

Four free surface algorithms for the 3D Navier–Stokes equations

Nils Reidar B. Olsen

ABSTRACT

Four algorithms are described for computing a steady free water surface with the solution of the three-dimensional (3D) Navier–Stokes equations. The numerical methods are used in hydraulic engineering cases, typically spillways and river modelling. The algorithms were tested against a laboratory experiment of a v-shaped broad-crested weir. The complex geometry of the weir introduced three-dimensional effects, which the numerical methods handled with varying degrees of success. One of the methods tested was the classical volume of fluid (VOF) approach, implemented in the OpenFOAM software with a fixed grid. The other three algorithms used an adaptive grid that followed the free water surface. These methods were coded in the SSIIM 2 program and were based on water continuity, pressure differences and an implicit solution of the diffusive wave equation. The VOF method gave the best results compared with the experiments. However, this method requires a very short time step. Two of the investigated methods compute the water surface location implicitly and can therefore use a much longer time step. The method based on the diffusive wave equation has the disadvantage that the results depend on a calibrated friction factor. All four methods predicted the water depth over the weir with an average accuracy below 14%.

Key words | computational fluid dynamics (CFD), implicit free surface algorithms, numerical modelling, OpenFOAM, SSIIM, weirs

Nils Reidar B. Olsen
Department of Environmental and Hydraulic
Engineering,
The Norwegian University of Science and
Technology,
S. P. Andersens vei 5,
7491 Trondheim,
Norway
E-mail: Nils.R.Olsen@ntnu.no

INTRODUCTION

Three-dimensional (3D) numerical models solving the Navier–Stokes equations are today increasingly applied in water engineering projects. A particular problem for rivers and channels is determining the location of the free surface (Aguilar *et al.* 2013). The Navier–Stokes equations require special algorithms for computing how the surface changes in space and time (Erdbrink *et al.* 2014). The most classical hydraulic problem is to predict the coefficient of discharge for a spillway (Savage & Johnson 2001; Jacobsen & Olsen 2010; Li *et al.* 2010). This is also one of the most common investigations for commercial laboratory model studies. Another hydraulic engineering problem related to computation of the free water surface is reservoir flushing. Then the water level is drawn down

to increase the water velocities and thereby erode the sediments in the reservoir (Haun & Olsen 2012). This problem requires an accurate and stable prediction method for the free surface. Predicting too low water levels will lead to a higher velocity and shear stress, giving too large erosion. Too high water levels reduce the computed erosion in the reservoir.

Laboratory studies are often used to test the accuracy of numerical models (Wilson *et al.* 2003). The test case can then be made fairly simple, ensuring that errors in the geometry modelling will not affect the results. A good test case for free surface algorithms is a weir, where the geometry is relatively simple compared to a natural river. Laboratory measurements of the water surface over the

weir have been made in the current study, enabling a base for testing the different free surface algorithms.

The most common method for computing the free water surface is the volume of fluid (VOF) approach used in a fixed grid. This method is included in all commercial general purpose 3D computational fluid dynamics (CFD) programs (Temeepattanapongsa *et al.* 2013; Castillo *et al.* 2014; Liu & Yang 2014), and also in the popular OpenFOAM open source program. The VOF method has the advantage that very complex water surfaces can be computed. Multiple surfaces can be computed, for example, flow over a sharp-crested weir (Lv *et al.* 2011; Abd El-Hady Rady 2011). A disadvantage with the VOF method is that a fairly dense grid has to be used near the water surface to get an accurate prediction of its location (Rüther *et al.* 2010; Feurich & Olsen 2012). Also, a relatively short time step has to be used, respecting the Courant criteria. The required size of the time step may then be only fractions of a second. Longer time steps produce instabilities. In sedimentation engineering, it is often desirable to predict sediment transport in a reservoir over a very long time, for example, 50–100 years. The computational time using the VOF method would then be too long.

In most water engineering cases where a river is modelled, the water surface is fairly horizontal (Tritthart & Gutknecht 2007). If this assumption is made, it is possible to derive alternative numerical algorithms for the determination of the free water surface. An advantage with this simplification is the possibility to employ an adaptive grid that follows the water surface (Olsen & Kjellesvig 1998). A coarser grid can then give the same accuracy as a finer fixed grid. Also, alternative equations can be solved, giving different accuracies and stabilities for the computation of the free surface. The most interesting approach for simulations over a long time would be implicit methods, where long time steps can be used. Two implicit and two explicit algorithms for computation of the free surface are described and tested in the current paper.

TEST CASE: V-SHAPED BROAD-CRESTED WEIR

An earlier study (Haun *et al.* 2011) investigated different algorithms for computing the water surface over a broad-

crested weir for a 2D situation. The current paper also uses a broad-crested weir, but with a more complex 3D geometry. This enabled 3D effects to be investigated. The weir is V-shaped looking both in plan view and in a cross-section. The geometry is given in Figure 1, showing a plan view (Figure 1(a)), longitudinal profile (Figure 1(b)) and a cross-section (Figure 1(c)). A physical model of this weir was built at the laboratory of the Department of Hydraulic and Environmental Engineering at the Norwegian University of Science and Technology. The weir was built of plexiglass, and was 60 cm wide, 90 cm long and 20 cm high. A photograph of the weir is shown in Figure 2. The model was

