A well-balanced numerical scheme for shallow water simulation on adaptive grids

HJ Zhang¹, JZ Zhou*, S Bi¹, QQ Li², Y Fan¹

¹School of Hydropower and Information Engineering, Huazhong University of science and Technology, Wuhan 430074, P R Chia

Email: Zhanghuajie@hust.edu.cn (HJ Zhang); jz.zhou@hust.edu.cn (JZ Zhou)

²Changjiang River Scientific Research Institute, Wuhan, 430010, P R China

ABSTRACT The efficiency of solving two-dimensional shallow-water equations (SWEs) is vital for simulation of large-scale flood inundation. For flood flows over real topography, local high-resolution method, which uses adaptable grids, is required in order to prevent the loss of accuracy of the flow pattern while saving computational cost. This paper introduces an adaptive grid model, which uses an adaptive criterion calculated on the basis of the water lever. The grid adaption is performed by manipulating subdivision levels of the computation grids. As the flow feature varies during the shallow wave propagation, the local grid density changes adaptively and the stored information of neighbor relationship updates correspondingly, achieving a balance between the model accuracy and running efficiency. In this work, a well-balanced (WB) scheme for solving SWEs is introduced. In reconstructions of Riemann state, the definition of the unique bottom elevation on grid interfaces is modified, and the numerical scheme is pre-balanced automatically. By the validation against two idealist test cases, the proposed model is applied to simulate flood inundation due to a dam-break of Zhanghe Reservoir, Hubei province, China. The results show that the presented model is robust and well-balanced, has nice computational efficiency and numerical stability, and thus has bright application prospects.

1 Introduction

The two-dimensional Shallow-Water Equations (2D SWEs) are widely used in simulating long wave flow phenomenon, where the movement in vertical direction is so small that can be neglected, and the flow can be assumed to be horizontal. Godunov type finite volume scheme associated with Riemann solvers has a good property for capturing shocks that is essential to preserve discontinuous. However, the accuracy of numerical solution in space mainly depends on the scale of cell size, and in order to gain high resolution results in areas of interest (AOI), tiny grid must be used, leading to large cell numbers and high computational cost. In most problems, flow features vary sharply in small area and the solutions may be relatively smooth in the rest of the computation domain. Hence it is necessary to design a grid structure that involves fine grids in AOI and coarsens grids in rest areas so as to obtain the balance between the accuracy of numerical solutions and computational cost.

The adaptive mesh refinement (AMR) is widely used to seek high spatial resolutions by organizing the computational grid in tree-like structure. One coarse grid is subdivided into small grids when

*Corresponding author. Tel.: +86 02787543127.
confronting a sharp slope of water surface. The most well known structured adaptive refinement grid model was developed initially by Berger [1] and later improved by many researchers [2-4, 7]. And many Godunov-type numerical schemes are developed to suit for adaptive quad-tree grids, e.g. [5, 6].

Recently, considerable effort has been made on incorporate the balance of discrete design of flux and source terms. Such scheme are termed well-balanced (WB), and are carefully designed to prevent an unreasonable flow due to spatial-varying topography. Researchers [8-9] have developed different well-balanced schemes using additional source terms. These extra terms makes the schemes well-balanced, but increase computational demands and reduces the running efficacy.

This paper aims to introduce a simplified and fast strategy for solving 2D SWEs with a WB scheme. Coarse rectangular grids fill the computational domain initially, and fine grids are the child grids of coarse grids and are attached according to the flow gradients. The additional terms are no longer needed under the modification of Riemann states reconstruction, simplifying the well-balanced scheme.

2 Dynamically adaptive grid model

Dynamically adaptive structured grid model is used for capturing wet/dry fronts, and free surface discontinuities. Each grid is a leaf-grid, or has four child-grids when been divided small to gain a higher resolution in space. This grid model is designed hierarchical and quadric, can be generated fast and automatically, and can be changed easily whether a grid is enriched into four or four child grids gather as one.

2.1 Grid generation

The grids are created in a recursive spatial decomposition of root grids according to following procedure:

Step 1 Generate root grids, and set the values of subdivision level to zero.

Step 2 Input seeding point data to describe the boundaries of the flow domain and any areas of interests, usually where the bed gradients are large.

Step 3 Enrich a grid, which contains seeding points, until the subdivision level reach a maximum level prescribed.

