Solution of shallow water equations using fully adaptive multiscale schemes

Philipp Lamby[†], Siegfried Müller[*‡] and Youssef Stiriba[§]

Institut für Geometrie und Praktische Mathematik, RWTH Aachen, D-52056 Aachen, Germany

SUMMARY

The concept of fully adaptive multiscale finite volume methods has been developed to increase spatial resolution and to reduce computational costs of numerical simulations. Here grid adaptation is performed by means of a multiscale analysis based on biorthogonal wavelets. In order to update the solution in time we use a local time stepping strategy that has been recently developed for hyperbolic conservation laws.

The adaptive multiresolution scheme is now applied to two-dimensional shallow water equations with source terms. The efficiency of the scheme is demonstrated on several problems with a general geometry, including circular damp breaks, oblique hydraulic jump, supercritical channel flows encountering sudden change in cross-section, and, finally, the bore wave and its interactions.

KEY WORDS: Shallow water equations; multiscale techniques; local grid refinement; finite volume methods.

1. INTRODUCTION

Shallow water equations (SWEs) are used to describe many physical problems of interest often encountered in environmental and hydraulic engineering: free surface flows caused by damp breaking, hydraulic jumps, open channel flows, bore wave propagation, tidal flows in estuary and costal zones are just a few examples. The SWEs are obtained through integration of the Navier-Stokes system over the depth of the fluid body by assuming hydrostatic pressure distribution.

*Correspondence to: Siegfried Müller, Institut für Geometrie und Praktische Mathematik, RWTH Aachen, D-52056 Aachen, Germany

[†]E-mail: lamby@igpm.rwth-aachen.de

[‡]E-mail: mueller@igpm.rwth-aachen.de

[§]E-mail: stiriba@igpm.rwth-aachen.de

Contract/grant sponsor: This work has been performed with funding by the Deutsche Forschungsgemeinschaft in the Collaborative Research Center SFB 401 "Flow Modulation and Fluid-Structure Interaction at Airplane Wings" of the RWTH Aachen, University of Technology, Aachen, Germany.

Various finite volume schemes developed to deal with Euler and Navier-Stokes systems have been extended successfully to the SWEs. For instance, Roe's scheme has been modified by Bermúdez and Vázquez [6] to include source terms. Anastasiou and Chan [2] developed and tested a second-order upwind finite volume method on unstructured meshes. Chipada et al. [13] used a Godunov-type method together with Roe's approximate Riemann solver. Liska and Wendroff [24] employed their composite scheme. Several numerical methods for SWEs have been reported, and we refer the interested reader to the aforementioned papers and references therein for specific details.

Typically, fluid flow problems contain a wide range of spatial scales such as shocks or wave fronts in high activity regions and stagnation zones in low hydrodynamic activity regions. To simulate accurately the different scales over large and complex domains a highly resolved mesh is required. For multidimensional problems, in particular, uniform grids are not feasible. However, due to the heterogeneity of the flow field high resolution is only needed in a small part of the computational domain whereas a moderate resolution is sufficient elsewhere. Therefore adaptive strategies can significantly improve the efficiency of the computation. Consequently, a number of adaptive mesh strategies have been investigated by several researchers in the past decade with the goal of concentrating cells *only* where they are most required, while maintaining acceptable CPU time. A recent approach are multiresolution methods.

The multiresolution framework was first introduced by Harten [21, 20] for one-dimensional conservation laws and later extended to multidimensional problems [7, 1, 16, 15, 29] on Cartesian, curvilinear and unstructured meshes, respectively. The basic idea is to represent cell averages associated with any given finite volume method into a different format that gives insight on the local smoothness of the solution. With this approach the computational time is reduced by using sophisticated numerical fluxes in regions where high resolution is needed and cheap interpolations in regions requiring less resolution. The solution is adequately resolved at the same accuracy of the reference scheme, i.e., the scheme on a given uniform highest level of resolution (reference mesh).

Parallel to Harten's approach a fully adaptive multiresolution method has been developed by Müller et al. [25, 14]. In this approach the cell average representation is used to adapt the computational mesh. The advantage of this method includes a significant reduction of the computational effort in terms of CPU time and memory requirements. In particular, the accuracy of the reference solution on the reference mesh is maintained. This approach has found many applications in compressible fluid flows, particularly, in steady- and unsteady-state solutions [11, 32].

The present work now extends and tests the fully adaptive multiscale finite volume methods to SWEs, combining a quadtree grid generation strategy based on B-splines and fully adaptive multiresolution methods. In contrast to previous work we employ a locally varying time stepping algorithm, recently developed in [26]. This algorithm uses different time steps at different resolution levels such that a CFL condition is satisfied locally using the same CFL number for all resolution levels. Furthermore, the number of grid adaptation steps is reduced since we perform less time steps for the coarser cells. By various test cases we show the benefits of the fully adaptive multiscale concept for several steady and unsteady flow problems with general geometry.

The paper is organized as follows. In Section 2, we present the governing equations together with the reference finite volume method based on ENO reconstruction and Roe's approximate Riemann solver. The fully adaptive multiresolution method will be described in Section 3. In

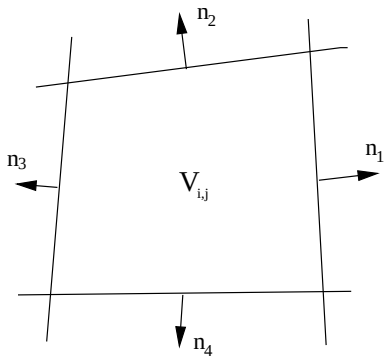

Figure 1. Quadrilateral finite control volume.

particular, we discuss the local grid adaptation and grid generation. Results of a collection of numerical experiments are presented and discussed in Section 4.

2. 2D SHALLOW WATER EQUATIONS AND FINITE VOLUME DISCRETIZATION

The shallow water equations are obtained by integration of the Navier-Stokes system over the depth of the fluid by assuming hydrostatic pressure distribution. Neglecting viscosity and turbulence effects, the coriolis force, and the bed and shear stress, the two-dimensional shallow water equations for any control volume V with boundary ∂V and outward unit normal $\mathbf{n} = (n_x, n_y)$ on the surface element $dS \subset \partial V$ can be written in integral form as

$$\frac{\partial}{\partial t} \int_V \mathbf{u}(t, \mathbf{x})\, dV + \int_{dS} \mathbf{f}^c(\mathbf{u}(t, \mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\, dS = \int_V \mathbf{s}(\mathbf{x}, \mathbf{u}(t, \mathbf{x}))\, dV \tag{1}$$

where $\mathbf{u}$ is the vector of conserved variables, $\mathbf{f}^c = (\mathbf{f}, \mathbf{g})^T$ is the convective flux vector and $\mathbf{s}$ is the vector of source terms. These are defined by

$$\mathbf{u} = \begin{pmatrix} h \\ h\,u \\ h\,v \end{pmatrix},\ \mathbf{f} = \begin{pmatrix} h\,u \\ h\,u^2 + \frac{1}{2}\,g\,h^2 \\ h\,u\,v \end{pmatrix},\ \mathbf{g} = \begin{pmatrix} h\,v \\ h\,u\,v \\ h\,v^2 + \frac{1}{2}\,g\,h^2 \end{pmatrix},\ \mathbf{s} = g\,h \begin{pmatrix} 0 \\ \frac{\partial b}{\partial x} \\ \frac{\partial b}{\partial y} \end{pmatrix} \tag{2}$$

where h is the water depth, u and v are the velocity components, b is the partial depth between a fixed reference level and the bed surface, and g is the acceleration due to gravity.

In the present approach, the physical domain Ω is discretized by a structured grid composed of quadrilateral cells $V_{i,j}$ with volume $|V_{i,j}| := \int_{V_{i,j}} 1\, dV$. For simplicity, the time is discretized by a constant time step size τ, i.e., $t^{n+1} = t^n + \tau$. For a quadrilateral finite control volume $V_{i,j}$, see Fig. 1, the integral equation (1) can be rewritten in the evolutionary form

$$\mathbf{u}_{i,j}^{n+1} = \mathbf{u}_{i,j}^{n} - \frac{\tau}{|V_{i,j}|}\, \mathbf{b}(t^n) + \tau\, \mathbf{s}(t^n) \tag{3}$$

where we introduce the averages

$$\mathbf{u}_{i,j}^n = \frac{1}{|V_{i,j}|} \int_{V_{i,j}} \mathbf{u}(t^n, \mathbf{x})\, d\mathbf{x}, \quad \mathbf{b}(t^n) = \frac{1}{\tau} \int_{t^n}^{t^{n+1}} \int_{\partial V_{i,j}} \mathbf{f}_{\mathbf{n}(\mathbf{x})}(\mathbf{u}(t^n, \mathbf{x}))\, dS\, dt, \tag{4}$$

$$\mathbf{s}(t^n) = \frac{1}{\tau} \frac{1}{|V_{i,j}|} \int_{t^n}^{t^{n+1}} \int_{V_{i,j}} \mathbf{s}(\mathbf{x}, \mathbf{u}(t^n, \mathbf{x}))\, d\mathbf{x}. \tag{5}$$

Here the flux in normal direction is determined by $\mathbf{f}_\mathbf{n} := \mathbf{f}^C \cdot \mathbf{n} = \mathbf{f}\, n_x + \mathbf{g}\, n_y$. Furthermore $\partial V_{i,j}$ denotes the boundary of the cell $V_{i,j}$, and the superscript represents the time level. Equation (4) is obtained by integrating the equation (1) over the control volume $V_{i,j} \times [t^n, t^n + \tau]$.

