INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **44**:789–810 (DOI: 10.1002/fld.681)

Solution of the shallow-water equations using an adaptive moving mesh method

Huazhong Tang*,†

LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China

SUMMARY

This paper extends an adaptive moving mesh method to multi-dimensional shallow water equations (SWE) with source terms. The algorithm is composed of two independent parts: the SWEs evolution and the mesh redistribution. The first part is a high-resolution kinetic flux-vector splitting (KFVS) method combined with the surface gradient method for initial data reconstruction, and the second part is based on an iteration procedure. In each iteration, meshes are first redistributed by a variational principle and then the underlying numerical solutions are updated by a conservative-interpolation formula on the resulting new mesh.

Several test problems in one- and two-dimensions with a general geometry are computed using the proposed moving mesh algorithm. The computations demonstrate that the algorithm is efficient for solving problems with bore waves and their interactions. The solutions with higher resolution can be obtained by using a KFVS scheme for the SWEs with a much smaller number of grid points than the uniform mesh approach, although we do not treat technically the bed slope source terms in order to balance the source terms and flux gradients. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: shallow water equations; kinetic flux-vector splitting scheme; adaptive grid method; surface gradient method

1. INTRODUCTION

The shallow-water equations (SWE) have wide applications in ocean and hydraulic engineering to describe the hydraulic jump, bore wave propagation, tidal flows in estuary and coastal water regions, and open channel flows, among others. Many numerical methods have been successfully developed for the SWEs. However, the inclusion of source terms relevant to the

*Correspondence to: H. Tang, School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China.
†E-mail: hztang@math.pku.edu.cn

Contract/grant sponsor: Special Funds for Major State Basic Research Projects of China
Contract/grant sponsor: Natural Science Foundation of China

Received 17 March 2003
Revised 15 October 2003

bed topography and the shear stress is often necessary in the study of realistic problems. In order to predict accurately a hydraulic jump, it is generally necessary and efficient to use the well-balanced methods, which employ a special treatment for the bed slope source terms that balanced the source terms and flux gradients, when the mesh is not very fine. Several authors have conducted this study, see References [1–8] and references therein. Zhou *et al.* [8] developed a robust and well-balanced scheme based on a Godunov-type method and the surface gradient method for the initial data reconstruction. The merit of their approach is that their initial reconstruction is based on the water surface level instead of the water depth. Xu [7] proposed a well-balanced kinetic scheme for the SWEs with a geometrical source term. In other methods, the schemes are also nicely designed based on the requirement of preserving the equilibria for the shallow water equations.

Adaptive moving mesh methods have important applications in fluid dynamics, see References [9–14] and others. The main reason is that the physical problems in this field usually develop discontinuities or nearly singular solutions in fairly localized regions, such as hydraulic jumps or shock waves, detonation waves, contact discontinuities, and boundary layers, etc. There has been important progress in moving mesh methods for partial differential equations (PDE) in the past several decades, including the variational approach of Winslow [15], Brackbill *et al.* [16, 17], Dvinsky [18], and Li *et al.* [19]; moving finite element methods of Millers [20], and Davis and Flaherty [21]; moving mesh PDEs of Cao *et al.* [22].

The purpose of this work is to extend our adaptive moving mesh method proposed in Reference [14] to the SWEs with source terms. Our moving mesh algorithm includes two parts: the PDEs evolution and the mesh redistribution. The PDEs solver may be any appropriate high resolution finite volume scheme on arbitrary meshes and the mesh redistribution is an iterative procedure. In each time iteration, grid points are first redistributed by a variational principle, and then the numerical solutions are updated on the resulting new meshes by a *conservative–interpolation* formula. In the present work, a traditional kinetic flux-vector splitting (KFVS) scheme will be used in combination with a surface gradient method for the data reconstruction, because of its simplicity. The KFVS scheme is well-designed based on gas-kinetic theory, and has been applied successfully in unsteady compressible flow simulations, see Reference [23] and references therein. Further numerical computations will be used to demonstrate that the adaptive moving mesh method is very efficient at obtaining accurate approximations for both unsteady and steady flow situations, even though we do not employ an additional well-balanced treatment and a much finer mesh.

The paper is organized as follows. In Section 2, we present the high resolution KFVS scheme for two-dimensional SWEs, based on the surface gradient method for the initial data interpolation. The adaptive moving mesh algorithm will be described in Section 3. More numerical experiments will be given in Section 4. Finally, we present conclusions in Section 5.