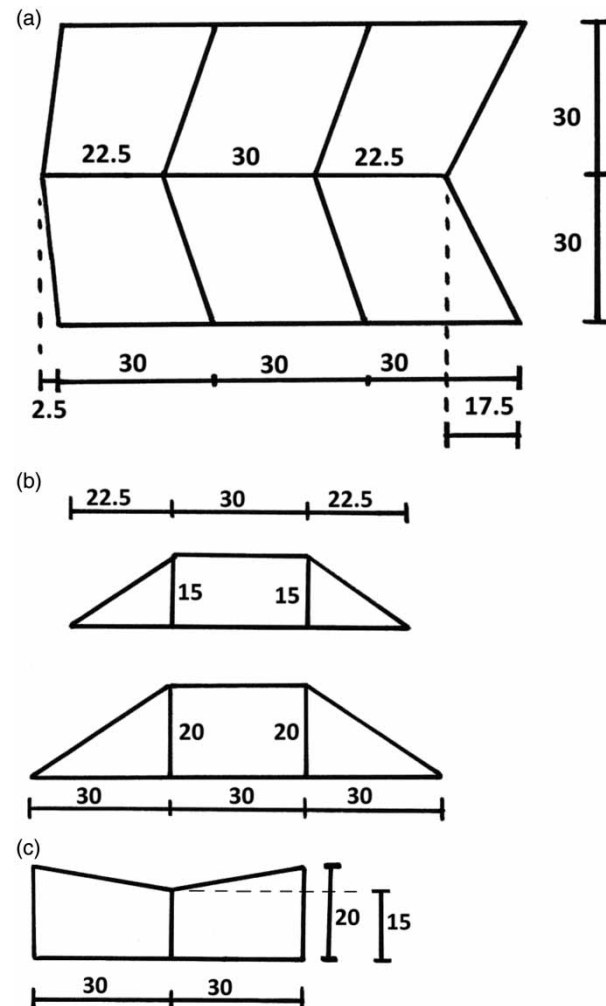


Figure 1 | Sketch of the weir, seen from above (a), in two longitudinal profiles (b) and in a cross-section (c). The longitudinal profile (b) is given in the centreline (upper figure) and at the sides (lower figure) of the flume/weir. The flow direction is from left to right in (a) and (b). The numbers are distances in cm.



Figure 2 | Photograph of the plexiglass weir.

installed in a flume which was 60 cm wide, 1 m high and 15 m long. A pump lifted water into the flume at a rate of 21.6 L/s. The downstream water level was kept at 15 cm above the flume bed. The physical model with water is shown in [Figure 3\(a\)](#) from the side and [Figure 3\(b\)](#) from above. The water flowed over the weir where critical flow occurred. The Froude number was, however, so low that a large hydraulic jump did not form. Since the geometry of the weir was V-shaped also in plan view, the border between the subcritical and supercritical flow followed a curve in the horizontal plane, as seen in [Figure 3\(a\)](#) and [3\(b\)](#). The water

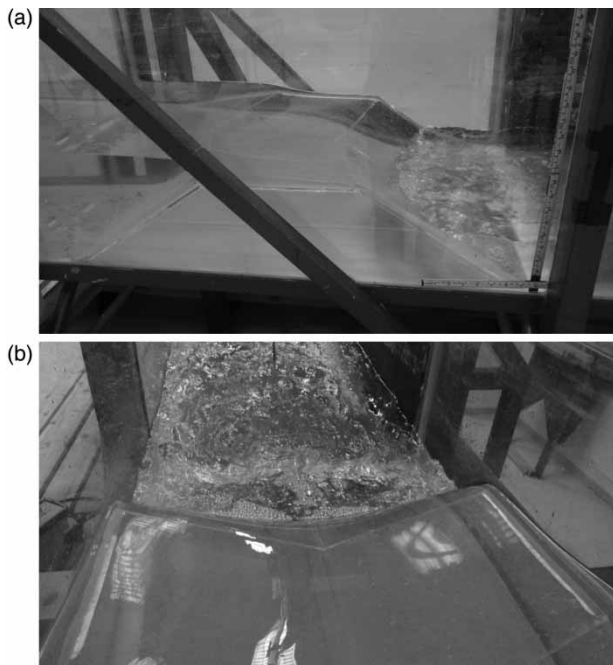


Figure 3 | Photograph of the water flowing over the weir in the flume, seen from the side (a) and in the streamwise direction (b).

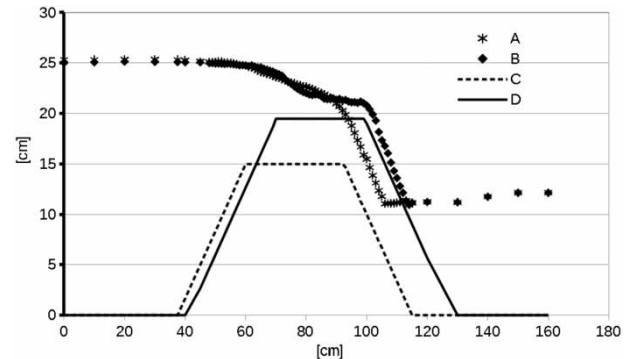


Figure 4 | Longitudinal profiles of measured water surface elevations over the weir at the centreline (A) and 3 cm from the side wall (B). The bed elevations of the top of the weir are also shown at the centreline (C) and 3 cm from the side wall (D).

level upstream of the weir was measured to be 25.4 cm above the flume bed.

The water level was measured from the top of the laboratory flume using a point gauge along two longitudinal profiles. One profile followed the centreline of the flume and another 3 cm from the side of the flume. The results from the laboratory measurements are shown in [Figure 4](#). The two profiles are not identical, since the bed level was 5 cm deeper at the centreline than at the walls. The wall sections were also longer than the central section of the weir. The side of the weir reached further downstream than the central section of the weir. This situation is seen in the difference between the water levels on the sides and in the centreline in [Figure 4](#). There were also other three-dimensional effects shown in the measurements, as lateral variation of the water surface at the top of the weir. Over the most upstream part of the weir, the water level was higher at the centre of the flume than at the sides. This reverted further downstream, before the longitudinal water level gradient became steep and ended in the small hydraulic jump.

