CHAPTER 135

NUMERICAL SIMULATION OF WAVE MOTION ON AND IN COASTAL STRUCTURES

J.W. van der Meer[1], H.A.H. Petit[1], P. van den Bosch[1], G. Klopman[1]
and R.D. Broekens[2]

Abstract

A 2-dimensional program for simulation of wave motion on coastal structures is described. The program is based on the Volume Of Fluid method and is able to describe fully plunging waves on all kind of structures.

Introduction

Simulation of wave motion on coastal structures, such as dikes and breakwaters, has traditionally been done by using physical small scale models. Most phenomena in these models reproduce nature fairly well. But phenomena such as porous flow, wave impacts and viscous effects, can not be modelled correctly. Furthermore, measurement of flow fields in breaking waves on a slope is difficult and may be easier to calculate by a numerical model.

The numerical simulation of wave motion on coastal structures will be presented in this paper. Most literature on this subject describes the 1-dimensional "bore approach", i.e. breaking waves are not modelled correctly. Kobayashi and Wurjanto (1989) described such a model. Verification of that model by Van der Meer and Klein Breteler (1990) showed that wave runup and depth-averaged velocities were simulated fairly well and that wave rundown and wave pressures on a slope could not be predicted. A similar and improved model, including porous flow, is given by Van Gent (1992).

Other 2-dimensional models are based on potential flow theory, such as described by Klopman (1987). These kind of models can simulate an overturning wave tongue, but calculations stop before the wave tongue hits the water or a structure.

---

1) Delft Hydraulics, P.O. Box 152, 8300 AD Emmeloord, The Netherlands
2) Rijkswaterstaat, P.O. Box 5044, 2600 GA Delft, The Netherlands

This paper will deal with a 2-dimensional description of the complete wave motion.

The method

The 2-dimensional program SAVOF, developed by the National Aerospace Laboratory in The Netherlands, computes incompressible flow with a free surface in a closed container. The flow is described by the Navier-Stokes equations and the program is based on the program SOLA-VOF, presented by Nichols and Hirt (1981). The programs use the volume of fluid (VOF) method which is, in contrast to surface tracking methods, capable to compute free surface flow when the fluid domain becomes multiply connected, i.e. when for example an overturning breaking wave hits the free surface. SAVOF has been modified and became the code as mentioned above. The new name for the code became SKYLLA.

The fluid is treated as incompressible and the resulting equation for the pressure is treated implicitly where in the original code artificial compressibility or limited compressibility was used in combination with an explicit solver. The results of a first calculation with the original SAVOF-program are shown in Fig. 1. The closed container was put on a slope of 1:4 and the calculation started with a "block" of water in the edge. Plots 5-10 show more or less a plunging breaker on a slope and the subsequent runup.

Possible applications

Various applications can be considered when a numerical simulation of wave motion in and on coastal structures is possible.
- Wave motion on impermeable (smooth or rough) slopes, giving water velocities, accelerations, pressures and runup levels.
- Wave overtopping on impermeable low-crested structures, giving water velocities and overtopping discharges.
- Wave motion on a submerged impermeable structure, giving wave transmission.
- Wave motion on and in a porous rubble structure, giving the same parameters as for an impermeable structure, but also the porous flow, phreatic line and wave transmission.
- Wave motion on vertical structures as caissons, giving wave forces and overtopping.
- Simulation of wave-current interaction on sloping beaches including bars.