2. THE KINETIC FLUX-VECTOR SPLITTING METHODS

The shallow-water equations in two-dimensions (2D) can be written as

$$\frac{\partial U}{\partial t}+\frac{\partial F(U)}{\partial x}+\frac{\partial G(U)}{\partial y}=S(\mathbf{x},U) \tag{1}$$

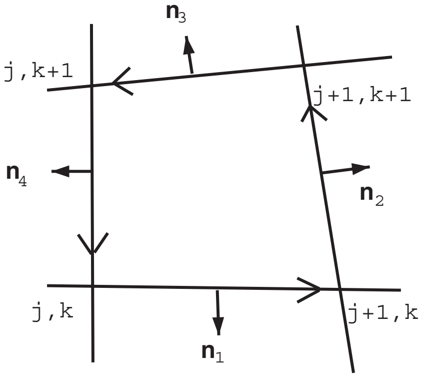

Figure 1. A 2D finite control volume $A_{J,K}$.

where

$$U = \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix}, \quad F = \begin{pmatrix} hu_x \\ hu_x^2 + \frac{1}{2}gh^2 \\ hu_x u_y \end{pmatrix}, \quad G = \begin{pmatrix} hu_y \\ hu_y u_x \\ hu_y^2 + \frac{1}{2}gh^2 \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ -gh\dfrac{\partial B}{\partial x} \\ -gh\dfrac{\partial B}{\partial y} \end{pmatrix} \tag{2}$$

and $\mathbf{x} = (x, y)$, g is the acceleration due to gravity, h is the total water depth, i.e. the height of the water above the wavy bottom characterized by the function $B(\mathbf{x})$ and $\mathbf{u} = (u_x, u_y)$ is the velocity vector. Here we have neglected the Coriolis force, the viscous stress, and the bed and wind shear stress.

Assume a partition $\{A_{J,K}\}$ of the physical domain $\Omega_{\rm p}$ is given, where $A_{J,K}$ is a quadrangle with four vertices $\mathbf{x}_{j+p,k+q}$, $0 \leqslant p$, $q \leqslant 1$, as shown in Figure 1, and the initial data is approximated by

$$U^0_{J,K} = \frac{1}{|A_{J,K}|} \int_{A_{J,K}} U_0(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{3}$$

where $|A_{J,K}|$ denotes area of the control volume $A_{J,K}$.

Integrating the SWEs (1) with respect to d$\mathbf{x}$ in a spatial element A may give

$$\frac{\partial}{\partial t}\int_A U\,\mathrm{d}\mathbf{x} + \int_{\partial A}(Fn_x + Gn_y)\,\mathrm{d}s = \int_A S(\mathbf{x}, U)\,\mathrm{d}\mathbf{x} \tag{4}$$

where ∂A denotes boundary surface of the control volume A, and $\mathbf{n} = (n_x, n_y)$ is the unit outward normal vector of the surface element ds. The finite volume scheme approximating the SWEs (1) will be developed based on the integral equation (4). The problem is how to give any appropriate approximation of the flux $F_{\mathbf{n}} = Fn_x + Gn_y$ on the surface ds.

If taking the quadrangle $A_{J,K}$ as a finite control volume, then we have

$$\frac{\partial}{\partial t}\int_{A_{J,K}} U\,\mathrm{d}\mathbf{x} + \sum_{l=1}^{4}\int_{s^l_{J,K}} F_{\mathbf{n}^l}\,\mathrm{d}s = \int_{A_{J,K}} S(\mathbf{x},U)\,\mathrm{d}\mathbf{x} \tag{5}$$

In our computations of the SWEs (1), the flow variables are updated according to

$$\begin{aligned} U^{n+1}_{J,K} = U^n_{J,K} - \frac{\Delta t_n}{|A_{J,K}|}\Bigg(&\sum_{l=1}^{4} F^+_{\mathbf{n}^l}(U^n_{J,K})|s^l_{J,K}| + F^-_{\mathbf{n}^1}(U^n_{J,K-1})|s^1_{J,K}| \\ &+F^-_{\mathbf{n}^2}(U^n_{J+1,K})|s^2_{J,K}| + F^-_{\mathbf{n}^3}(U^n_{J,K+1})|s^3_{J,K}| \\ &+F^-_{\mathbf{n}^4}(U^n_{J-1,K})|s^4_{J,K}|\Bigg) + \Delta t_n S^{n+1/2}_{J,K} \end{aligned} \tag{6}$$

where the source term $S^{n+1/2}_{J,K}$ is defined as

$$S^{n+1/2}_{J,K} = -\frac{g}{2}(h^n_{J,K} + h^{n+1}_{J,K})\begin{pmatrix} 0 \\ \dfrac{\partial B}{\partial x}(\mathbf{x}_{J,K}) \\ \dfrac{\partial B}{\partial y}(\mathbf{x}_{J,K}) \end{pmatrix} \tag{7}$$

and

$$\begin{aligned} |s^1_{J,K}|\mathbf{n}^1 &= (y_{j+1,k} - y_{j,k}, x_{j,k} - x_{j+1,k}) \\ |s^2_{J,K}|\mathbf{n}^2 &= (y_{j+1,k+1} - y_{j+1,k}, x_{j+1,k} - x_{j+1,k+1}) \\ |s^3_{J,K}|\mathbf{n}^3 &= (y_{j,k+1} - y_{j+1,k+1}, x_{j+1,k+1} - x_{j,k+1}) \\ |s^4_{J,K}|\mathbf{n}^4 &= (y_{j,k} - y_{j,k+1}, x_{j,k+1} - x_{j,k}) \end{aligned}$$

Although scheme (6) is implicit, we may solve it explicitly.

