatment (obstacle boundaries) :

$$\frac{\partial(q\,u_i)}{\partial x_i}=0$$

$$\frac{\partial(q\,u_i)}{\partial t}+q\,u_j\frac{\partial(q\,u_i)}{\partial x_j}=\frac{q}{r}\frac{\partial p}{\partial x_i}+q\,g_i+q\frac{\partial}{\partial x_j}t_{ij}$$

where *q* is the openness function, relation between the cell surface open to the flow and the total cell surface. The only difference between the openness function and the VOF function is that the VOF function varies with time whereas the opening function does not.

3. PROCEDURE FOR COBRAS APPLICATION


Problem configuration
(data preparation)
1. Mesh definition :
• In relation with waves
• In relation with the required information
2. Obstacles description (if necessary)
3. Porous media description (if necessary)
4. Boundary conditions definition :
• Waves, flow, reflection condition…
5. Required information definition
6. Turbulence model type definition
• Is breaking expected? Turbulent dissipation?
7. Definition of the points where pollutant is introduced (if necessary)
VOF calculation
(for t_{end}=0.0)
Review:
• Mesh generation
• Obstacles generation
• Porous media generation
VOF calculation
without turbulence model
with turbulence model
Results processing
Restart?
End

4. INPUT FILE

The input file is divided into 17 sections. Each of them includes information about a specific aspect of the case considered and requires a certain number of data. All of the sections, except the first one, are clearly identified in the input file. These sections are :

1. Name of the project (comments)
2. Numerical parameters
3. Fluid parameters
4. Mesh generation
5. Obstacles
6. Free surface
7. Graphics
8. Wave parameters
9. Output format
10. Porous media
11. Turbulence model
12. Boundary type
13. Sponge layer
14. Free surface tracking method
15. Pollutant and sediment transport parameters
16. Restart controlling parameter
17. Overtopping calculation

In Appendix 1 an example of the input file is presented. The values preceded by the "~" symbol, are characteristic values, proposed or recommended. For each value, the corresponding validity range is also indicated.

4.1. Name of project

The first line of the input file is a descriptive text of the project or the case study considered, 80 characters maximum.

4.2. Numerical parameters

Formed by 7 lines of data.

DELT, TWFIN, PRTDT, PLTDT

DELT (>0.0, ~0.005) Initial time step of calculation. Once the calculation begins, DELT can change from its initial value if the automatic time step adjustment flag AUTOT ≠ 0.

TWFIN (>0.0) Final simulation time.

PRTDT (>0.0) Time step for data writing in Type I files. If PRTDT > TWFIN, data will not be written. For Type I files description, see section 2.1.

PLTDT (>0.0) Time step for data writing in Type II files. If PLTDT > TWFIN, data will not be written. For Type II files description, see section 2.2.

ALPHA, BETA, KL, KR, KT, KB, AUTOT

ALPHA (~0.3) Controls the finite differences representation of the advective terms in the momentum equation :

ALPHA	Momentum advective terms
0.0	Centered Difference
0.0 < ALPHA < 1.0	Interpolated Donor Cell
1.0	Donor Cell (upwind)
ALPHA > 1.0	2nd-order van Leer

BETA (1.0) Test parameter, **not to be changed**

KL, KR, KT, KB Boundary condition flag for the left, right, top and bottom boundaries (L, R, T and B), respectively :

KL, KR, KT, KB	Boundary condition
1	Rigid Free-Slip
2	Rigid No-Slip
3	Continuative Outflow (open)
4	Periodic
5	Applied Pressure
6	Specified Inflow / Outflow

AUTOT (~1.0) Automatic time step adjustment during the simulation (AUTOT ≠ 0). If AUTOT is equal to 0, the time step is DELT.

NPACK, CON, DMPDT

NPACK (~0) flag set nonzero to activate a biased packing (vertical, top-to-bottom) of the VOF function after advection. The fluid is forcibly packed down until all surface cells lie above all fluid cells.
This flag should be used with caution since the packing alters the free surface without regards to the current fluid flow field.
0 = false, 1 = true.

CON (0.0 < CON ≤ 0.5, ~0.3) Maximum allowable Courant number.

DMPDT (> 0.0, ~greater than TWFIN) Time step for binary files writing.

DTMAX, IDIV

DTMAX (>0.0, ~0.5) Maximum allowed time step.

IDIV (~0) Flag for divergence correction to the VOF function after advection.
0 : no correction, ≠ 0 : correction.

ERRICCG, FCVLIM, NRESTART

ERRICCG (> machine minimum, ~1.0d-08) Convergence criterion for Poisson pressure equation solution.

FCVLIM (0.0 < FCVLIM ≤ 0.5, ~0.39) Courant number above which a recomputation of the current cycle is performed with a time step reduced by 20%.

NRESTART Restart dump number
≤ 0 : New calculation starts from t=0.
>0 : Calculation restarts from binary dump.
-1.0 : **not to be changed**.

GFNCTN, FRCTN, CONSERVE

GFNCTN (~.true.) Logical flag controlling the $g_{i,j}$ function form.
.true. : $g_{i,j}=2F_{i,j}$
.false. : $g_{i,j}=1$

FRCTN (0.0 < FRCTN < 1.0, ~1.0 e-02) Fraction of the fluid density RHOF below which solutions to the Navier Stokes equations are not obtained. Pressures and velocities in those cells satisfying ρ < FRCTN * RHOF are set to PSAT and 0.0, respectively.

CONSERVE Logical flag for the momentum advection terms.
~.false. Nonconservative finite differences expressions are employed for the whole mesh.
.true. Conservative finite differences expressions are employed for the whole mesh. Useful only for uniform meshes.

CRAY, ITMXICCG, SYM

CRAY (~.false.) Logical flag indicating the use of a CRAY.

ITMXICCG (>0, ~500) Maximum number of iterations allowed for the Poisson pressure equation solution.

SYM (~.true.) Logical flag informing about the symmetry of the matrix to be inverted for the Poisson pressure equation resolution :

.true.	Symmetric
.false.	Asymmetric. SYM needs to be .false. only when the mixed Dirichlet and Neumann pressure boundary conditions are prescribed.

4.3. Fluid parameters

Formed by 7 lines of data.

XNU, ICYL, GX, GY, UI, VI

XNU	(>0.0, ~1.0e-06) Fluid kinematic viscosity.
ICYL	(~0) Problem geometry indicator : 0 : cartesian coordinates. 1 : cylindrical coordinates.
GX	(~0.0) Gravity acceleration in the positive x (r) direction.
GY	(~ -9.81) Gravity acceleration in the positive y (z) direction.
UI	(~0.0) Initial fluid velocity in the positive x (r) direction.
VI	(~0.0) Initial fluid velocity in the positive y (z) direction.

UTOP, PSAT, SIGMA, ISURF10, RHOF

UTOP	(0.0) **Not to be changed**.
PSAT	(~0.0d0.0) Constant pressure in empty cells. The fluid pressures are scaled relatively to this value.
SIGMA	(>0.0, ~22.6) Surface tension coefficient.
ISURF10	Surface tension indicator. If ISURF10 ≠ 0, surface tension is imposed on all free surfaces. Surface tension will be employed only for very small cases, in which surface tension is relevant.
RHOF	(>0.0, ~1.0) Fluid density in the F = 1.0 regions. RHOF is a relative value, 1.0 value can be used for fresh water. Pressure and forces results will be scaled according to RHOF.

CANGLER, CANGLEL, CANGLET, CANGLEB

CANGLER	(0.0 < CANGLER < 180.0, ~90.0) Equilibrium contact angle (in degrees) used on the right boundary for wall adherence effects.

CANGLEL (0.0 < CANGLER < 180.0, ~90.0) Equilibrium contact angle (in degrees) used on the left boundary for wall adherence effects.

CANGLET (0.0 < CANGLER < 180.0, ~90.0) Equilibrium contact angle (in degrees) used on the top boundary for wall adherence effects.

CANGLEB (0.0 < CANGLER < 180.0, ~90.0) Equilibrium contact angle (in degrees) used on the bottom boundary for wall adherence effects.