Development of the research code SKYLLA

A feasibility study was performed (Broekens and Petit, 1991) on the modifications required or relevant for the application of SKYLLA on the simulation of wave motion on and in coastal structures. The main modifications were:
- prescription of incident waves and a weakly reflecting boundary condition
- description of an impermeable slope
- description of porous flow

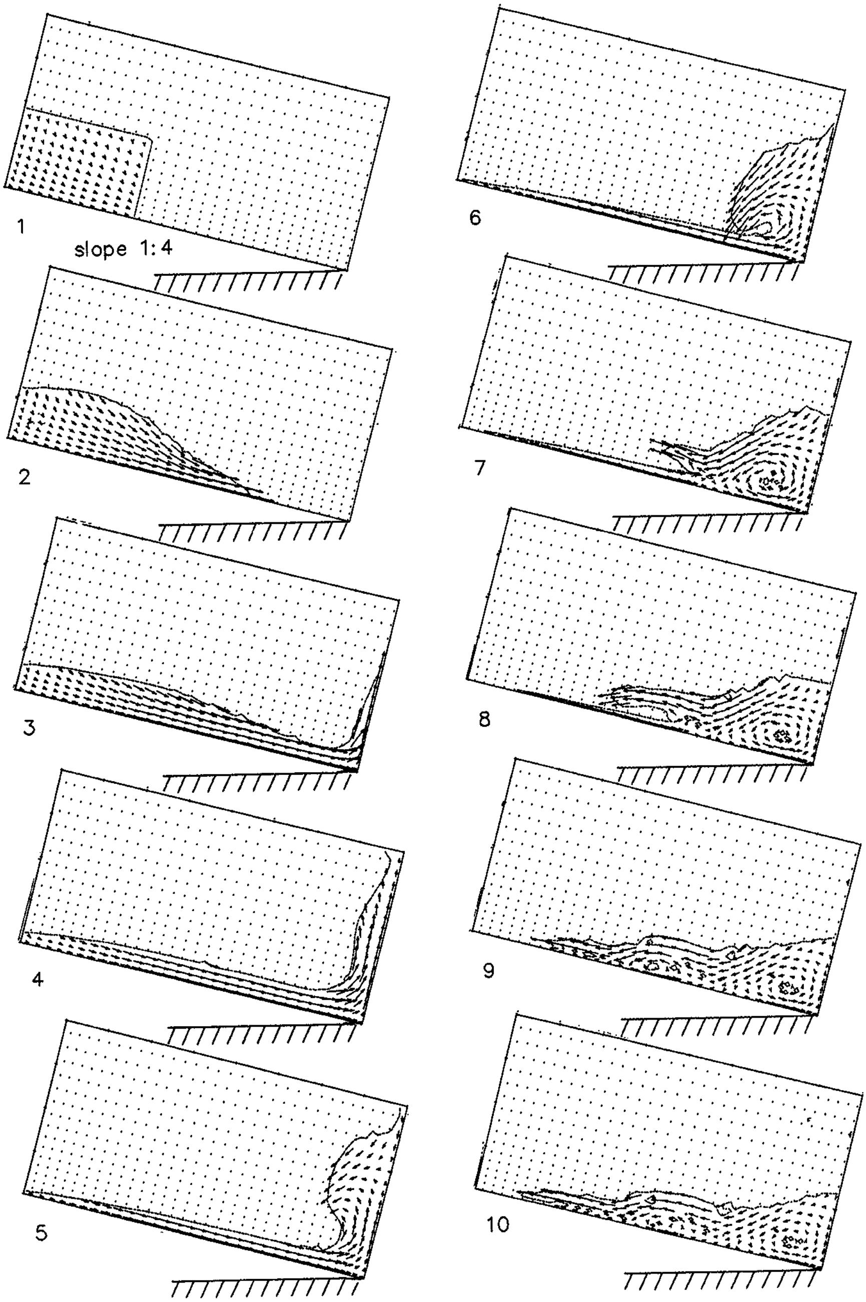

Figure 1 First calculations with SAVOF

The first two modifications will be summarized here and are more fully described in Petit and Van den Bosch (1992).