The kinetic flux-vector splitting scheme for the SWEs is derived by solving a collisionless Boltzmann equation

$$\frac{\partial f}{\partial t} + v_x\frac{\partial f}{\partial x} + v_y\frac{\partial f}{\partial y} = 0 \tag{8}$$

where f denotes the velocity distribution function, and $\mathbf{v} = (v_x, v_y)$ denotes the particle velocity vector. The idea of the construction of the scheme is similar to that used by Pullin [24] who studied the compressible Euler equations. For the SWEs (1) without the source terms, the (local) equilibrium state may be taken as [7]

$$f = h\left(\frac{\lambda}{\pi}\right)^{1/2} \mathrm{e}^{-\lambda(\mathbf{v}-\mathbf{u})^2} \tag{9}$$

where $(\mathbf{v}-\mathbf{u})^2=(v_x-u_x)^2+(v_y-u_y)^2$, and $\lambda=1/gh$. The connection between the (local) equilibrium state f and the macroscopic flow variables and the corresponding flux is

$$\begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix} = \iint_{R^2} \begin{pmatrix} 1 \\ v_x \\ v_y \end{pmatrix} f\,\mathrm{d}\mathbf{v} \tag{10}$$

and

$$F_{\mathbf{n}^l} = \begin{pmatrix} hu_{\mathbf{n}^l} \\ hu_x u_{\mathbf{n}^l} + \frac{1}{2}\,gh^2_{n^l_x} \\ hu_y u_{\mathbf{n}^l} + \frac{1}{2}\,gh^2_{n^l_y} \end{pmatrix} = \iint_{R^2} v_{\mathbf{n}^l} \begin{pmatrix} 1 \\ v_x \\ v_y \end{pmatrix} f\,\mathrm{d}\mathbf{v} = \iint_{R^2} v_{\mathbf{n}^l} \begin{pmatrix} 1 \\ v_x \\ v_y \end{pmatrix} f_{\mathbf{n}^l}\,\mathrm{d}v_{\mathbf{n}^l}\,\mathrm{d}v_{\mathbf{t}^l} \tag{11}$$

In the above, $u_{\mathbf{n}^l}$ and $u_{\mathbf{t}^l}$ (or $v_{\mathbf{n}^l}$ and $v_{\mathbf{t}^l}$) denote the depth-averaged (or particle) velocity components normal and tangential to the boundary s^l, computed by

$$u_{\mathbf{n}^l} = \cos(\alpha^l)u_x + \sin(\alpha^l)u_y, \quad u_{\mathbf{t}^l} = -\sin(\alpha^l)u_x + \cos(\alpha^l)u_y$$

where α^l is the angle between the normal direction $\mathbf{n}^l$ and the x-co-ordinate, measured in the anti-clockwise direction. Suppose that the initial data $W^{\mathrm{T}}=(h+B,\mathbf{u})(\mathbf{x},0)$ are given as a piecewise constant over the cell $A_{J,K}$, and let

$$(f_{\mathbf{n}^l})_{J,K} \equiv f(\mathbf{x}_{J,K},\mathbf{v}) = h_{J,K}\left(\frac{\lambda_{J,K}}{\pi}\right)^{1/2} \mathrm{e}^{-\lambda_{J,K}(v_{\mathbf{n}^l}-(u_{\mathbf{n}^l})_{J,K})^2-\lambda_{J,K}(v_{\mathbf{t}^l}-(u_{\mathbf{t}^l})_{J,K})^2} \tag{12}$$

be a Maxwellian distribution in the cell $A_{J,K}$, where $\lambda_{J,K}=1/gh_{J,K}$. For the collisionless scheme, the initial Maxwellian inside each cell corresponding to the surface s^l is

$$f_0(\mathbf{x},\mathbf{v}) = (f_{\mathbf{n}^l})_{J,K}, \quad \mathbf{x}\in A_{J,K} \tag{13}$$

In the evolution stage, the initial data are propagated according to the collisionless Boltzmann equation (8), and the solution of the equation is

$$f(\mathbf{x},t,\mathbf{v}) = \begin{cases} (f_{\mathbf{n}^l})_{J,K} & \text{if } v_{\mathbf{n}^l}>0 \\ (f_{\mathbf{n}^l})_{J+p,K+q} & \text{if } v_{\mathbf{n}^l}<0 \end{cases} \tag{14}$$

for $\mathbf{x}\in s^l$ and $t_n \leqslant t < t_{n+1}$, where $(p,q)=(0,-1)$, $(1,0)$, $(0,1)$, and $(-1,0)$ corresponding to $l=1,2,3,4$, respectively. Using formula (11), we obtain the splitting fluxes

$$F^{\pm}_{\mathbf{n}^l} = \int_{\mathrm{R}}\int_{R^{\pm}} v_{\mathbf{n}^l} \begin{pmatrix} 1 \\ v_x \\ v_y \end{pmatrix} g_{\mathbf{n}^l}\,\mathrm{d}v_{\mathbf{n}^l}\,\mathrm{d}v_{\mathbf{t}^l} = h \begin{pmatrix} [v^1_{\mathbf{n}^l}]^{\pm} \\ \cos(\alpha^l)[v^2_{\mathbf{n}^l}]^{\pm} - \sin(\alpha^l)[v^1_{\mathbf{n}^l}]^{\pm}[v^1_{\mathbf{t}^l}] \\ \sin(\alpha^l)[v^2_{\mathbf{n}^l}]^{\pm} + \cos(\alpha^l)[v^1_{\mathbf{n}^l}]^{\pm}[v^1_{\mathbf{t}^l}] \end{pmatrix} \tag{15}$$