UINF(n), VINF(n)

UINF(n) (~0.0) Flow velocity in the positive x (r) direction at a specified inflow or outflow boundary when parameters KB, KT, KL or KR are equal to 6. The value of index n in array UINF corresponds to the specific boundary :

1 = bottom
2 = top
3 = left
4 = right

VINF(n) (~0.0) Flow velocity in the positive y (z) direction at a specified inflow or outflow boundary when parameters KB, KT, KL or KR are equal to 6. The value of index n in array VINF corresponds to a specific boundary :

1 = bottom
2 = top
3 = left
4 = right

PBC(n)

PBC(n) (~0.0) Constant pressure to be applied at the boundary when parameters KB, KT, KL or KR are equal to 5. The value of index n in array PBC corresponds to the specific boundary :

1 = bottom
2 = top
3 = left
4 = right

4.4. Mesh generation

The mesh generation depends on the case study, on the computer capacities and on the problem geometry. The computational mesh is rectangular, and the cells spacing can be variable. Still, the use of uniform meshes is recommended, whenever possible, as the finite differences scheme is first order and the numerical error due to variations in the cells dimensions can be considerable.

The computing mesh is constructed from a number of "submeshes" defined at each coordinate direction. The mesh generation section of the input file specifies the boundaries, the number of cells, the minimum cell dimension, and the convergence point ("central" point) of each submesh. Arbitrary variable cell spacing is achieved by linking submeshes together. The number of cells in each submesh is specified by supplying cell numbers to the left and right of the convergence point. Cells directly adjacent to the convergence point will have a cell spacing equal to the minimum value, specified as either DXMN or DYMN. The cell spacing is then expanded quadratically from these cells to the left and right edges of the submesh in accordance with the desired number of cells (NXR, NXL or NYR, NYL) in the input list. If the number of cells specified on the left (right) should produce a uniform cell size less than the minimum size DXMN (or DYMN), a uniformly spaced mesh is generated with δx < DXMN (or δy < DYMN). The chosen mesh spacing is then determined by the user-specified number of cells (NXR, NXL or NYR, NYL). The number of cells to the left and right of the convergence point need not to be equal, but there must be at least one on both sides.

Because reciprocals of coordinates are used in COBRAS, a calculation will fail with a "divide-by-zero" error if a cell face or cell center coordinate in a real fluid cell coincides with zero. This circumstance is most easily prevented by the requirement that XL(m) and YL(m) be non-negative, increasing sequence of numbers. Of course, in cylindrical coordinates it is also required that $x \geq 0$. The trigonometric functions used for obstacles and free surfaces definition require as well that XL(m), YL(m) ≥ 0.

When two or more submeshes are linked together, the location of the left edge of the right submesh must coincide with the location of the right edge of the left submesh. Large disparities in cell spacing should be avoided across submeshes and within a submesh itself. As a general rule, the spacing of adjacent cells should not differ by more than ~10-20%. Cells aspect ratios ($\delta x/\delta y$) greater than 0.5 and less than 2 are more desirable.

Brief descriptions of the mesh generation sections, formed by 7 lines of data, is given below.

NKX, XL(NKX), XC(NKX), NXL(NKX), NXR(NKX)

NKX (>0) Number of submesh regions in the x-direction.

XL (m=1,NKX+1) (≥0.0) Location of the left edge of submesh m. NKX+1 values of XL(m) are necessary because the right edge of submesh m is equivalent to the left edge of submesh m+1, XL(m+1). The values should be given as an increasing sequence of numbers.

XC (m=1,NKX) (>0.0) x-coordinate of the convergence point in submesh region m.

NXL (m=1,NKX) (>0) Number of cells between locations XL(m) and XC(m) in submesh region m.

NXR (m=1,NKX) (>0) Number of cells between locations XC(m) and XL(m+1) in submesh m.

DXMN(m)

DXMN(m=1,NKX) (>0.0) Minimum space increment in the x-direction, for submesh m.

NKY, YL(NKY+1), YC(NKY), NYL(NKY), NYR(NKY)

NKY (>0) Number of submesh regions in the y-direction.

YL (m=1,NKY+1) (≥0.0) Location of the bottom edge of submesh m.
NKY+1 values of YL(m) are necessary because the top edge of submesh m is equivalent to the bottom edge of submesh m+1, YL(m+1). The values should be given as an incresaing sequence of numbers.

YC (m=1,NKY) (>0.0) y-coordinate of the convergence point in submesh region m.

NYL (m=1,NKY) (>0) Number of cells between locations YL(m) and YC(m) in submesh region m.

NYR (m=1,NKY) (>0) Number of cells between locations YC(m) and YL(m+1) in submesh m.

DYMN(NKY)

DYMN(m=1,NKY) (>0.0) Minimum space increment in the y-direction, for for submesh m.

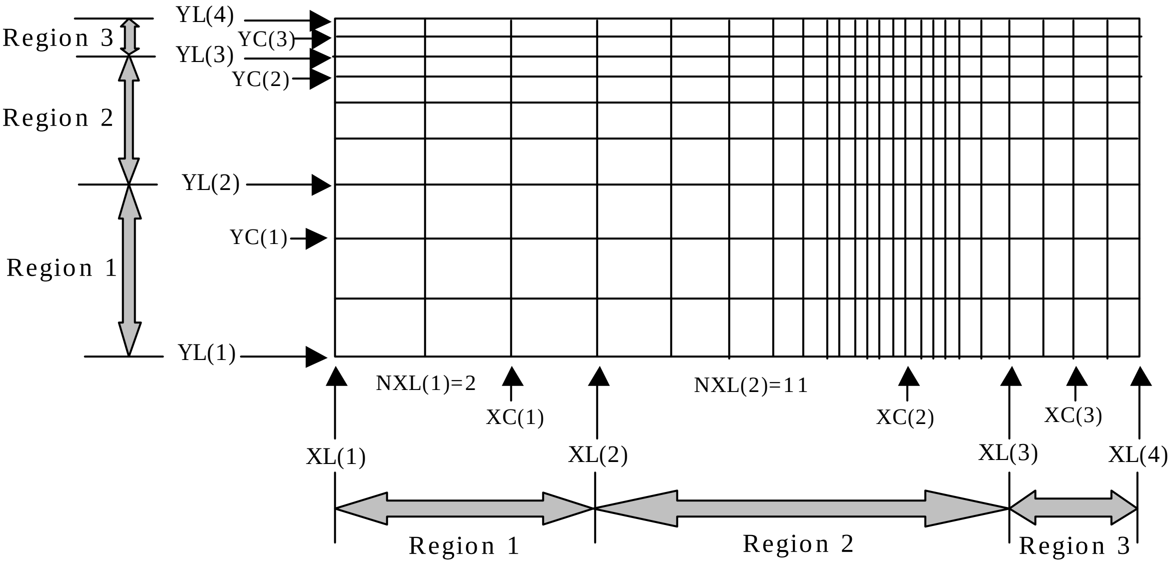


Figure 1. Nonuniform mesh definition.

When wave steepness is high, and so is the aspect ratio, a problem of *false breaking* can arise. This problem occurs when various full fluid cells (in other words, not free surface cells) are adjacent to empty cells. In that case, due to the VOF function (F) convection, the empty cells "receive" a flow that does not exist in reality (figure 2).

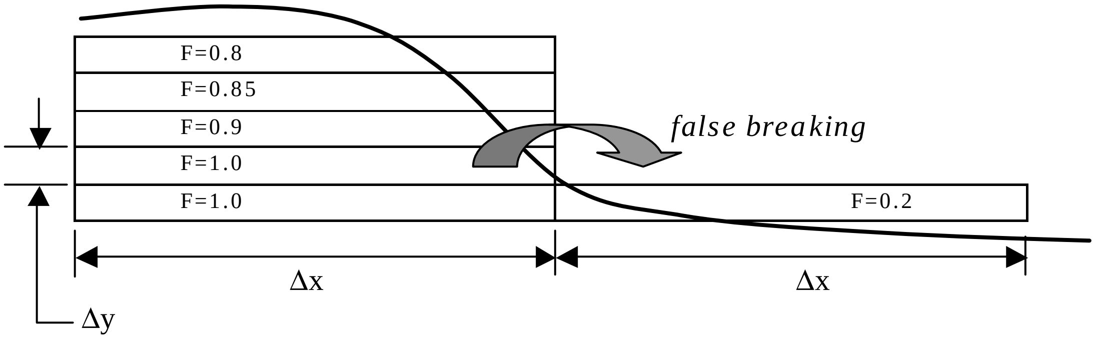


Figure 2. "False breaking" phenomenon.