Weakly reflecting boundary condition

In order to give an idea of how a weakly reflecting boundary condition and an impermeable slope were implemented in the SKYLLA model first the pressure equation for the VOF method will be derived. The Navier-Stokes equations for the momentum in x and y direction respectively are:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial P}{\partial x} + \nu\nabla^2 u \tag{1}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial P}{\partial y} + \nu\nabla^2 v - g \tag{2}$$

where u and v are velocities in the x and y direction respectively and P denotes the reduced pressure $P = p/\rho$, with p = pressure and ρ = mass density of the fluid. Conservation of mass is, for constant ρ, expressed by:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3}$$

By discretizing the momentum equations in time the following equations are found:

$$\frac{u^{n+1} - u^n}{\Delta t} + u^n\frac{\partial u^n}{\partial x} + v^n\frac{\partial u^n}{\partial y} = -\frac{\partial P^{n+1}}{\partial x} + \nu\nabla^2 u^n \tag{4}$$

$$\frac{v^{n+1} - v^n}{\Delta t} + u^n\frac{\partial v^n}{\partial x} + v^n\frac{\partial v^n}{\partial y} = -\frac{\partial P^{n+1}}{\partial y} + \nu\nabla^2 v^n - g \tag{5}$$

Notice that the pressure is taken at the new time level n+1 while both the convection and the viscous terms are taken at the old time level n. Furthermore, the conservation of mass at the new time level n+1 is required:

$$\frac{\partial u^{n+1}}{\partial x} + \frac{\partial v^{n+1}}{\partial y} = 0 \tag{6}$$

From these equations the pressure Poisson equation can be derived by differentiating Eq. (4) to x, Eq. (5) to y and use Eq. (6) to eliminate the velocities at the time level n+1. The result is:

$$\begin{aligned}\nabla^2 P^{n+1} = {} & \frac{1}{\Delta t}\left(\frac{\partial u^n}{\partial x} + \frac{\partial v^n}{\partial y}\right) + \frac{\partial}{\partial x}\left(-u^n\frac{\partial u^n}{\partial x} - v^n\frac{\partial u^n}{\partial y} + \nu\nabla^2 u^n\right) \\ & + \frac{\partial}{\partial y}\left(-u^n\frac{\partial v^n}{\partial x} - v^n\frac{\partial v^n}{\partial y} + \nu\nabla^2 v^n - g\right)\end{aligned} \tag{7}$$

In space the VOF method uses a staggered grid like given in Fig. 2, where the velocities are given at the centre of the cell faces and the pressure is given at the cell centre. At the cell centre also the F value is given which indicates the fraction of the cell which is filled with fluid.

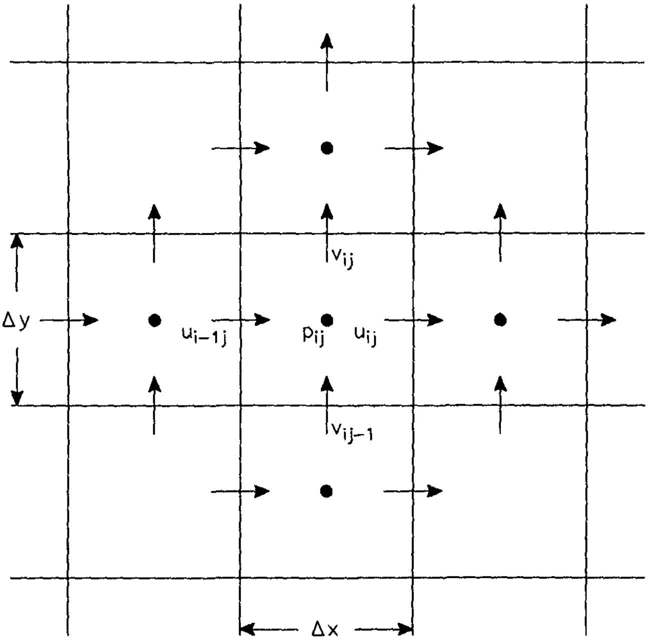

Figure 2 Staggered grid with velocities and pressures

By also discretizing in space the Navier-Stokes equations can be written (here, for simplicity, in the case of an equidistant grid in both x and y direction) as:

$$u_{ij}^{n+1} + \Delta t \frac{p_{i+1j}^{n+1} - p_{ij}^{n+1}}{\Delta x} = \tilde{u}_{ij} \tag{8}$$

$$\text{where } \tilde{u}_{ij} = u_{ij}^{n} + \Delta t \, \mathrm{DISU}_{ij}\left(-u^{n} \frac{\partial u^{n}}{\partial x} - v^{n} \frac{\partial u^{n}}{\partial y} + \nu \nabla^{2} u^{n} \right) \tag{9}$$

and DISU_{ij} stands for an operator that discretizes at the U velocity point.