From the evolution equation (3) we deduce the finite volume discretization

$$\mathbf{v}_{i,j}^{n+1} = \mathbf{v}_{i,j}^n - \frac{\tau}{|V_{i,j}|} \mathbf{B}_{i,j}^n + \tau\, \mathbf{S}_{i,j}^n \tag{6}$$

where we replace the flux balance $\mathbf{b}(t^n)$ and the source term $\mathbf{s}(t^n)$ by approximations. The numerical flux balance is approximated by

$$\mathbf{B}_{i,j}^n := \sum_{l=1}^{4} |\Gamma_l| \;\; \mathbf{F}_{\mathbf{n}_l}^n. \tag{7}$$

Here $|\Gamma_l|$ denotes the length of the cell side with normal vector $\mathbf{n}_l$, and $\mathbf{F}_{\mathbf{n}_l}^n$ the corresponding numerical flux in normal direction, see Fig. 1. Since the inviscid SWEs are similar to the Euler system that governs inviscid compressible fluid flow, various numerical finite volume methods developed to deal with the latter have been extended with great success to SWEs. In the present work we use Roe's approximate Riemann solver that has been extended to SWEs used in [6]. In order to avoid non-physical expansion shocks we use Harten's entropy fix [19]. The spatial and temporal accuracy are improved by using a quasi one-dimensional second-order ENO reconstruction according to [23]. Here the reconstruction is applied to the characteristic variables instead of the conserved variables.

Finally, the numerical source term is approximated by

$$\mathbf{S}_{i,j}^n = \mathbf{s}(\hat{\mathbf{x}}_{i,j}, \mathbf{v}_{i,j}^n) \equiv g\, h_{i,j}^n \begin{pmatrix} 0 \\ \frac{\partial b}{\partial x}(\hat{\mathbf{x}}_{i,j}) \\ \frac{\partial b}{\partial y}(\hat{\mathbf{x}}_{i,j}) \end{pmatrix} \tag{8}$$

where $\hat{\mathbf{x}}_{i,j} := \frac{1}{|V_{i,j}|} \int_{V_{i,j}} \mathbf{x}\, dV$ denotes the centroid of $V_{i,j}$, and $h_{i,j}^n$ approximates the water depth.

3. THE ADAPTIVE MULTISCALE METHOD

The efficiency of the finite volume scheme presented in Section 2 is now to be improved by employing local grid refinement. For this purpose, we employ recent multiresolution techniques first proposed by Harten in the context of finite volume schemes for hyperbolic conservation laws, cf. [21]. In contrast to Harten's original idea, we do not use these techniques to control the flux computation but to trigger the grid adaptation process. For details we refer the reader to the book [25]. In the following we confine only to the basic ideas.

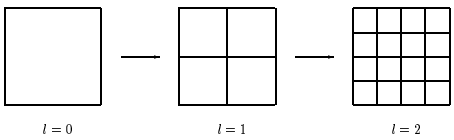

Figure 2. Sequence of nested grids

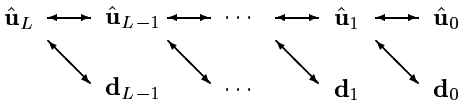

Figure 3. Multiscale transformation

3.1. Multiscale analysis

In order to analyze the local regularity behavior of the flow field we employ a multiresolution analysis. For this purpose we may use the concept of biorthogonal wavelets [12] or the concept of prediction and reconstruction, cf. [3].

The fundamental idea is to present cell averages on a given uniform highest level of resolution (*reference mesh*) associated with any given finite volume discretization (*reference scheme*) as cell averages on some coarse level where the fine scale information is encoded in arrays of *detail coefficients* of ascending resolution. This decomposition is performed on a hierarchy of *nested* grids. In the present work we confine to structured two-dimensional grids though the general framework is not restricted to this configuration but can also be applied to *unstructured* grids and *irregular* grid refinements in arbitrary space dimensions. Since the meshes are assumed to be structured we may enumerate the cells on each level l by a multi-index, i.e., $\boldsymbol{k} = (k_1, k_2)$. Then a hierarchy of partitions $\mathcal{G}_l = \{V_{l,\boldsymbol{k}}\}_{\boldsymbol{k}\in I_l}$ with increasing resolution $l = 0, \ldots, L$ of the computational domain $\Omega \subset \mathbb{R}^2$, i.e., $\Omega = \bigcup_{\boldsymbol{k}\in I_l} V_{l,\boldsymbol{k}}$, is determined by dyadic grid refinement, i.e.,

$$V_{l,\boldsymbol{k}} = \bigcup_{\boldsymbol{e}\in E} V_{l+1,2\boldsymbol{k}+\boldsymbol{e}}, \quad E := \{0,1\}^2. \tag{9}$$

See Figure 2 for an illustration. Relative to the partitions $\mathcal{G}_l$ we introduce the cell averages of a scalar, integrable function $u \in L^1(\Omega)$, i.e.,

$$\hat{u}_{l,\boldsymbol{k}} := \frac{1}{|V_{l,\boldsymbol{k}}|} \int_{V_{l,\boldsymbol{k}}} u(\boldsymbol{x})\, d\boldsymbol{x}. \tag{10}$$

Now the ultimate goal is to transform the array of cell averages $\boldsymbol{u}_L := (\hat{u}_{L,\boldsymbol{k}})_{\boldsymbol{k}\in I_L}$ corresponding to a finest uniform discretization level L into a sequence of coarse grid data $\boldsymbol{u}_0 := (\hat{u}_{0,\boldsymbol{k}})_{\boldsymbol{k}\in I_0}$ and details $\boldsymbol{d}_l := (d_{l,\boldsymbol{k}})_{\boldsymbol{k}\in I_l}$, $l = 0, \ldots, L-1$, representing the successive update from a coarse resolution to a high resolution. This is to be realized successively by a *multiresolution transformation* where we proceed levelwise from fine to coarse, i.e.,

$$\hat{u}_{l,\boldsymbol{k}} = \sum_{\boldsymbol{e}\in E} \frac{|V_{l+1,2\boldsymbol{k}+\boldsymbol{e}}|}{|V_{l,\boldsymbol{k}}|}\, \hat{u}_{l+1,2\boldsymbol{k}+\boldsymbol{e}}, \quad d_{l,\boldsymbol{k}} = \sum_{\boldsymbol{r}\in\mathcal{A}} \alpha_{\boldsymbol{r}}\, \hat{u}_{l+1,2\boldsymbol{k}+\boldsymbol{r}} \tag{11}$$

where the first relation results from the nestedness of the grids and the linearity of the integration operator. Then (11) provides an equivalent representation which can be reversed by the *inverse multiresolution transformation*

$$\hat{u}_{l+1,k} = \sum_{\boldsymbol{r}\in\mathcal{B}} \beta_{\boldsymbol{r}}\, \hat{u}_{l,\lfloor \boldsymbol{k}/2\rfloor+\boldsymbol{r}} + \sum_{\boldsymbol{r}\in\mathcal{C}} \gamma_{\boldsymbol{r}}\, d_{l,\lfloor \boldsymbol{k}/2\rfloor+\boldsymbol{r}}. \tag{12}$$

Here the coefficients $\alpha_{\boldsymbol{r}}$, $\beta_{\boldsymbol{r}}$ and $\gamma_{\boldsymbol{r}}$ only depend on the geometric data but not on u. However, they may depend on the level l and the position $\boldsymbol{k}$. The same holds true for the sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. Furthermore, we emphasize that there are 2^{d-1} (here d=2) detail coefficients corresponding to one cell. Therefore the coefficients $d_{l,\boldsymbol{k}}$ in principle represent a vector of details. In Figure 3 we sketch graphically the procedure of the multiscale transformation.

Note that the construction of an appropriate multiresolution analysis is subject to three conditions, namely, (i) *reversibility*, i.e., transformations (11) and (12) are equivalent, (ii) *efficiency*, i.e., the cardinality of the index sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ is uniformly bounded, and (iii) *stability*, i.e., the linear multiresolution operator is uniformly bounded in the spectral norm.

3.2. Local grid adaptation

It can be shown that the details become small with increasing refinement level when the underlying function is smooth. This motivates to neglect all sufficiently small details in order to compress the original data. The idea is simply to discard all coefficients $d_{l,\boldsymbol{k}}$ that fall in absolute value below a certain threshold. For this purpose, we introduce the index set

$$\mathcal{D}_{L,\varepsilon} := \left\{(l,\boldsymbol{k}) \ ; \ |d_{l,\boldsymbol{k}}| > \varepsilon_l, \ \boldsymbol{k} \in I_l, \ l \in \{0,\ldots,L-1\}\right\}$$

corresponding to what will be referred to as *significant details*. Here $\varepsilon_l = 2^{l-L}\varepsilon$ is a level–dependent threshold value. By means of the set $\mathcal{D}_{L,\varepsilon}$ a locally refined grid is determined. For this purpose, we recursively check proceeding levelwise from coarse to fine whether there exists a significant detail to a cell. If there is one, then we refine the respective cell. Note that this procedure only works provided that the set $\mathcal{D}_{L,\varepsilon}$ corresponds to graded tree. We finally obtain the locally refined grid with hanging nodes represented by the set $\mathcal{G}_{L,\varepsilon} \subset \{(l,\boldsymbol{k})\,;\,\boldsymbol{k} \in I_l, \ l = 0,\ldots,L\}$, i.e., $\Omega = \bigcup_{(l,k)\in\mathcal{G}_{L,\varepsilon}} V_{l,\boldsymbol{k}}$. This set is composed of the leaves corresponding to the tree of significant details. Of course, the crux in this context is to arrange this procedure in such a way that at no stage of the computation there is ever made use of the fully refined uniform mesh. For this purpose, the multiresolution transformations (11) and (12) are performed locally to provide the details on the old time step and the cell averages on the new grid.

