

## CHAPTER 38

### NUMERICAL SIMULATION AND VALIDATION OF PLUNGING BREAKERS USING A 2D NAVIER-STOKES MODEL

H.A.H. Petit<sup>1)</sup>, P. Tönjes<sup>2)</sup>, M.R.A. van Gent<sup>3)</sup>, P. van den Bosch<sup>1)</sup>

#### ABSTRACT

The numerical model SKYLLA, developed for simulation of breaking waves on coastal structures is described. The model is based on the Volume Of Fluid method and solves the two-dimensional (2DV) Navier-Stokes equations. Weakly reflecting boundary conditions allow waves to enter and leave the computational domain. Impermeable boundaries can be introduced to simulate a structure. A two-model approach can be used to simulate overtopping over a low crested structure. Results obtained with the model are compared with those obtained with physical model tests for waves on a 1:20 slope of a submerged structure.

#### INTRODUCTION

Traditionally, wave motion on coastal structures was studied by means of physical small-scale model tests. Some phenomena can be studied quite well on a small scale whereas others, like those which involve effects of viscosity, cannot. Numerical models do not have the disadvantage of scaling however, they have the disadvantage that the equations they solve represent a simplification of reality. Most models used to simulate wave motion on structures either solve the shallow water equations or potential flow formulations. For examples of the first we refer to Kobayashi et al.(1987) and Van Gent (1994). For examples of methods based on potential flow we refer to Klopman (1987) for the two dimensional case and to Broeze (1993) for a solver for three-dimensional flow. The shallow water equation

---

<sup>1)</sup> Delft Hydraulics, P.O.Box 152, 8300 AD, Emmeloord, The Netherlands.

<sup>2)</sup> Ministry of Transport, Public Works and Water Management. Directorate-General of Public Works and Water Management (Rijkswaterstaat), Road and Hydraulic Engineering Division, P.O. Box 5044, 2600 GA, Delft, The Netherlands.

<sup>3)</sup> Delft University of Technology, Dept. Civil Engrg., P.O. Box 5048 GA Delft, The Netherlands.

solvers cannot directly simulate wave breaking but need to add extra dissipation to simulate the wave height reduction caused by breaking. The potential flow solvers can solve the flow very accurately up to the moment where the flow domain becomes multiply connected as a result of the breaking process. After that moment these methods become unstable and the calculation breaks down. Solvers based on the MAC (Marker And Cell) or the VOF (Volume Of Fluid) method can solve the Navier-Stokes equation for breaking waves.

## THE 2D-NAVIER-STOKES MODEL

The Volume Of Fluid method (Hirt and Nichols, 1981) has been made applicable for simulation of wave and flow phenomena on coastal structures (Petit & Van den Bosch, 1992 and Van der Meer et al., 1992). The model solves the two-dimensional incompressible Navier-Stokes equations with a free surface.

For the treatment of the free surface a redistribution of water contained in the cells of the computational grid has to take place once the velocity is known. The method called FLAIR (Ashgriz and Poo, 1991) has been adopted for this purpose. Arbitrary free-slip boundaries can be introduced in the model to simulate breaking waves on impermeable coastal structures. The numerical simulation of the breaking process is not limited to the moment where the fluid domain becomes multiply connected.

## IMPROVEMENTS

Recent improvements of the model involve the use of weakly reflecting boundary conditions that allow nonlinear waves based on a Rienecker and Fenton (1981) (R&F) formulation to enter the domain. Further improvements allow the simulation of overtopping at a dike, not only with respect to the volume of water, but also a detailed simulation of water running down the rear of the dike (Petit et al. 1994). Furthermore, the simulation of flow through permeable structures has been made possible for the model (Van Gent et al. 1993) which, however, is beyond the framework of this paper.

## WEAKLY REFLECTING BOUNDARY CONDITIONS

In Figure 1 we show a situation where weakly reflective boundary conditions are needed at both sides of the model.