To prevent this problem it is recommended to use a number of cells per wave length such that a breaking limit wave steepness can be represented. Experimentally, it has been observed that the maximum angle of the free surface before breaking varies between 23° and 25°. In the one hand, the wave should be vertically represented by at least 10 cells. On the other hand, the maximum wave steepness before breaking is :

$$\left(\frac{H}{L}\right)_{max} = \tan \quad 23^{o} \ ,$$

thus, if *nx* is the number of cells per wavelength, and *ny* is the number of cells per wave height, the wave steepness associated to the mesh would be:

$$\frac{H/ny}{L/nx} > tan\,23^{o}, \qquad \frac{\Delta y}{\Delta x} > tan\,23^{o}, \qquad \Delta x < 2.36\,\Delta y\ ,$$

rounded off : $\Delta x<2.5\Delta y$

For instance, if the wave height is 2 m, and 10 vertical divisions are considered, $\Delta y = 0.2$ m, thus $\Delta x < 0.5$ m.

If breaking is not expected, the *false breaking* effect can be verified performing a calculation and observing the turbulence generation. Another method consists of verifying the convergence by making a calculation with a specific Δx and afterwards another calculation using a smaller Δx. If convergence is reached, the results of both calculations should be identical.

When a variable mesh is used, changes in the dimension of each cell have to be less than 5%. This criterion is achieved if the second derivate of the coordinates of each cell satisfies :

$\Delta^2 x < 0.05$
$\Delta^2 y < 0.05$

To estimate the number of cells in a submesh, the following calculation can be done :

$$\frac{dx_i}{n} \cong \frac{\Delta x_i + \Delta x_{i+1}}{2},$$

where δx_i=XC(i)-XL(i) or XL(i+1)-XC(i) is the submesh region dimension, n the number of cells in the sub-region and Δx_i and Δx_{i+1} are the cells dimensions at the beginning and at the end of the sub-region, respectively. The same criterion is used in the y-direction.

To create a variable mesh, at least 3 regions are required. A way of understanding the configuration of a variable mesh consists of plotting the cells dimension along the coordinates axis. The submesh regions with uniform cells are represented by a horizontal line and the submesh regions with variable dimension cells are represented by an inclined line, see figure 3.

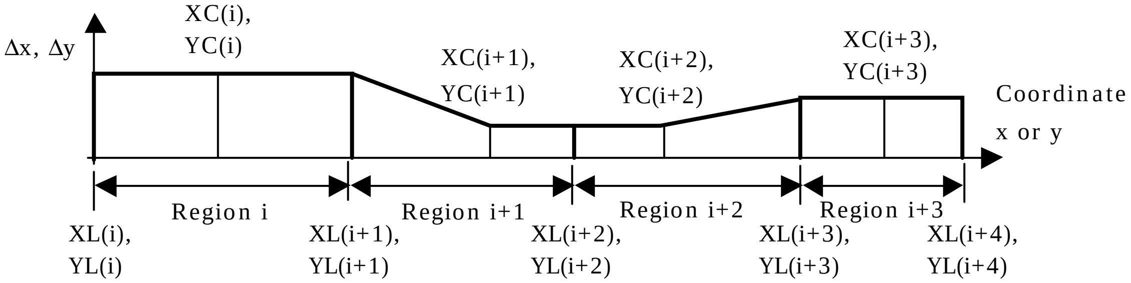


Figure 3. Submesh regions cells dimensions.

It is recommended that the mesh contains a space between the structures or objects under study of, at least, half a wavelength, more if possible, at both sides of the mesh, depending on the case.

4.5. Obstacles

Interior obstacles are defined as any nonflow regions within the computational mesh. Obstacle surfaces are generated with a series of conic sections that may overlap one another to form boundaries that are arbitrarily complex. Cells are then designated as flow or nonflow regions, depending upon which side of the surface *f(x,y)* = 0 they reside, where *f(x,y)* is a modified conic function of the form :

$$\begin{aligned} f(x,y) = {} & a_1 x + a_2 x^2 + b_1 y + b_2 y^2 + c_1 + c_2 xy + \\ & + d_1 \cos\left[\left(n_x k_{xm}\right)\ x\right] + d_2 \operatorname{sen}\left[\left(m_x k_{xm}\right)\ x\right] + \\ & + e_1 \cos\left[\left(n_y k_{yxm}\right)\ y\right] + e_2 \operatorname{sen}\left[\left(m_y k_{ym}\right)\ y\right] \end{aligned}$$

where: $k_{xm} = \dfrac{2\pi}{x_{\max}}$, $k_{ym} = \dfrac{2\pi}{y_{\max}}$ are the mesh wave numbers in the x- and y-directions, respectively,
x_{max} = XL(NKX+1), y_{max} = YL(NKY+1) are the maximum x and y mesh coordinates, respectively,
n_x, m_x, n_y and m_y can be used to generate k_{xm} and k_{ym} harmonics (defined as real numbers).

The cells that satisfy the conditions *f(x,y)< 0* (for any *(x,y)* within the cell) respond to the condition of the IOH flag associated with each function, being totally opened or closed to the flow. Hence, each obstacle is defined by a group of conic functions with their associated indicator flag.

The obstacles generation follows the order in which the functions are defined. Adjacent obstacles should be superposed to guarantee that all the frontier cells are included. Even if no obstacles are defined, the whole 5 lines have to be filled.

NOBS

NOBS (≥ 0) Number of conic functions defining any flow obstacles interior to the computational mesh.

NC(NOBS)

NC Number of conic functions to define obstacle. NC is an array with NOBS data.

OA2(NOBS), OA1(NOBS), OB2(NOBS), OB1(NOBS)

OA2(n) Coefficient of the x^2 term in obstacle function n.
OA1(n) Coefficient of the x term in obstacle function n.
OB2(n) Coefficient of the y^2 term in obstacle function n..
OB1(n) Coefficient of the y term in obstacle function n.

OC2(NOBS), OC1(NOBS), IOH(NOBS)

OC2(n) Coefficient of the xy term in obstacle function n.
OC1(n) Constant term in obstacle function n.
IOH(n) Interior obstacle indicator flag on cells satisfying $f(x,y)<0$ (for any (x,y) within the cell) for obstacle function n :

≠0 adds obstacle
=0 eliminates obstacle

OD2(n), OD1(n), OE2(n), OE1(n)

OD2(n) Amplitude of the x-direction sine term in obstacle function n.
OD1(n) Amplitude of the x-direction cosine term in obstacle function n.
OE2(n) Amplitude of the y-direction sine term in obstacle function n.
OE1(n) Amplitude of the y-direction cosine term in obstacle function n.

NXO(n), MXO(n), NYO(n), MYO(n)

NXO(n) (Real number) Mesh wavenumber coefficient in the x-direction (k_{xm}), for the x-direction cosine term in obstacle function n.

MXO(n) (Real number) Mesh wavenumber coefficient in the x-direction (k_{xm}), for the x-direction sine term in obstacle function n.

NYO(n) (Real number) Mesh wavenumber coefficient in the y-direction (k_{ym}), for the y-direction cosine term in obstacle function n.

MYO(n) (Real number) Mesh wavenumber coefficient in the y-direction (k_{ym}), for the y-direction sine term in obstacle function n.

Note : There is not one single way to define obstacles for a determined simulation case geometry. The same obstacle can be defined differently depending on the user.

4.6. Free surface

Fluid surfaces can be defined in the same way as the obstacles. The conic functions are defined the same way that the *f(x,y)* function.

Formed by 11 lines of data. Even if no free surfaces are defined, these 11 lines of data have to be included.

NFRSRF

NFRSRF (≥ 0) Number of conic functions used in intializing free surfaces. Fluid initially occupies the entire domain if conic functions are used to specify the free surface.

FA2(n), FA1(n)

FA2(n) Coefficient of the x^2 term in free surface function n.

FA1(n) Coefficient of the x term in free surface function n.

FB2(n), FB1(n)

FB2(n) Coefficient of the y^2 term in free surface function n.

FB1(n) Coefficient of the y term in free surface function n.

FC2(NFRSRF), FC1(NFRSRF), IFH(NFRSRF)

FC2(n) Coefficient of the xy term in free surface function n.

FC1(n) Constant term in free surface function n.

IFH(n) Fluid indicator flag on cells satisfying *f(x,y)<0* (for any (x,y) within the cell) for free surface function n :

≠0 adds fluid
=0 eliminates fluid

FD2(NFRSRF), FD1(NFRSRF)

FD2(n) Amplitude of the x-direction sine term in free surface function n.

FD1(n) Amplitude of the x-direction cosine term in free surface function n.

FE2(NFRSRF), FE1(NFRSRF)

FE2(n) Amplitude of the y-direction sine term in free surface function n.

