European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate and J. Périaux (Eds)

ACCURATE AND EFFICIENT COMPUTATION OF STEADY WATER FLOW WITH SURFACE WAVES AND TURBULENCE

Jeroen Wackers* **and Barry Koren†**

*CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
e-mail: Jeroen.Wackers@cwi.nl

†CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
and
Delft University of Technology, Faculty of Aerospace Engineering
P.O. Box 5058, 2600 GB Delft, The Netherlands
e-mail: Barry.Koren@cwi.nl

Key words: Steady water waves, Surface capturing, RANS, Menter model, Multigrid, Defect correction

Abstract. *A surface capturing method is developed for steady water-air flow with gravity. Second-order accuracy is obtained with flux limiting and turbulence is modeled with Menter's model. The model is solved efficiently with a combination of multigrid and defect correction. Results for two test cases confirm the efficiency and accuracy of the method.*

1 INTRODUCTION

Numerical simulation of steady water flow plays a continuously increasing role in the development of ships and offshore structures. Accurate predictions of a ship's wave pattern, for example, can help in reducing the ship's drag during the design, and thus minimise operating costs [7, 9].

A key choice in the development of a numerical model for water flow is the representation of the water surface. The most widely used approach is surface fitting: the computational mesh is deformed during the computation, such that it coincides with the water surface. Another approach is the surface capturing technique (the volume of fluid and level set methods are well-known examples) where the grid is fixed, but the location of the water surface on the grid can change. These methods are more flexible than surface fitting methods; they allow steeper waves and more complex object geometries. A disadvantage is that, for steady flow problems, the classical surface capturing methods are hard to solve efficiently. Currently, they are usually solved by time-marching the unsteady flow equations to convergence, which is a time-consuming process.

In previous work, we have shown that efficient solution of a surface capturing model is possible [13]. We developed a volume-of-fluid method based on conservation laws only, which is solved with classical multigrid. Very fast convergence of the solution process was obtained. But it also became clear that the first-order, laminar flow model was not sufficient to accurately compute realistic water flow.

Therefore, the method is improved by making it second-order accurate and by adding a turbulence model. The second-order model is solved with a defect correction procedure, that uses the efficiency of the first-order multigrid method. Turbulence is modeled with Menter's model [10]: a simple yet accurate one-equation turbulence model that has already been applied in some ship flow solvers.

This paper briefly describes the first-order accurate discretisation and focuses on the extension to second-order accuracy and on the inclusion of the turbulence model. The flow equations are given in section 2, the discretisation of the different terms in section 3. Section 4 describes the defect correction and the special multigrid method that are used to solve the difficult two-fluid flow equations with turbulence. Finally, the practical application of the method is illustrated in section 5 with two test cases: the flow over a flat plate and the water flow in a channel with a bottom bump.

2 FLOW EQUATIONS

The flow equations used here are based on the Reynolds-averaged Navier-Stokes (RANS) equations. The distinction between the water w and air a is made by adding a mass conservation equation for the water. Water and air are both considered incompressible: they have constant densities. Defining α as the volume fraction of water, the mixture density is $\rho = \rho_w \alpha + \rho_a (1-\alpha)$. The resulting system has conservation laws only, which makes it suitable for multigrid solution.

When the mixture density is substituted in the steady compressible (variable-density) 2D RANS equations with gravity, we get the following, incompressible flow equations:

$$
\begin{aligned}
\frac{\partial}{\partial x}\left(p+\rho u^2\right) + \frac{\partial}{\partial y}\left(\rho u v\right) &= \frac{\partial}{\partial x}\left(\left(\mu+\mu_T\right)2u_x\right) + \\
&\quad \frac{\partial}{\partial y}\left(\left(\mu+\mu_T\right)\left(u_y+v_x\right)\right) && (x\text{-mom.}),\\
\frac{\partial}{\partial x}\left(\rho u v\right) + \frac{\partial}{\partial y}\left(p+\rho v^2\right) &= \frac{\partial}{\partial x}\left(\left(\mu+\mu_T\right)\left(u_y+v_x\right)\right) + \\
&\quad \frac{\partial}{\partial y}\left(\left(\mu+\mu_T\right)2v_y\right) - \rho g && (y\text{-mom.}),\\
\frac{\partial}{\partial x}\left(u\right) + \frac{\partial}{\partial y}\left(v\right) &= 0 && (\text{tot. mass}),\\
\frac{\partial}{\partial x}\left(u\alpha\right) + \frac{\partial}{\partial y}\left(v\alpha\right) &= 0 && (\text{water mass}).
\end{aligned}
\tag{1}
$$

The last equation is equivalent to a volume-of-fluid equation, but it is completely in conservation form.