with

$$[v^1_{\mathbf{n}^l}]^{\pm} = \frac{u_{\mathbf{n}^l}}{2}\,\mathrm{erfc}\left(\mp\sqrt{\lambda}u_{\mathbf{n}^l}\right) \pm \frac{\mathrm{e}^{-\lambda u^2_{\mathbf{n}^l}}}{2\sqrt{\pi\lambda}} \tag{16}$$

$$[v^2_{\mathbf{n}^l}]^{\pm} = \left(\frac{u^2_{\mathbf{n}^l}}{2} + \frac{1}{4\lambda}\right)\mathrm{erfc}\left(\mp\sqrt{\lambda}u_{\mathbf{n}^l}\right) \pm \frac{u_{\mathbf{n}^l}\mathrm{e}^{-\lambda u^2_{\mathbf{n}^l}}}{2\sqrt{\pi\lambda}} \tag{17}$$

$$[v^1_{\mathbf{t}^l}] = u_{\mathbf{t}^l}, \quad \mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} \mathrm{e}^{-t^2}\,\mathrm{d}t \tag{18}$$

Remark 2.1

We have proved in Reference [5] that scheme (6) is positivity-preserving in one-dimensional case under a suitable CFL condition, i.e. $h^{n+1} \geqslant 0$ if $h^n \geqslant 0$, but it is not well-balanced. Numerical results also demonstrated these properties there. In this paper, we will prove that the above scheme combined with the adaptive moving mesh algorithm can give solutions at a higher resolution, even though the geometrical source terms have been included.

The spatial accuracy of scheme (6) can be improved by using the initial reconstruction technique. Following References [25, 8], the piecewise linear function

$$\bar{W}^n_{J,K}(\mathbf{x}) = W^n_{J,K} + \mathbf{S}^n_{J,K}\cdot(\mathbf{x}-\mathbf{x}_{J,K}), \quad \mathbf{x}\in A_{J,K} \tag{19}$$

can be employed to replace the previous piecewise constant function $W^n_{J,K}$ at each time level t_n, where $\mathbf{S}^n_{J,K}$ is an approximation of gradient ∇W. Based on these, we can obtain a high accuracy finite volume scheme as

$$\begin{aligned} U^{n+1}_{J,K} = U^n_{J,K} - \frac{\Delta t}{|A_{J,K}|}\Bigg(&\sum_{l=1}^{4} F^{+}_{\mathbf{n}^l}(\bar{U}^n_{J,K}(\mathbf{x}_{\mathrm{c}^l})) + F^{-}_{\mathbf{n}^1}(\bar{U}^n_{J,K-1}(\mathbf{x}_{\mathrm{c}^1}))\\ &+F^{-}_{\mathbf{n}^2}(\bar{U}^n_{J+1,K}(\mathbf{x}_{\mathrm{c}^2})) + F^{-}_{\mathbf{n}^3}(\bar{U}^n_{J,K+1}(\mathbf{x}_{\mathrm{c}^3})) + F^{-}_{\mathbf{n}^4}(\bar{U}^n_{J-1,K}(\mathbf{x}_{\mathrm{c}^4}))\Bigg)\\ &+\Delta t S^{n+1/2}_{J,K} =: U^n_{J,K} + \Delta t L_{J,K}(U^n) \end{aligned} \tag{20}$$

where $\mathbf{x}_{\mathrm{c}^l}$ denotes the co-ordinates of the middle of the lth edge of the control element $A_{J,K}$, and $\bar{U}^n_{J,K}(\mathbf{x}) = \bar{W}^n_{J,K}(\mathbf{x}) - (B(\mathbf{x}),0,0)^{\mathrm{T}}$.

In practice, a slope limiter should be used to suppress appearance of the numerical oscillation. For example, interpolation (19) may be modified as

$$\bar{W}^n_{J,K}(\mathbf{x}) = W^n_{J,K} + \psi_{J,K}\mathbf{S}^n_{J,K}\cdot(\mathbf{x}-\mathbf{x}_{J,K}), \quad \mathbf{x}\in A_{J,K} \tag{21}$$

where $\psi_{J,K}$ is any limiter. To save the computational cost, we will only employ the initial reconstruction technique along each normal direction of the edge of the finite control volume.

Moreover, the second order TVD Runge–Kutta time discretization

$$U^*_{J,K} = U^n_{J,K} + \Delta t L_{J,K}(U^n) \tag{22}$$

$$U^{n+1}_{J,K} = \tfrac{1}{2}\left(U^n_{J,K} + U^*_{J,K} + \Delta t L_{J,K}(U^*)\right) \tag{23}$$

is also implemented to improve accuracy of scheme (20) in time.