FE1(n) Amplitude of the y-direction cosine term in free surface function n.

NXF(NFRSRF), MXF(NFRSRF)

NXF(n) (Real number) Mesh wavenumber coefficient in the x-direction (k_{xm}), for the x-direction cosine term in free surface function n.

MXF(n) (Real number) Mesh wavenumber coefficient in the x-direction (k_{xm}), for the x-direction sine term in free surface function n.

NYF(NFRSRF), MYF(NFRSRF)

NYF(n) (Real number) Mesh wavenumber coefficient in the y-direction (k_{ym}), for the y-direction cosine term in free surface function n.

MYF(n) (Real number) Mesh wavenumber coefficient in the y-direction (k_{ym}), for the y-direction sine term in free surface function n.

NSMOOTH, IEQUIB, FLHT, UPRIGHT

NSMOOTH (> 0, ~1) Number of times per computational cycle that the VOF function (F) is smoothed when SMOOTH = .true.

IEQUIB (~0) Flag set nonzero for initializing the fluid free surface in an equilibrium meniscus configuration, subject to the contact angle CANGLER and a nonzero fluid height FLHT. This initialization requires that NFRFSRF=0, and is valid only in the absence of interior obstacles along the left and right portion of the problem boundaries that contain the free surface.

FLHT (0.0 < FLHT < y_{max}) Vertical fluid height in the positive y-direction relative to the y=ymin position. Applies only when NFRSRF = 0.

UPRIGHT (~.true.) Logical variable for FLHT variable definition. Applies only when NFRSRF = 0 :

.false. Fluid is initialized in an inverted position, the surface height is relative to y=y_{max}.

.true. Fluid is initialized in a normal position.

SMOOTH, EMIN, DELTEMIN

SMOOTH .true. Smoothed VOF function is used for the calculation of free surface curvature.

~.false. VOF function is not smoothed.

EMIN (~.false.) Logical flag. When .true., allows the user-initialized free surface to change shape and to evolve towards a minimal energy configuration before the calculation begins. Useful in cases with surface tension.

DELTEMIN (> 0.0, ~0.0) Initial iteration time step used in the surface energy minimization.

ITMXEMIN, ITSKPEMN

ITMXEMIN (> 0, ~500) Total number of iterations used in the surface energy minimization.

ITSKPEMN (0 < ITSKPEMN ≤ ITMXEMIN, ~50) Number of iterations to skip in the surface energy minimization before resetting fluid velocities to 0.0.

4.7. Graphics

This section of data comes from the initial RIPPLE code. It is formed by 6 lines of data.
In the new version of the COBRAS model, the entire section is not used any more and thus must not be changed.

4.8. Wave parameters

The number of lines of data is variable, and depends on the type of wave that is defined.

AA, HO, NCENTER

AA Wave height.
HO Depth of wave definition.
NCENTER Wave type:

≤ 3: Wave profile is defined through free surface definition.
XCENTER Crest position.

4: Stokes II order
XXT Wave period
ARETURN Parameter (~1.0)

5: Solitary wave
FXSTART Caracteristic length:

$$l = \frac{FXSTART * HO}{\sqrt{AA/HO}} \quad (\sim 2.12)$$

DEAN & DALRYMPLE (1991), p.315, ec. (11.87)

6: Paddle-generated wave. Reads from ETA_PAD file (3 columns with information of time, level and position of the paddle, 550 points).

7: Horizontal jet.
UJET Jet center position.
VJET Jet width.
WJET Horizontal velocity.
YJET Vertical velocity.

8: Vertical jet.
UJET Jet center position.
VJET Jet width.
WJET Horizontal velocity.
YJET Vertical velocity.

14: Paddle-generated Stokes V order wave.

24: Cnoidal wave.
XXT Wave period
ARETURN Parameter (0.5-1.0, ~0.6)

34: Linear wave.
XXT Wave period
ARETURN Parameter (0.5-1.0, ~1.0)

100: Source function (see Figure).
ISOURCES Initial cell (x)
ISOURCEE Final cell (x)
JSOURCES Initial cell (y)
JSOURCEE Final cell (y)
NSOURCE Wave type

4: Stokes II order
5: Solitary wave
FXSTART (~2.12)
14: Stokes V order
24: Cnoidal wave
34: Linear wave
44: Irregular wave
NWAVE # of waves
AAWAVE(NWAVE) Amplitude
TWAVE(NWAVE) Period
TSOURCE Wave period

Waves are generated at the left boundary (except when NCENTER=100, or when another type of condition, different to wave generation, is specified). In that case, the boundary has to be defined as KL=6.

The source function generates a system of waves that propagates in two directions. The top limit of the source function has to remain under the wave trough, and its vertical dimension has to be about half the depth. The source function has to be placed at a distance of ~L/2 from the left boundary and at a distance of ~h/3 from the bottom boundary. Its width has to be <4 cells or ~L/20, its height is recommended to be around h/4 (see Figure).

4.9. Ouput results

Formed by 1 to 4 lines of data, according to the type of results that are required.

TSTART, TFINISH, IBG, IEG, JBG, JEG, ITERX, ITERY, LOUT, NANIMATION, NMEAN

TSTART Initial time for Type I files.
TFINISH Final time for Type I files.
IBG Initial cell for data writing (x).
IEG Final cell for data writing (x).
If information on the whole mesh is required :
IBG=1, IEG=NXL(1..NKX)+NXR(1..NKX)+2
JBG Initial cell for data writing (y).
JEG Final cell for data writing (y).
If information on the whole mesh is required:
JBG=1, JEG=NYL(1..NKY)+NYR(1..NKY)+2
ITERX Data writing interval in Type I files (Δx).
ITERY Data writing interval in Type I files (Δy).
LOUT Gages files (Type IV, see section 2.4).
0 = no, 1 = yes.
NANIMATION Animation files (Type III).
0 = no, 1 = yes.

NMEAN	Vertical-averaged files (Type V, see section 2.5). 0 = no, 1 = yes.		
If LOUT=1:			
	NLOC	Number of gages.	
	XOUT(NLOC)	Gages position (x).	
	XXXF	Initial time (writing).	
	TTEND	Final time (writing).	
	PRTDT_T	Time interval (writing).	
	If PRTDT_T<0.0		
		NDT	Number of intervals.
		DET_K(NDT)	Time interval.
If NANIMATION=1:			
	TSTART_A	Initial time (writing).	
	TFINISH_A	Final time (writing).	
	PRTDT_A	Time interval (writing).	
If NMEAN=1:			
	ISTART	Initial cell (x).	
	IEND	Final cell (x).	
	IINTERVAL	Writing interval (Δx).	
	TMSTART	Initial time (writing).	
	TMEND	Final time (writing).	
	TINTERVAL	Time interval (writing).	