The turbulent viscosity μ_T is computed with Menter's turbulence model [10], adapted for non-constant density flow.

$$\frac{\partial(\tilde{\nu}_T u)}{\partial x} + \frac{\partial(\tilde{\nu}_T v)}{\partial y} = \frac{\partial}{\partial x}\left(\left(\nu + \frac{\tilde{\nu}_T}{\sigma_m}\right)\frac{\partial \tilde{\nu}_T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\left(\nu + \frac{\tilde{\nu}_T}{\sigma_m}\right)\frac{\partial \tilde{\nu}_T}{\partial y}\right) + P - D, \qquad (2)$$

with $\nu = \mu/\rho$. The actual μ_T to be used in the RANS equations is scaled, to give better behaviour close to walls:

$$\mu_T = \rho\nu_T = \rho\left(1 - e^{-\left(\frac{\tilde{\nu}_T}{A^+\kappa\nu}\right)^2}\right)\tilde{\nu}_T. \qquad (3)$$

A^+ and κ are constants. The production P and the dissipation D of the turbulence are:

$$P = c_1\frac{\nu + \nu_T}{\nu + \tilde{\nu}_T}\tilde{\nu}_T\sqrt{2(u_x^2 + v_y^2) + (u_y + v_x)^2}, \qquad D = c_2 c_3 D_{BB}\tanh\left(\frac{D_{k-\varepsilon}}{c_3 D_{BB}}\right), \qquad (4)$$

with

$$D_{k-\varepsilon} = \tilde{\nu}_T^2\frac{(u_{xx} + u_{yy})^2 + (v_{xx} + v_{yy})^2}{u_x^2 + u_y^2 + v_x^2 + v_y^2}, \qquad D_{BB} = (\tilde{\nu_T}_x)^2 + (\tilde{\nu_T}_y)^2. \qquad (5)$$

The model constants have the values $c_1 = 0.144$, $c_2 = 1.86$ and $A^+ = 13.0$. The von Karman constant $\kappa = 0.41$. Furthermore, we take $c_3 = 7$ and $\sigma_m = 1$.

3 DISCRETISATION

The system (1) is discretised with a cell-centered finite-volume technique on structured, curvilinear grids. The convective and diffusive fluxes and the source terms are discretised separately, to independently control the stability of each part. Both first- and second-order accurate fluxes are used.

The derivation of the first-order fluxes is given in [13]. These fluxes are described briefly here, together with a description of the second-order accurate fluxes and the turbulence source terms.

3.1 Convective fluxes

The convective fluxes are computed by reconstructing the states $\mathbf{q} = [u, v, p, \tilde{\nu}_T, \alpha]^T$ on the left and right side of the cell faces from the states in the cell centres. These states are then put in a flux function, that gives the flux over the cell face.

For the first-order accurate fluxes, the state at the cell faces is taken equal to the state in the cell centres. For the second-order accurate fluxes, the states are reconstructed with a slope limiter. A proper choice of the limiter guarantees the stability of the second-order fluxes. It is known that the Minmod limiter is unsuitable for use with defect correction [6]. Instead, we use the $\kappa = \frac{1}{3}$ limiter proposed by Koren [8].

Near the walls, where turbulent boundary layers develop, the grids are usually made highly stretched. Here, it is important to take the non-uniformity of the grid into account when computing the slopes for the limiter. Otherwise, significant errors are introduced.

The flux function itself is an approximate Riemann solver, based on artificial compressibility [2, 12] and comparable to the flux used in [3]. The flux is derived from a solution of the Riemann problem: the flow in a shock tube with a discontinuous initial condition. The initial left and right states are the states $\mathbf{q}_0$ and $\mathbf{q}_1$ on each side of the cell face. The output flux is the flux at the location of the initial discontinuity.