NAVIER–STOKES EQUATIONS AND THE GRID

Both computer programs used in the current study solved the Navier–Stokes equations in three dimensions using the k-epsilon turbulence model (Lauder & Sharma 1974). A control volume method was used for the discretization of the equations. The pressure was computed based on the

SIMPLE method (Patankar 1980) and wall laws were used to resolve the steep gradients close to the walls/bed (Schlichting 1979). The boundary conditions were a specified velocity and turbulence value at the inlet and zero gradients for all variables at the outlet. These numerical algorithms have been verified by a number of earlier studies where the results were compared with field or laboratory data (Fischer-Antze *et al.* 2008; Baranya *et al.* 2012; Đorđević 2013).

The three-dimensional grid was based on a 2D depth-averaged grid of the spillway and the flume bed, as given in Figure 5(a). The grid had 150 cells in the longitudinal direction and 20 cells in the lateral direction. An initial horizontal water surface was specified above the bed and the 3D grid was made between these surfaces using vertical grid lines. This structured, non-orthogonal grid was made with the SSIIM 2 program and used for all the computations. The two-phase VOF method used this grid for the whole computation, and the other methods used this grid as a starting point before grid changes. The adaptive grid methods changed the grid after each time step, as the water level moved. This ensured that only the water fluid was modelled. The number of cells in the vertical direction changed during the computation, as seen in

Figure 5(b). The adaptive grid method is described in detail by Olsen (2003).

VOF METHOD

VOF is the most commonly used method to predict the free surface with the solution of the 3D Navier–Stokes equations (Soleiman Beygi *et al.* 2014). A fixed grid with a horizontal upper boundary was used for the VOF computations in the current study. The OpenFOAM program was used to compute the two-phase flow of air and water. The most commonly used method to create grids with OpenFOAM is to place a structured, orthogonal grid around the geometry, which is defined by STL files. Then the OpenFOAM program snappyHexMesh is used to cut the orthogonal grid to its final shape. Another approach was used in the current study. A non-orthogonal grid was made (Figure 5(c)) that followed the bed of the flume and top of the weir. The grid was made by the SSIIM 2 program and exported to OpenFOAM in a blockMeshDict file. The interFoam solver was used to compute the two-phase flow of air and water. The top of the grid was then set to a level 29 cm above the bed level of the flume. The inflow

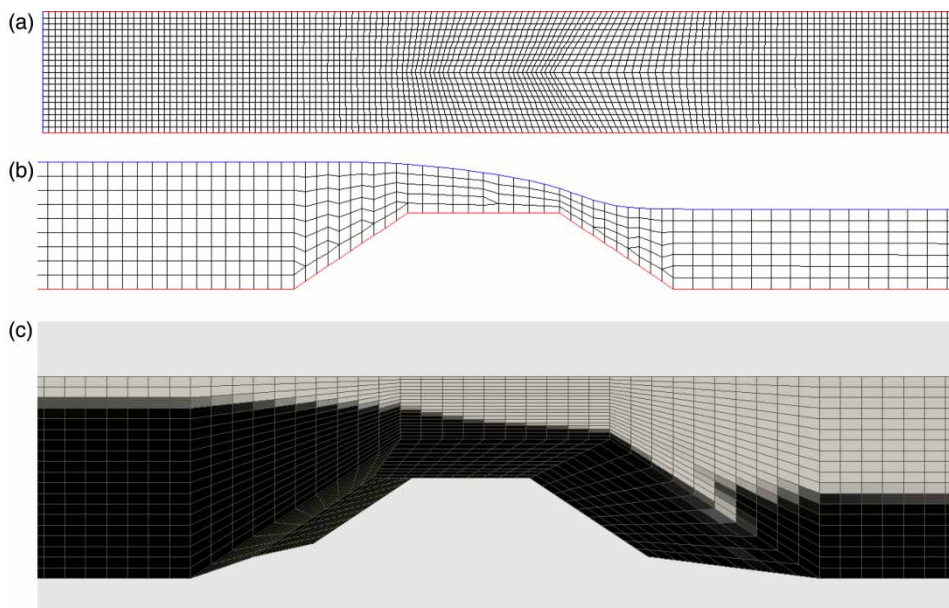


Figure 5 | (a) Grid seen from above. The flow direction is from left to right. There are 20 cells in the lateral direction and 130 cells in the longitudinal direction. The computational domain is 60 cm wide and 4.5 m long. (b) Longitudinal profile of the adaptive grid, through the centreline of the flume. (c) Three-dimensional grid seen from the side, used for the VOF computations. The dark areas are water and the white areas are air. Grey areas are partly filled with water and partly with air.

region was set to cover the whole height of 29 cm. The outflow region initially only covered the lower 15 cm, corresponding to the measured downstream water level.

The VOF method computed both water and air in a two-phase model on a fixed grid. A transient convection equation was solved for the volume fraction, α , of water in the grid:

$$\frac{\partial \alpha}{\partial t} + U_j \frac{\partial \alpha}{\partial x_j} = 0 \quad (j = 1, 2, 3) \quad (1)$$

The time is denoted t and x is a spatial distance. U_j is the fluid velocity component in direction j . The initial conditions for the volume fraction was $\alpha = 1$ behind the weir and $\alpha = 0$ downstream of the weir.

The VOF method requires a fairly short time step to give accurate and stable results. A time step of 0.001 s was used in the current study.

The OpenFOAM program produced a file called *alpha1* for a number of time steps during the computation. The file contained the α values in each grid cell. The file was read back into SSIIM 2, where an interpolation algorithm changed the surface to correspond with α values equal to 0.5. The SSIIM 2 grid was then regenerated accordingly, and the resulting water surface written to a file. The SSIIM 2 program was thereby used as a post-processor for OpenFOAM. The resulting water elevations are shown in Figure 6, compared with the measured values.

A problem with the OpenFOAM computations was the downstream boundary condition. The downstream water level was kept at 15 cm in the physical model, and therefore the outlet area was initially set to be 15 cm above the bed in the numerical model. However, false diffusion caused the water level downstream of the weir to become higher than 15 cm. To solve the problem, the water level for the VOF computations was only allowed to flow out of the lower 10 cm of the downstream end of the grid. This caused a correct water level to form downstream of the weir during the first part of the simulations, as seen in Figure 6. However, when the simulations were done over a very long time, the downstream water level would rise. The water level over the weir and upstream of the weir was not affected by this problem.