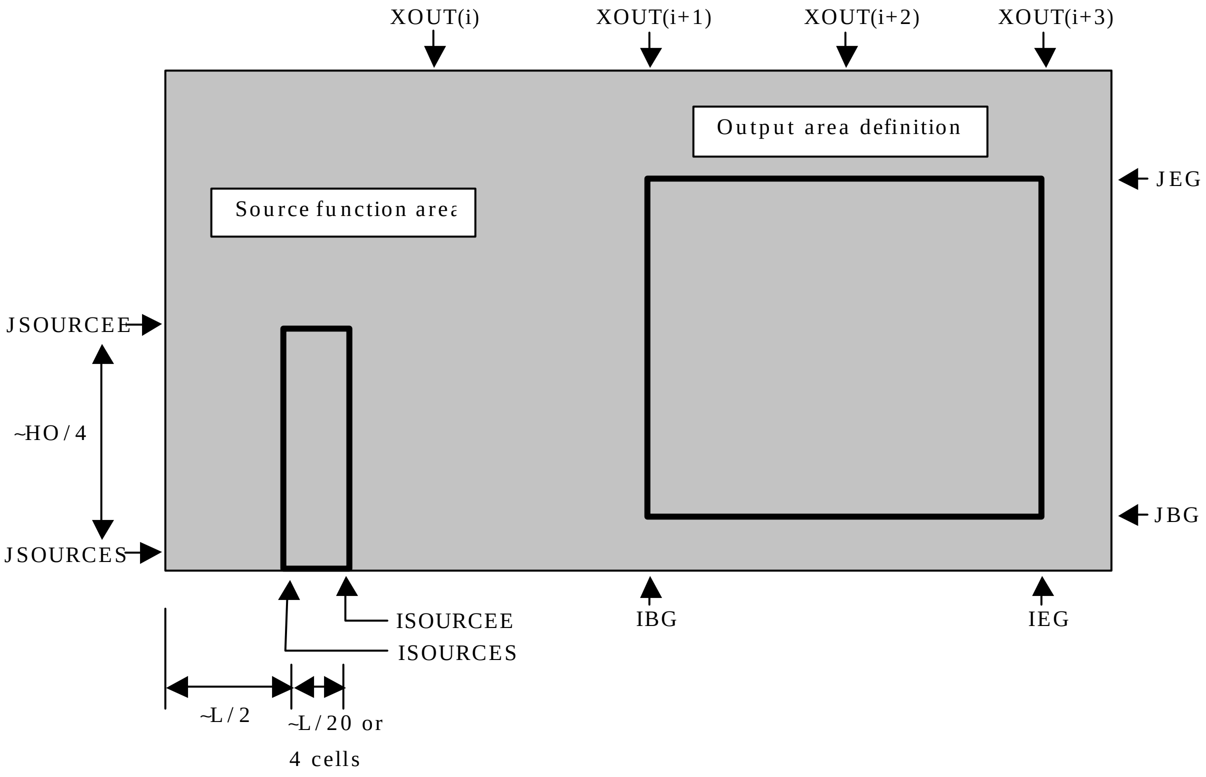


Figure 4. Representation of computational domain.

4.10. Porous media

Formed by a variable number of lines, depending on the type of porous media, the number of defined areas, etc.

NPOR

NPOR Type of porous media:

0 = no porous media.

1 = flow through porous media solved with Laplace equation (linear friction).

2 = flow through porous media solved with Navier-Stokes averaged equations (nonlinear friction).

If NPOR≠0:

NPORTYPE Number of porous media.

Group of data for each porous media.

XPERM, XPOROSITY

XPERM Permeability.

XPOROSITY Porosity

If NPOR=2:

XALPHA, XBETA, XGAMMA, D50

XALPHA Linear friction coefficient (~200.0).

XBETA Nonlinear friction coefficient (~1.1).

XGAMMA Added mass coefficient (0.34).

D50 Porous material diameter.

NPOROUS Number of elements of a specific porous media.

PA2(NPOROUS), PA1(...), PB2(...) PB1(...)
PC2(NPOROUS), PC1(...), IPR(...)
PD2(NPOROUS), PD1(...), PE2(...), PE1(...)
NXP(NPOROUS), MXP(...), NYP(...), MYP(...)

Defined the same way as obstacles.

4.11. Turbulence model

Formed by 1 to 2 lines of data.

KEMODEL

KEMODEL Type of turbulence model :

0 : No turbulence model is used.

1 : Linear eddy viscosity (isotropic).

2 : 1-equation (*k*-*l*) model.

3 : Reynolds tensor.

4 : Nonlinear eddy viscosity (anisotropic).

If KEMODEL≠0

TICF, EDDYCOEF, KOPEN

TICF Turbulence seed parameter (~0.5).

EDDYCOEF Eddy viscosity behavior parameter (~50.0):

KOPEN Parameter for boundary layer turbulence resolution. A value of 1.0 is recommended in cases of grid size inadequate to define the boundary layer width.

4.12. Reflection conditions at boundaries

Formed by 1 or 2 lines of data :

NWEAKREF, NOPEN, ISLIP, ISLIPCV

NWEAKREF Weak reflection :
0 : No weak reflection.
1 : Conventionnal wave reflection.
2 : Considers mass conservation for wave reflection.

NOPEN Radiation condition :
0 : Standard.
1 : Modified by PENGZHI, 1998.

ISLIP Sliping condition :
0 : Sliping.
1 : Intermediate.
2 : No sliping .

ISLIPCV Sliping condition for convective terms.
0 : Sliping.
1 : No sliping.
The value of 0 is recommended only for cases of waves propagating over flat bottom, without structures.

If KR=3, NOBS≠0 y NOPEN≠0:
H0R Depth at the right boundary.

4.13. Sponge layer

Formed by 1 or 2 lines of data.

NSPONGE Flag for sponge layer mode.

ISP If NSPONGE=1: sponge layer final cell.

The sponge layer is a zone of the computational domain aimed at damping wave energy that propagates from the source function to left boundary. Its width should be at least ~2L. Sponge layer final cell should be one before source function.

4.14. Free surface tracking method

Formed by 1 line of data.

NFREE Type of tracking :

1: VOF method.
2: VOF method. Creates ETA output file.
3: MAC method.
≥4: Young method (more accurate, but non-conservative method).

4.15. MAC method control parameters

Formed by 1 line of data.

NTRACER Type and number of profiles.

0: The MAC method is not used.
≠0: MAC method : opens a file with the free surface coordinates (number of profiles=NTRACER).

4.16. Pollutant and sediment transport parameters

Formed by 1 or 2 lines of data.

NPOLLUTANT, NABVIN, NROUGH, MIRRORT

NPOLLUTANT Number of points where pollutant substance is introduced :

≠0:

XPOL(NPOLLUTANT)	Coordinate x.
YPOL(NPOLLUTANT)	Coordinate y.
CONC(NPOLLUTANT)	Concentration.

NABVIN, NROUGH, MIRRORT are parameters linked to pollutant transport.

sD, DENS, FALLV, NBUOY, SEDTIME, JSEDB, JSEDF

These are parameters linked to sediment transport.

This "Pollutant and sediment transport parameters" section is currently under development. It will be explained with more details in a future version of this manual.

4.17. Restart controlling parameter

Formed by 1 line of data.

NRS 0: No restart.

1: Reads the binary output file tape10 and calculation resumes.

Process :

1. When a normal calculation is over, a binary file is created, tape8, containing all the information (u, v, vof, turbulence, pressure, etc.).
2. tape8 file is copied to tape10.
3. Initial and final times of the whole input file are corrected, according to the new calculation.
4. The NRS parameter is changed to 1.
5. The calculation is initialized and will start at the final time of the previous calculation.

4.18. Overtopping calculation

Formed by 1 or 2 lines of data.

NOVERTOP 0: No overtopping calculation

1: Overtopping calculation:

XOVERTOP X coordinate of the overtopping calculation point. The result is written in the CONSERVE (Type II) output file.

5. OUTPUT FILES

5 types of output files have been defined, according to the required information. In each section the unit number of each file has been included (FORTRAN).

5.1. Type I files.

These files are unconditionally generated and come from the original code of the RIPPLE program. They contain the complete information about all of the variables. The values are written according to the window defined in the output format (section 4.9.), from initial time TSTART to final time TFINISH and for each time interval PRTDT (defined in the numerical parameters, section 1.2), in a single file. Data are written, in the x-direction, from the IBG cell to the IEG cell every ITERX cell and, in the y-direction, from the JBG cell to the JEG cell every ITERY cell.

Type I files are:

		Unit
EDDY	Eddy viscosity (n_t).	30
EPS	Turbulent dissipation (e).	28
K	Turbulent kinetic energy ($\sqrt{2k}$).	27
PS	Pressure ($<p>$).	24
UX	Horizontal velocity ($<u>$).	22
VOF	VOF function (F).	21
VORTEX	Vorticity (w).	29
VY	Vertical velocity ($<v>$).	23
If NPOLLUTANT=1		
POL	Pollutant concentration ($<c>$).	33
MAC	Free surface tracking.	36
If KEMODEL=3		
UU	Reynolds tensor ($<u'u'>$).	40
VV	Reynolds tensor ($<v'v'>$).	41
WW	Reynolds tensor ($<w'w'>$).	42
UV	Reynolds tensor ($<u'v'>$).	43
If KEMODEL=2		
UV	Vertical eddy viscosity (ny_t).	43

The information corresponds to the right (or top) face of the cell.

5.2. Type II files

These unconditionally generated files contain information about the calculation, the input data, mass and energy conservation, etc.

Type II files are :

CONSERVE Unit 12. Writes, line after line, averaged information for the whole mesh:

TIME

t

MASS

$$\int_{ymin}^{ymax}\int_{xmin}^{xmax} F\,n\,dx\,dy$$

KINETIC ENERGY

$$\int_{ymin}^{ymax}\int_{xmin}^{xmax} (u^2+v^2)\frac{F\,n}{2}\,dx\,dy$$

POTENTIAL ENERGY

$$\int_{ymin}^{ymax}\int_{xmin}^{xmax} (y+F\,dy-h)\,g\,F\,n\,dx\,dy$$

TURBULENT ENERGY

$$\int_{ymin}^{ymax}\int_{xmin}^{xmax} k\,F\,n\,dx\,dy$$

X NET FORCE

$$\int_{obstáculos} p\,dy$$

Y NET FORCE

$$\int_{obstáculos} p\,dx$$

OVERTOPPING MASS

$$\int_{\mathrm{ymin}}^{\mathrm{ymax}} \mathrm{F}\ \ \mathrm{n}\ \ \mathrm{dy}\ \Big|_{\mathrm{x=XOVERTOP}}$$

For each PLTDT time interval.