Our solver is a linearised Osher-type flux function, it couples the normal velocity and the pressure on the two sides. The resulting state $\mathbf{q}_{\frac{1}{2}}$ at the cell face (with normal velocity u and tangential velocity v) is:

$$\begin{aligned} u_{\frac{1}{2}} &= u_0 + \frac{p_1 - p_0 + \rho_1 \lambda_1^- (u_1 - u_0)}{\rho_1 \lambda_1^- - \rho_0 \lambda_0^+}, \\ p_{\frac{1}{2}} &= p_0 - \rho_0 \lambda_0^+ \frac{p_1 - p_0 + \rho_1 \lambda_1^- (u_1 - u_0)}{\rho_1 \lambda_1^- - \rho_0 \lambda_0^+}, \end{aligned} \tag{6}$$

with the wave speeds defined as $\lambda^- = \frac{1}{2}u - \sqrt{c^2/\rho + \left(\frac{1}{2}u\right)^2}$ and $\lambda^+ = \frac{1}{2}u + \sqrt{c^2/\rho + \left(\frac{1}{2}u\right)^2}$, for a constant c. The other three state variables v, $\tilde{\nu}_T$ and α are chosen purely upwind:

$$\begin{aligned} v_{\frac{1}{2}} &= v_0, \qquad \tilde{\nu}_{T\frac{1}{2}} = \tilde{\nu}_{T0}, \qquad \alpha_{\frac{1}{2}} = \alpha_0 \qquad \text{if } u_{\frac{1}{2}} \geq 0, \\ v_{\frac{1}{2}} &= v_1, \qquad \tilde{\nu}_{T\frac{1}{2}} = \tilde{\nu}_{T1}, \qquad \alpha_{\frac{1}{2}} = \alpha_1 \qquad \text{if } u_{\frac{1}{2}} < 0. \end{aligned} \tag{7}$$

We construct convective fluxes with these state variables.

3.2 Diffusive flux

The diffusive fluxes in the momentum equations are modeled with central differences. These are both stable and second-order accurate, therefore they are used for both the first- and the second-order accurate discretisation. The horizontal fluxes are:

$$\begin{aligned} (\mu u_x)_{\frac{1}{2}} &= \left(\mu_{\frac{1}{2}} + \mu_{T\frac{1}{2}}\right) \frac{u_{i+1} - u_i}{\Delta x}, \\ (\mu v_x)_{\frac{1}{2}} &= \left(\mu_{\frac{1}{2}} + \mu_{T\frac{1}{2}}\right) \frac{v_{i+1} - v_i}{\Delta x}, \end{aligned} \tag{8}$$

with $\mu_{\frac{1}{2}} = \alpha_{\frac{1}{2}}\mu_w + (1 - \alpha_{\frac{1}{2}})\mu_a$ and with $\mu_{T\frac{1}{2}}$ computed from $\tilde{\nu}_{T\frac{1}{2}}$. For non-cartesian grids and for the cross-diffusion terms, the velocity derivatives are computed with a control volume approach.

3.3 Source terms

Two different source terms appear in the system (1). The gravity force in the y-momentum equation is discretised as $-\rho_i g \, \Delta x \, \Delta y$. The production and dissipation terms

in the turbulence equation contain both first-order and second-order derivatives of the state variables, which are computed with finite differences in a local cartesian grid: two orthogonal coordinate axes are positioned in each cell, aligned with the cell orientation. Then the locations of the neighbour cell centres are projected on these axes and second-order finite difference schemes are constructed with these projected locations.

When the grid is smooth and locally (close to) orthogonal, then this approach is second-order accurate. It is used for both the first-order and the second-order accurate schemes.

4 MULTIGRID

For the solution of nonlinear systems, multigrid is one of the most powerful techniques known today. Its application to fluid flow solutions is mature and well studied [4, 5, 11]. The multigrid principle is to apply a simple solution technique or smoother to the solution on the finest grid and to copies of this solution on underlying coarser grids, to eliminate high-frequency and low-frequency errors, respectively. Smoothers can be, for example, time stepping, Jacobi or Gauss-Seidel relaxation.

4.1 Choice of smoother

When multigrid is used to solve RANS equations, it is usually combined with time stepping. Either a time step smoother is used, or multigrid is used to compute the individual steps in an implicit time stepper. This procedure is robust and stable.

We are not going to use time stepping, for two reasons. First, we think that the fastest solution techniques are obtained when the steady flow equations are solved directly with multigrid. Our analysis shows that this is indeed possible for the two-fluid RANS equations and that other smoothers than the time stepper can be used.

And second, time stepping is inefficient as a smoother for flow with gravity waves, because these flows have transient waves that damp out very slowly in time. Also, the Reynolds number of these flows is usually high, which means that there is little damping by the viscosity. In our previous research [13], we found that a nonlinear line Gauss-Seidel smoothing is very effective for water flow. It gives excellent smoothing when the flow is convection-dominated and it is not prohibitively expensive, since the flow equations for each line of cells can be solved effectively with Newton-Raphson iteration, combined with a Thomas algorithm for the direct solution of the block-tridiagonal systems that appear therein.