$$v_{ij}^{n+1} + \Delta t \frac{p_{ij+1}^{n+1} - p_{ij}^{n+1}}{\Delta y} = \tilde{v}_{ij} \tag{10}$$

$$\text{where } \tilde{v}_{ij} = v_{ij}^{n} + \Delta t \, \mathrm{DISV}_{ij}\left(-u^{n} \frac{\partial v^{n}}{\partial x} - v^{n} \frac{\partial v^{n}}{\partial y} + \nu \nabla^{2} v^{n} - g \right) \tag{11}$$

Here DISV_{ij} is an operator that discretizes at the V velocity point.

The discretized version of Eq.(6) is given by:

$$\left(u_{ij}^{n+1} - u_{i-1j}^{n+1}\right)\Delta y + \left(v_{ij}^{n+1} - v_{ij-1}^{n+1}\right)\Delta x = 0 \tag{12}$$

By using this equation the velocities at time level n+1 can be eliminated and the discretized version of the pressure Poisson equation is found:

$$\frac{P_{i+1j}^{n+1} - 2P_{ij}^{n+1} + P_{i-1j}^{n+1}}{\Delta x^2} + \frac{P_{ij+1}^{n+1} - 2P_{ij}^{n+1} + P_{ij-1}^{n+1}}{\Delta y^2} = \frac{1}{\Delta t}\left(\frac{\tilde{u}_{ij} - \tilde{u}_{i-1j}}{\Delta x} + \frac{\tilde{v}_{ij} - \tilde{v}_{ij-1}}{\Delta y}\right) \tag{13}$$

The velocity arrows which are shown in Fig. 2 indicate all the velocities that are used in the discretization of the right hand side of Eq. (13). At an impermeable boundary, which coinsides with a grid-line, the velocity at the boundary at time-level n+1 can be left out in Eq. (12) and the velocities outside the flow domain, which are needed to calculate DISU or DISV at the boundary, can be chosen such that e.g. a free slip boundary condition is met. These velocities are called virtual velocities.

Once the pressure equation is solved Eq. (8) and (10) can be used to find the velocities at the new time level.

In Fig. 3 the situation at a left boundary is sketched. The velocities v_{0j}^n and v_{0j-1}^n are virtual velocities.

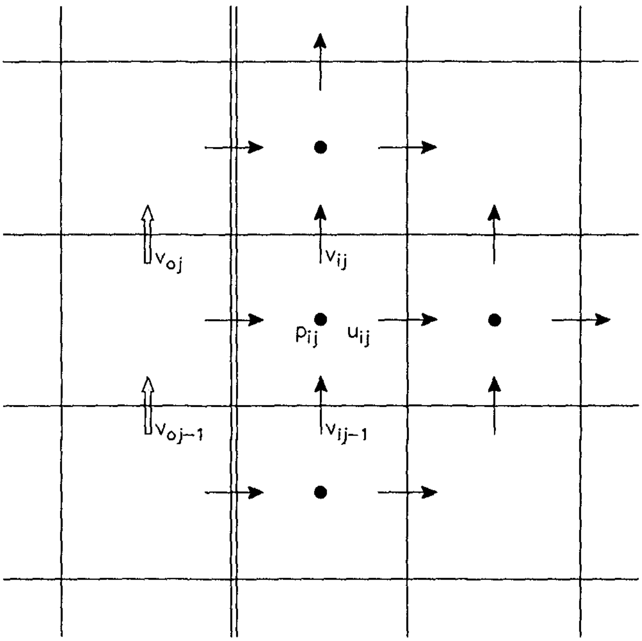

Figure 3 Left side boundary

In order to allow waves to enter and leave at the left boundary, the following equations were discretized:

$$\frac{\partial u}{\partial t} - c\frac{\partial u}{\partial x} = \frac{\partial u_{in}}{\partial t} - c\frac{\partial u_{in}}{\partial x} \tag{14}$$

$$\frac{\partial v}{\partial t} - c\frac{\partial v}{\partial x} = \frac{\partial v_{in}}{\partial t} - c\frac{\partial v_{in}}{\partial x} \tag{15}$$

$$\frac{\partial \eta}{\partial t} - c\frac{\partial \eta}{\partial x} = \frac{\partial \eta_{in}}{\partial t} - c\frac{\partial \eta_{in}}{\partial x} \tag{16}$$

The discretization of Eq.(14) was done upwind for the outgoing waves to yield an explicit expression for u_{0j}^{n+1}. Note that the right hand side of this equation involves the incoming wave and is supposed to be known. Eq. (15) was discretized implicitely using timelevels n and n-1, yielding an explicit expression for the virtual velocity v_{0j}^{n}. Equation (16) was also discretized upwind for the outgoing waves and explicitely in time where η in Eq. (16) can be related to the F values in the first two colomns by:

$$\eta_i^n = \sum_{j=2}^{j_{si}} F_{ij}^n \Delta y_j \quad \text{for } i = 1,2 \tag{17}$$

where j_{si} has the property:

$F_{ij} = 1$ for $j = 1(1)j_{si} - 1$

$0 < F_{ij_{si}} < 1$

$F_{ij} = 0$ for $j > j_{si}$

Here it was implicitely assumed that near to the weakly reflecting boundary the surface is a single valued function of x.

Impermeable slope

Again, by setting virtual velocities, an impermeable free-slip slope could be included in SKYLLA, where the slope is allowed to intersect the grid arbitrarily. In Fig. 4 the four possible cell intersections of a climbing slope with pressure cells is shown. Cases where the slope intersects the cell at a corner can numerically be treated as one of the four cases. For each of the cases 2), 3) and 4) two virtual velocities were defined such that the velocity stencil given in Fig. 2 can be used at each cell which contains fluid. These virtual velocities are defined such that a free- or a no-slip boundary condition is satisfied at a given location point on the slope in the intersected cell.

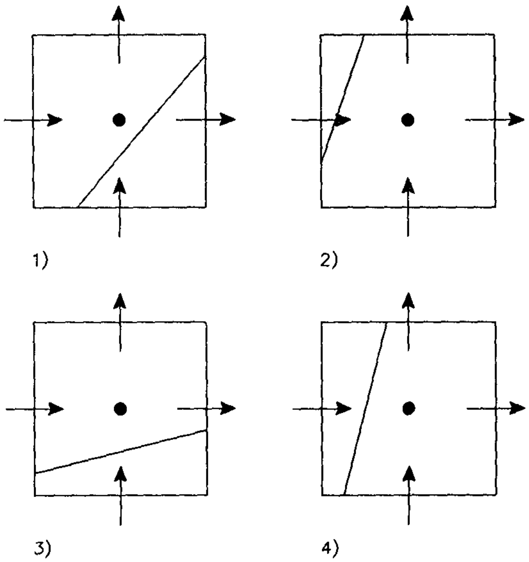

Figure 4 Cells with an impermeable slope

The advantage of this approach over the pressure-velocity iteration technique (Viecelli, 1969), is that an elliptic solver (e.g. Conjugate Gradient Squared (CGS)) can be used to solve the pressure equation instead of the (Succesive Over Relaxation (SOR) like) process of artificial compressibility needed for the pressure-velocity iteration technique. Since solving the pressure equation is by far the most time consuming process in the VOF solver, the use of versions of the CGS method that where specially built for vector computers meant a significant improvement of the performance of SKYLLA.