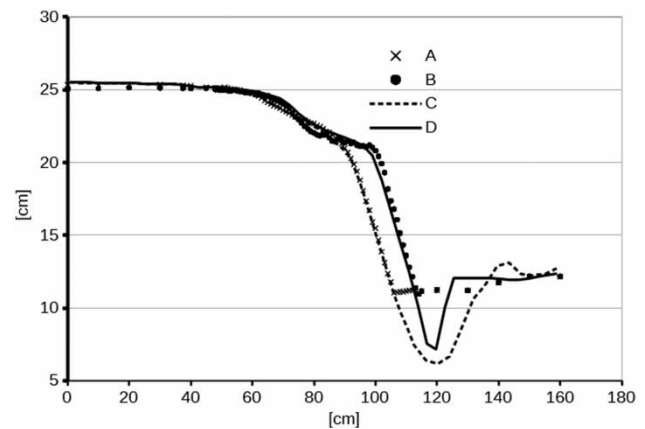


Figure 6 | Longitudinal profiles of measured water surface elevations at the centreline of the flume (A) and 3 cm from the flume side (B), together with computed water surface elevations at the centreline of the flume (C) and 3 cm from the flume side (D). The computations are done with the VOF method.

NUMERICAL METHOD BASED ON THE CONTINUITY EQUATION (CGA)

The adaptive grid methods can be divided into two groups: one which includes gravity in the Navier–Stokes equations and one which omits this term directly. The latter methods take gravity into account when computing changes in the water surface.

The method using the continuity equation with gravity in the Navier–Stokes equations and an adaptive grid is in the following given the abbreviation CGA (continuity, gravity and adaptive grid). The algorithm was first developed by Olsen & Kjellesvig (1998), who used it successfully to compute the water flow over a spillway. A deviation of 0.5% was observed between the predicted and measured coefficient of discharge. Olsen (2013) also used the method to compute the movement of soil in a sand slide due to a water reservoir drawdown.

The main idea of the algorithm is based on an ad hoc method to compute the pressure in the cells close to the free surface instead of using the SIMPLE method. The method interpolates the pressure in the cell closest to the surface from the cell below and the zero water pressure at the free surface. The SIMPLE method is based on water continuity being fulfilled, so when it is not used, the water continuity is not fulfilled in the surface cells. The water continuity defect, ΔQ , can be computed from the water fluxes

flowing in and out of the cell. The continuity defect is then used to move the water surface vertically a distance Δz according to the following formula:

$$\Delta z = \frac{\Delta Q \Delta t}{A_z} \quad (2)$$

The time step is denoted Δt and A_z is the area of the cell projected in the horizontal plane. Since the formula is proportional to the time step, the method will require a short time step to be stable. A time step of 0.001 s was used in the current study, similar to what was used in the VOF method. The initial downstream water level was given as 0.3 m above the bed, and drawn down to 0.15 m after 2 s.

The results are given in Figure 7. The computed water surface profiles follow the measurements well over the weir. Upstream of the weir, the water level is slightly lower than the measurements. The water level downstream of the weir shows a dip at the hydraulic jump, a feature that was not observed in the physical model. Comparing the VOF and the CGA methods in Figures 6 and 7, the results are fairly similar, although the CGA approach gives slightly lower water levels upstream of the weir. The computational time was similar for the two methods.

The CGA method produced small instabilities at the surface, similar to small waves. A smoothing formula was used

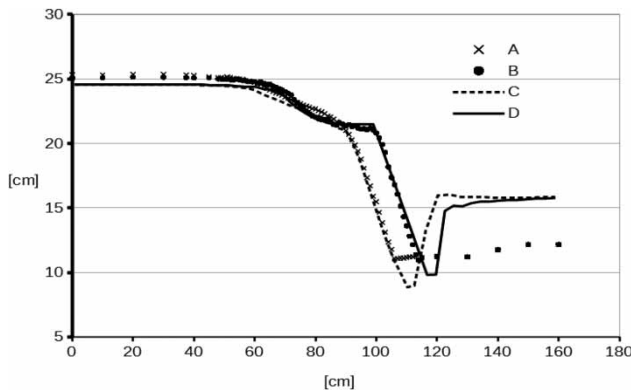


Figure 7 | Longitudinal profiles of measured water surface elevations at the centreline of the flume (A) and 3 cm from the flume side (B), together with computed water surface elevations at the centreline of the flume (C) and 3 cm from the flume side (D). The free surface computations are done with the CGA method.

to remove the small oscillations:

$$z_{i,j} = (1.0 - r) * z_{i,j} + r(z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1})/4 \quad (3)$$

The indexes (i, j) denote the grid intersection between cross-section line i and longitudinal profile j . The parameter r is a relaxation coefficient, set to 0.01. Equation (3) was applied after each grid update.

IMPLICIT NUMERICAL METHOD BASED ON PRESSURE DIFFERENCES (IPDA)

Methods allowing for longer time steps are based on other approaches. The VOF and CGA methods use the Navier–Stokes equations to find the free surface through the computed velocity field. They are thereby based on a momentum balance. Instead, it is possible to use the computed pressure to find the water surface elevation, z_p , in a surface cell with index p as a function of the water surface elevation in the neighbour cell, indexed i :

$$z_p = z_i + \frac{(P_p - P_i)}{\rho g} \quad (4)$$

The water density is denoted ρ , and g is the acceleration of gravity. The pressure, P , in each cell is computed from the SIMPLE method, when solving the Navier–Stokes equations. Equation (4) then gives the elevation difference between two cells as a function of the pressure difference. The equation can be used for each cell and solved implicitly. The method is abbreviated IPDA (implicit pressure differences with adaptive grid).