Therefore, we solve the discretised flow equations (1) with direct multigrid and alternating line Gauss-Seidel smoothing.

4.2 Linear multigrid

The usual multigrid technique for nonlinear systems is nonlinear multigrid. Here, discretised flow equations are constructed on the coarser grids in exactly the same way as on the finest grid. Then these flow equations are smoothed, using the residual in the

solution on the finer grids as a source term.

For the two-fluid RANS equations, this approach gives two problems:

1. The two-fluid flow equations cannot handle large arbitrary source terms. For example, a source term could specify a sink for water in a cell whose neighbours contain only air. The only way to get a net inflow of water in that cell is to take a negative amount of water in the cell itself. This may lead to negative densities.

2. For nonlinear multigrid to work well, the flow equations on the coarser grids must resemble the flow equations on the finest grid. Here, this is not the case. The discontinuity in the volume fraction α, at the water surface, may lie in a different location on the finer grids and on the coarser grids. Furthermore, the Menter turbulence equation is dominated by the production and dissipation terms; with their second-order derivatives and the high curvature of the solution in boundary layers, these terms may depend strongly on the grid size. This happens mainly when the grid is too coarse to resolve a boundary layer accurately (less than about 16 cells over the thickness of the boundary layer). So the boundary layers on the coarse grids do not resemble those on the finer grids.

We overcome these problems by switching to *linear* multigrid. The flow equations on the finest grid are linearised around the current solution and this linearised operator is copied directly to the coarser grids. These linear coarse grid operators can be solved for any source term and they always resemble the fine grid operator reasonably well.

The algorithm is the following. The finest grid is called K and the underlying coarser grids are $0, \cdots, K-1$. The RANS equations (1), discretised on K, are denoted by the operator F_K, so the problem to be solved is $F_K \mathbf{q}_K = 0$. We call the nonlinear line smoothing operator M_K. The linearised operator L_K is defined as the Jacobian of F_K:

$$L_K = \partial F_K / \partial \mathbf{q}_K. \tag{9}$$

On the coarse grids, we do not compute full solutions $\mathbf{q}$, but (small) corrections $\mathbf{u}$. For these, a prolongation operator P_{k-1}^k is defined that moves a correction from one cell on grid $k-1$ to the four cells on grid k that lie in the same location. For finite volumes, this operator is:

$$(\mathbf{u}_k)_{2i(+1),2j(+1)} = P_{k-1}^k (\mathbf{u}_{k-1})_{i,j} = (\mathbf{u}_{k-1})_{i,j}\,. \tag{10}$$

In the same way, a restriction operator R_k^{k-1} is defined for the defect $\mathbf{f}$:

$$(\mathbf{f}_{k-1})_{i,j} = R_k^{k-1} (\mathbf{f}_k)_{2i(+1),2j(+1)} = (\mathbf{f}_k)_{2i,2j} + (\mathbf{f}_k)_{2i+1,2j} + (\mathbf{f}_k)_{2i,2j+1} + (\mathbf{f}_k)_{2i+1,2j+1}\,. \tag{11}$$

The linear operators on the coarse grids are Galerkin operators, i.e.

$$L_{k-1}\mathbf{u}_{k-1} = R_k^{k-1} L_k P_{k-1}^k \mathbf{u}_{k-1}. \tag{12}$$

In practice, the entries in L_{k-1} are found by summing entries from L_k. A line smoothing step for the linear operator is called M_k^L.

Then the multigrid procedure is defined recursively as follows. It is started on the finest grid K.

$\mathbf{q}_K^{n+1} = \textbf{function } LMG(\mathbf{q}_K^n):$

$\tilde{\mathbf{q}}_K^n = (M_K)^{q_1}\,\mathbf{q}_K^n$	q_1 pre-relaxation steps,
$L_K^n = \partial F_K / \partial \tilde{\mathbf{q}}_K^n$	linearisation,
$\mathbf{f}_{K-1}^n = R_K^{K-1} F_K \tilde{\mathbf{q}}_K^n$	source term,
$L_{K-1}^n = R_K^{K-1} L_K^n P_{K-1}^K$	linearised system,
$\mathbf{u}_{K-1} = 0$	initial solution,
$\mathbf{u}_{K-1}^{n+1} = LMGC\left(K-1, \mathbf{u}_{K-1}^n, \mathbf{f}_{K-1}^n, L_{K-1}^n\right)$	MG on coarser grid,
$\tilde{\tilde{\mathbf{q}}}_K^n = \tilde{\mathbf{q}}_K^n + P_{K-1}^K \mathbf{u}_{K-1}^{n+1}$	correction,
$\mathbf{q}_K^{n+1} = (M_K)^{q_2}\,\tilde{\tilde{\mathbf{q}}}_K^n$	q_2 post-relaxation steps.