In order to achieve a more accurate free surface update after new velocities are determined the FLAIR method (Ashgriz and Poo, 1991) was adopted. Updating the fluid domain near the position where the free furface meets the slope proved to be rather difficult however.

Computational results

Various calculations have been done with SKYLLA on smooth impermeable structures in order to test the flexibility and the robustness of the program. Results of one calculation will be given here. No validation tests have been performed until now which means that the results are only output of a computer program and the correspondence with nature has not been verified.

The calculations showed that the grid size is of paramount importance to the results especially when breaking waves occur. The result is that in order to describe breaking waves on a slope small cells and time steps are required. This is not a drawback of the VOF-method as used in SKYLLA, but a direct result from the fact that a nonlinear highly instationary process is simulated.

Fig. 5 gives the cross-section of a dike with a berm. The upper and down slopes are 1:4 and the berm with a length of 5 m has a slope of 1:15 and is located just beneath the still water level. The water depth is 3.4 m at the toe of the structure. This cross-section was used for computation. The generated wave had a wave height of 1.2 m and a wave period of 4.5 s.

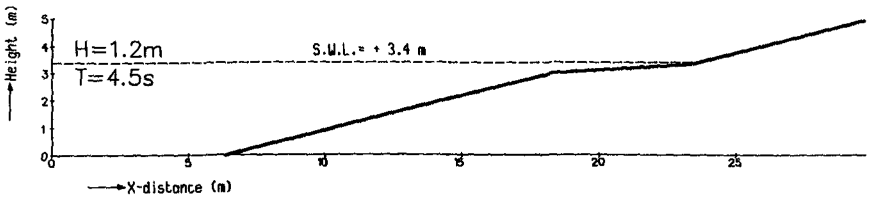

Figure 5 Cross-section used for computations with SKYLLA

In total 350 cells were used in the x-direction with a decreasing cell size from left to right from 0.101 m to 0.050 m. In the y-direction 69 cells were used with a refinement near the still water level (range: 0.041 m to 0.258 m). The basic time step was 0.025 s and the minimum time step needed was 0.0015 s (at 8.6 s, see Fig. 6). In total 17 s wave motion was simulated which required 5977 s of cpu time on a CONVEX C3820.

Figs. 6-9 show results of the calculations at 4 different time steps. Each Fig. has 3 subfigures. At the top the F function is shown where the black colour corresponds with cells that are completely filled with fluid (F = 1) and white cells that are empty (F = 0). The middle plot shows the velocity field in the wet domain. The lowest plot gives the tangential velocity at the free-slip slope as a function of the x-coordinate.

In Fig. 6, at t = 8.6 s, the overturning wave tongue is about to fall on the backwash, thereby multiply connecting the region of the filled cells. Fig. 6c gives the location of the separation point. Fig. 7 gives the result 0.2 s later at t = 8.8 s. The wave tongue has hit the water surface and a horizontal jet emerges from this process. The downstream velocity under the enclosed cylinder of "vacuum" increased. At about x = 18 m the velocity has changed to a shoreward direction.

In reality the enclosed cylinder of air will change to large air-bubbles and escape rapidly upwards out of the fluid. Fig. 7 does not show this escape due to the fact that vacuum was modelled and not air. This is certainly a difference with nature.

Fig. 8 shows the results 2.3 s later when another wave arrives at the toe of the structure and steepens its slope, partly due to the back wash from the previous wave. Further on the slope the breaking goes on which is caused by the nearly horizontal berm and the up and back rushing water. Runup velocities are about 1 m/s. In Fig. 9 the wave starts to break again.