The mesh total MASS indicates whether the simulation is conservative or not, as the initial resting mass (basic mass) has been deduced, so the time-averaged MASS has to be approximately 0.0. The OVERTOPPING MASS is the mass of fluid flowing through the XOVERTOP section, with respect to the x-axis sign.

ERRORS Unit 9. For each cycle, writes information about the calculation, for each line :

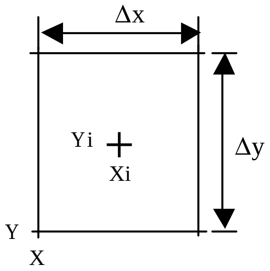
NCYC	Cycle
NI	X cell
NJ	Y cell
T	Time
XX	X coordinate
YY	Y coordinate
XMAXXK	Maximum turbulence intensity
XMAXXEP	Maximum dissipation
PRODMAX	Maximum production

XMAXXNUT	Maximum eddy viscosity. The (x,y) coordinates are those of the point where the eddy viscosity is maximum.

Also writes if a numerical or convergence error occurs.

EDIT Unit 13. Checking of the data reading and the mesh generation. First writes all the information about the calculation, fluid, mesh and obstacles. Then, for each x coordinate :

X()	X coordinate
RX()	1/x
DELX()	X increment, Δx
RDX()	1/Δx
Xi()	X coordinate of the center of the cell.



Next, for each y coordinate :

Y()	Y coordinate
DELY()	Y increment, Δy
RDY()	1/Δy
Yj()	Y coordinate of the center of the cell.

The time step calculation criterion is also written for each cycle :

ITER		Iteration.
TIME		Time.
Time step	DTCX(i,j)	Per Courant in x, in the i,j cell.
	DTCY(i,j)	Per Courant in y, in the i,j cell.
	DTV(i,j)	Per eddy viscosity, in i,j.
	DTG(i,j)	Per increasing, in i,j (see FCVLIM).
	DTM(i,j)	Per maximum increment, in i,j (see DTMAX).
CYCLE		Consecutive cycle.
VCHGT		VOF function error.

5.3. Type III files

For each variable and for each PRTDT_A time interval a file is created, from TSTART_A time to TFINISH_A time. The file name corresponds to the variable and is filled consecutively. At the beginning of the calculation files are generated at t=0 (resting condition):

T0000 Unit 1000. VOF function (F).

K0000	Unit 1001. Turbulent kinetic energy ($\sqrt{2k}$).
P0000	Unit 1002. Pressure (or vorticity) (<p> ó ω).
If NPOLLUTANT≠0	
C0000	Unit 1003. Substances concentration (<c>).
U0000	Unit 1004. Velocity in x-direction (<u>).
W0000	Unit 1005. Velocity in y-direction (<v>).

Then, the files corresponding to the time intervals (T0001, T0002, and so on) are generated, for all the variables. The values correspond to the center of the cells, in the x-direction, from the IBG cell to the IEG cell with an ITERX interval, and in the y-direction, from the JBG cell to the JEG, with an ITERY interval. The type III files are aimed at animation sequence generation.

5.4. Type IV files

A file is generated for each gage, with information about the VOF function, velocities, turbulent kinetic energy, etc., vertically from the first cell to the last one, for a constant x coordinate, XOUT. Each line corresponds to a PRTDT_T time interval, from XXXF to TTEND.

		Unit
XTn	Eddy viscosity	900+(n-1)*5+4
XFn	VOF function (F).	700+(n-1)*7+1
XUn	Horizontal velocity (<u>).	700+(n-1)*7+2
XVn	Vertical velocity (<v>).	700+(n-1)*7+3
XKn	Turbulent kinetic energy ($\sqrt{2k}$).	700+(n-1)*7+4
XPn	Pressure (<p>).	700+(n-1)*7+5
XWn	Vorticity (ω).	700+(n-1)*7+6
If NPOLLUTANT≠0		
XZn	Substances concentration (<c>).	700+(n-1)*7+7
HEIGH	Free surface position.	8700+(n-1)*5+1
EG	Energy flux.	1700+(n-1)*5+1

Note : The last two type IV files, HEIGH and EG, have a different structure from the others. They are formed by two columns and contain information of the variable magnitude (integrated on the vertical) at the considered section XOUT, for each PRTDT_T time interval from XXXF to TTEND.

5.5. Type V files

Similar to Type II files, they contain vertical-averaged information, for a mesh section (in x-direction, from the cell ISTART to the cell IEND every IINTERVAL cells). A line is written form TMSTART to TMEND every TINTERVAL, in seconds.

Generates the files :

		Unit
EFLUX	Energy flux.	70

$$EFLUX = \int_{-h}^{h} (p + rgz)u\,dz$$

DISP	Dissipation.	71

$$DISP = \int_{-h}^{h} r\,e\,dz$$

ENERGY	Total energy.	72

$$ENERGY = \int_{-h}^{h} (E_k + E_p + E_T)dz = \int_{-h}^{h} r\,\frac{u^2 + v^2}{2}\,dz + \int_{-h}^{h} rg\,\frac{(h+h)^2}{2}dz + \int_{-h}^{h} r\,k\,dz$$

ETA	Free surface.	73

$$ETA = \int_{-h}^{h} VOF\,dz$$

EFLUXT	Energy flux.	74

$$EFLUXT = \int_{-h}^{h} p\,u\,dz$$

5.6. Other files

		Unit
INPUT	Input file.	5
TAPE8	Output binary file.	8
TAPE10	Input binary file.	9
XC	File containing the left coordinate of cell 1 to cell n+1.	31
YC	File containing the bottom coordinate of cell 1 to cell m+1.	32
POROUS	File containing the whole mesh porosity information : (n+1) x (m+1) data. The values correspond to the left coordinate of each cell.	26
OBS	File containing the whole mesh obstacles information : (n+1) x (m+1) data. (0= obstacles, 1= no obstacles) The values correspond to the left coordinate of each cell.	25

6. REFERENCES

Hsu, T.-J., T. Sakakiyama and P.L.-F. Liu (2002). A numerical model for wave motions and turbulence flows in front of a composite breakwater. Coastal Engineering, 46, 25-50.

Lin P., P.L.-F. Liu (1998). A numerical study of breaking waves in the surf zone. Journal of Fluid Mechanics 359, 239-264.

Liu, P.L.-F. and P. Lin (1997). A numerical model for breaking wave : the Volume of Fluid method. Research Report. No. CACR-97-02, Center for Applied Coastal Research, Ocean Engineering Laboratory, University of Delaware.

Liu, P.L.-F., P. Lin, K.A. Chang and T. Sakakiyama (1999). Numerical modeling of wave interaction with porous structures. J. Waterway, Port, Coastal and Ocean Engineering, ASCE 125 (6), 322-330.

RIPPLE : A computer program for incompressible flows with free surfaces. Los Alamos National Laboratory Report LA-12007-MS.

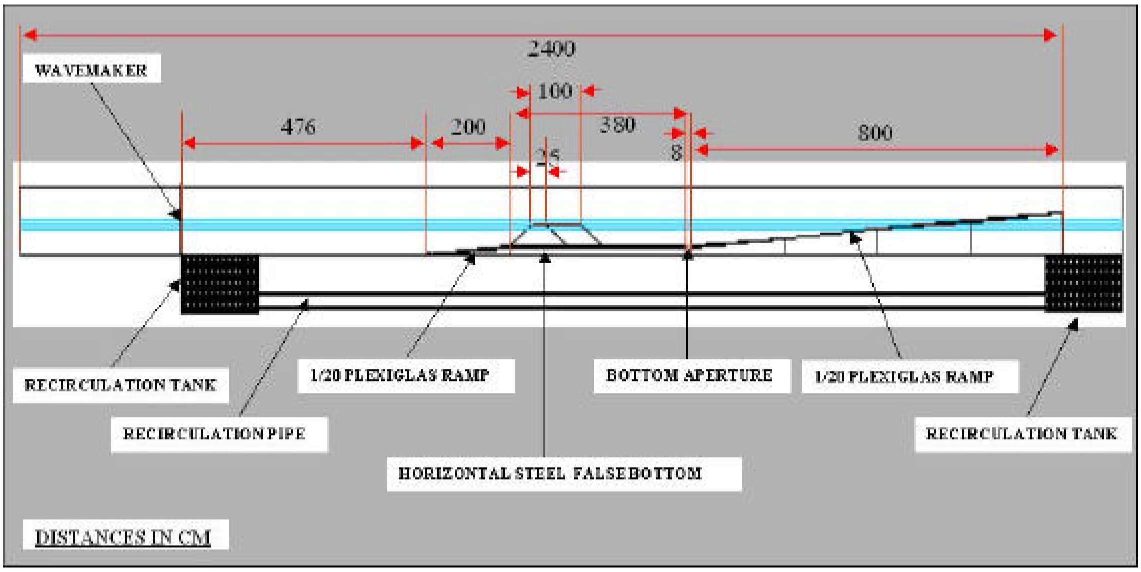
Appendix I.