The waves are assumed to enter the domain at the left. They are given by the free surface elevation  $\eta_{in}(x_0, t)$ , and the velocity components  $u_{in}(x_0, y, t)$  and  $w_{in}(x_0, y, t)$  in x- and y direction respectively. At the right boundary the incoming waves are set to zero, although the model allows waves to be sent in from both sides. The equations which prescribe the weakly reflecting boundary conditions at the left boundary are:

$$\frac{\partial}{\partial t}(\eta - \eta_{in}) - C \frac{\partial}{\partial x}(\eta - \eta_{in}) = -r(\eta - \bar{\eta}) \quad (1)$$

$$\frac{\partial}{\partial t}(u - u_{in}) - C \frac{\partial}{\partial x}(u - u_{in}) = -r(u - \bar{u}) \tag{2}$$

$$\frac{\partial}{\partial t}(w - w_{in}) - C \frac{\partial}{\partial x}(w - w_{in}) = 0 \tag{3}$$

Here, it is assumed that at the boundary the free surface elevations and the velocity components can be decomposed as the sum of a wave travelling to the right and a wave travelling to the left. For the surface elevation at the left boundary this becomes:

$$\eta(x,t) = \eta_{out}(x+Ct) + \eta_{in}(x-Ct)$$

In the case where  $r=0$  this signal satisfies the weakly reflecting boundary condition (1) perfectly. If, again for the case of  $r=0$ ,  $\eta$  is replaced by  $\eta + d$  where  $d$  is a constant, equation (1) will still be satisfied. This means that a change in the time averaged water level caused by inaccuracies in the code will not be corrected by the boundary conditions. The same problem occurs for the weakly reflective boundary condition for the x- velocity component (2). We

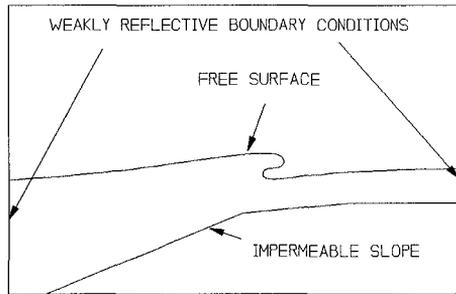


Figure 1 Model application with two weakly reflective boundary conditions

have experienced that in using free slip boundary conditions at the bottom unrealistic average velocities can develop during lengthy computations. By choosing  $r$  equal to a small positive constant a time averaged value for the free surface elevation  $\eta$  and for the velocity in x direction  $\bar{u}$  can be prescribed. Although the amplitude of the incoming signals will be reduced for positive values of  $r$ , small values like  $r = \omega/5$  which theoretically reduce the amplitude by a factor of 0.995 prove to work quite well.

In order to test the quality of the weakly reflecting boundary conditions with incoming nonlinear waves, the velocities and the free surface elevation from the R&F solutions were used. At the left boundary of a numerical wave flume with a constant water depth, these waves were generated using a weakly reflecting boundary condition. At the right boundary again a weakly reflecting boundary condition was used to allow the waves to leave the domain undisturbed. At both boundaries of the Navier-Stokes model the velocities and the surface elevation were calculated and compared with the incoming signal. For a flume with the length of one wave length (wave height 0.2 m, period 3.0 s) the time series of the free surface elevation at the right and the left boundary are shown in Figure 2. Here we can see that, once the

initial disturbances have left the domain, incoming and outgoing signals match nicely.

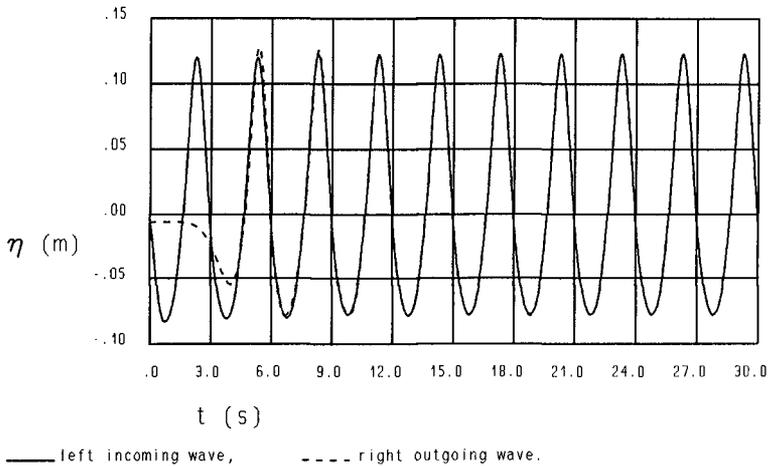


Figure 2 Time series of free surface elevation in a numerical wave flume in order to test the weakly reflecting boundary conditions at both sides

### IMPERMEABLE BOUNDARIES

The velocity components used in the VOF method are defined at the centres of the cell faces. In order to discretize the spacial derivatives in the Navier-Stokes equations, velocity components at several locations are used. They are indicated by the arrow in Figure 3 for the case of the momentum equations in the horizontal direction.