Each 2D depth-averaged cell has eight close neighbours, so a question is which cell to choose as the i cell. The method proposed by Haun & Olsen (2012) uses a weighted average of the eight neighbour cells:

$$a_p z_p = \sum_{i=1}^8 a_i (z_i + \frac{(p_p - p_i)}{\rho g}) \quad \text{where} \quad a_p = \sum_{i=1}^8 a_i \quad (5)$$

A formula for the weighting coefficients, a_i , is then required. In supercritical flow, the upstream cells should be given more weight than the downstream cells, and vice

versa for subcritical flow. In the current study, the following formula was used:

$$a_{i,0} = \frac{1 - \tanh(-20w)}{2} \quad (6a)$$

$$a_i = \left\{ \begin{array}{ll} a_{i,0} & w > 0 \\ \max[a_{i,0}, 10(Fr - 1)] & w < 0 \quad 1.1 > Fr > 1 \\ \max[a_{i,0}, 1.1/(1 - Fr)] & w < 0 \quad Fr < 1 \end{array} \right\} \quad (6b)$$

The Froude number is denoted Fr , and w is the normalized dot product of the velocity vector and the direction vector, \vec{s} , from cell p to cell i :

$$w = \frac{\vec{U} \cdot \vec{s}}{|\vec{U}| |\vec{s}|} \quad (7)$$

In addition, the coefficient, a_i , in Equation (6a) was set to zero if cell p was upstream of a hydraulic jump, and cell i was downstream of the jump. Equation (6b) was made on the assumption that the downstream coefficient should be close to zero for supercritical flow. For subcritical flow, the downstream coefficient should be higher than the upstream coefficient to improve convergence. The tangent hyperbolic function in Equation (6a) was made to ensure a smooth variation in the a_i coefficient for w values close to zero. Except for this, the derivation of Equation (6) was based on trial and error. It performed better than a similar equation used by Haun & Olsen (2012).

Equations (5)–(7) were solved implicitly using a Gauss–Seidel method. The boundary conditions were zero gradients at all sides, except for the downstream boundary. The initial downstream water level was equal to 0.3 m, the constant initial value for the whole grid. The downstream water level was drawn down to 0.15 m after the first time step and was kept fixed there during the computations.

Equations (5)–(7) do not contain the time step, so the method is well suited for quasi-steady simulations with long time steps. The current study used a time step of 100 s. However, there were still stability problems with the method. The solution has to start with an initially guessed water level. This is usually not the correct one, and the pressure field associated with the initial grid is then not the same as the pressure field for the correct solution. Using Equation (5) directly will therefore easily lead to instabilities, especially for high Froude

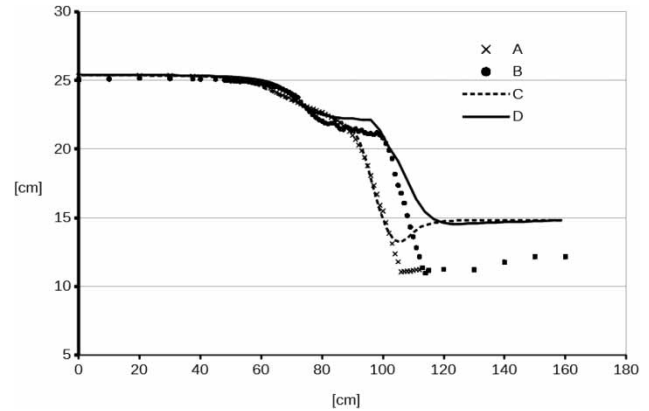


Figure 8 | Longitudinal profiles of measured water surface elevations at the centreline of the flume (A) and 3 cm from the flume side (B), together with computed water surface elevations at the centreline of the flume (C) and 3 cm from the flume side (D). The free surface computations are done with the IPDA method.

numbers. A relaxation procedure was therefore used, where the movement of the water surface, Δz , between each grid update was computed from the following equation:

$$\Delta z = r(z_k - z_{k-1}) \quad (8)$$

The index k is here used to denote the iteration number. The current iteration is k and $k-1$ is the previous iteration. The relaxation factor, r was set to 0.01 in the current formula. This did not eliminate the instabilities completely. A smoothing procedure was therefore used, with Equation (3). A smoothing factor of 0.03 gave good results as shown in Figure 8. Smaller smoothing factors gave instabilities and larger smoothing factors gave lower water level gradients.

Figure 8 shows good results for the water level behind the weir, and also reasonable results over the weir. Downstream of the weir, there is a dip in the water surface elevation where the hydraulic jump is located. The water level downstream of the jump is also too high. The reason is probably that the flow causing the energy loss over the jump is complex and therefore not computed correctly.

IMPLICIT NUMERICAL METHOD BASED ON THE DIFFUSIVE WAVE EQUATIONS (IDWA)

A large number of numerical methods and publications have been made on the solution of the shallow water equations (Ying & Wang 2010; Cozzolino *et al.* 2014). Almost all

methods are based on an explicit solution of the equations, giving stability problems for larger time steps. However, an approach pioneered by Casulli (1990) solved the equations partly implicitly, allowing longer time steps. The method was further developed by Kramer & Stelling (2008). The main idea is to make an equation for the water level, z , in a cell p as a weighted average of the water levels in the four neighbouring 2D depth-averaged cells, i :

$$a_p z_p = \sum_{i=1}^4 a_i z_i + S_p \quad \text{where} \quad a_p = \sum_{i=1}^4 a_i \quad (9)$$

The method will be stable for long time steps if the weighting coefficients, a_i , are positive and the source term, S_p , in the equation is not proportional to the time step.

The shallow water equations and the water continuity equation have in principle three unknowns: the velocities in two horizontal directions and the water depth. The equations are usually solved in a coupled manner. The approach used in the current study decouples the equations, so that only one equation is solved for the water level. The other variables, the velocities, are taken from the solution of the Navier–Stokes equations.