Since the flow field and, hence, the cell averages, evolves in time, grid adaptation is performed after each evolution step to provide the adaptive grid on the *new* time level. Since the corresponding averages, respectively details are not yet available, we have to *predict* all details on the new time level $n+1$ that may become significant due to the evolution by means of the details on the *old* time level n. In order to guarantee the adaptive scheme to be *reliable* in the sense that no significant future feature of the solution is missed, the prediction set $\tilde{\mathcal{D}}^{n+1}_{L,\varepsilon}$ has to satisfy

$$\mathcal{D}^{n}_{L,\varepsilon} \cup \mathcal{D}^{n+1}_{L,\varepsilon} \subset \tilde{\mathcal{D}}^{n+1}_{L,\varepsilon}, \tag{13}$$

where, of course $\mathcal{D}^{n+1}_{L,\varepsilon}$ is not known at the old time level. In [21] Harten suggests a heuristic approach that could not be rigorously verified to satisfy (13). However, in [14] a slight modification of Harten's prediction strategy has been shown to lead to a reliable prediction strategy in the sense of (13).

3.3. Efficient representation of a nested grid hierarchy

The multiscale setting outlined in Section 3.1 is based on a *hierarchy of nested grids*. In the curvilinear case this sequence can be efficiently realized by a *parametric mapping* $\mathbf{x} : [0,1]^2 \to \Omega$

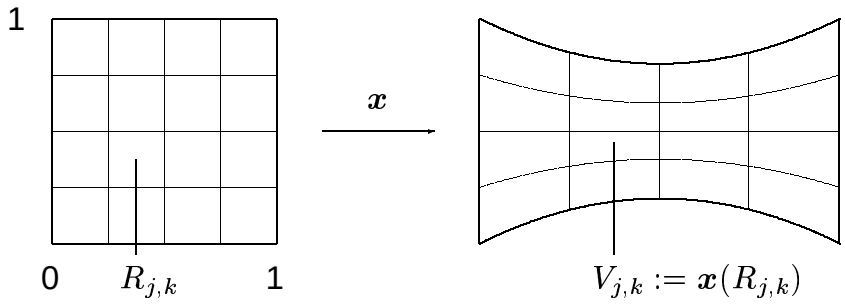

Figure 4. Parametric Mappings

from a logical space to the physical domain Ω. In this setting grid cells are the images of the corresponding cells in logical space, cf. Figure 4. Then grid refinement can be interpreted simply as function evaluation. Furthermore the grid generation process can completely be separated from the discretization process because the grid generator needs only to provide a (possibly sparse) representation of the grid function. For the representation of such a parameter mapping we use tensor product B-splines. These functions seem to be a very appropriate tool for this task, because they possess excellent approximation properties. Modeling is intuitive and the evaluation is fast and numerically stable. For references on splines we refer to [17, 28]. In order to enhance the flexibility we may embed several of such mappings into a multiblock concept to handle complex geometries, cf. [11].

For $i = 0, 1, \ldots N$ we denote with $N_{i,p,T}$ the i-th normalized B-spline of order p with respect to the knot vector T. Here $T = (t_i)_{i=0}^{N+p-2}$ is a non-decreasing and non-stationary sequence of real numbers, i.e., $t_i \leq t_{i+1}$ and $t_i < t_{i+p}$. The B-splines are piecewise polynomials of degree $p-1$ and can be defined by the recursion

$$N_{i,1,T}(t) = \chi_{[t_i,t_{i+1})}(t) = \begin{cases} 1 & \text{if } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{14}$$

$$N_{i,p,T}(t) = \frac{t - t_i}{t_{i+p-1} - t_i} N_{i,p-1}(t) + \frac{t_{i+p} - t}{t_{i+p} - t_{i+1}} N_{i+1,p-1}(t). \tag{15}$$

From this we build planar grids by tensor products of the form

$$\boldsymbol{x}(u, v) = \sum_{i=0}^{N} \sum_{j=0}^{M} \boldsymbol{p}_{i,j}\, N_{i,p_u,U}(u)\, N_{j,p_v,V}(v). \tag{16}$$

The $\boldsymbol{p}_{ij}$ are called *control points*. Generally, they are not grid points but can be considered as discrete approximation of the grid function, see Figure 5. For curved configurations we usually choose $p = 4$, i.e., cubic splines [8, 9, 10]. However, for the applications in Section 4 we use $p = 2$, i.e., linear splines.

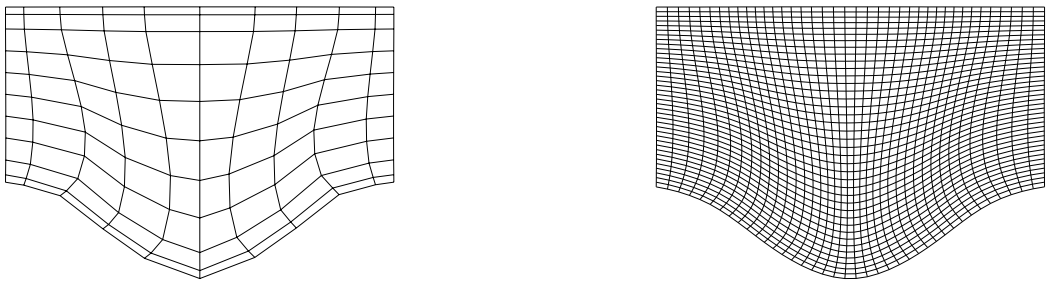

Figure 5. Control Points and Evaluation of Grid Function

3.4. Adaptive multiresolution finite volume schemes

Finally we have to provide the time evolution on the locally refined grid. Here the ultimate goal is aiming at reducing the computational costs with regard to both computational time *and* memory requirements but still maintaining the accuracy of the reference scheme. For this purpose we apply the multiscale transformation to the reference scheme. For details we refer the reader to [14, 25].

Since the reference scheme (6) is assumed to use an explicit time discretization, the time step size is bounded due to the CFL condition by the smallest cell in the grid. Hence τ is determined by the highest refinement level L, i.e., $\tau = \tau_L$. However, for cells on the coarser scales $l = 0, \ldots, L-1$ we may use $\tau_l = 2^{L-l}\,\tau_L$ to satisfy locally the CFL condition. In [26] a local time stepping strategy has been incorporated recently to the standard adaptive multiresolution finite volume scheme as proposed in [25]. The basic idea is to evolve each cell on level l with the level-dependent time discretization $\tau_l = 2^{L-l}\,\tau_L$, $l = 0, \ldots, L$. Obviously, all cell averages correspond to the same integration time after having performed 2^l time steps with τ_l, i.e., the cells are *synchronized*. This is schematically sketched in Figure 6. Therefore

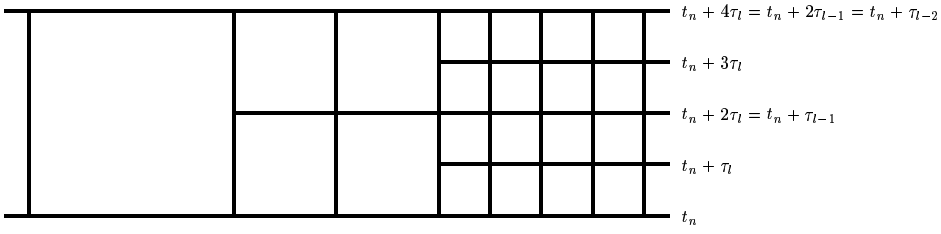

Figure 6. Synchronization on multilevel grid

one macro time step with $\tau_0 = 2^L\,\tau_L$ consists of 2^L intermediate time steps corresponding to the time level $t_{n+i\,2^{-L}}$, $i = 0, \ldots, 2^L - 1$, with step size τ_L. Obviously, at time $t_{n+i\,2^{-L}}$ the *smallest synchronization level* is determined by

$$l_i := \min\{l\,;\, 0 \le l \le L,\ i \bmod 2^{L-l} = 0\}.$$

The intermediate time steps $i = 0, \ldots, 2^L - 1$, then take the form

$$\mathbf{v}_{l,\boldsymbol{k}}^{n+(i+1)\,2^{-L}} = \mathbf{v}_{l,\boldsymbol{k}}^{n+i\,2^{-L}} - \frac{\tau_L}{|V_{l,\boldsymbol{k}}|}\,\mathbf{B}_{l,\boldsymbol{k}}^{n+i\,2^{-L}} + \tau_L\,\mathbf{S}_{l,\boldsymbol{k}}^{n+i\,2^{-L}} \tag{17}$$

for any cell $(l, \boldsymbol{k}) \in \mathcal{G}_{L,\varepsilon}$ of the current locally adapted grid. Similar to (7) of the reference scheme the numerical flux balance is determined by

$$\mathbf{B}_{l,\boldsymbol{k}}^{n+i\,2^{-L}} = \mathbf{F}_{l,\boldsymbol{k}+\boldsymbol{e}_1}^{n+i\,2^{-L}} - \mathbf{F}_{l,\boldsymbol{k}}^{n+i\,2^{-L}} + \mathbf{G}_{l,\boldsymbol{k}+\boldsymbol{e}_2}^{n+i\,2^{-L}} - \mathbf{G}_{l,\boldsymbol{k}}^{n+i\,2^{-L}}$$