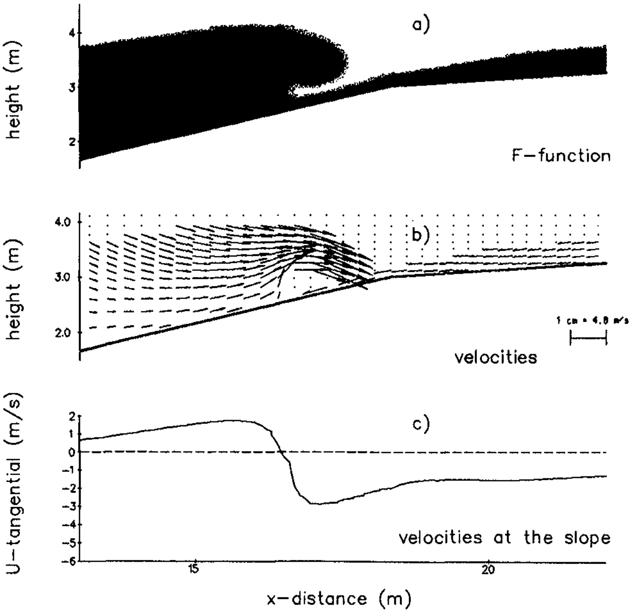

Figure 6 SKYLLA results at t = 8.6 s

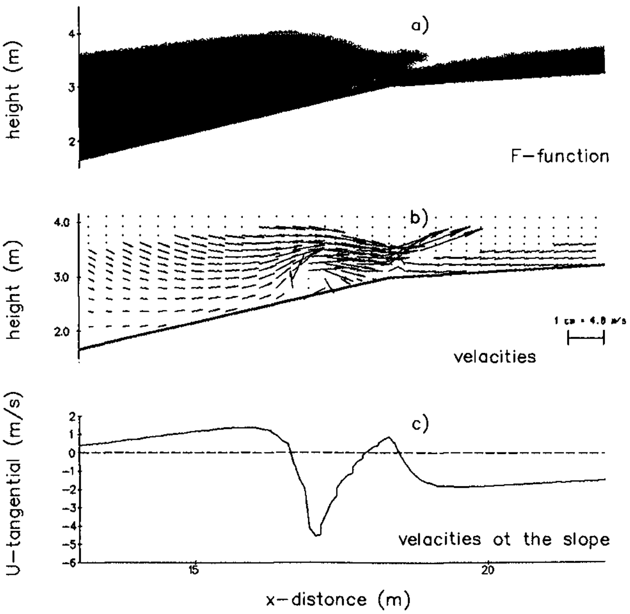

Figure 7 SKYLLA results at t = 8.8 s

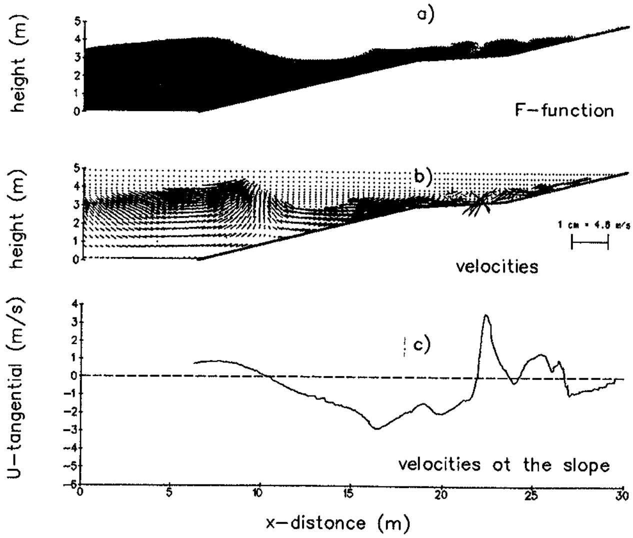

Figure 8 SKYLLA results at t = 11.1 s

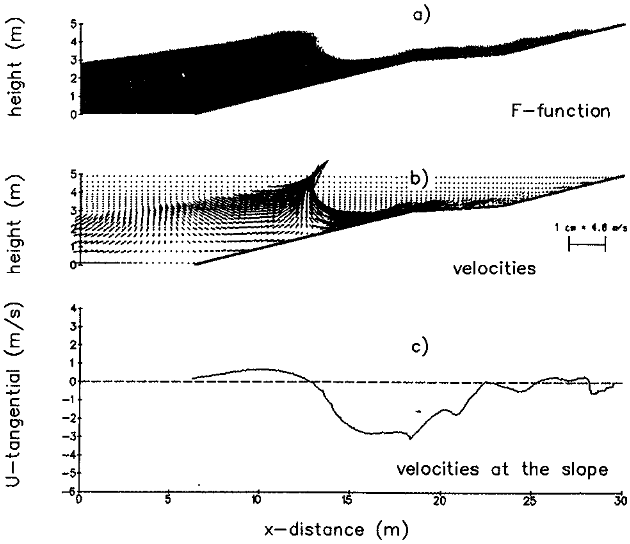

Figure 9 SKYLLA results at t = 12.0 s

Conclusions and recommendations

The results of the computations show that it is possible to simulate breaking waves on a slope. The next step, however, is to perform physical model tests in order to validate the accuracy of the results. The wave profiles at various time steps can then be compared and possibly the whole velocity field using a Partical Image Processing technique.

Attention has to be paid to the effect that vacuum "bubbles" do not escape from the water.

Calculations are still costly. By applying better solvers for solving the Poisson equation on a vector computer the SKYLLA-code has already become 4 times faster than the original code. With the expected increase in calculation speed of supercomputers in the (very) near future and the better solvers that are being developed it is expected that computation time will become less important.

The existing program can be developed further to cope with the possible applications which were mentioned earlier. Wave overtopping can be included and also a porous medium like a breakwater. Recent research on porous flow modelling (the same MAST research, see Acknowledgement) can easily be included, and suggestions to adjust the Navier-Stokes equations for a porous medium have been given in Broekens and Petit (1991). It seems well possible and even straight forward to add extra terms to the momentum equations and to include these in a VOF-method.

Acknowledgements

The work has been performed under a project of the Commission of the European Communities, called MAST - G6 Coastal Structures, project 0032 and was further funded by the Dutch Public Works Department (Rijkswaterstaat) and Delft Hydraulics. Institutes and universities from Denmark, Greece, United Kingdom and The Netherlands were involved in the project.