The derivation of the equation follows the method of Kramer & Stelling (2008), by looking at the 1D momentum equation between two cells, w and p :

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} - g \frac{\partial z}{\partial x} - c_f \frac{U|U|}{h} = 0 \quad (10)$$

The velocity, U , is here assumed normal to the surface between the two cells. The water depth is denoted h , and z is the vertical location of the water level. The parameter c_f is a friction coefficient and g is the acceleration of gravity. The equation has four terms. From the left, a transient acceleration term, then a convective term, then a term related to the surface slope and then a friction loss term. The current approach uses a finite difference method for the discretization. The equation is solved with respect to the velocity and applied to a location between two depth-averaged cells, p and i :

$$\frac{U^{t+1} - U^t}{\Delta t} + U^{t+1} \frac{(U_p^t - U_i^t)}{\Delta x} - g \frac{(z_p^{t+1} - z_i^{t+1})}{\Delta x} - c_f \frac{U^{t+1} |U^t|}{h^t} = 0 \quad (11)$$

The velocities without index are located between the cells p and i . The velocities at time t are taken from Navier–Stokes equations at the previous time step, and are known when the equation is solved. The distance between the centre of the two cells is given as Δx .

The current algorithm is designed for a quasi-steady state situation with long time steps. Then the transient acceleration term in Equation (10) is small and can be neglected. Taking it out of the equation and solving with respect to the velocity at the next time step, U^{t+1} gives:

$$U^{t+1} = \frac{g((z_p^{t+1} - z_i^{t+1})/(\Delta x))}{((c_f |U^t|)/(h^t)) + ((U_p - U_i)/(\Delta x))} \quad (12)$$

The water flux between two cells, w and i , is then given by this velocity multiplied with the area of the border between the two cells. With cell p having four neighbour cells in a 2D depth-averaged grid, there are four fluxes around cell p . The continuity equation can then be written as:

$$\sum_{i=1}^4 U_i A_i = \frac{A_z \Delta z}{\Delta t} \quad (13)$$

This gives the following equation for the weighting coefficient in Equation (9):

$$a_i = \frac{A_i}{A_z} \frac{g}{\Delta x} \left(\frac{h_i}{c_f |U_i^t|} \right) \quad (14)$$

The convective term is here omitted, so that the coefficient always will be positive. This is done for stability reasons. Removing both the convective and the transient acceleration terms will reduce the shallow water equation to a diffusive wave equation. The current approach is therefore abbreviated IDWA (implicit diffusive wave with adaptive grid).

In the current study, the Manning equation is used to compute the friction loss instead of a drag formula. Equation (14) can then be transformed to:

$$a_w = \frac{A_i}{A_z} \frac{1}{\Delta x} \frac{h_i^{4/3}}{n^2 |U_i^t|} \quad (15)$$

Equation (15) includes the effects of the water slope and the bed friction. Manning's friction coefficient is denoted n .

The velocity in the equation is taken from the 3D Navier–Stokes equations, and is known when the coefficient is computed.

The source term in Equation (9) includes the right-hand side of Equation (13) as a boundary condition for the upstream and downstream cells:

$$S_p = \frac{Q}{A_z} \quad (16)$$

The downstream boundary condition is most often a specified value of the water level. In the current model, this is implemented by giving the following boundary condition to the downstream cells:

$$\begin{aligned} S_p &= z_{\text{Specified}} 10^{50} \\ a_p &= 10^{50} \end{aligned} \quad (17)$$

The method was also used as boundary condition for the IPDA method.

Equation (9) was solved with a standard Gauss–Seidel solver. A time step of 100 s was used, with 100 inner iterations per time step for the implicit solution of the Navier–Stokes equations. Similar to the IPDA computations, the initial downstream water level was 0.3 m, and drawn down to 0.15 m after the first time step, where it remained fixed.

The result is shown in Figure 9. The water level upstream of the weir is computed fairly well and so is the water level in the steepest area upstream of the hydraulic jump. The maximum overprediction of the water level is 120%, at the acceleration zone on the side at the top of the weir. There are steep gradients in the water velocity at these locations, so the omission of the convective term could contribute to the discrepancy between measured and computed water levels. The largest velocity gradients are however at the hydraulic jump downstream of the weir. Downstream of the hydraulic jump, the water depth is overpredicted by more than 35%. The current grid is too coarse to resolve the flow field in this area properly, but also other factors complicate the computation of the hydraulic jump. The shallow water equation is based on a hydrostatic pressure assumption, which is not present in

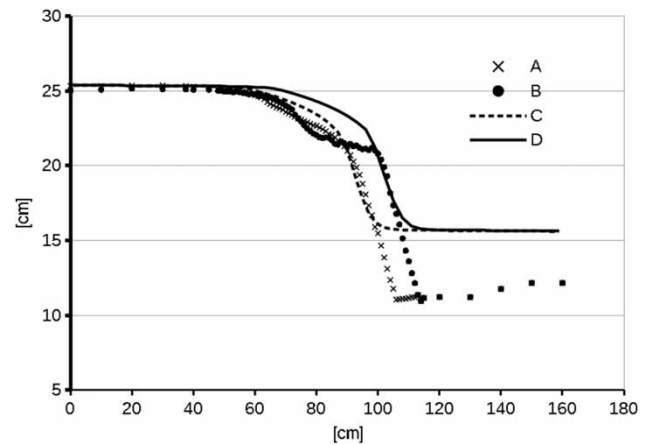


Figure 9 | Longitudinal profiles of measured water surface elevations at the centreline of the flume (A) and 3 cm from the flume side (B), together with computed water surface elevations at the centreline of the flume (C) and 3 cm from the flume side (D). The free surface computations are done with the IDWA method.

the hydraulic jump. There are also significant vertical velocity components upstream and in the hydraulic jump, pushing the water level down. A hydraulic jump may entrain a significant amount of air, complicating the physics. All these factors significantly reduce the accuracy of the solution of the shallow water equation for hydraulic jumps.