with the numerical fluxes

$$\mathbf{F}_{l,\boldsymbol{k}}^{n+i\,2^{-L}} = \begin{cases} \mathbf{F}(\mathbf{v}_{l,\boldsymbol{k}-q\boldsymbol{e}_1}^{n+i\,2^{-L}}, \ldots, \mathbf{v}_{l,\boldsymbol{k}+q\boldsymbol{e}_1-1}^{n+i\,2^{-L}}) & , \quad l_i \le l \le L \text{ and no hanging node at interface} \\ \mathbf{F}_{l+1,2\boldsymbol{k}}^{n+i\,2^{-L}} + \mathbf{F}_{l+1,2\boldsymbol{k}+\boldsymbol{e}_2}^{n+i\,2^{-L}} & , \quad l_i \le l \le L \text{ and hanging node at interface} \\ \mathbf{F}_{l,\boldsymbol{k}}^{n+(i-1)\,2^{-L}} & , \quad \text{elsewhere} \end{cases}$$

and

$$\mathbf{G}_{l,\boldsymbol{k}}^{n+i\,2^{-L}} = \begin{cases} \mathbf{G}(\mathbf{v}_{l,\boldsymbol{k}-q\boldsymbol{e}_2}^{n+i\,2^{-L}}, \ldots, \mathbf{v}_{l,\boldsymbol{k}+q\boldsymbol{e}_2-1}^{n+i\,2^{-L}}) & , \quad l_i \le l \le L \text{ and no hanging node at interface} \\ \mathbf{G}_{l+1,2\boldsymbol{k}}^{n+i\,2^{-L}} + \mathbf{G}_{l+1,2\boldsymbol{k}+\boldsymbol{e}_1}^{n+i\,2^{-L}} & , \quad l_i \le l \le L \text{ and hanging node at interface} \\ \mathbf{G}_{l,\boldsymbol{k}}^{n+(i-1)\,2^{-L}} & , \quad \text{elsewhere} \end{cases}$$

Here we distinguish between the numerical fluxes $\mathbf{F}$ and $\mathbf{G}$ in the first and second parameter direction of the underlying parametric grid mapping, cf. Section 3.3. For lower scales we use the flux of the previous intermediate time step. In the other case we distinguish whether there is a hanging node at the corresponding cell edge. If there is no hanging node we have to compute the quasi one-dimensional numerical fluxes $\mathbf{F}(\mathbf{w}_1, \ldots, \mathbf{w}_{2q})$ and $\mathbf{G}(\mathbf{w}_1, \ldots, \mathbf{w}_{2q})$, respectively, of the reference scheme due to Section 2. The stencil of numerical fluxes is characterized by the parameter q. For our second order scheme it is $q = 2$. Note that locally non-accessible data in the adaptive grid have to be computed by performing a local refining or coarsening step applying the two-scale transformation according to (11) and (12). In the other case the numerical flux is determined by the fluxes of the higher scale to maintain conservation of the scheme. Here we employ the fact that the refinement level of two adjacent cells differs by at most one, i.e., there is at most one hanging node at one edge. This can be ensured by a grading process of the adaptive grid, cf. Section 3.2. To ensure that the fluxes at an interface with hanging nodes have already been computed when determining the corresponding flux on the coarser level, we perform for each intermediate time level the time evolution first for the cells on the highest level and then successively for the coarser levels. This procedure is similar to the predictor-corrector method in [27].

The sources are determined by

$$\mathbf{S}_{l,\boldsymbol{k}}^{n+i\,2^{-L}} = \begin{cases} \mathbf{s}(\hat{\mathbf{x}}_{l,\boldsymbol{k}}, \mathbf{v}_{l,\boldsymbol{k}}^{n+i\,2^{-L}}) & , \quad l_i \le l \le L \\ \mathbf{S}_{l,\boldsymbol{k}}^{n+(i-1)\,2^{-L}} & , \quad 0 \le l < l_i \end{cases} .$$

Note that for the lower levels $0, \ldots, l_i - 1$ we do not compute new fluxes or sources. This makes the local time stepping version of the adaptive multiresolution concept more efficient then the standard approach using a global time step size.

Finally, we have to comment on the grid adaptation step. The ultimate goal is to provide after one macro time step with $\tau_0 = 2^L \tau_L$ as good an approximation as having performed

2^L time steps with the reference scheme on the reference mesh using the time step size τ_L. Therefore we have to make sure that the solution is adequately resolved at each intermediate time step. For this purpose we perform a grid adaptation step according to Section 3.2 before each even intermediate time step, i.e., $i = 0, 2, \ldots, 2^L - 2$. However, we do not apply the adaptation process for the whole computational domain, but only for the cells on the levels $l = l_i, \ldots, L$, i.e., level l_i is considered to be the coarsest scale in the multiscale analysis. Note, that only for this range of scales new fluxes and sources have to be recomputed. This process provides us with the sets $\mathcal{G}_{L,\varepsilon}^{n+(i+1)\,2^{-L}}$ for which we perform the evolution step (17). For the odd intermediate time steps we use the same grid as in the previous step, i.e., $\mathcal{G}_{L,\varepsilon}^{n+i\,2^{-L}} = \mathcal{G}_{L,\varepsilon}^{n+(i-1)\,2^{-L}}$, $i = 1, 3, \ldots, 2^L - 1$. Hence, it is possible to track, for instance, the shock position on the intermediate time levels instead of a-priori refining the whole range of influence.

4. NUMERICAL EXPERIMENTS

In this section, various numerical experiments are considered to show the benefit of the fully adaptive multiscale concept for various steady and unsteady flow problems. We compare the results for some numerical tests with analytical and numerical results reported in the literature.

We take $g = 9.8\ m/s^2$ in our computations. The time discretization is determined by the CFL number. Here we choose $CFL_0 = 0.8 = 2^L\ CFL_L$ on the coarser level for the adaptive locally varying time stepping method. This implies that for the explicit adaptive finite volume scheme we have to perform 2^L time step with $\tau = \tau_L$. For comparison, we perform computations with the standard scheme using a global time size corresponding to $CFL_L = 0.8$ for all levels. For the grid adaptation we choose wavelets with three vanishing moments. The threshold value for the multiscale analysis is $\varepsilon = 10^{-3}$. We display only numerical results obtained by the adaptive locally varying time stepping strategy. Note that all computations have been performed on a PC with an Intel Pentium IV processor and 2.8 GHz.

4.1. Supercritical channel flows

The first series of test problems concerns supercritical (Froude number $F_r = u\ /\ \sqrt{g\ h} > 1$) flows through channels with sudden change in cross section. We use three different channel configurations utilized by several authors to test and validate their numerical methods, see for instance [24, 13]. We apply slip boundary conditions at the solid wall, i.e., the normal velocity is set to zero $(u, v) \cdot \mathbf{n} = 0$. These problems lead to the formation of steady state flows with hydraulic jumps (shock waves) and negative jumps (rarefaction waves).

4.1.1. Oblique hydraulic jump. In the first test, we consider supercritical flow in a channel constricted from one side wall. The geometry of the problem is shown in Figure 7. It can be represented by a bilinear spline with knot vectors $U = (0,\ 0,\ 0.25,\ 1,\ 1)$, $V = (0,\ 0,\ 1,\ 1)$ and control points

$$(\boldsymbol{p}_{i,j}) = \begin{pmatrix} (0,0) & (10,0) & (40, 30\sin\theta) \\ (0,30) & (10,30) & (40,30) \end{pmatrix}, \tag{18}$$

where the angle of constriction is $\theta = 8.95^o$. The initial conditions are the height $h = 1\ m$, velocity $u = 8.57\ m\ s^{-1}$ and $v = 0$. We use inflow conditions at $x = 0$, slip boundary conditions

along $y-$direction, and outflow conditions at $x = 40$. The problem leads to the formation of a stationary hydraulic jump at a particular angle α.

Computations were initialized with a structured grid with a resolution of 15×10 cells and were run to steady state. Figure 7 shows the contours of the steady state water depth using $L = 6$ refinement levels. The shock is sharply captured without oscillations. The approximate shock angle $\alpha \approx 29.753$ agrees very well with exact analytical solution $\alpha = 30^o$. Numerical values of the height $h = 1.499$ and the velocity $|\mathbf{u}| = 7.951$ behind the shock are also in very good agreement with the analytical solution $h = 1.5$, $|\mathbf{u}| = 7.955$.

In Table I we compare the CPU times for the fully adaptive finite volume scheme using global C_{GTS} and local time stepping C_{LTS}, respectively, where we vary the number of refinement levels L and fix the coarse grid discretization. The speed up is improved with increasing L. For the highest number of refinement levels the speed-up factor is approximately 2. By the spatial adaptation the data are significantly reduced. For instance, the uniform discretization with $L = 6$ refinement levels consists of 614400 cells, whereas the final adaptive mesh consists of 23283 cells only.

L	C_{GTS}	C_{LTS}
4	1976	1229
5	8423	5217
6	32765	20218
7	145821	84683

Table I. CPU time (sec) vs. number of refinement levels.

In the second test, the angle of restriction is set to $\theta = 15^o$. The initial conditions are the height $h = 1\,m$ and the Froude number $Fr = 3$. We use the same boundary conditions as in the previous example. Figure 7 displays the water depth together with the adaptive grid. Again the shock angle $\alpha \approx 34.1^o$ agrees well with the analytical solution $\alpha = 34.36^o$.

4.1.2. Channel constricted on both sides. In this test case the channel is symmetrically constricted on both sides with angle $\theta = 5^o$. The geometry of the channel is depicted in Figure 8 where we impose supercritical inflow conditions at $x = 0$ with height $h = 1\ m$ and Froude number $Fr = 2.5$. We use an initial grid with a resolution of 45×20 and a maximum refinement level of $L_{max} = 5$.