EXAMPLE OF INPUT FILE.

The aim of appendix I is to present an example of the COBRAS model input file.

The input file described hereafter has been used for the simulation of one of the laboratory test performed at the University of Cantabria for the DELOS EU project. The tests consisted of the propagation of paddle-generated regular waves over a two-layer rubble-mound low-crested breakwater, and their purpose was to characterize the hydrodynamics around and within the porous media.

The geometry and experimental set-up of the DELOS tests performed in a wave flume at the University of Cantabria are depicted on figure 1.



The test presented in this appendix considered wide-crested (1 m) rubble mound and maximum water depth (40 cm). Furthermore, the bottom aperture was covered and no return flow through the recirculation pipe was considered.

The following three pages include the complete input file defined for the simulation of that test.

```
Example of input file - DELOS test 222
 $numparam
   0.005,80.,1000,0.2        <==delt,twfin,prtdt,pltdt
   0.3,1.0,3,1,1,1,1.0       <==alpha,beta,kl,kr,kt,kb,autot
   0,0.3,300.0               <==npack,con,dmpdt
   0.05,0                    <==dtmax,idiv
   1.0d-8,0.39,-1            <==erriccg,fcvlim,nrestart
   .true.,1.0e-02,.false.    <==gfnctn,frctn,conserve
   .false.,500,.true.        <==cray,itmxiccg,sym
 $end
 $fldparam
   1.0e-06,0,0.0,-9.81,0.0,0.0     <==xnu,icyl,gx,gy,ui,vi
   0.0, 0.0d0, 22.6, 0, 1.0        <==utop,psat,sigma,isurf10,rhof
   90.0,90.0,90.0,90.0             <==cangler,canglel,canglet,cangleb
   0.,0.,0.,0.,0.,0.,0.,0.         <==uinf(1:4),vinf(1:4)
   0.0,0.0,0.0,0.0                 <==pbc(1:4)
 $end
 $mesh
   4, 0.0,8.0,12.76,18.64,26.64,4.0,12.75,15.70,18.65,100,200,294,1,100,1,294,300<==nkx,xl(nkx+1),xc(nkx),nxl(nkx),nxr(nky)
   0.04,0.01,0.01,0.01,0.01                                          <==dxmn(nkx)
   1,0.0,0.6,0.3,30,30                                               <==nky,yl(nky+1),yc(nky),nyl(nky),nyr(nky)
   0.01                                                              <==dymn(nky)
 $end
 $obstcl
   2                                                 <==nobstype
   3                                                 <==nc(nobs)
   0.,0.,0.,0.,0.05,-1.,0.,0.,0.,1.,-1.,0.           <==oa2(nobs),oa1(nobs),ob2(nobs),ob1(nobs)
   0.,0.,0.,-0.10,-0.638,18.56,1,0,0                 <==od2(nobs),od1(nobs),oe2(nobs),oe1(nobs)
   0,0,0,0,0,0,0,0,0,0,0,0                           <==nxo(nobs),mxo(nobs),nyo(nobs),myo(nobs)
   2                                                 <==nc(nobs)
   0.,0.,-1.,0.05,0.,0.,0.,-1.                       <==oa2(nobs),oa1(nobs),ob2(nobs),ob1(nobs)
   0.,0.,18.64,-0.832,1,0                            <==oc2(nobs),oc1(nobs),ioh(nobs)
   0.,0.,0.,0.,0.,0.,0.,0.                           <==od2(nobs),od1(nobs),oe2(nobs),oe1(nobs)
   0,0,0,0,0,0,0,0                                   <==nxo(nobs),mxo(nobs),nyo(nobs),myo(nobs)
 $end
```

```
$freesurf
   0                                     <==nfrsrf
   0.,0.,0.0,0.0,-1.0,0.0                <==fa2(nfrsrf),fa1(nfrsrf)
   0.,0.,0.0,-1.0,0.0,1.0                <==fb2(nfrsrf),fb1(nfrsrf)
   0.,0.,0.0,0.80,25.0,10000.0,1,1,0     <==fc2(nfrsrf),fc1(nfrsrf),ifh(nfrsrf)
   0.,0.,0.,0.,0.,0.                     <==fd2(nfrsrf),fd1(nfrsrf)
   0.,0.,0.,0.,0.,0.                     <==fe2(nfrsrf),fe1(nfrdrf)
   0,0,0,0,0,0,0,0,0,0                   <==nxf(nfrsrf),mxf(nfrsrf)
   0,0,0,0,0,0,0,0,0,0                   <==nyf(nfrsrf),myf(nfrsrf)
   1,0,0.40,.true.                       <==nsmooth,iequib,flht,upright
   .false., .false., 0.0                 <==smooth, emin, deltemin
   500,50                                <==itmxemin,itskpemn
 $end
 $graphics
   .true.,.false.                        <==plots,dump
   1,1,0,0,1,1,0,0,0,0,0,0,0             <==iout(1:13)
   0,0,0,0,0,0,0,0,0,0,0,0,0             <==iout(14:26)
   0,0,0,1,1,1,1                         <==iout(27:33)
   2.0,1.0d-3,.false.,0                  <==scale,vmxfrctn,unfrmmsh,ixsymplt
   1,1,1,0                               <==iysymplt,ixskip,iyskip,icolor
 $end
$wave parameter$
0.07,0.40,100                 <==aa,h0,ncenter
150,152,13,23,14,1.6          <==isources,isourcee,jsources,jsourcee,nsource,tsource
$output format$
0.0,180.,1,1292,1,62,1,1,1,1,0,    <==ts,tf,ib,ie,jb,je,ix,iy,lout,na,nm
14,13.89,14.04,14.26,14.76,15.01,15.035,15.26,15.76,15.897,16.26,16.362,17.305,17.745,17.895,0.0,180.0,0.05 <==nloc,xout(nloc),xxxf,ttend
0.0,180.0,.1                       <==tstart_a,tfinish_a,predt_a
$porous material information$
2                                                          <==npor
2                                                          <==porous layers
1.0,0.49                                                   <==permeability,porosity
1000.,1.2,0.34,0.0118                                      <==xalpha,xbeta,xgamma,d50
3                                                          <==nporous
0.,0.,0.,0.5,-0.5,0.,0.,0.,0.,1.,-1.,-1.                   <==pa2,pa1,pb2,pb1
0.,0.,0.,-0.35,-7.28,8.4725,1,0,0,                         <==pc2,pc1,ipr
```

```
0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.                        <==pd2,pd1,pe2,pe1
0,0,0,0,0,0,0,0,0,0,0,0                                    <==nxp,mxp,nyp,myp

1.0,0.53                                                   <==permeability,porosity
1000.,0.8,0.34,0.0387                                      <==xalpha,xbeta,xgamma,d50
3                                                          <==nporous
0.,0.,0.,0.5,-0.5,0.,0.,0.,0.,1.,-1.,-1.                   <==pa2,pa1,pb2,pb1
0.,0.,0.,-0.35,-7.28,8.4725,1,0,0,                         <==pc2,pc1,ipr
0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.                        <==pd2,pd1,pe2,pe1
0,0,0,0,0,0,0,0,0,0,0,0,                                   <==nxp,mxp,nyp,myp
$turbulence model information$
4                       <==kemode
0.5,5,1                 <==ticf,eddycoef,kopen
$boundary type$
0,1,0,1                 <==nweakref,nopen,islip,islipcv
$sponge layer$
1                       <==nsponge
149                     <==isp
$free surface tracking method$
1                                  <==nfree
$MAC method controlling parameter$
0                                  <==ntracer
$pollutant transport parameter$
0,0,0,0                            <==npollutant,nabvin,nrough,mirrort
0.0002,2.66,0.026,0,50.1,5,30      <==sD,dens,fallv,nbuoy,sedtime,jsedb,jsedf
$restart controlling parameter$
0                                  <==nrs
$overtopping calculation$
0                                  <==novertop
```

In the following paragraphs, a short explanation about the simulation parameters is given, section by section.