In order to model an impermeable boundary as indicated by the line, one could choose to change the stencil of velocity components such that none of the velocity components needed in the discretization is beneath the impermeable boundary. The disadvantage of this approach is that on a vector computer the vectorization of the computational process would be frustrated by the different treatment of the equations inside the fluid and at the boundaries. We wanted to avoid this problem and decided to define virtual velocities at those positions beneath the impermeable boundary. They are indicated by the dotted vectors in Figure 3. In Figure 4 an example of a submerged structure is shown where only the virtual velocities are given. The virtual velocities which are to be defined beneath the surface of the structure are determined by the boundary conditions at the surface.

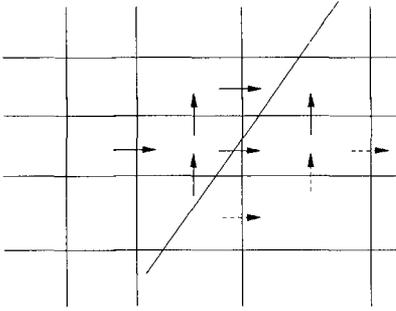


Figure 3 Velocity components needed to discretize the horizontal momentum equations

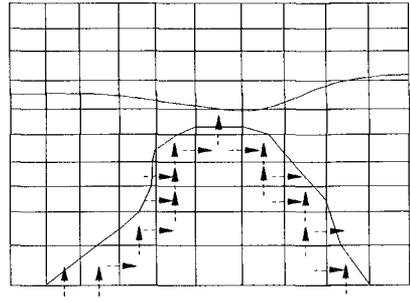


Figure 4 Virtual velocity components beneath the surface of the structure

In the program only the free slip boundary condition was implemented. Both conditions at the impermeable surface now become:

$$\frac{\partial u_{\tau}}{\partial n} = 0 \tag{4}$$

$$u_n = 0 \tag{5}$$

where  $u_n$  is the velocity component in normal direction to the impermeable surface,  $n$  the coordinate in this direction and  $u_{\tau}$  the velocity component along the surface.

The 14 cell categories which are identified in the program are shown in Figure 5. As can be seen here the impermeable boundary is to be modelled as a straight line inside each cell. For the case of category 4 we will examine how the virtual velocities can be determined. The velocity components shown in this figure are those which are used to discretize the impermeability and free slip condition.

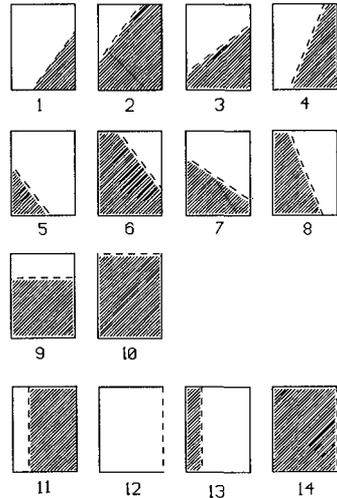


Figure 5 Examples of cell categories used in SKYLLA

By using the components of the normal unit vector at the part of the slope in this cell,  $n_x$  and  $n_y$ , the equations (4) and (5) can be rewritten as:

$$n_x^2 \frac{\partial w}{\partial x} - n_y^2 \frac{\partial u}{\partial y} - n_x n_y \left( \frac{\partial u}{\partial x} - \frac{\partial w}{\partial y} \right) = 0 \tag{6}$$

$$n_x u + n_y v = 0 \tag{7}$$

The velocity component shown in Figure 6 can be used to find a first order accurate approximation of the derivatives in the free slip condition at the collocation point indicated by the small circle in Figure 6. At his same position the impermeability of the slope can be approximated second order accurately by using linear interpolation. In this way two linear equations are found from which the virtual velocities can be determined. For each cell category the two virtual velocities involved are chosen such that all velocities needed for the discretization of the Navier-Stokes equation are available. Furthermore, the virtual velocities determined for one cell do not coincide with the virtual velocities of another cell.