Figure 8 shows the steady state water depth and the corresponding adaptive grid, respectively. The cross-wave pattern due to constriction is well resolved. Again we can compare numerical values of the height with the analytical solution. The numerical water depths of the first and second plateau are $h = 1.249\ m$ and $h = 1.525\ m$, respectively. Those results compare very well with the analytical ones $h = 1.254\ m$ and $h = 1.55\ m$. The CPU times in second were 107448 s and 64237 s for the adaptive scheme using global and local time step, respectively. The final adaptive grid consists of 93084 cells in comparison to 921600 of the uniform finest discretization.

4.1.3. Channel with variable width. In the last example, we consider a more complicated symmetric channel constricted on both sides from $x = 10$ to $x = 30$. The constriction angle is

$\theta = 15^o$. Figure 9 displays the geometry and the initial grid, whose parameter representation is given by the bilinear spline with knot vectors U = (0, 0, 1/9, 1/3, 1, 1), V=(0, 0, 1, 1) and control points

$$(\boldsymbol{p}_{i,j}) = \begin{pmatrix} (0,0) & (10,0) & (30,\delta) & (90,\delta) \\ (0,40) & (10,40) & (30,40-\delta) & (90,40-\delta) \end{pmatrix}, \tag{19}$$

where $\delta = 20 \sin\theta$. We apply the same boundary conditions as before. The inflow conditions at $x = 0$ are the height $h = 1\ m$ and the Froude number $Fr = 2.5$. We again use an initial grid with a resolution of 45×20, a maximum refinement level of $L_{max} = 5$, and run the model to steady state.

Figure 9 shows the steady state water depth and the corresponding adaptive grid, respectively. The cross-wave pattern now includes both hydraulic jumps and rarefaction waves due to the presence of concave corners as observed in [24, 13]. One can see that the fully adaptive scheme resolves well hydraulic jumps, negative waves and their interactions. The CPU times are 163489 s and 95406 s for the adaptive scheme using global and local time stepping, respectively. The final adaptive grid consists of 133605 cells in comparison to 921600 of the uniform finest discretization.

4.2. Circular dam break

In this example we consider the circular dam break problem. A variety of numerical results by different schemes are available in [24, 2] and references therein. The initial conditions are two states of water separated by a cylinder of radius $r = 11\,m$. The depth of the water inside the dam is $10\,m$ and outside the dam is $1m$. Here we are interested in the instance when the cylindrical wall is removed and the subsequent time evolution of the flow.

The computational domain is a square $[0, 50\,m] \times [0, 50\,m]$. The coarsest grid is discretized by 15×15 cells. The model was run up to time $T = 0.7\,s$. Figure 10 shows the water depth contours and the corresponding adaptive grid with $L = 7$ refinement levels. We display also the water surface elevation in three dimensions in Figure 11. One can observe that the circular symmetry of the front is well preserved without irregular depressing waves. The results of contours of water surface elevations are in close agreement with those published in [24, 2], but the present computations require less computer time. The CPU time for the adaptive scheme using global and local time stepping are 22737 s and 12461 s, respectively. The final adaptive grid consists of 79788 cells in comparison to 921600 of the uniform finest discretization.

4.3. Bore wave past a hump

This test gives unsteady results of a right-going bore wave past a downward hump. The physical domain is shown in Figure 12. The Froude number is $Fr = 2$, and the initial conditions to the left and the right of the bore are similar to those in [34]

$$h_R = 1\ m, \quad u_R = v_R = 0\ m.s^{-1} \tag{20}$$

$$h_L = \frac{h_R}{2}\left(\sqrt{1+8Fr^2}-1\right)\ m, \quad u_L = Fr\left(1-\frac{h_R}{h_L}\right)\sqrt{g\ h_R}\ m.s^{-1}, \quad v_L = 0\ m.s^{-1}. \tag{21}$$

The initial discontinuity is located at $x = -5.5$. The bottom function $b(x, y)$ is defined by

$$b(x,y) = -0.4\ e^{0.2(12.5-x^2-y^2)}. \tag{22}$$

We use inflow boundary conditions at the left boundary and outflow boundary conditions for the other boundaries.

The computation uses 7 levels of refinement and the coarse initial grid corresponds to 15×15 cells. Figure 12 shows the adaptive grid and the numerical approximation of the water depth with 30 equally spaced contours at time $T = 2$. A 3D plot of the results is depicted in Figure 13. The required CPU time for the adaptive scheme using global and local time stepping are 19375 s and 9888 s, respectively. The final adaptive grid consists of 261720 cells in comparison to 3686400 of the uniform finest discretization.

4.4. Single Mach reflection at a slopping wall

The final test concerns the bore reflection at a sea wall [36, 34]. The geometry of the problem is shown in Figure 14 where a right traveling bore wave along the basin interacts with a wall deflected by an angle 25^o. The initial Froude number and the left and right states to the bore are the same as in the previous example. The initial shock is located at the point $x = -10$. For the lower boundary we use reflecting conditions and outflow conditions for the upper boundary. On the left boundary we impose inflow conditions and outflow conditions on the right.

The computation is initialized by a structured grid with a resolution of 30×10 cells and we run to time $T = 8$. We use $L = 6$ levels of refinement and a threshold value $\varepsilon = 10^{-2}$. Figures 15 and 16 show the adaptive grid, the water depth and the x-velocity component with 35 equally spaced contours, respectively. The reflective pattern is well resolved in comparison with results presented in [36, 34]. The bore wave has been captured sharply. The computations require 4542 s for the adaptive scheme using local time stepping and 8615 s for the same scheme using global time stepping. The speed-up factor is approximately 2. The uniform mesh contains 1843200 cells. Due to adaptation the final adaptive grid consists of 47430 cells only resulting in a significant reduction of the computational costs.

5. CONCLUSIONS

In this paper, we have extended and tested the concept of the fully adaptive multiscale finite volume schemes to two-dimensional shallow water equations with general geometry. The concept combines a quadtree grid generation strategy using a B-spline method and fully adaptive multiresolution methods. The essential feature of the presented scheme is the use of a local time stepping strategy at different resolution levels of the grid hierarchy. By this strategy we, in particular, save an additional factor of about 2. Several test problems have been presented to show the high efficiency and the high accuracy of the presented method.

REFERENCES

1. Abgrall R, Harten A. Multiresolution Representation on unstructured meshes. *SIAM Journal of Numerical Analysis* 1998 **35**(6):2128–2146.
2. Anastasiou K, Chan CT. Solution of 2D shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids* 1997 **24**:1225–1245.
3. Arandiga F, Donat R, Harten A. Multiresolution based on weighted averages of the hat function I: Linear reconstruction techniques. *SIAM J. Numer. Anal.* 1998 **36**(1):160–203.

4. Berger M, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* 1984 **53**:484–512.
5. Berger M, and Colella P. Local Adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* 1989 **82**:64–84.
6. Bermudez A, Vazquez ME. Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids* 1994 **23**:1094–1071.
7. Bihari B, Harten A. Multiresolution schemes for the numerical solution of 2D conservation laws *SIAM Journal of Scientific Computing* 1997 **18**(2):315–354.
8. Brakhage K-H. High Quality Mesh Generation and Sparse Representation Using B-Splines. *Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations, Chateau Whistler Resort, British Columbia* 2000. Soni BK, Thompson JF, Häuser J, Eiseman, P (eds.), 753-762.
9. Brakhage K-H, Lamby Ph. CAGD Tools for High Quality Grid Generation and Sparse Representation. *Proceedings of the 9th International Conference on Numerical Grid Generation in Computational Field Simulations, Waikiki Beach, Hawaii* 2002. Soni BK, Thompson JF, Häuser J, Eiseman, P (eds.), 599-608.
10. Brakhage K-H, Müller S. Algebraic-hyperbolic grid generation with precise control of intersection of angles. *International Journal for Numerical Methods in Fluids* 2000 **33**:89-123.
11. Bramkamp F, Lamby Ph, Müller S. An adaptive multiscale finite volume solver for unsteady and steady state flow computations. *Journal of Computational Physics* 2004 **197**(2):460–490.
12. Carnicer JM, Dahmen W, Peña JM. Local decomposition of refinable spaces and wavelets. *Appl. Comput. Harmon. Anal.* 1996 **3**:127-153.
13. Chippada S, Dawson CN, Martinez ML, Wheeler MF. A Godunov-type finite volume method for the system of Shallow water equations. *Comput. Methods Appl. Engrg.* 1998 **151**:105–129.
14. Cohen A, Kaber SM, Müller S, Postel M. Fully adaptive multiresolution finite volume schemes for conservation laws. *Mathematics of Computation* 2003 **72**(241):183–225.
15. Cohen A, Dyn N, Kaber SM, Postel M. Multiresolution finite volume schemes on triangles. *Journal of Computational Physics* 2004 **161**:264–286.
16. Dahmen W, Gottschlich–Müller B, Müller S. Multiresolution schemes for conservation laws. *Numerische Mathematik* 2003 **88**(3):399–443.
17. de Boor C. A Practical Guide To Splines. Springer, 1978.
18. Greaves DM, Borthwick AGL. On the use of adaptive hierarchical meshes currents meshes for numerical simulations of separated flows. *International Journal for Numerical Methods in Fluids* 1998 **26**:303–322.
19. Harten A. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* 1983 **49**:357–393.
20. Harten A. Adaptive multiresolution schemes for shock computations. *Journal of Computational Physics* 1994 **115**:319–338.
21. Harten A. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.* 1995 **48**(12):1305–1342.
22. Harten A. Multiresolution representation of data: A general framework. *SIAM J. Numer. Anal.* 1995 **33**(3):1205–1256.
23. Harten A, Engquist B, Osher S, Chakravarthy SR. Uniformly high order accurate essentially non–oscillatory schemes III. *Journal of Computational Physics* 1987 **71**:231–303.
24. Liska R, Wendroff B. Two-dimensional shallow water equations by composite schemes. *International Journal for Numerical Methods in Fluids* 1999 **30**:461–479.
25. Müller S. Adaptive Multiscale Schemes for Conservation Laws. *Lecture Notes on Computational Science and Engineering*. Springer, 2002. **27**.
26. Müller S, Stiriba Y. Fully adaptive multiscale scheme for conservation laws employing locally varying time stepping. *IGPM–Report* 2004 **238**.
27. Osher S, Sanders R. Numerical approximations to nonlinear conservation laws with locally varying time and space grids. *Math. Comp* 1983 **41**:321–336.
28. Piegl L, Tiller W. The NURBS book. Springer, 2nd edition, 1997.
29. Rault A, Chiavassa G, Donat R. Shock-vortex interactions at high Mach numbers. *Journal of Scientific Computing* 2003 **19**:347–371.
30. Roe P. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981 **43**:357–372.
31. Rogers B, Fujihara M, Alistair G, Borthwick GL. Adaptive Q-tree Godunov-type scheme for shallow water equations. *International Journal for Numerical Methods in Fluids* 2001 **35**:247–280.
32. Roussel O, Schneider K, Tsigulin A, Bockhorn H. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *Journal of Computational Physics* 2003 **188**(2):493-523.
33. Vazquez ME. Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *Journal of Computational Physics* 1999 **23**:497–526.