The grids generated in Step 1 are coarse and usually contain undesired grids where water flow never reaches. Therefore, seeding points are added to describe the irregular domain boundaries and gain a higher resolution in areas of interests, such as regions where initial water surface or bottom elevations have steep slopes. A primary root grid is then divided into four equal quadrant grids recursively, when it contains more than one seeding points. Boundary identification cut off regions, which are far away from the route of water flow, and minimizes the total number of cells and hence reduces memory storage requirement and overall computational cost.

2.2 Adaption criterion

For leaf grids, the grid adaption variable is set to zero in dry cells, and calculated in wet cells based on the gradient of free surface elevation

\[ \theta = \sqrt{\left(\frac{\partial \eta}{\partial x}\right)^2 + \left(\frac{\partial \eta}{\partial y}\right)^2} \]  

and for grids containing with sub-grids, \( \theta \) is set to the maximum of those of its child grids.

A grid is subdivided when \( \theta \) is greater than \( \theta_{\text{max}} \) and the subdivision level is less than \( L_{\text{max}} \), in which \( \theta_{\text{max}} \) and \( L_{\text{max}} \) are prescribe constants. Grid coarsening for wet grids is as follows: each grid with one-depth value is identified, and a Boolean status is set false if \( \theta \) is less than a predefined value \( \theta_{\text{min}} \). The efficiency of an ARM algorithm is related to the reliability of the mesh adaption procedure, which is controlled by the threshold parameters. However these threshold values of \( \theta_{\text{max}} \) and \( \theta_{\text{min}} \) are usually problem dependent, and are determined by trial and error.
2.3 Variable reassignment

For a shallow-water simulation over complex topography, bottom elevations distribute spatially, and therefore, it’s quite necessary to calculate bottom values of each child grids by interpolating from raw data of Digital Elevation Model (DEM) during grid enrichment. While the bed elevation changes, the flow values (especially $h$ and $\eta$) of child grids should be assigned properly to maintain the conservation of mass and momentums. The bottoms at grid centers do not usually equal to the average of those in child grids, thus the new water lever changes to prevent the violation of maintain mass conservation. Giving the initial value as $\eta_p$, water surface elevation alters according (2). The iterative computing procedure ends within four loops, when the increment of water lever is zero.

$$\Delta \eta = (\eta_p - b_p) - \frac{1}{4} \sum_i \max(\eta - b_i, 0)$$

$$\eta' = \eta + \Delta \eta \tag{2}$$

The coarsen procedure compute the flow variables of parent grid directly through conservation laws.

$$\eta = b_p + \frac{1}{4} \sum_i \max(h_i, 0) \tag{3}$$

3 Numerical model

3.1 Governing Equations

The 2D SWEs, which can be derived from the Navier-Stokes equations by average integrating over water depth, under the assumptions that a pressure distribution is hydrostatic and vertical accelerations can be ignored, are widely used for simulations of flood flows. These equations can be expressed in matrix form as follows.

$$\frac{\partial U}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = s_f + s_b \tag{4}$$

with

$$U = \begin{bmatrix} h \\ u \\ v \\ v 
\end{bmatrix}, f = \begin{bmatrix} u^2 + \frac{1}{2} gh^2 \\ uh \\ uvh \\ v^2 + \frac{1}{2} gh^2 
\end{bmatrix}, g = \begin{bmatrix} v \\ uh \\ v \\ u^2 + \frac{1}{2} gh^2 
\end{bmatrix}, s_f = \begin{bmatrix} 0 \\ -gS_{fx} \\ -gS_{fy} 
\end{bmatrix}, s_b = \begin{bmatrix} 0 \\ g S_{bx} \\ g S_{by} 
\end{bmatrix} \tag{5}$$

in which, $h$ is the water depth; $u$ and $v$ are the flow velocities in each Cartesian coordinate; $g$ represents the acceleration due to gravity, and depends on latitude of computation grid , with a default value of 9.81 m/s$^2$ as; $S_b$ and $S_f$ are bottom and frictional slops, respectively. Viscosity, wind effects and the Coriolis term can be included in the source term, but are neglected in this paper.