$\mathbf{u}_k^{n+1} = \textbf{recursive function } LMGC(k, \mathbf{u}_k^n, \mathbf{f}_k^n, L_k^n):$

$\tilde{\mathbf{u}}_k^n = \left(M_k^L\right)^{q_1}\,\mathbf{u}_k^n$	q_1 pre-relaxation steps,
if $k \neq 0$ **then**	
$\mathbf{f}_{k-1}^n = R_k^{k-1} L_k \tilde{\mathbf{u}}_k^n$	source term,
$L_{k-1}^n = R_k^{k-1} L_k^n P_{k-1}^k$	linearised system,
$\mathbf{u}_{k-1} = 0$	initial solution,
$\mathbf{u}_{k-1}^{n+1} = LMGC\left(k-1, \mathbf{u}_{k-1}^n, \mathbf{f}_{k-1}^n, L_{k-1}^n\right)$	MG on coarser grid,
$\tilde{\tilde{\mathbf{u}}}_k^n = \tilde{\mathbf{u}}_k^n + P_{k-1}^k \mathbf{u}_{k-1}^{n+1}$	correction,
end if	
$\mathbf{u}_k^{n+1} = \left(M_k^L\right)^{q_2}\,\tilde{\tilde{\mathbf{u}}}_k^n$	q_2 post-relaxation steps.

We have chosen this unusual combination of linear multigrid on the coarser grids with nonlinear line smoothing on the finest grids, because the nonlinear smoother is very robust. It can correct small unphysical solutions ($\tilde{\nu}_T < 0$ etc.) that arise from the linear coarse grid corrections.

A disadvantage of the method is, that it requires much computer memory for the storage of the nonlinear operators. Also, Galerkin operators can reduce the convergence rate of multigrid for convective flows ([11], section 5.4). However, in practice this effect was observed but it caused no problems.

On the other hand, the method can solve flows whose features cannot be resolved accurately on coarse grids. Another advantage is, that the coarse grid corrections are cheap. The linear line smoothing is less expensive than nonlinear line smoothing and the

computation of L_K takes little extra work, as the linearisation is already needed for the line smoothing on K.

4.3 Full multigrid

As initial condition, we choose $\tilde{\nu}_T$ very close to zero. Then, when the solution process is started, the turbulence intensity has to grow. Typically, we see the residual in the turbulence equation increase at first, because the turbulence intensity increases. Then, when the boundary layers have more or less developed, it starts to decrease. Experiments show that multigrid relaxation is of little use in the first stage.

Therefore, full multigrid (FMG) is essential to our method. We do not start the solution on the finest grid, but on the coarsest grid on which the boundary layers can be accurately resolved. When the solution on this grid is computed, it is prolongated as initial solution to the next finest grid where the solution is computed too. This is continued until the finest grid is reached. Thus, all but the first computations start at the second stage, with developed boundary layers. The time-consuming development of turbulence is only needed on the first grid.

4.4 Second-order accuracy with defect correction

It is difficult to find good smoothers for the direct multigrid solution of second-order accurate convection equations. Therefore, defect correction is often used [6, 11]. In an iterative process, the residual of the second-order accurate discretisation is computed and used as a source term in the first-order multigrid smoother. Theoretically, this process converges to the second-order solution.

In practice, defect correction does not always converge. But when the process is started from the first-order solution, a great improvement is obtained in the first iterations and the solution is usually second-order accurate after a few steps. So convergence is not needed for a good solution.

For efficiency, we perform the multigrid step in the defect correction with the linearised operators, even on the finest grid. These operators are not updated during the process, since they are stable and close enough to the converged solution to give good smoothing.

5 RESULTS

The practical aspects of the method are illustrated with two test cases. One is a simple test: a turbulent flat-plate boundary layer. The other is a challenging turbulent two-fluid problem: the flow in a channel with a bottom bump.

5.1 Flat plate boundary layer

Single-fluid flow is computed over a flat plate with a Reynolds number of 10^7 based on the length of the plate. This test case is used in Menter's paper [10]. Our grid has 128×128 cells and is highly stretched in y-direction. Multigrid computations are done

on six grids, the coarsest grid is 4×4 cells.