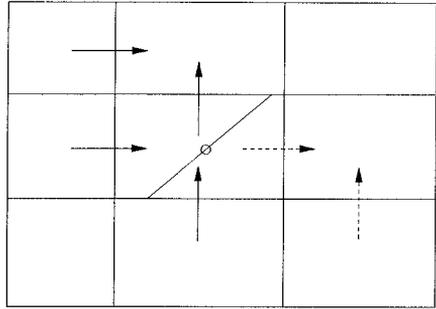


Figure 6 Virtual velocity components for a cell of category 4

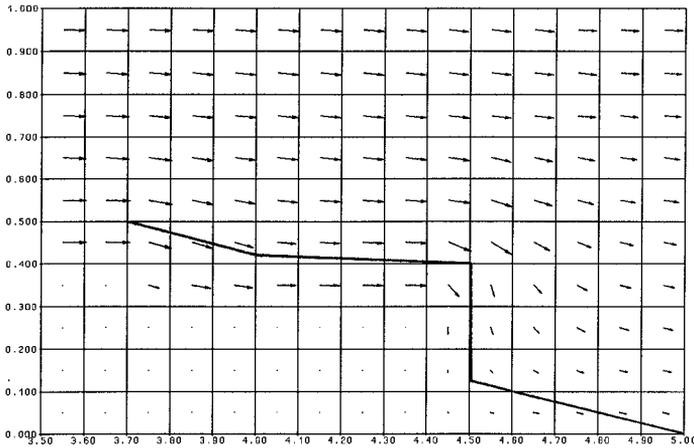


Figure 7 Test for impermeable free-slip boundaries

In Figure 7 we show the result of a computation with a falling slope. The components of the velocity vectors shown here were determined as the averaged values of the velocity components at the boundaries of each cell. The velocities beneath the impermeable boundary are partially determined by the virtual velocities. As can be seen in this figure, the resulting flow near the structure is well aligned with the surface of the structure.

## OVERTOPPING BOUNDARY CONDITIONS

Computations with the VOF method are very costly. Especially if the cell sizes are small the explicit time solver will need very small time steps to keep the computations stable. In each time step a Poisson pressure equation needs to be solved to ensure the incompressibility of the fluid. This leads to a set of equations to be solved for the pressure in each cell. The computational effort to solve the pressure equations is roughly proportional to  $(N*M)^{2.5}$  where  $N$  is the number of cells in horizontal direction and  $M$  the number in vertical direction. In cases where a low-crested structure is to be modelled the computation can be carried out applying two separate computational domains, provided that the flow at the top of the crest has supercritical velocity.

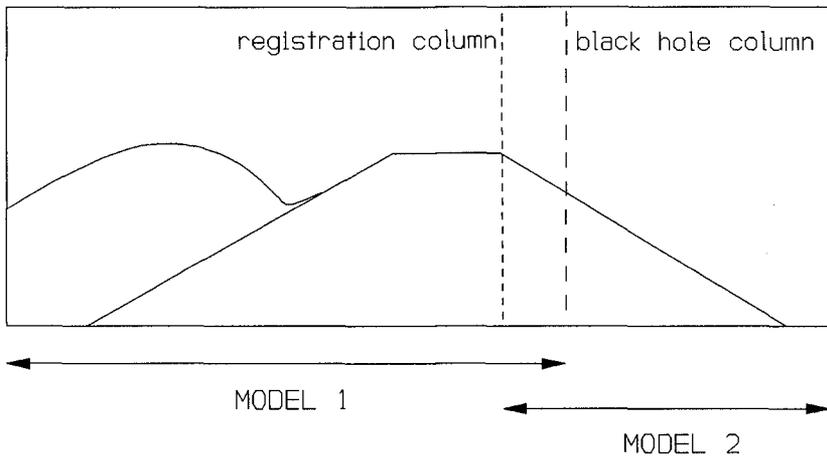


Figure 8 Registration and black hole column at different locations used in the two model approach

The first part of the computation takes place in model 1 as indicated in Figure 8. At the right boundary indicated by 'black hole column' we use the boundary conditions  $u_x = 0$  and  $w_x = 0$ . Furthermore, we set the  $F$  value equal to zero in this column each time step. The  $F$  values and velocity components are registered during the

computation in the 'registration column' each time step. The registered quantities are used during the second computation which involves the flow in model 2 as indicated in Figure 9.

In a strict sense the incompressibility condition can only be satisfied in a simply connected domain by solving the pressure Poisson equation. We expect, however, that in the case where free slip boundary conditions are used and the layer of water at the crest of the structure flows with a supercritical velocity, the errors introduced by using this method will be small.