3. THE ADAPTIVE MOVING MESH METHOD

The basic idea of our adaptive mesh algorithm can be summarized as the following:

(i) Suppose a logically rectangular spatial grid is given on which the approximation to the SWEs solutions exists on cell-centre;
(ii) Update the grid by iterating an elliptic grid generator derived from the variational approach. Simultaneously update the approximate solution U on the new grid by using a high-resolution conservative interpolation formula; and
(iii) Compute the solution to the SWEs with a physical time Δt_n by using the KFVS finite volume method in the physical domain.

In the remaining part of this section, we will mainly describe Step (ii) above.

3.1. Mesh-redistribution based on variational methods

The calculus of variations have been proven to provide an excellent opportunity to create new techniques for generation of grids or mesh-redistribution by utilizing the idea of optimization of grid characteristics modelled through appropriate functionals. The grid characteristics include grid smoothness, cell skewness, cell volume, and departure from orthogonality or conformality.

In our calculus of variations, the functional is defined by the integral

$$E[\mathbf{x}] = \int_{\Omega_c} \sum_{i=1}^{2} (\tilde{\nabla} x_i)^{\mathrm{T}} G_i \tilde{\nabla} x_i \, \mathrm{d}\boldsymbol{\xi} \tag{24}$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2) = (\xi, \eta)$ and $\mathbf{x} = (x_1, x_2) = (x, y)$ are the logical and physical co-ordinates, respectively, Ω_c denotes a logical domain, $\tilde{\nabla} = (\partial_{\xi_1}, \partial_{\xi_2})^{\mathrm{T}}$, and G_i $(i = 1, 2)$ are given symmetric positive definite matrices, called *monitor functions*. In general, the monitor functions depend on the underlying solution and its derivatives. It is very important to choose suitably the monitor function in the moving mesh algorithms in order to increase the concentration of grid points in any localized region, or improve the quality of the adaptive meshes.

An optimal transformation $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ for functional (24) is a solution of following system of the Euler–Lagrange equations

$$\tilde{\nabla} \cdot (G_i \tilde{\nabla} x_i) = 0, \quad i = 1, 2 \tag{25}$$

in the interior points of the domain Ω_c. Solving numerically the Euler–Lagrange equations (25) on the logical domain Ω_c will give directly a co-ordinate transformation $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ from

Ω_c to the physical domain Ω_p. It is more convenient to solve numerically the mesh equation (25), because the logical domain Ω_c chosen is usually regular, and divided into square meshes.

In this paper, the monitor function G_i is taken as

$$G_1 = G_2 = \begin{pmatrix} w & 0 \\ 0 & w \end{pmatrix} \tag{26}$$

with the following control function:

$$w = \sqrt{1 + \alpha|\psi|^2 + \beta \sum_{i=1}^{2} \left| \frac{\partial \psi}{\partial \xi_i} \right|^2} \tag{27}$$

where α and β are usually positive constants, and ψ is one of the dependent variables. Generally, we will take $\psi = h$ unless otherwise stated.

Before ending this subsection, we give a finite difference or volume discretization of the mesh equation (25) which is further reduced to

$$\frac{\partial}{\partial \xi}\left(w \frac{\partial x_i}{\partial \xi} \right) + \frac{\partial}{\partial \eta}\left(w \frac{\partial x_i}{\partial \eta} \right) = 0, \quad i = 1,2 \tag{28}$$

due to (26). Divide the logical domain $\Omega_c = \{(\xi,\eta) \mid 0 \leqslant \xi, \eta \leqslant 1\}$ into the square meshes

$$\{(\xi_j, \eta_k) \mid \xi_j = j\Delta\xi, \eta_k = k\Delta\eta; 0 \leqslant j \leqslant N_x, 0 \leqslant k \leqslant N_y\}$$

where $\Delta\xi = 1/N_x$ and $\Delta\eta = 1/N_y$. Correspondingly, the numerical approximations to $\mathbf{x} = \mathbf{x}(\xi,\eta)$ are denoted by $\mathbf{x}_{j,k} = \mathbf{x}(\xi_j, \eta_k)$, and Equation (25) is discretized by a second-order central difference scheme as

$$\frac{\Delta_-^j (w_{j+1/2,k} \Delta_+^j (x_i)_{j,k})}{(\Delta\xi)^2} + \frac{\Delta_-^k (w_{j,k+1/2} \Delta_+^k (x_i)_{j,k})}{(\Delta\eta)^2} = 0, \quad i = 1,2 \tag{29}$$

where Δ_+^v and Δ_-^v denote forward and backward difference operators in v-direction, $v = j$ or k, and

$$w_{j+1/2,k} = \tfrac{1}{2}\left((w_1)_{J,K} + (w_1)_{J,K-1}\right)$$

$$w_{j,k+1/2} = \tfrac{1}{2}\left((w_2)_{J,K} + (w_2)_{J-1,K}\right)$$

The discrete system (29) subject to Dirichlet boundary condition can be solved by an explicit iteration methods with a fixed number of iterations, such as Gauss–Seidel iteration.

Remark 3.1

Gauss–Seidel iteration is applied in the present computations, and is more robust than the Jacobi iteration, even though the latter has a good symmetry. Other iteration methods are also available, but the moving speed of grid points should not be too large, otherwise the conservative solution–updating on new meshes will become unstable.