34. Tang H. Solution of the shallow-water equations using an adaptive moving mesh method. *Int. J. Numer. Meth. Fluids* 2004 **44**:789-810.
35. Wille SO. On the use of adaptive hierarchical meshes currents meshes for numerical simulations of separated flows. *Comput. Methods Appl. Engrg.* 1998 **166**:379–390.
36. Zhou JG, Causon DM, Mingham CG, Ingram DM. The surface gradient method for the treatment of source terms in the shallow water equations. *Journal of Computational Physics* 2001 **24**:1–25.

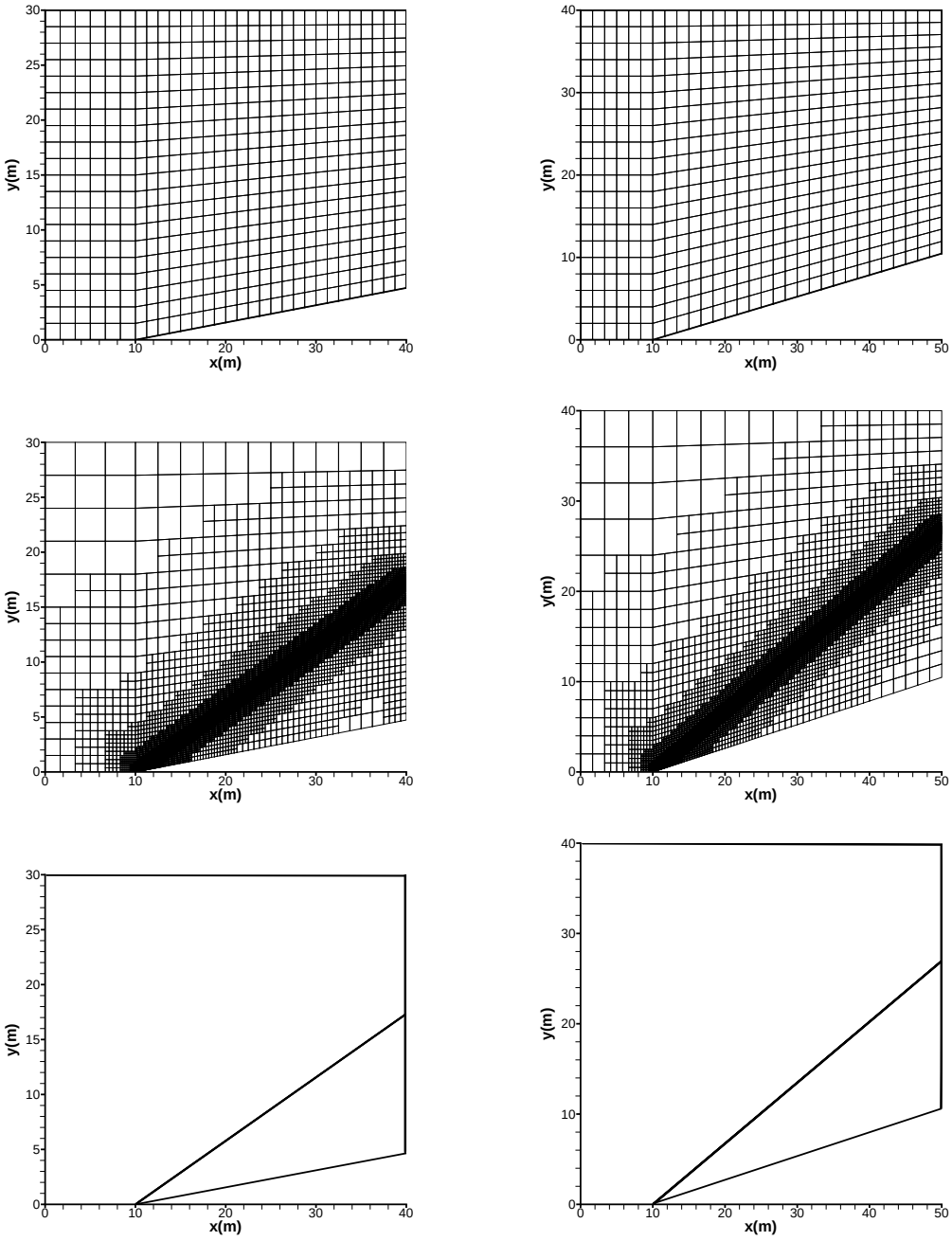

Figure 7. The oblique hydraulic jump problems. Left: $\theta = 8.95^o$. Right: $\theta = 15^o$. Top: initial grid, middle: final grid using 6 refinement levels, bottom: water depth.

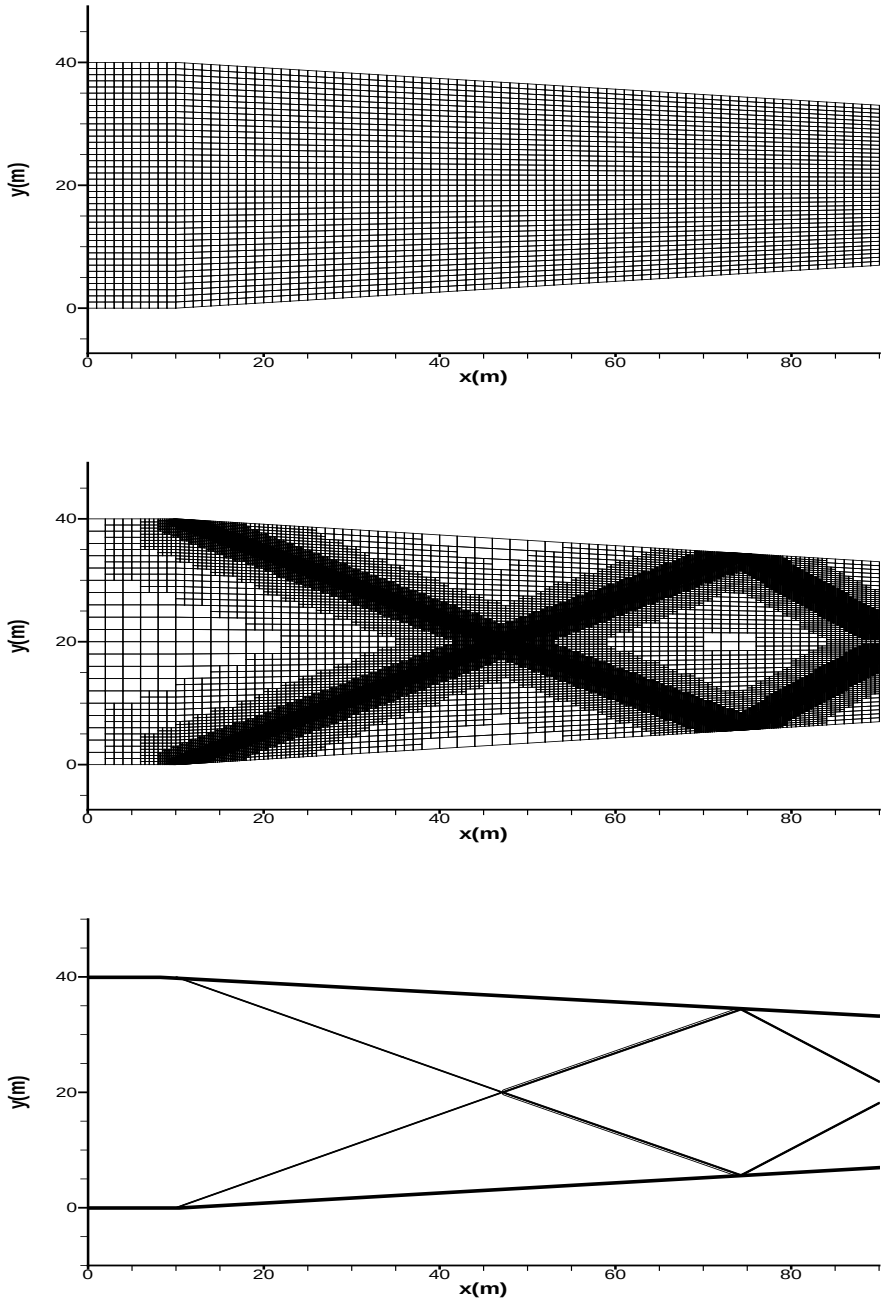

Figure 8. Computational grids for the channel constricted on both sides problem. *Top*: initial grid, *middle*: final grid using 5 refinement levels, *bottom*: contours of the water depth.