3.2 Finite volume method

Integrating (4) over a finite volume $\Omega$, the integral form of the SWEs obtained by applying Green theorem is given by

$$\frac{\partial}{\partial t} \int_{\Omega} Ud\Omega + \oint_{\partial \Omega} F \cdot n d\ell = \int_{\partial \Omega} s d\Omega \tag{6}$$

The integrand $F \cdot n$ is the outward normal flux vector, in which $F = [f, g]$. The finite volume method based on (6) ensures that the mass and momentum will be conserved across a discontinuity. Discretizing (6) in space and time, the equation becomes
\[ U_{n+1}^i = U^n_i + \Delta t \left( -\frac{f_E - f_W}{\Delta x} - \frac{f_N - f_S}{\Delta y} + s \right) \]  

(7)

where the superscript \( n \) and \( n+1 \) represent the time level. The normal flux \( f \) can be estimated in different approximate Riemann technologies, and the HLLC approximate Riemann solver associated with wave speed estimated by two rarefactions assumption [7] is suggested in following work.

3.3 Well-balanced scheme

A new WB scheme is proposed in this section. The WB scheme is obtained by modifying bottom reconstruction values of Riemann states on cell interfaces. In reconstruction, flow variables on interfaces are estimated with a non-linear slope limitation for a spatial second-order accuracy. The minmod slope limiter [7] is used for restrain the flow gradients, given by

\[ \Psi_{x,i} = \text{minmod}(U_{x,i} - U_{x,i+1}, U_{x,i} - U_{x,i-1}), \quad \Psi_{y,i} = \text{minmod}(U_{y,i} - U_{y,i+1}, U_{y,i} - U_{y,i-1}), \]  

and the interpolation is

\[ U_i(\Delta x, \Delta y) = U_i + \Psi_{x,i} \Delta x + \Psi_{y,i} \Delta y. \]  

Consider the computation of flux from cell \( i \) to its east cell \( e \) in one dimension, the interface values are given by

\[ \eta_L = \eta_i + \frac{1}{2} \Psi^{(0)}_{x,i}, h_L = h_i + \frac{1}{2} \Psi^{(0)}_{x,i}, b_L = \eta_L - h_L \]  

(8)

\[ \eta_R = \eta_i - \frac{1}{2} \Psi^{(0)}_{x,e}, h_R = h_i - \frac{1}{2} \Psi^{(0)}_{x,e}, b_R = \eta_R - h_R \]  

(9)

Based on the face values from (8) and (9), Riemann states are modified to ensure non-negative of water depth. A unique value of bottom elevation at the interface may be defined, and the water depth on either side are re-computed subsequently.

\[ b_i = \min(\eta_L, \eta_R, \max(b_L, b_R)) \]  

(10)

\[ h_i^* = \eta_L - \max(b_L, b_R), h_i^* = \eta_R - \max(b_L, b_R) \]  

(11)

Obviously, \( h_i^* \) and \( h_i^* \) given by the above procedure are non-negative. Correspondingly, the reconstruction values of water surface elevation and unit discharge is then given by

\[ \eta_i^* = h_i^* + b_i, \eta_i^* = h_i^* + b_i \]  

(12)

The single value of bottom elevation at interface \( b^* \), and the reconstruction values of \( \eta_i^*, \eta_i^* \) are recorded for later bed slope computation. The reconstruction procedure can be extended to other faces in a similar way. These Riemann states are then employed by the HLLC approximate Riemann solver to compute the interfaces fluxes \( f_E, f_W, f_N, \) and \( f_S \).

3.4 Bed slope source term

The bed slope source term is given by

\[ s_b = [0, gh_x \frac{\partial b}{\partial x}, gh_y \frac{\partial b}{\partial y}] \]  

where \( \eta \) is the average of the reconstructed values of water surface elevation on the two interfaces, and slopes of bottom elevation are directly discretized.

\[ gh_x \frac{\partial b}{\partial x} = g \cdot \frac{\frac{h_E^*}{2} + \frac{h_W^*}{2} + \frac{b_E^* - b_W^*}{\Delta x}}{2} \]  

(13)

The proof or well-balance property is straightforward. Assuming a steady state, the values of velocities \( u \) and \( v \) equal to 0 everywhere, It is easy to find the equivalence between convective flux terms and bottom source terms, by simply substituting (5) with (8)–(13). The scheme is pre-balanced, and no special treatments should be done on a wet/dry front.
3.5 Time marching

In the framework of Godunov-type finite volume scheme, the time marching formula is show as (7). However, it gives diffusive results with first-order accuracy in time. In this paper, predictor-corrector technique [5], which computes Riemann flux only once during one time-marching step, will be employed to achieve a second-order accuracy in time. In summary, the time marching process is as follows.