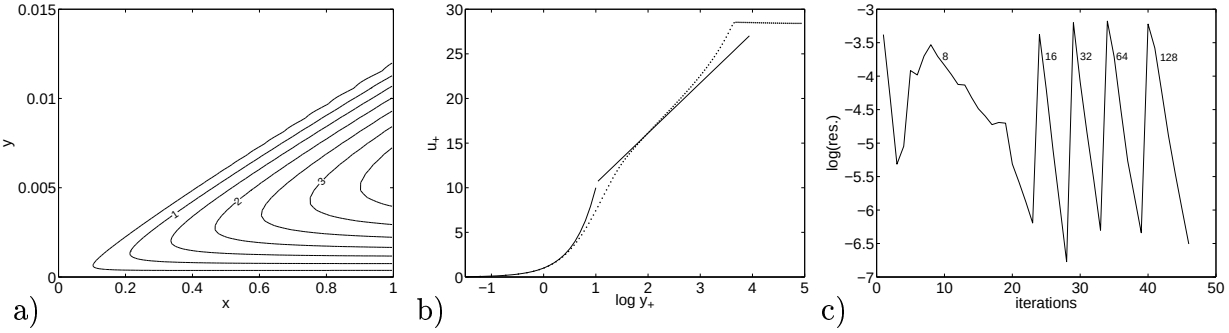

Figure 1: Turbulent flat plate flow at $Re = 10^7$. Turbulent viscosity $\tilde{\nu}_T \cdot 10^5$ (a), velocity profile at $x = 1$ in wall coordinates ($\cdots$) compared with analytical solution (—) (b) and FMG convergence (c).

Results are given in figure 1. The first figure gives the turbulence intensity $\tilde{\nu}_T$. We see that this intensity is (almost) zero in the far field, then increases in the boundary layer and returns to zero at the wall. The second figure gives the velocity profile in wall coordinates; good agreement is found with the theoretical profiles in the viscous sublayer and the logarithmic layer.

Finally, figure 1c gives the convergence of the residual during the multigrid computation. The finest grid is very fine for a boundary layer (128 cells in vertical direction), so the flow can be resolved on most coarse grids too. Therefore, we start the FMG computation on the second (8×8) grid. Here, the development of the boundary layer, when the residual rises and falls, is clearly seen. On all subsequent grids, the convergence is excellent. The convergence rate does not deteriorate much on the finer grids, which means that the Galerkin operators cause few problems.

5.2 Channel flow with bottom bump

As a second test, the flow in a channel with a bottom bump is computed. Experimental results for this test are given by Cahouet [1]. The flow has a Froude number $Fr = 2.05$ and a Reynolds number $Re = 1.9 \cdot 10^5$ (based on inflow water height). The top wall is modeled as a slip wall, the bottom is a no-slip wall. The bump has a thickness of 44% of the water height. The curvilinear grid has 128×512 cells and is compressed near the boundary layer and the water surface.

In the velocity plot (figure 2a), we see low-velocity regions near the leading and trailing edge of the bump and high-velocity regions above the bump and in the air region near the top wall. The turbulent viscosity (figure 2b) shows the boundary layer. The volume fraction is given in figure 2c, here it can be seen that the water surface is smeared out a little.

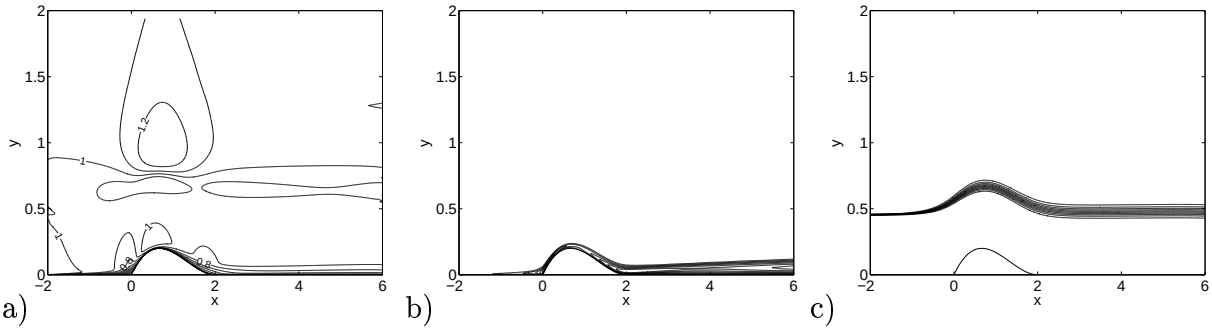

Figure 2: Cahouet test case, $Fr = 2.05$. Speed (a), turbulent viscosity $\tilde{\nu}_T$ (b), and volume fraction α (c). The y-coordinate is stretched for clarity.