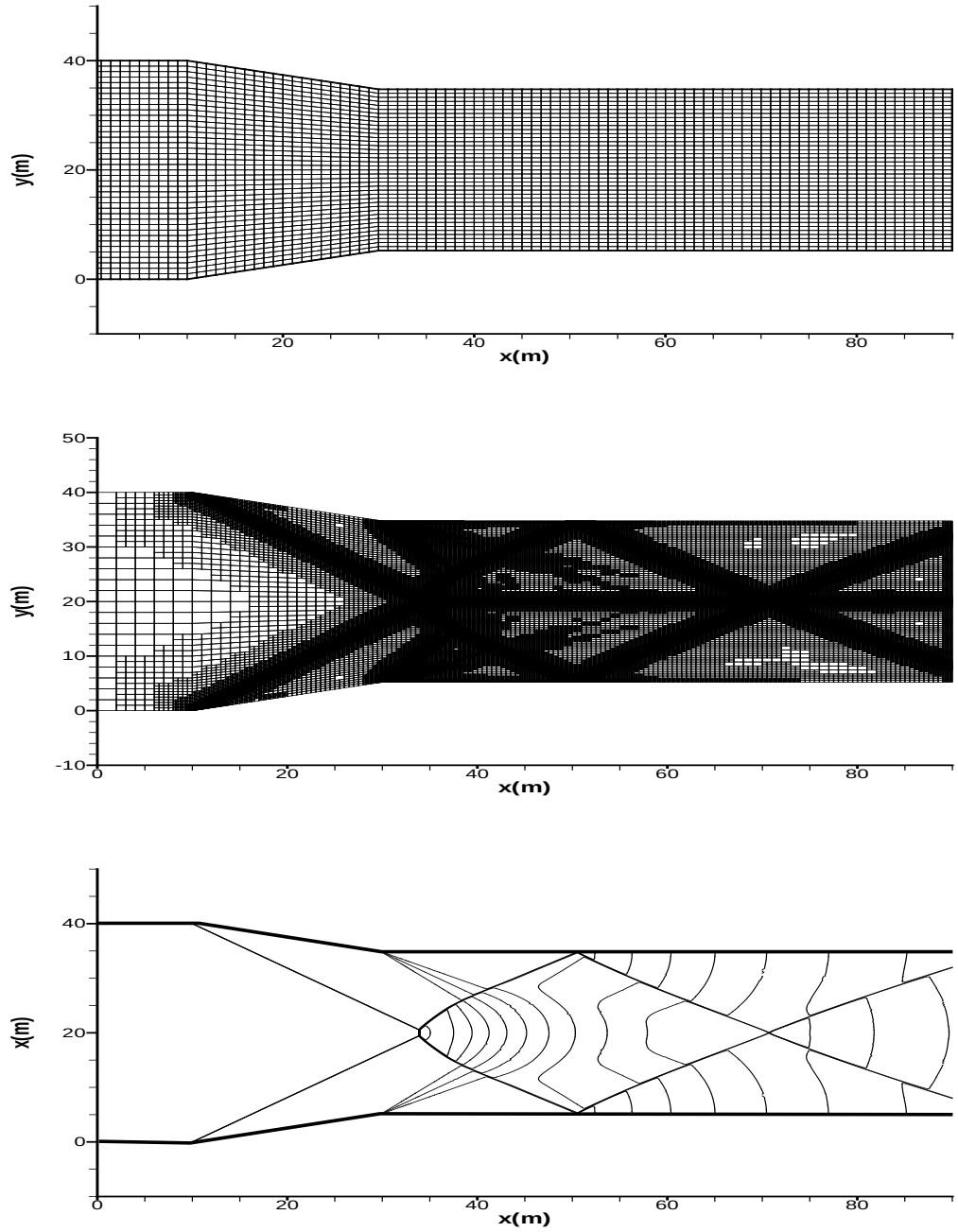

Figure 9. Contours of the water depth for the symmetry channel with variable width problem. *Top*: initial grid, *middle*: final grid using 5 refinement levels, *bottom*: contours of the water depth.

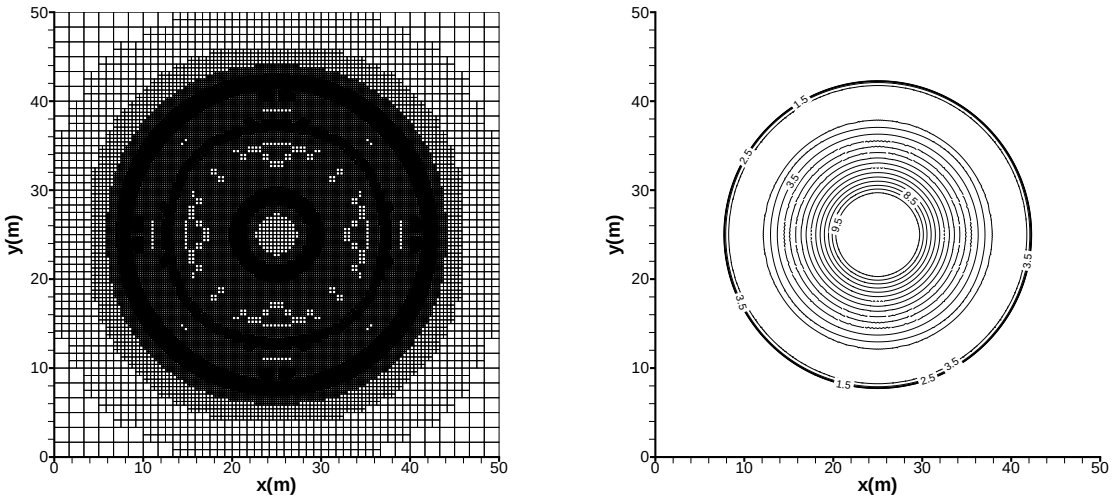

Figure 10. Circular dam break problem. *Left*: final grid using 6 refinement levels, *right*: contours of the water depth.

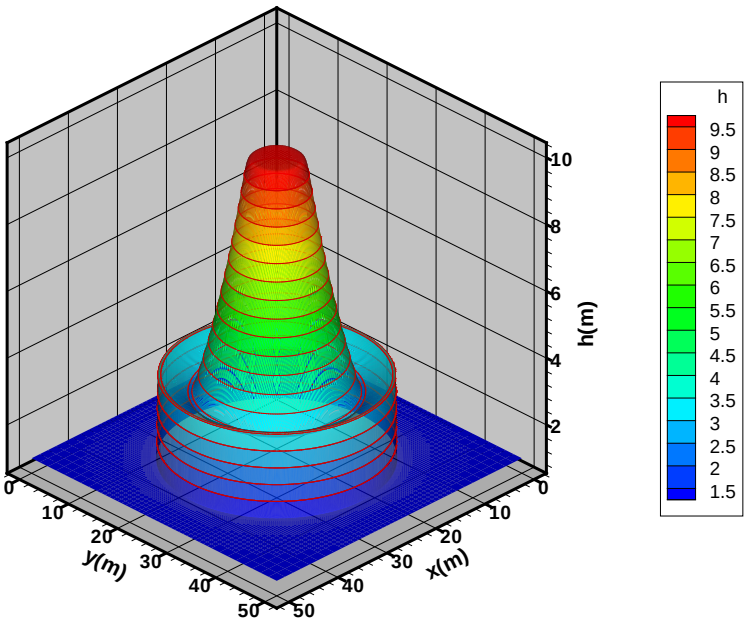

Figure 11. Water surface profile after breaking of circular dam.

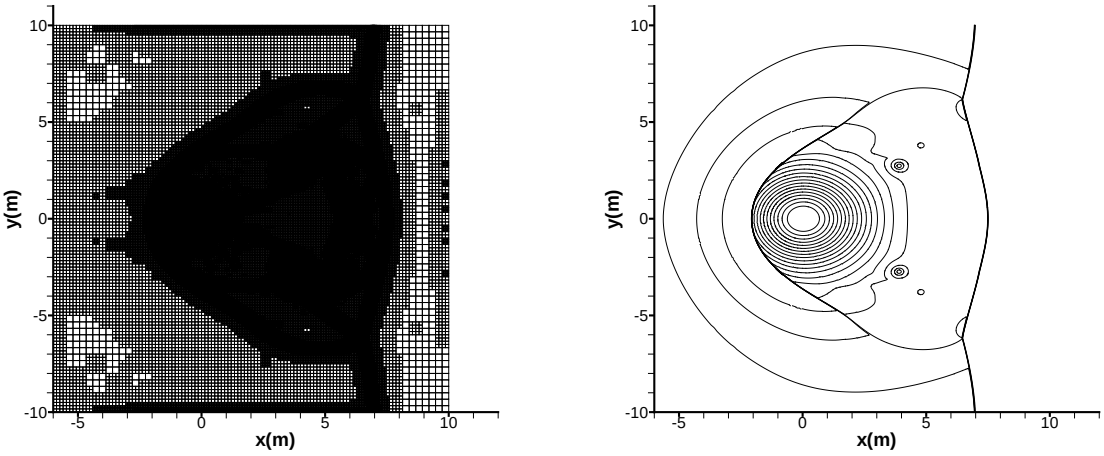

Figure 12. Bore wave past a hump problem. *Left*: final grid using 6 refinement levels, *right*: contours of the water depth.