**Step 1** For all leaf cells, refine the grids according to $\theta$ values when $\theta$ larger than $\theta_{\text{max}}$.
**Step 2** For the grids, whose children are all leaf grids, remove the child grids when $\theta$ is smaller than $\theta_{\text{min}}$.
**Step 3** Generate an array to store the information of all leaf cells (for later use in Step 4~8).
**Step 4** Compute the gradients of flow variables.
**Step 5** Evaluate the update rate of conserved variables, and predict them to an intermediate time step.
**Step 6** Reconstruct Riemann states and estimate the fluxes for each interface.
**Step 7** Calculate source terms.
**Step 8** Correct flow variables of leaf grids and step to a new time step.
**Step 9** Update non-leaf cells and step to next time step.
**Step 9 Return to Step 1 until the process is terminated by $t \geq t_{\text{end}}$.

3.6 Stability criteria

The stability of the explicit numerical model is governed by the Courant-Friedrichs-Lewy (CFL) condition. For a two-dimension Cartesian grid, the CFL criterion for choosing an appropriate time step may be expressed as

$$\Delta t = C \cdot \min(\Delta t_x, \Delta t_y)$$  \hspace{1cm} (14)

with

$$\Delta t_x = \min \left\{ \frac{M_i}{|u_i| + \sqrt{gh_i}} \right\}, \Delta t_y = \min \left\{ \frac{N_i}{|v_i| + \sqrt{gh_i}} \right\}$$ \hspace{1cm} (15)

where $C$ is the Courant number in the range $0 < C \leq 1$. In the present scheme, an adaptive time step based upon (14) takes Courant number to 0.5. The overall time step is generally constrained by the smallest cells. Higher computational efficiency may be achieved by implementing a local time step [10-12].

4 Results and discussions

The numerical scheme is validated against several benchmark tests, and results are compared with analytical solutions. The test cases included preservation of still water, water sloshing over one-dimensional parabolic containers. In all cases, default values are specified as: $g=9.81 \text{m/s}^2$, $\theta_{\text{max}}=0.08$, $\theta_{\text{min}}=0.03$.

4.1 Preservation of still water over a two-dimensional bump

The shallow flow solver associated with adaptive refinement mesh is validated in an initially still water case to prove that the present model preserves the flow steady with wet/dry front over non-uniform bottom topography. The bump is a cone located at the center of a 2m×2m square container. The boundaries are solid walls, and no friction is imposed by the smooth bottom, which is defined by

$$b(x, y) = 0.8 \exp\left\{ -25 \times [(x-1)^2 + (y-1)^2] \right\} \text{m}$$ \hspace{1cm} (16)

The computational domain is approximated in 1600 rectangular cells ($dx=0.05\text{m}$), and no grid adaption is employed. Initially the water inside the container is at rest, and the bump is partially or
absolutely submerged. The steady state should be maintained, and velocities in the cells must be zero anywhere and anytime. Two initial situations are tested for the validation, and both are performed for 100s. In Case 1a, bump is partially submerged as the water surface elevation is 0.5m. In Case 1b, the water surface is 1.0m and the bump is absolutely under water. Figure 1 profiles the numerical solutions and analytical values for water depth along y=0m in Case 1a and 1b, and shows the steady state is well preserved, verifying the C-property of the numeric model.

Figure 1. Preservation of still water: comparison between numerical and analytical results of water surface profiled along y=0m.