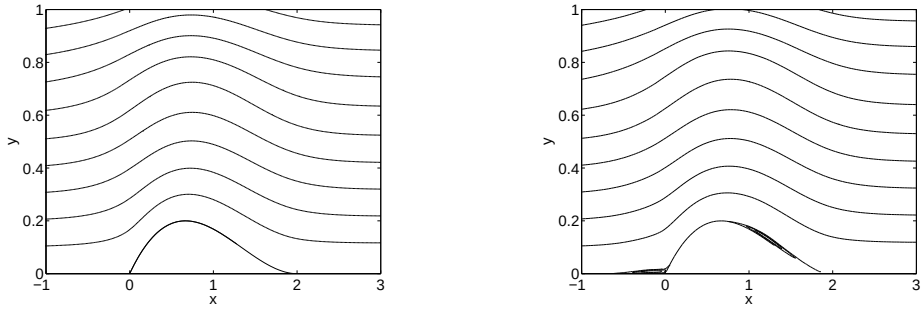

Figure 3: Streamline plot, close-up of the bottom bump. With turbulence (left) and without turbulence(right).

Effect of turbulence In the Introduction, it was stated that turbulence modeling is necessary for the simulation of water flow. This is confirmed by figure 3. The Cahouet bottom bump was designed such, that no separation occurs over its surface and indeed, the streamlines for the turbulent flow show no separation. But when the turbulence model is switched off, the resulting laminar flow has separation bubbles, both in front of the leading edge and over the back of the bump. Thus, the turbulence model is needed to get a physically correct flow.

Multigrid convergence The boundary layer grid for the bottom bump test case is not as fine as for the flat plate. Therefore, no solutions on really coarse grids are possible. The solution is computed on four grids, a convergence result is given in figure 4. We see bad convergence on the first two grids, apparently the boundary layer on the second grid is so different from the first grid that boundary layer development is needed on the second grid too. The convergence on the last two grids is acceptable. Compared with a solution on a single grid, the number of (expensive) fine-grid iterations is reduced by a factor 12. The total computation time is reduced by a factor 8. The current method does not converge so fast in difficult locations, like stagnation points, but we believe this can be solved by

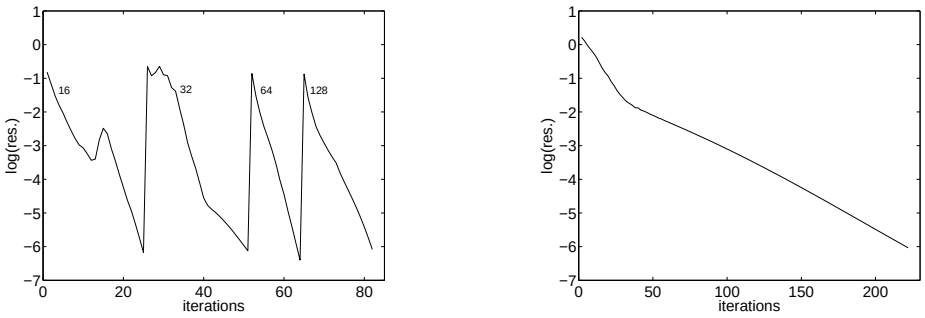

Figure 4: Cahouet test case, convergence of multigrid (FMG on four grids, left) and line Gauss-Seidel smoothing on a single grid (right).

simple fixes: therefore, the convergence can likely be improved even further.