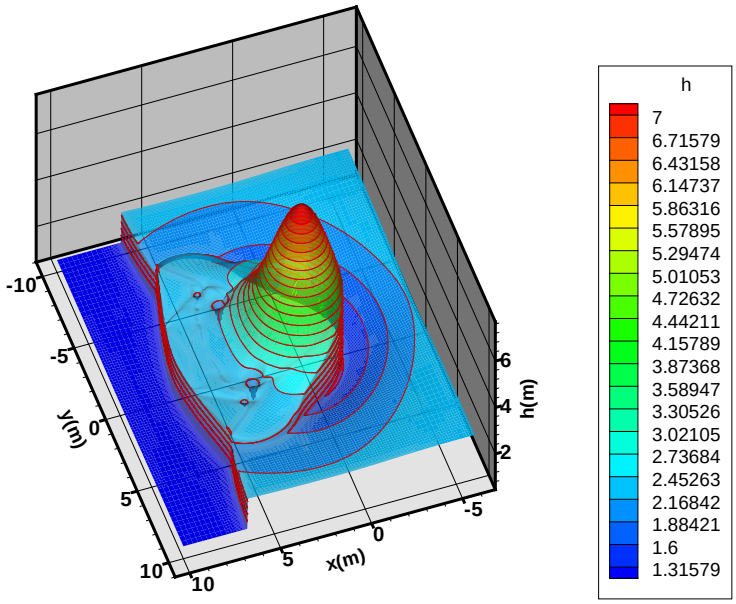

Figure 13. Water surface profile.

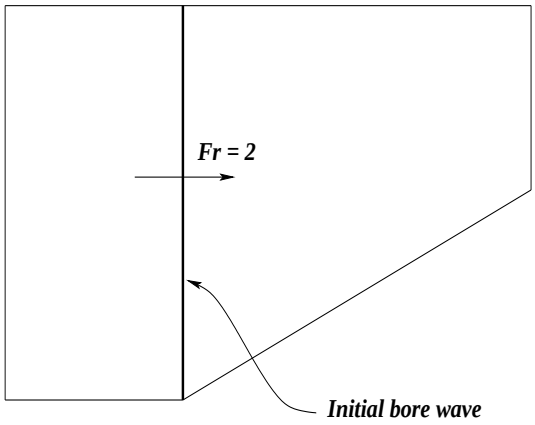

Figure 14. Bore reflection problem.

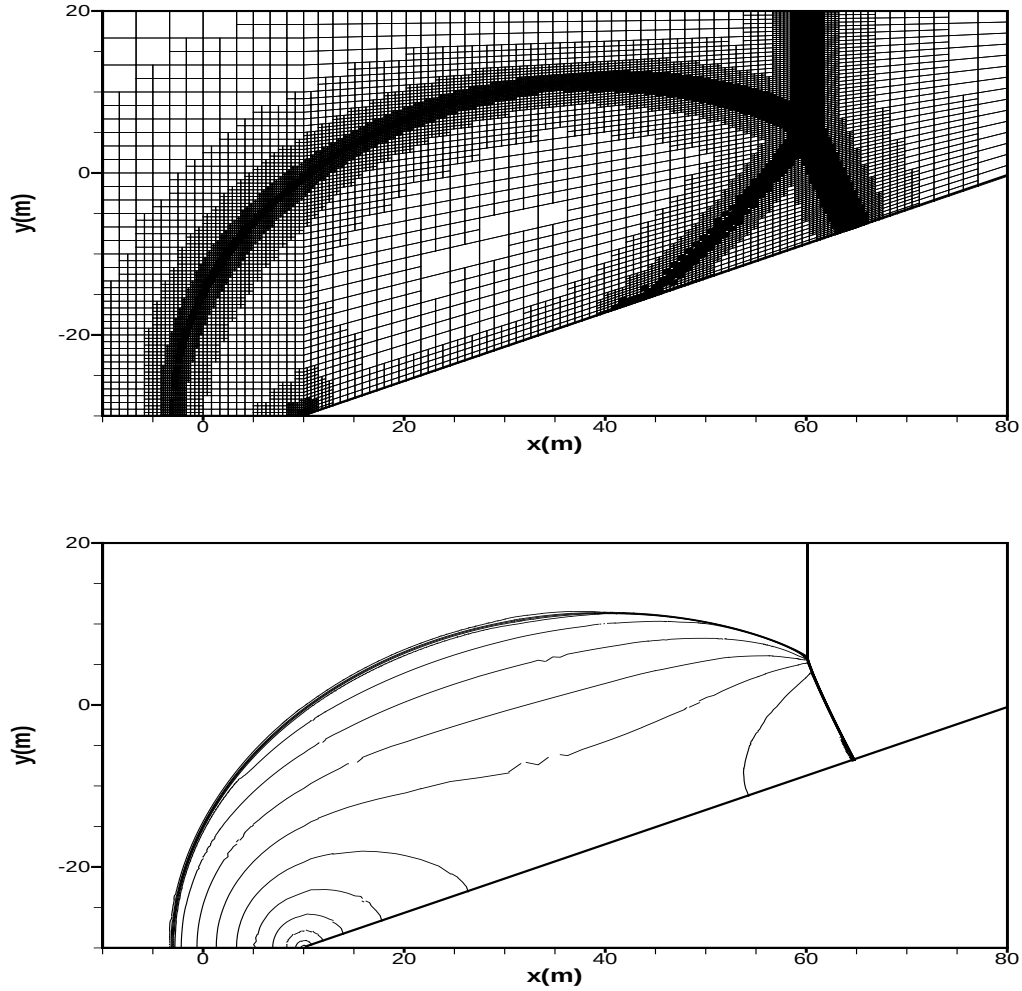

Figure 15. Bore reflection problem. *Top*: final grid using 6 refinement levels, *bottom*: contours of the water depth.

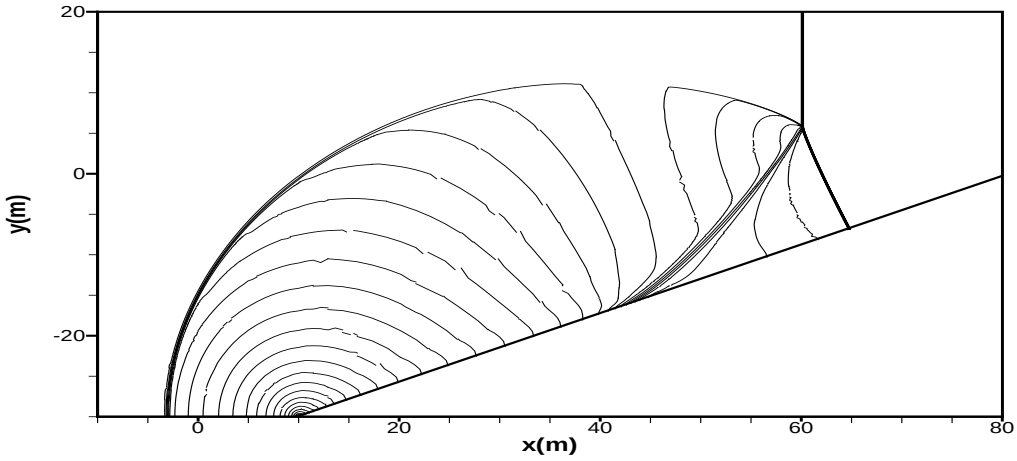

Figure 16. Bore reflection problem: contours of the x-velocity.

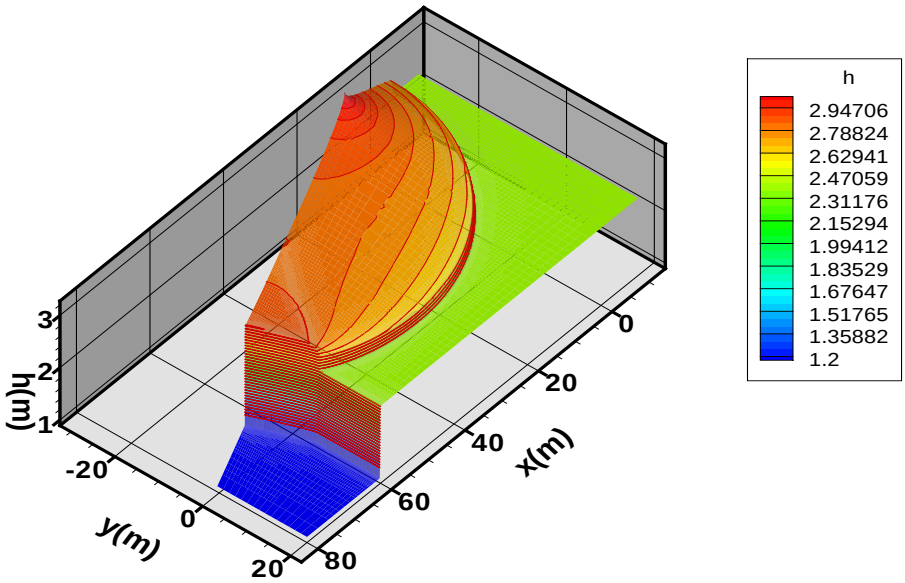

Figure 17. The water surface profile for the bore reflection problem.