### VALIDATION OF WAVES ON A 1:20 BAR

In order to gain insight in the performance of the numerical model, physical model tests were performed with waves on a submerged bar with a front slope of 1:20. Here we did not use the two-model approach as the velocities at the top of the structure would not be supercritical. Incident regular waves broke on this bar as weakly plunging breakers. Figure 9 shows the experimental set-up used. The numerical set-up used in the verification runs was simpler because at the time the verification took place the falling slope option had not been implemented. Figure 10 shows the left part of the slope used in the experiment.

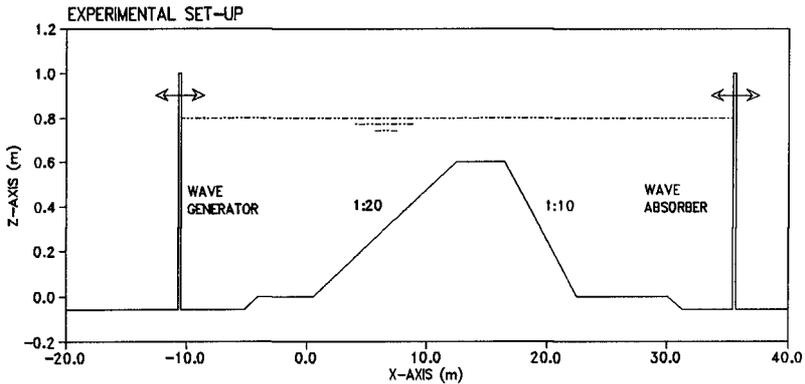


Figure 9 Bottom topography used in wave flume

Velocity profiles at a large number of locations were measured with Laser-Doppler Velocity meters. Those positions are indicated by the blocks in Figure 10. The wave profile was recorded using a video camera. The position of the free surface was determined electronically from the video registration, which resulted in two or more lines in regions with much air entrainment. As the position of the free surface is not a variable in the VOF method the free surface had to be defined using the

F-function. This quantity is the volume fraction of the cell which is filled with fluid (which explains the name Volume Of Fluid). We chose the value  $F = 1/2$  to define the location of the free surface. To send in waves into the numerical model we used solutions obtained by the R&F (1981) method. The parameters used to get this solution were obtained by comparing the free surface as prescribed by R&F with the measured free surface of the incoming waves assuming the reflected waves to be negligibly small.

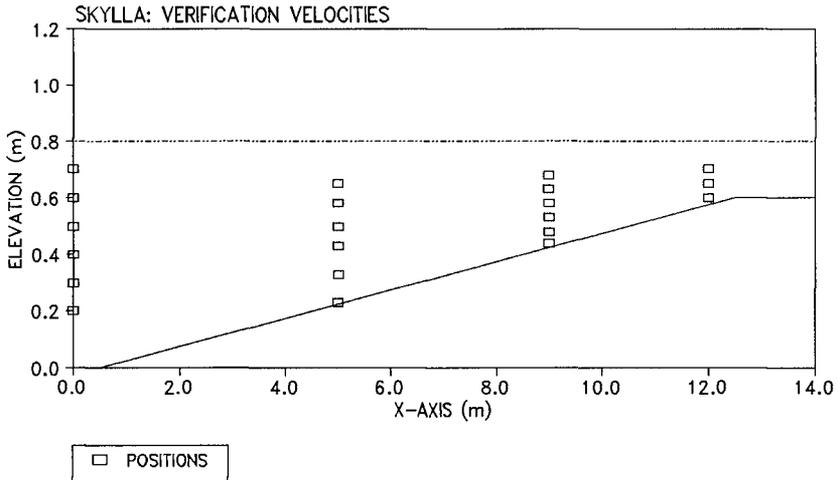


Figure 10 Schematized bottom used in numerical simulation

The following wave parameters were found:

Wave height : 0.29 m  
 Wave period : 1.80 s  
 Still water level : 0.80 m

For the R&F solution we used 16 Fourier components. The mean Eulerian velocity was set to zero m/s. The resulting wave length of the incoming waves was 4.41 m. Comparison of the velocity profiles of the R&F solutions and the measured velocities showed that the crest velocities were somewhat too large whereas the trough velocities were underestimated in an absolute sense. We expect this to be caused by the fact that the undertow is assumed to be uniformly distributed over the vertical in the potential model solved by R&F. In practice, however, smaller velocities occur near the bottom and higher velocities more upward in the vertical.