The objective of the MAST-project was: the development of a physically based numerical formulation of the water motion on a smooth slope (runup, rundown, overtopping, water velocities, water pressures, wave breaking), and a formulation of the same water motion on, but also in a porous rubble structure (phreatic line, pressures and porous flow).

The authors are grateful to A. van der Ploeg who supplied us with a vectorizable preconditioner for the CGS method that was used to solve the pressure equation.

References

Ashgriz, N. and Poo, J.Y. (1991). Axisymmetric Liquid Sloshing under low-g Conditions: Numerical Simulation Method. National Aerospace Laboratory Report NLR TR 86057 L, The Netherlands.

Broekens, R.D. and Petit, H.A.H. (1991). SKYLLA: wave motion in and on coastal structures. Feasibility study on the application of SAVOF. Delft Hydraulics Report H 1351.

Hirt, C.W. and Nichols, B.D. (1981). Volume of fluid method for the dynamics of free boundaries. Journ. of Computational Physics 39, 201-225.

Klopman, G. (1987). Numerical simulation of breaking waves on steep slopes. Coastal Hydrodynamics, ed. R.A. Dalrymple.

Kobayashi, N. and Wurjanto, A. (1989). Numerical model for design of impermeable coastal structures. Research Report No. CE-89-75. University of Delaware.

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 H 1351.

Van der Meer, J.W. and Klein Breteler, M. (1990). Measurement and computation of wave induced velocities on a smooth slope. Proc. ASCE, 22nd ICCE, Delft, The Netherlands.

Van Gent, M.R.A. and Engering F.P.H. (1992). The modelling of wave action on and in coastal structures. Proc. Final workshop MAST G6S, Lisbon.

Viecelli, J.A. (1969). A Method for Including Arbitrary External Boundaries in the MAC Incompressible Fluid Computation Technique, J. Comp. Phys. 4, 543-551.