3.2. Conservative solution–updating on new meshes

Let $\mathbf{x}_{j,k}$ and $\tilde{\mathbf{x}}_{j,k}$ be co-ordinates of the old and new grid points, respectively. That is to say, grid point $\mathbf{x}_{j,k}$ will move to position $\tilde{\mathbf{x}}_{j,k}$, after one time iteration of the discrete mesh equations. Use $A_{J,K}$ and $\tilde{A}_{J,K}$ to denote quadrangles (finite control volumes) with four vertices $\mathbf{x}_{j+p,k+q}$, and $\tilde{\mathbf{x}}_{j+p,k+q}$, $0 \leqslant p,\ q \leqslant 1$, respectively. These are of similar setup to that shown in Figure 1.

Using a perturbation method, we can derive a *conservative* scheme to evaluate approximately cell average values $\tilde{U}_{J,K}$ of the underlying solution $U(\mathbf{x})$ over control volume $\tilde{A}_{J,K}$, according to the known cell average values $U_{J,K}$ over control volume $A_{J,K}$.

By assuming $\tilde{\mathbf{x}} = \mathbf{x} - \tilde{\mathbf{u}}$ with small magnitudes of speed $\tilde{\mathbf{u}} = (\tilde{u}_x, \tilde{u}_y)$, we have

$$\begin{aligned}
\int_{\tilde{A}_{J,K}} U(\tilde{\mathbf{x}})\,\mathrm{d}\tilde{\mathbf{x}} &= \int_{A_{J,K}} U(\mathbf{x}) \det\left(\frac{\partial(\tilde{x},\tilde{y})}{\partial(x,y)}\right)\mathrm{d}\mathbf{x} \\
&\approx \int_{A_{J,K}} \left(U(\mathbf{x}) - \tilde{u}_x \frac{\partial U}{\partial x} - \tilde{u}_y \frac{\partial U}{\partial y}\right)\left(1 - \frac{\partial \tilde{u}_x}{\partial x} - \frac{\partial \tilde{u}_y}{\partial y}\right)\mathrm{d}\mathbf{x} \\
&\approx \int_{A_{J,K}} \left(U(\mathbf{x}) - \frac{\partial \tilde{u}_x U}{\partial x} - \frac{\partial \tilde{u}_x U}{\partial y}\right)\mathrm{d}\mathbf{x} \\
&= \int_{A_{J,K}} U(x,y)\,\mathrm{d}\mathbf{x} - ((\tilde{u}_{\mathbf{n}^2} U)_{j+1,k+1/2} + (\tilde{u}_{\mathbf{n}^4} U)_{j,k+1/2} \\
&\quad + (\tilde{u}_{\mathbf{n}^3} U)_{j+1/2,k+1} + (\tilde{u}_{\mathbf{n}^1} U)_{j+1/2,k})
\end{aligned} \tag{30}$$

where the higher order terms have been neglected, $\tilde{u}_{\mathbf{n}^l} := \tilde{u}_x n_x^l + \tilde{u}_y n_y^l$ with the normal outward vector $\mathbf{n}^l = (n_x^l, n_y^l)$, defined as follows:

$$\begin{aligned}
\tilde{u}_{\mathbf{n}^1} &= \tfrac{1}{2}(\tilde{\mathbf{u}}_{j,k} + \tilde{\mathbf{u}}_{j+1,k}) \cdot (y_{j+1,k} - y_{j,k}, x_{j,k} - x_{j+1,k}) \\
\tilde{u}_{\mathbf{n}^2} &= \tfrac{1}{2}(\tilde{\mathbf{u}}_{j+1,k} + \tilde{\mathbf{u}}_{j+1,k+1}) \cdot (y_{j+1,k+1} - y_{j+1,k}, x_{j+1,k} - x_{j+1,k+1}) \\
\tilde{u}_{\mathbf{n}^3} &= \tfrac{1}{2}(\tilde{\mathbf{u}}_{j+1,k+1} + \tilde{\mathbf{u}}_{j,k+1}) \cdot (y_{j,k+1} - y_{j+1,k+1}, x_{j+1,k+1} - x_{j,k+1}) \\
\tilde{u}_{\mathbf{n}^4} &= \tfrac{1}{2}(\tilde{\mathbf{u}}_{j,k+1} + \tilde{\mathbf{u}}_{j,k}) \cdot (y_{j,k} - y_{j,k+1}, x_{j,k+1} - x_{j,k})
\end{aligned}$$

and $(\tilde{u}_{\mathbf{n}} U)_{j+p,k+1/2}$ and $(\tilde{u}_{\mathbf{n}} U)_{j+1/2,k+q}$ denote the values of $\tilde{u}_{\mathbf{n}} U$ at corresponding surface of the control volume $A_{J,K}$, $p, q = 0$ or 1. They will be approximated by using an upwind scheme. For example, the term $(\tilde{u}_{\mathbf{n}^2} U)_{j+1,k+1/2}$ may be approximated by

$$(\tilde{u}_{\mathbf{n}^2} U)_{j+1,k+1/2} = \frac{\tilde{u}_{\mathbf{n}^2}}{2}(U_{J+1,K} + U_{J,K}) - \frac{|\tilde{u}_{\mathbf{n}^2}|}{2}(U_{J+1,K} - U_{J,K}) \tag{31}$$

which is only first-order accurate in space. For this, the initial data reconstruction introduced in Section 2 should also be used in order to avoid too much numerical dissipation.