4.2 Moving shorelines over a frictional parabolic bottom topography

In 2006, Sampson derived analytical solution of the SWEs for a perturbed flow in a container with a one-dimensional parabolic bed bottom with friction. This benchmark test is adopted to validate the stability of the numerical model with moving wet/dry front, as well as provide a further check for source terms, including bed friction and bed slopes. The bed of the domain is defined by

\[ b(x) = h_0(x/a)^2 \]  \hspace{1cm} (17)

in which, \( h_0 \) and \( a \) are constants. The bottom is uniform in \( y \)-direction. The analytical solution describes a bed resistance in parameter \( \tau \) (relation with bed friction is given by \( C_f = \mu \tau / \sqrt{u^2 + v^2} \), or \( S_f = ut \tau \) in an equivalent form) and a hump parameter \( p = (8gh_0/a^2)^{0.5} \). We consider a case where \( \tau < p \), and the analytical solution of the water stage is

\[ \eta(x,t) = h_0 - \frac{B \nu}{4 g} \left[ B \cos(st) + \frac{B \nu}{2} \sin(st) \right] x + \frac{a^2 B \nu}{8 g^2 h_0} \left[ -s \tau \sin(2st) + \left( \tau^2 / 4 - s^2 \right) \cos(2st) \right] \] \hspace{1cm} (18)

where \( B \) is a constant and \( s = 0.5 \times (p^2 - \tau^2)^{0.5} \). As \( t \) tends to infinite, the water level along the channel converges to \( h_0 \), defining the still water elevation above the datum. And the perturb flow is damped by bed friction.

The computation domain is 10000m×500m. Slip conditions are invoked at the lateral boundaries. The coefficients are \( a = 3 \) km, \( h_0 = 10 \) m, \( \tau = 0.001 \) s\(^{-1} \) and \( B = 5 \) m/s. Since the water surface is quite smooth in spatial, the grid adaption is not carried on, and the computation domain is filled with cells in uniform size of 100m×100m. The simulation lasts 6000s. Figure 2 shows a good agreement between the numerical and analytical solutions for water depth at observed points located at A(\( x = 50 \), \( y = 0 \)), B(\( x = -2750 \), \( y = 0 \)), C(\( x = 2750 \), \( y = 0 \)).
4.3 Application: flood wave propagation due to dam-break of Zhanghe Reservoir

Zhanghe River is a branch of Yangtze River. The river length is 202km, and the watershed area is about 400km$^2$. Zhanghe Reservoir was built maintaining huge water of nearly twenty billion tons. We will make a general analysis on flood propagation due to an imaginary dam-break.

The computational domain should contain both reservoir area and downstream river. However the dam-breach type is not defined uniquely, and the bathymetry of the reservoir is not available, therefore we only concern the river valley and floodplain on the downstream side of the reservoir. The computational domain is about 129km$^2$, and five survey points along the river.

The discharge at the dam site must be predefined, and a suggested inflow discharge plan is obtained by using empirical formulation. The flood water flushing from reservoir into the domain for 4.5h. The discharge of tributaries is relatively small and can be neglected during the flood inundation, thus the lateral boundaries are set to slip wall condition. And the downstream boundary is set to be free out.

Since the domain contains both narrow river valley and relatively wide floodplain, the domain is suggested to be discretized into coarse cells in floodplain and fine grids in river valley. The max cell size cannot be too large, otherwise river valley is unable for modelling. Considering that the average width of river is 300m, we suggest 100m as the max cell size and 25m as the smallest size.

The simulation runs in different cases, fixed fine mesh, and dynamic mesh. Table 1 shows the detail configurations and computational cost of time in each case. In Case 3a, the domain is discretization by the finest grids, no cell-density change is made, and the numerical results for water depth can be considered relatively rea, and thus be used as reference values in the subsequent comparison. In Case 3b, configuration of the 100m×100m coarse cells with 2 max subdivision level is employed, and the grid dynamically varies according the adaptive criterion in Section 2.2.

The number of leaf cells in Case 3b increases gradually from 13,689 to 44,919, and is much smaller than the fixed cells numbers in Case 3a, which is 206,903. The relative time $t_r$ calculated by $t_r(i)=t_{CPU(i)}/t_{CPU(a)}$, is shown in Table 1, indicating an improvement in computational efficiency of time.
by the local refined grids. The AMR algorithm saves 60% in time, gaining an excellent running efficiency.

Table 1. Grid configuration of dam-break wave propagation over Zhanghe floodplain

| Case | AMR mode | cell size /m | $L_{\text{max}}$ | cells number | $t_{\text{CPU}}$ / s | $t_1$  |
|------|-----------|--------------|-----------------|--------------|-----------------|--------|
| 3a   | static    | 25           | --              | 206,903      | 21,307          | 100.0% |
| 3b   | dynamic   | 100          | time varying    | 8,319        | 39.0%           |        |

Figure 3 and Figure 4 compares numerical solutions for water depth and unit discharge at survey points respectively. The comparisons show a general agreement between the three cases. The information of flood arrival time and peak time is easily found from Figure 3. Flood waves arrive Point A and B soon after the dam-break, and a sharp rise in water depth is observed at these points. Due to a tributary flowing into Zhanghe River from the left bank, the area of debouchment can store amount of water during the flood propagation. The Point D and E, which locate at the downstream of the debouchment, has later arrival time and smaller peak values in unit width discharge, reflecting an attenuate peak discharge along the floodplain.