For the computation 480 cells were used in horizontal direction and 50 in the vertical. The kinematic viscosity was set to  $0.001 \text{ m}^2/\text{s}$ .

After the numerical solutions had become periodic we started the comparison. Figure 11 shows the first of the comparisons. Here we see that once the waves start climbing the slope the wave length of the numerical waves become smaller than the measured value.

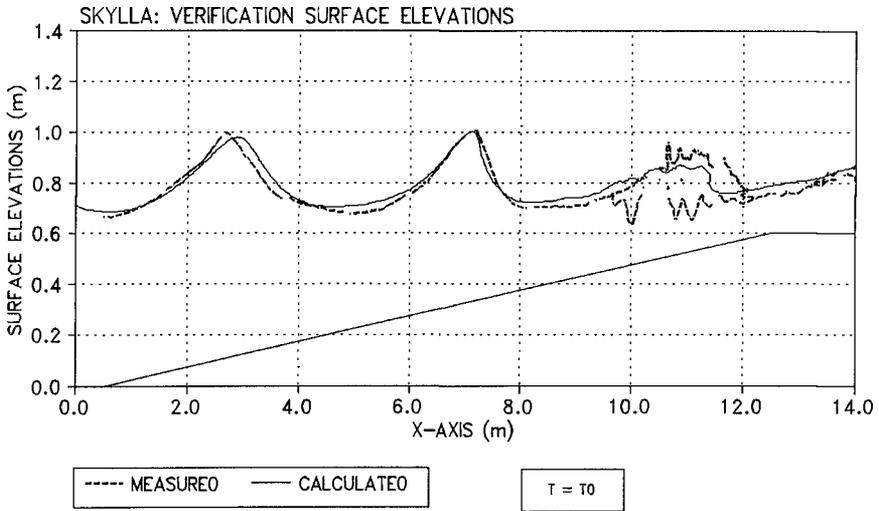


Figure 11

Here we can also see that, the breaking process in the numerical model takes place at the right position. Figures 12 and 13 show the comparison at time intervals of 0.48 s.

The effect of shoaling which is clearly visible in Figure 12 for the measured spilling wave at 8.5 m is not represented well in the numerical simulation. Furthermore, it can be seen in Figure 13 that the breaking process itself develops faster in the numerical model as the decrease in wave height is faster. The transmitted waves at the right boundary, however, were found to be rather accurate.

In Figure 14 we show the measured and computed horizontal and vertical velocities at the left boundary of the model. The problems which arise when in using R&F solutions regarding undertow which were mentioned earlier, reduce the absolute velocity at the trough of the waves. In Figures 15 and 16 we show the comparison of these velocities at 5 and 9 m from the left boundary of the computational domain. All velocities shown here were measured at about 0.5 m from the zero level of the wave flume.