From (30), a conservative-interpolation can be obtained

$$|\tilde{A}_{J,K}|\tilde{U}_{J,K} = |A_{J,K}|U_{J,K} - ((\tilde{u}_{\mathbf{n}^2}U)_{j+1,k+1/2} + (\tilde{u}_{\mathbf{n}^4}U)_{j,k+1/2} + (\tilde{u}_{\mathbf{n}^3}U)_{j+1/2,k+1} + (\tilde{u}_{\mathbf{n}^1}U)_{j+1/2,k}) \qquad (32)$$

where $|\tilde{A}|$ and $|A|$ denote the areas of the control volumes $\tilde{A}$ and A, respectively. It can be verified that the above solution-updating scheme satisfies the following mass-conservation:

$$\sum_{j,k} |\tilde{A}_{J,K}|\tilde{U}_{J,K} = \sum_{j,k} |A_{J,K}|U_{J,K} \qquad (33)$$

Remark 3.2

The conservative-interpolation (32) is available, when the magnitudes of the grid speed $\tilde{\mathbf{u}}$ are not too big, see its analysis in Reference [14]. It is also possible to propose a conservative-interpolation from a geometrical viewpoint. In 1D, we have

$$\int_{\tilde{x}_j}^{\tilde{x}_{j+1}} U(x)\,\mathrm{d}x = \int_{x_j}^{x_{j+1}} U(x)\,\mathrm{d}x + \left(\int_{\tilde{x}_j}^{x_j} U(x)\,\mathrm{d}x - \int_{\tilde{x}_{j+1}}^{x_{j+1}} U(x)\,\mathrm{d}x \right)$$

i.e.

$$(\tilde{x}_{j+1} - \tilde{x}_j)\tilde{U}_{j+1/2} = (x_{j+1} - x_j)U_{j+1/2} + \left(\int_{\tilde{x}_j}^{x_j} U(x)\,\mathrm{d}x - \int_{\tilde{x}_{j+1}}^{x_{j+1}} U(x)\,\mathrm{d}x \right)$$

According to the locations of the new points $\tilde{x}_j$ and $\tilde{x}_{j+1}$, we can give a suitable approximation of the integrals at the right-hand side in the above, and derive the conservative-interpolation used in Reference [14]. It will become very complicated and expensive to construct multi-dimensional conservative-interpolation from geometry. However, we believe that this multi-dimensional interpolation will be more accurate and useful for computations of the complex problems. This will be studied in the future.

3.3. Solution procedure

The solution procedure of our adaptive mesh strategy for two-dimensional hyperbolic problems can be summarized as follows:

Step i: Give an initial partition $\mathbf{x}_{j,k}^{[0]} = (x_{j,k}^{[0]}, y_{j,k}^{[0]}) := (x_{j,k}, y_{j,k})$ of the physical domain Ω_{p} and a uniform (fixed) partition of the logical domain Ω_{c}, and compute grid values $U_{J,K}^{[0]}$ by cell average over control volume $A_{J,K}^{[0]}$ based on the initial data $U(\mathbf{x}, 0)$.

Step ii: For $v = 0, 1, 2, \ldots, M$, where M is a fixed number, do the following:

(a) Move grid points $\mathbf{x}_{j,k}^{[v]}$ to $\mathbf{x}_{j,k}^{[v+1]}$ by solving (29) with one time Gauss–Seidel iteration.

(b) Compute $\{U_{J,K}^{[v+1]}\}$ over the new control volume $A_{J,K}^{[v+1]}$ based on the conservative-interpolation (32).

Step iii: Evaluate the SWEs using a 2D high-resolution finite volume method (see Section 2) on the mesh $\mathbf{x}_{j,k}^{[M+1]}$ to obtain the numerical approximations $U_{J,K}^{n+1}$ at the time level t_{n+1}.

Step iv: If $t_{n+1} < T$, then let $U_{J,K}^{[0]} := U_{J,K}^{n+1}$ and $\mathbf{x}_{j,k}^{[0]} := \mathbf{x}_{j,k}^{[M+1]}$, and go to *Step ii.*

3.4. Boundary redistribution

In practical flow situations, discontinuities may initially exist at boundaries or move to boundaries in a later time. As a consequence, the movement of boundary points should be controlled in order to improve the quality of the nearby adaptive solutions. A simple redistribution strategy is proposed as follows. For convenience, our attention is restricted to the case that the physical domain $\Omega_{\rm p}$ is rectangular. Assume a new set of grid points $\tilde{\mathbf{x}}_{j,k}$ is obtained in $\Omega_{\rm p}$ by solving the moving mesh equation (29). Then the speeds of internal grid points $\mathbf{x}_{j,k}$ are given by

$$\tilde{\mathbf{u}} := \mathbf{x}^{[v]} - \mathbf{x}^{[v+1]} \quad \text{for} \quad 1 \leqslant j < N_x, \;\; 1 \leqslant k < N_y$$