One phenomenon is observed at Point E, that as adaptive mesh refinement technology is employed, the flood arrives about 5 minutes earlier in the hydrograph line, yet the curve of unit width discharge arises nearly 8 minutes later, in contrast with fixed fine mesh. An explanation is that flood information transfers downstream faster in adaptive refined cells. However, water moves at a similar speed behind the wave front in different cases, herein the rise of unit width discharge lags. In order to simulate the flood propagation more accurately, the mesh adaption criterion is worth studying in future work.

**Figure 3.** Hydrograph of observing points during dam-break wave propagation.
5 Conclusions

A well-balanced scheme is developed in this paper. The presented scheme modifies the bed elevations at cell interfaces during the reconstructions of Riemann states, and need no special treatment on source terms, thus makes the numeric procedure easier to actualize. Adaptive grid model is introduced and well designed for seeking local high-resolution, reducing overall computational cost, and improving the running efficiency of large-scale shallow-water simulation over irregular topography.

The well-balanced property of proposed model is tested in two test cases of the preservation of still water with or without wet/dry fronts. Besides, the numerical model is validated against another two theoretical benchmark tests and a laboratory experiment. The simulated predictions agree excellently with analytical solutions or observed values in the validation tests. Finally, the model is applied to Zhanghe Reservoir, where a flood propagation event occurs due to an imaginary dam-break. Comparisons between different AMR strategies are made, and show that the AMR algorithm improves overall efficiency of numeric computations, and can account for flood propagation over highly irregular topography. Therefore the proposed model is suitable for problems with complex topography and thus has bright application prospects.

Acknowledgement

This work was financially supported by a grant from the Key Program of the National Natural Science Foundation of China (No. 51239004); the Research Fund for the Doctoral Program of Higher Education of China (No.20100142110012).

Reference

[1] Berger MJ and Oliger J 1984 Adaptive mesh refinement for hyperbolic partial differential equations. J. Comput. Phys. 53 484-512

[2] Rogers B, Fujihara M and Borthwick AGL 2001 Adaptive Q-tree Godunov-type scheme for shallow water equations Int. J. Numer. Methods Fluids 35 247-280
[3] Liang Q, Borthwick AGL and Stelling G 2004 Simulation of dam- and dyke-break hydrodynamics on dynamically adaptive quadtree grids *Int. J. Numer. Methods Fluids* **46** 127-162

[4] Baeza A and Mulet P 2006 Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations *Int. J. Numer. Methods Fluids* **52** 455-471

[5] Liang Q 2012 A simplified adaptive Cartesian grid system for solving the 2D shallow water equations *Int. J. Numer. Methods Fluids* **69** 442-458

[6] Ji H, Lien F and Yee E 2010 A new adaptive mesh refinement data structure with an application to detonation *J. Comput. Phys.* **229** 8981-8993

[7] Song L, Zhou J, Guo J, Zou Q, and Liu Y 2011 A robust well-balanced finite volume model for shallow water flows with wetting and drying over irregular terrain *Adv. Water Resour.* **34** 915-932

[8] Liang Q and Marche F 2009 Numerical resolution of well-balanced shallow water equations with complex source terms *Adv. Water Resour.* **32** 873-884

[9] An H and Yu S 2012 Well-balanced shallow water flow simulation on quadtree cut cell grids *Adv. Water Resour.* **39** 60-70

[10] Crossley AJ and Wright NG 2005 Time accurate local time stepping for the unsteady shallow water equations *Int. J. Numer. Methods Fluids* **48** 775-799

[11] Krámer T and Józsa J 2007 Solution-adaptivity in modelling complex shallow flows *Comput. Fluids* **36** 562-577

[12] Sanders BF 2008 Integration of a shallow water model with a local time step *J. Hydraul. Res.* **46** 466-475