Note that the choice of the Manning's friction coefficient will affect the slope of the water surface. The energy losses over the weir are due to complex physics and the Manning's friction coefficient must take the losses into account. The value of the coefficient can therefore not be equal to what is usually given for a smooth wall, although the weir and side walls of the flume were made of plexiglass. The result in Figure 9 was obtained with a Manning's friction coefficient of $n=0.1$. Using a value of 0.2 gave a 3.2 cm higher water level behind the weir.

DISCUSSION

The OpenFOAM version used in the computations was made by the company BlueCFD. It was the single core version 2.0 for Windows. The computation was done on a 3.4 GHz PC with the Intel i7-4770 processor. The computational time was about 20 h until a steady situation was reached, as given in Figure 6. Note that most OpenFOAM versions can run on multiple cores, so on a four core PC,

it is likely that the computational time could be reduced considerably.

The SSIIM computations were run on the same computer as the OpenFOAM program, but on two cores. The SSIIM program is parallelized with OpenMP, which gives good scaling for coarse grids. The computational speed was then about 40 time steps/minute, giving about 10 h of computational time until a steady solution emerged using the IDWA method. The IPDA method required much less computational time. Note that the adaptive grid will use roughly half the number of cells compared with the VOF method, reducing the computational time.

Figure 10 shows a time series of the water levels upstream of the weir for the three methods with an adaptive grid. The CGA method has an initial rise in the water levels that could come from an instability when the water hits the weir. As the time step is 0.001 s, the total simulated time in Figure 10 for the CGA method is only 20 s. The implicit methods give a faster convergence than the explicit method, with fewer time steps. The IPDA method converges much faster than the IDWA method. The reason is that the a_i coefficients in Equations (5) and (9) are higher in the downstream direction than the upstream direction for the IPDA method. This means that the information about the lowered downstream water surface travels faster upstream than in the IDWA method, which has the same a_i coefficients in the upstream and downstream directions. However, the IPDA method is more unstable. Note that smoothing and relaxation was required to get the IPDA to

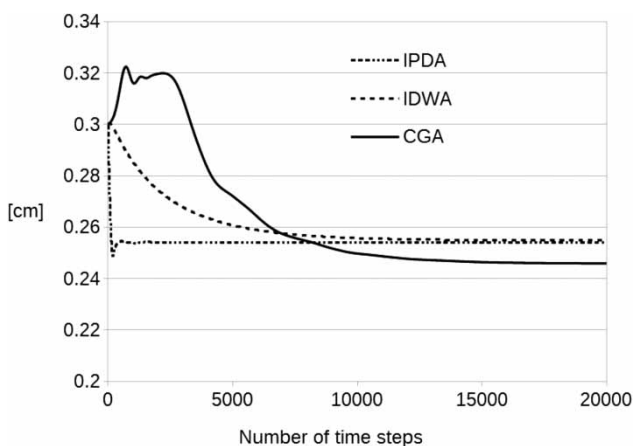


Figure 10 | Time series of computed water surface elevations behind the weir.

be stable, but this was not needed for the IDWA method in the current study.

Also note that since the IPDA and IDWA methods do not contain the time step, they will not give the correct solution for cases where the water level varies fairly rapidly, for example waves.

The accuracy of the four methods was assessed by calculating the difference between the measured water surface, z_m , and the values from the numerical model, z_n :

$$D = \frac{\sum_1^c |z_m - z_n|}{cH} \quad (18)$$

The computations were done over the weir, upstream of the hydraulic jump. The parameter D is dimensionless, scaled with the average water depth, H , at the weir. The number of measured water levels over the weir is denoted c . Table 1 gives the D values for the different algorithms. The VOF method gives the best result, and the IDWA method has the poorest accuracy. This corresponds well with the visual observations from Figures 6–9.

CONCLUSIONS

The VOF method used in, for example OpenFOAM, gives a very good estimate of the water surface over the weir. The main problem with the VOF and CGA method is the need for a relatively short time step. An adaptive grid method can use fewer cells with the same accuracy, but it has other drawbacks. The CGA method did not compute the water levels as accurately as the VOF method.

The implicit methods with adaptive grids (IPDA, IDWA) give reasonable results for the water level over the weir, although not as good as the VOF method. However, the implicit methods can use very long time steps, which is beneficial for simulating processes that take place over a long time. An important disadvantage with the IDWA method

Table 1 | Estimated dimensionless errors for the different algorithms

| Algorithm | VOF | CGA | IPDA | IDWA |
|-----------|-----|-----|------|------|
| D (%) | 4.5 | 4.8 | 13.5 | 5.3 |

is that the resulting water level is a function of a friction coefficient, which requires calibration. All the other methods use a very low roughness for the skin friction and compute the other energy losses directly. Then calibration of a friction coefficient is not needed.

None of the free surface algorithms compute the water slope in the hydraulic jump correctly. This is probably not a serious problem for computation of reservoir flushing and other flows in most natural rivers. Large hydraulic jumps usually only occur downstream of hydraulic constructions. The coefficient of discharge for a weir is only based on the upstream water level, which all methods computed reasonably well.

The study also shows that a CFD program can be used to predict the water surface over a spillway on a relatively coarse grid and still produce high quality results. The computational times for all the tested algorithms were reasonable for engineering purposes on a desktop PC.

ACKNOWLEDGEMENTS

I want to thank Geir Tesaker for the construction work on the physical model of the weir and Heide Reppenhausen for assistance with the laboratory measurements.