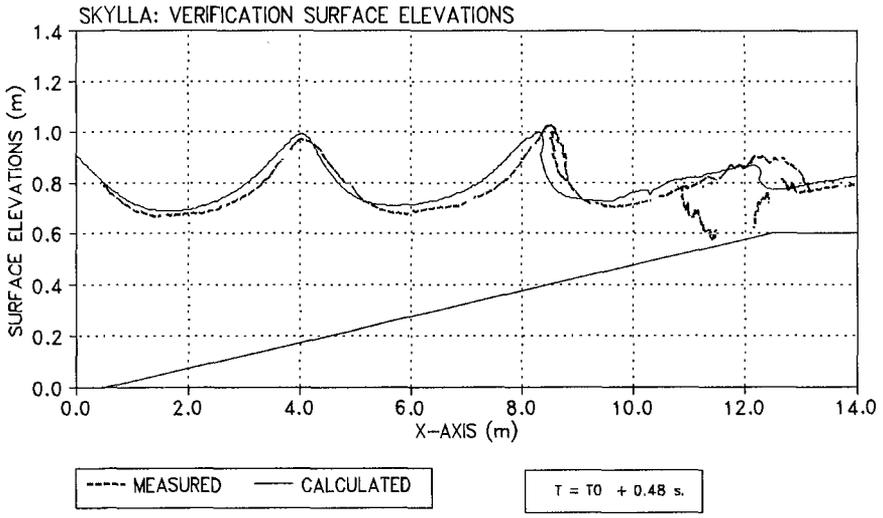


Figure 12

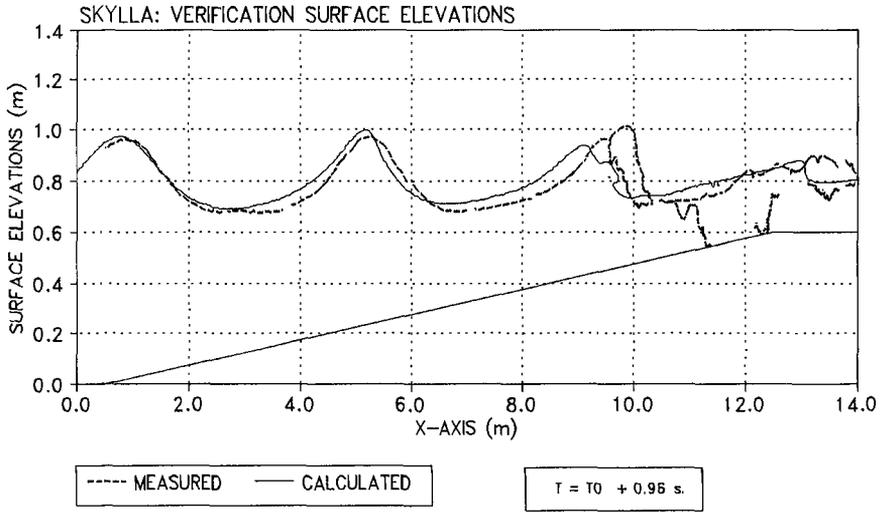


Figure 13

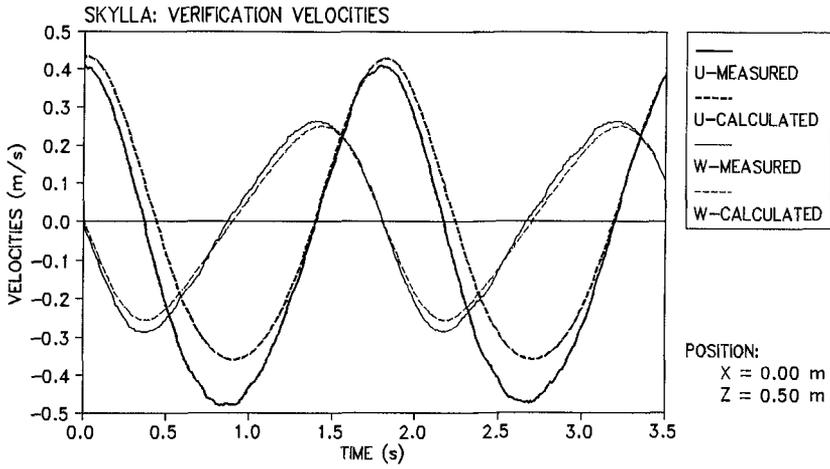


Figure 14 Comparison of measured and calculated velocities; u denotes horizontal velocities, w denotes vertical velocities

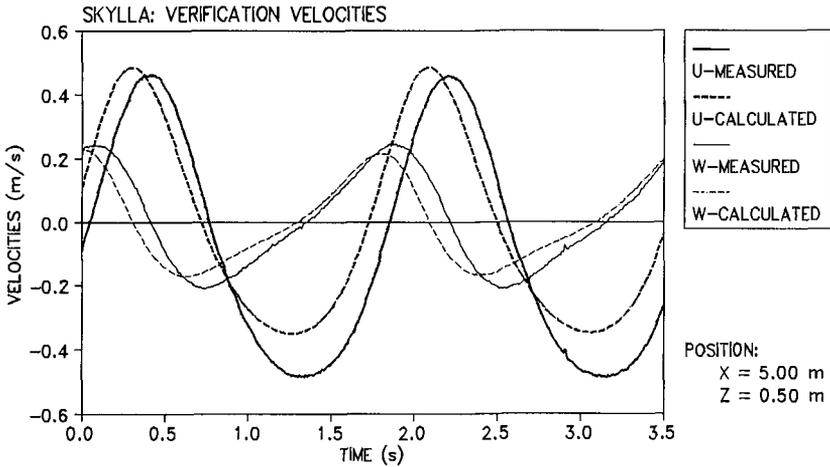


Figure 15 Comparison of measured and calculated velocities; u denotes horizontal velocities, w denotes vertical velocities