Numerical parameters

A 80 seconds calculation is performed, with automatic time step adjustment (initial time step of 0.005 s). Information is not written in type I files (prtdt > twfin). In type II files information is written every 0.2 seconds.
Left boundary is open, the other three boundaries are closed.

Fluid parameters

Surface tension model is not used and fluid density is 1.0.
Initial fluid velocity is zero, and neither flows nor pressures are defined at the boundaries.

Mesh generation

See figures A1a and A1b.
A 26.64 m x 0.6 m variable mesh is generated, with 4 regions in x-direction and 1 region in y-direction. The whole mesh is uniform in the y-direction : Δy (cell dimension in y-direction) = 1 cm. The cell dimension in x-direction (Δx) is constant in the first and the third regions.
Region 1 is 8 m long and corresponds to the sponge layer and a "flow regularization zone". Δx (region 1) = 4 cm.
Region 3 is 5.88 m long and corresponds to the study zone (first 1/20 sloped ramp + horizontal false bottom with rubble mound). Δx (region 3) = 1 cm.

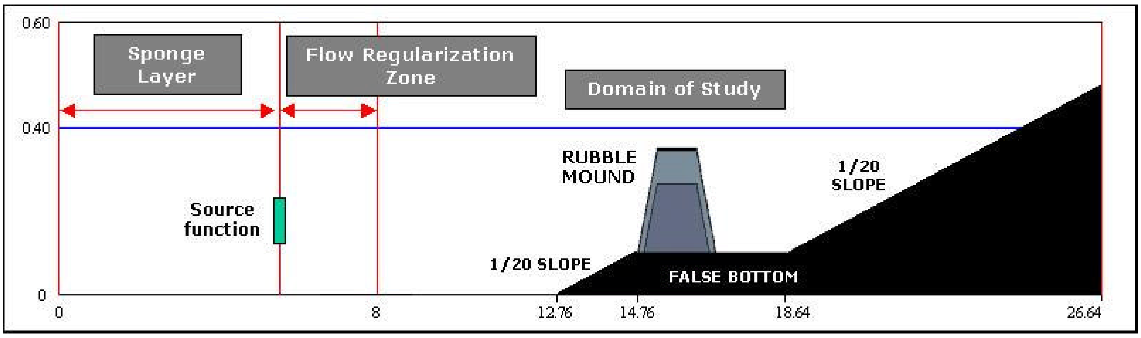


Figure A1a. Computational domain zones and elements.

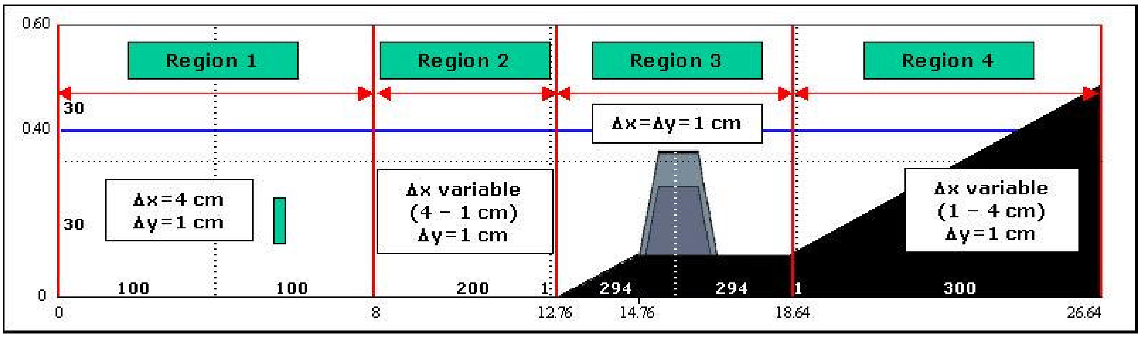


Figure A1b. Submesh regions and cell dimensions.
The dot lines indicate the location of each submesh region convergence coordinates. On both sides of the dot lines the number of cells is included.

Obstacles

Two obstacles are defined. The first one is defined by 3 functions and represents the first 1/20 sloped ramp and the 10 cm high horizontal false bottom. Obstacle 1 definition functions are presented as an example on figure 2.

The second one is defined by 2 functions and represents the final 1/20 sloped ramp.

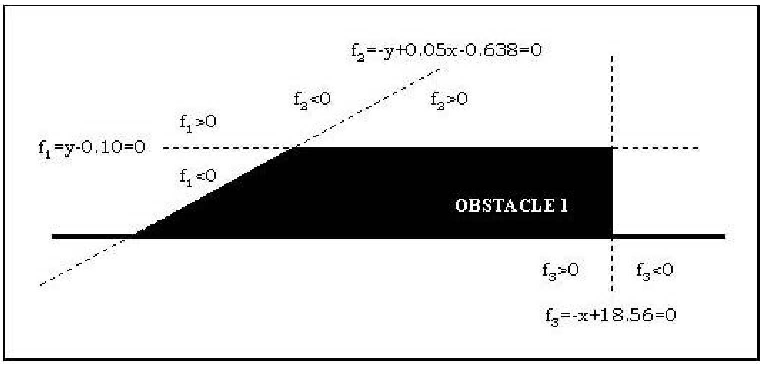


Figure A2. Example of obstacle definition : obstacle 1.

Obstacle 1 :

f_1 = y-0.10	fill for f1<0
f_2 = -y+0.05x-0.638	eliminate for f2<0
f_3 = -x+18.56	eliminate for f3<0

Obstacle 2 :

g_1 = -x+18.64	fill for g1<0
g_2 = -y+0.05x-0.832	eliminate for g2<0

Free surface

No conic functions are defined for free surface and as a consequence the free surface level is defined at 40 cm.

Note : The first six sections of the input file, directly inherited from former RIPPLE code, have to be concluded by the "end" statement (see the input file printout).

Wave parameters

The source function for wave generation is chosen. The source function is 12 cm long (cells 150 to 152 in x-direction) and 11 cm high (cells 13 to 23 in y-direction), and it is located at 6.08 m from the left boundary. Wave height and period are 7 cm and 1.6 s respectively. Stokes V order waves are generated.

Output results

Results of the whole mesh are written in Type I files from the beginning till the end of the simulation.
The considered mesh has 100+200+294+1+100+1+294+300 = 1290 cells in the x-direction and 30+30 = 60 cells in the y-direction. Thus, the total number of points with results in the x- and y-direction are 1290+2 and 60+2 respectively.
Results are written in Type IV files (gages files) from the beginning till the end of the simulation every 0.5 s. The total number of gages considered is 18.
Results are written in Type III files (animation files) from the beginning till the end of the simulation every 0.1 s.

Porous media

Two types of porous media are defined. The first one, with a 0.49 porosity and a D_{50} = 1.18 cm, represents the core of the rubble mound. The second one, with a 0.53 porosity and a D_{50} = 3.87 cm, represents the first layer of the rubble mound.
They are both defined by 4 conic functions, with the same criteria as the obstacles.

<u>Porous media 1:</u>

$f_1 = y+0.5x-0.2726$	fill for $f1<0$
$f_2 = -y-0.5x-7.3574$	eliminate for $f2<0$
$f_3 = -y+8.3951$	eliminate for $f3<0$

Porous media 2:

g_1 = y+0.5x-0.35	fill for g1<0
g_2 = -y-0.5x-7.28	eliminate for g2<0
g_3 = -y+8.4725	eliminate for g3<0

Turbulence model

Nonlinear eddy viscosity (anisotropic) model is used.

Reflection conditions at boundaries

No weak reflection model at the left boundary, radiation modified model at the right boundary and sliping condition are used.

Sponge layer

The sponge layer used is 5.96 m long (149 cells).

Free surface tracking method

VOF method is used.

MAC method controlling parameters

As nfree is 1, ntracer is 0.

Pollutant and sediment transport parameters

No substances are introduced.

Restart controlling parameter

Calculation begins from quiescent situation (no previous simulation results are used).

Overtopping calculation

Overtopping is not calculated.