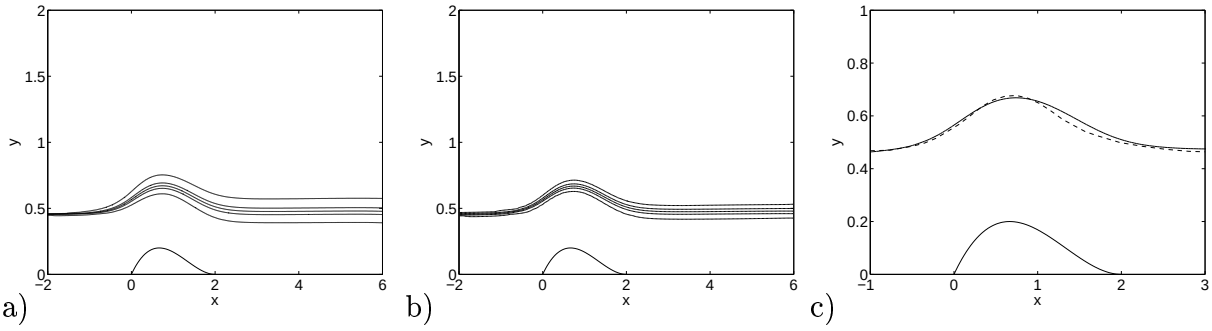

Figure 5: Volume fraction α, five isolines equally spaced between 0.01 and 0.99. First-order accurate (a) and second-order accurate with defect correction (b). Figure (c) gives a comparison of the second-order $\alpha = 0.5$ isolines (—) with Cahouet's experiment (averaged wave height, - -).

Second-order accuracy In figure 5, the volume fraction solution is compared for the first-order (a) and second-order (b) discretisation, after five defect correction steps. The thickness of the water surface is reduced by a factor 2. It is expected that different choices of limiters can reduce this thickness even more.

Figure 5c gives a comparison of the second-order solution with Cahouet's measurements. Overall, the agreement is very good. The slight difference at the back of the wave is probably due to the fact that the boundary layer in Cahouet's experiment is more developed when it hits the bump, as Cahouet's tunnel has a longer inflow length than our domain.

6 CONCLUSION

We presented an accurate two-fluid finite-volume method, based on conservation laws only. The distinction between the fluids is made with a conservative volume-of-fluid equa-

tion. Second-order accurate fluxes are obtained by flux limiting and Menter's turbulence model is used.

The flow equations are solved with multigrid. A line Gauss-Seidel relaxation is used, to get good smoothing for convection-dominated flows with gravity waves. Linear multigrid is used, to circumvent problems that appear when the solutions on coarser grids do not match the solution on the finest grid. Second-order accuracy is obtained with defect correction.

Two test cases confirm that the multigrid and defect correction method work. Good agreement is obtained with analytical solutions and with experiments. A study of special limiters is planned, to further improve the sharpness of the water surface.

REFERENCES

[1] J. Cahouet, *Etude numérique et expérimentale du problème bidimensionnel de la résistance de vagues non-linéaire,* Technical report 185, Ecole Nationale Supérieure de Techniques Avancées, Paris, 1984.

[2] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comp. Phys.* 2 (1976) 12–26.

[3] E. Dick and J. Linden, A multigrid method for steady incompressible Navier-Stokes equations based on flux-difference splitting, *Int. J. Num. Meth. Fluids* 14 (1992) 1311–1323.

[4] C. Frohn-Schauf, *Flux-Splitting-Methoden und Mehrgitterverfahren für hyperbolische Systeme mit Beispielen aus der Strömungsmechanik*, Berichte der Gesellschaft für Mathematik und Datenverarbeitung nr. 211, R. Oldenbourg Verlag, 1993.

[5] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, 1985.

[6] P.W. Hemker and J.-A. Désidéri, Convergence behaviour of defect correction for hyperbolic equations, *J. Comp. Appl. Math.* 45 (1993) 357–365.

[7] T. Hino (ed.), *CFD Workshop Tokyo 2005*, 2005.

[8] B. Koren, A robust upwind discretization method for advection, diffusion and source terms, In C.B. Vreugdenhil and B. Koren (eds.), *Numerical Methods for Advection-Diffusion Problems*, Vieweg, 1993, pp. 117–138.

[9] L. Larsson, F. Stern and V. Bertram (eds.), *Gothenburg 2000, A Workshop on Numerical Ship Hydrodynamics*, Chalmers Univ. of Technology, Göteborg, 2000.

[10] F. R. Menter, Eddy viscosity transport equations and their relation to the $k - \varepsilon$ model, *J. Fluids Engineering* 119, pp. 876–884, 1997.

[11] U. Trottenberg, C.W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, 2001.

[12] E. Turkel, Preconditioned methods for solving the incompressible and low speed compressible equations, *J. Comp. Phys.* 72 (1987) 277–298.

[13] J. Wackers and B. Koren, A surface capturing method for the efficient computation of steady water waves, In P. Bergan, J. García, E. Oñate and T. Kvamsdal (eds.), *Proceedings of the International Conference on Computational Methods in Marine Engineering*, Oslo, pp. 419–428. 2005.