REFERENCES

- Abd El-Hady Rady, R. M. 2011 2D-3D Modeling of flow over sharp-crested weirs. *J. Appl. Sci. Res.* **7** (12), 2495–2505. <http://www.flow3d.com/wp-content/uploads/2014/08/2D-3D-Modeling-of-Flow-Over-Sharp-Crested-Weirs.pdf>.
- Aguilar, J. V., Langarita, P., Linares, L., Gómez, M. & Rodellar, J. 2013 An adaptive predictive approach for river level forecasting. *J. Hydroinform.* **13** (2), 232–245.
- Baranya, S., Olsen, N. R. B., Stoesser, T. & Sturm, T. 2012 Three-dimensional RANS modeling of flow around circular piers using nested grids. *Eng. Appl. Comput. Fluid Mech.* **6** (4), 648–662.
- Castillo, L. G., Carrillo, J. M. & Sordo-Ward, A. 2014 Simulation of overflow nappe impingement jets. *J. Hydroinform.* **16** (2), 922–940.
- Casulli, V. 1990 Semi-implicit finite difference methods for the two-dimensional shallow water equations. *J. Comput. Physics* **86** (1), 56–74.
- Cozzolino, L., Cimorelli, L., Covelli, C., Della Morte, R. & Pianese, D. 2014 Boundary conditions in finite volume schemes for the solution of shallow-water equations: the non-submerged broad-crested weir. *J. Hydroinform.* **16** (6), 1235–1249.
- Dordevic, D. 2013 Numerical study of 3D flow at right-angled confluences with and without upstream platform curvature. *J. Hydroinform.* **15** (4), 1073–1088.
- Erdbrink, C. D., Krzhizhanovskaya, V. V. & Sloat, P. M. A. 2014 Free-surface flow simulations for discharge-based operation of hydraulic structure gates. *J. Hydroinform.* **16** (1), 189–206.
- Feurich, R. & Olsen, N. R. B. 2012 Finding free surface of supercritical flows – numerical investigation. *Eng. Appl. Comput. Fluid Mech.* **6** (2), 307–315.
- Fischer-Antze, T., Olsen, N. R. B. & Gutknecht, D. 2008 Three-dimensional CFD modeling of morphological bed changes in the Danube River. *Water Resour. Res.* **44** (9). DOI: 10.1029/2007WR006402.
- Haun, S. & Olsen, N. R. B. 2012 Three-dimensional numerical modelling of reservoir flushing in a prototype scale. *Int. J. River Basin Manage.* **10** (4), 341–349.
- Haun, S., Olsen, N. R. B. & Feurich, R. 2011 Numerical modelling of flow over a trapezoidal broad-crested weir. *Eng. Appl. Comput. Fluid Mech.* **5** (3), 397–405.
- Jacobsen, J. & Olsen, N. R. B. 2010 Numerical modelling of the capacity for a complex spillway. *Water Manage.* **163** (WM6), 283–288.
- Kramer, S. C. & Stelling, G. S. 2008 A conservative unstructured scheme for rapidly varied flows. *Int. J. Numer. Meth. Fluids* **58**, 183–212.
- Lauder, B. E. & Sharma, B. I. 1974 Application of the energy dissipation model of turbulence to the calculation of flow near a spinning disc. *Lett. Heat Mass Transfer* **1** (2), 131–138.
- Li, S., Cain, S., Wosnik, M., Miller, C., Kocahan, H. & Wyckoff, R. 2010 Numerical modeling of probable maximum flood flowing through a system of spillways. *J. Hydraul. Eng.* **137** (1), 66–74.
- Liu, T. & Yang, J. 2014 Three-dimensional computations of water-air flow in a bottom spillway during gate opening. *Eng. Appl. Comput. Fluid Mech.* **8** (1), 104–115.
- Lv, X., Zou, Q. & Reeve, D. 2011 Numerical simulation of overflow weirs using a hybrid level set/VOF method. *Adv. Water Resour.* **34** (10), 1320–1334.
- Olsen, N. R. B. 2003 3D CFD modeling of a self-forming meandering channel. *J. Hydraul. Eng.* **129** (5), 366–372.
- Olsen, N. R. B. 2013 Numerical algorithms for predicting sediment slides in water reservoirs. *Electron. J. Geotech. Eng.* **18** (Y) Paper 2013.496.
- Olsen, N. R. B. & Kjellesvig, H. M. 1998 Three-dimensional numerical flow modelling for estimation of spillway capacity. *J. Hydraul. Res.* **36** (5), 775–784.
- Patankar, S. V. 1980 *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill Book Company, New York.
- Rüther, N., Jacobsen, J., Olsen, N. R. B. & Vatne, G. 2010 Prediction of the three dimensional flow field and bed shear stresses in a regulated river in Mid-Norway. *Hydrology Res.* **41** (2), 145–152.

- Savage, B. M. & Johnson, M. C. 2001 [Flow over ogee spillway: physical and numerical case study](#). *J. Hydraul. Eng.* **127** (8), 640–649.
- Schlichting, H. 1979 *Boundary Layer Theory*. McGraw-Hill Book Company, New York.
- Soleiman Beygi, N., Hakimzadeh, H. & Chenaglou, M. R. 2014 [Simulation of free surface flows using volume of fluid method and genetic algorithm](#). *J. Hydroinform.* **16** (5), 1110–1124.
- Temeepattanapongsa, S., Merkley, G. P., Barfuss, S. L. & Smith, B. L. 2013 [Generic free-flow rating for cutthroat flumes](#). *J. Hydraul. Eng.* **139** (7), 727–735.
- Tritthart, M. & Gutknecht, D. 2007 3-D Computation of flood processes in sharp river bends. *Proc. Inst. Civil Engrs: Water Maritime Eng.* **160** (4), 233–247.
- Wilson, C. A. M. E., Stoesser, T., Olsen, N. R. B. & Bates, P. D. 2003 Application and validation of numerical codes in the prediction of compound channel flows. *Proc. Inst. Civil Engrs: Water Maritime Eng.* **153**, 117–128.
- Ying, X. & Wang, S. S. Y. 2010 [Evaluation of 2D shallow-water model for spillway flow with a complex geometry](#). *J. Hydraul. Res.* **48** (2), 265–268.

First received 13 January 2015; accepted in revised form 29 May 2015. Available online 9 July 2015