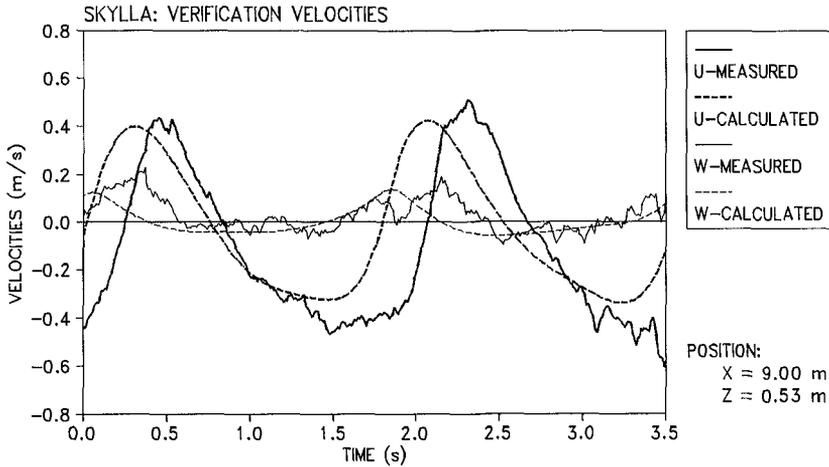


Figure 16 Comparison of measured and calculated velocities;  $u$  denotes horizontal velocities,  $w$  denotes vertical velocities.

## CONCLUSIONS

The VOF method has been made applicable for the computation of breaking waves on coastal structures. Verification with measurements has shown that the program can fairly well simulate waves on a structure. The differences with measurements found in the comparison can partly be explained by the way the boundary conditions were used to define the incoming waves.

## ACKNOWLEDGEMENT

This work was funded by the Ministry of Public Works of the Netherlands, Rijkswaterstaat, Road and Hydraulic Engineering Division.

## REFERENCES

- Ashgriz, N. and J.Y. Poo (1991). "FLAIR: Flux Line-Segment Model for Advection and Interface Reconstruction", *J. of Comp. Physics* 93.
- Broeze, J. (1993). "Numerical Modelling of Nonlinear Free Surface Waves with a 3D Panel Method", Ph.D. Thesis, Twente University, Enschede.

- Hirt, C.W. and Nichols, B.D. (1981). "Volume Of Fluid method for the dynamics of free boundaries". *J. of Comp. Physics* 39.
- Klopman, G. (1987). "Numerical simulation of breaking waves on steep slopes", *Coastal Hydrodynamics*, ed. R.A. Dalrymple.
- Kobayashi, N., Otta A.K. and Roy I. (1987). Wave reflection and run-up on rough slopes, *J. of Water ways, Ports, Coastal and Ocean Engrg.*, ASCE, Vol. 113, no. 3.
- Petit, H.A.H. and Van den Bosch, P. (1992). SKYLLA: Wave motion in and on coastal structures. Numerical analysis of program modifications. Delft Hydraulics Report H1351.
- Petit, H.A.H, Van den Bosch, P and Van Gent, M.R.A. (1994). SKYLLA: Wave motion in and on coastal structures, Implementation of impermeable slopes and overtopping-boundary conditions. Delft Hydraulics, report no H1780.11.
- Rienecker, M.M. and Fenton, J.D. (1981). "A Fourier Method for Steady Water Waves", *J. of Fluid Dynamics*, Vol. 104.
- Van Gent, M.R.A., De Waal, J.P., Petit, H.A.H. and Van den Bosch, P. (1994). SKYLLA: Wave motion in and on coastal structures, Verification of kinematics of breaking waves on an offshore bar. Delft Hydraulics, report no H1780.03.
- Van Gent, M.R.A. (1994). "The modelling of wave action on and in coastal structures", *Coastal Engineering*, Elsevier, Amsterdam. Vol. 22.
- Van Gent, M.R.A., Petit, H.A.H., Van den Bosch, P. (1993). SKYLLA: Wave motion in and on coastal structures, Implementation and verification of flow on and in permeable structures. Delft Hydraulics, report no H1780.
- Van der Meer, J.W., Petit, H.A.H., Van den Bosch, P., Klopman, G. and Broekens, R.D. (1992). "Numerical Simulation of Wave Motion on and in Coastal Structures". ASCE, Proc. 22nd ICCE, Venice, Italy.