We assume that the points on the boundary move in its tangential direction at the same speed as the tangential velocity components at internal points immediately adjacent to the boundary, namely

$$\tilde{\mathbf{u}}_{0,k} = (0, (\tilde{u}_y)_{1,k}), \quad \tilde{\mathbf{u}}_{N_x,k} = (0, (\tilde{\mathbf{u}}_y)_{N_x-1,k}) \quad 1 \leqslant k < N_y$$

$$\tilde{\mathbf{u}}_{j,0} = ((\tilde{\mathbf{u}}_x)_{j,1}, 0), \quad \tilde{\mathbf{u}}_{j,N_y} = ((\tilde{\mathbf{u}}_x)_{j,N_y-1}, 0) \quad 1 \leqslant j < N_x$$

Thus the new boundary points $\tilde{\mathbf{x}}_{0,k}$, $\tilde{\mathbf{x}}_{N_x,k}$, $\tilde{\mathbf{x}}_{j,0}$ and $\tilde{\mathbf{x}}_{j,N_y}$ are defined by

$$\tilde{\mathbf{x}}_{j,k} = \mathbf{x}_{j,k} - \tilde{\mathbf{u}}_{j,k}$$

for $j = 0$ or N_x with $1 \leqslant k < N_y$, and $k = 0$ or N_y with $1 \leqslant j < N_x$. Numerical experiments show that the above procedure to move the boundary points is useful in improving the solution resolution.

If the physical domain $\Omega_{\rm p}$ is not a rectangular, then we may introduce an intermediate or parameter space $\Omega_{\rm i}$ with Cartesian co-ordinates $\zeta = (\zeta_1, \zeta_2)$ where $\Omega_{\rm i}$ is taken as the unit square. This idea is similar to one used in numerical grid generations [26].

At $t = 0$, we construct an initial map $\mathbf{x} = \mathbf{x}(\boldsymbol{\zeta}) : \Omega_l \to \Omega_{\rm p}$ by calculus of variations or using algebraic grid generation method, if there does not exist any map between $\Omega_{\rm p}$ and $\Omega_{\rm i}$ with an explicit algebraic expression. For $t > 0$, we first move the grid points in $\Omega_{\rm i}$ according to the previous algorithm, i.e. to solve the following moving mesh equations:

$$\frac{\Delta_-^j((w_1)_{j+1/2,k}\Delta_+^j(\zeta_i)_{j,k})}{(\Delta\xi)^2} + \frac{\Delta_-^k((w_2)_{j,k+1/2}\Delta_+^k(\zeta_i)_{j,k})}{(\Delta\eta)^2} = 0, \quad i = 1, 2 \tag{34}$$

Next, we update the Cartesian co-ordinates $\mathbf{x}$ of the physical space by a non-conservative high resolution interpolation [27], i.e., using a high resolution Hamilton–Jacobi solver to compute the following discrete equation:

$$\mathbf{x}(\tilde{\zeta}) = \mathbf{x}(\zeta) + (\tilde{\zeta} - \zeta) \cdot \nabla_\zeta \mathbf{x} \tag{35}$$

Here $\mathbf{x}$ has been considered as the functions in terms of ζ, and $\mathbf{x}_{j,k}$ is nodal discretization. Finally, the underlying solutions are interpolated using formula (32). The above non-conservative interpolation is fast and computationally efficient.

4. NUMERICAL EXPERIMENTS

In this section, we present several numerical examples to validate the adaptive grid methods coupled to the high resolution KFVS scheme for the SWEs (1).

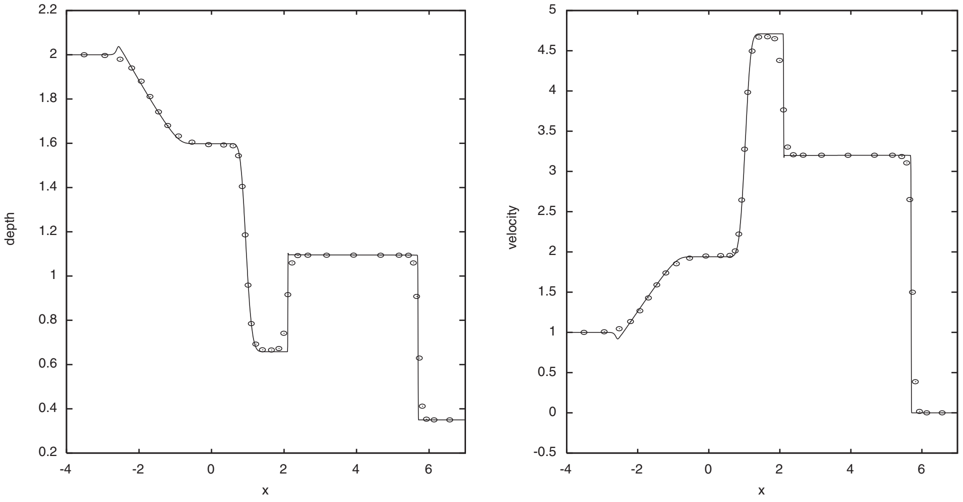

Figure 2. The water depth h and the fluid velocity u at $t = 1$ of Example 4.1 with a non-flat bed obtained by using the adaptive grid method with 51 grid points.

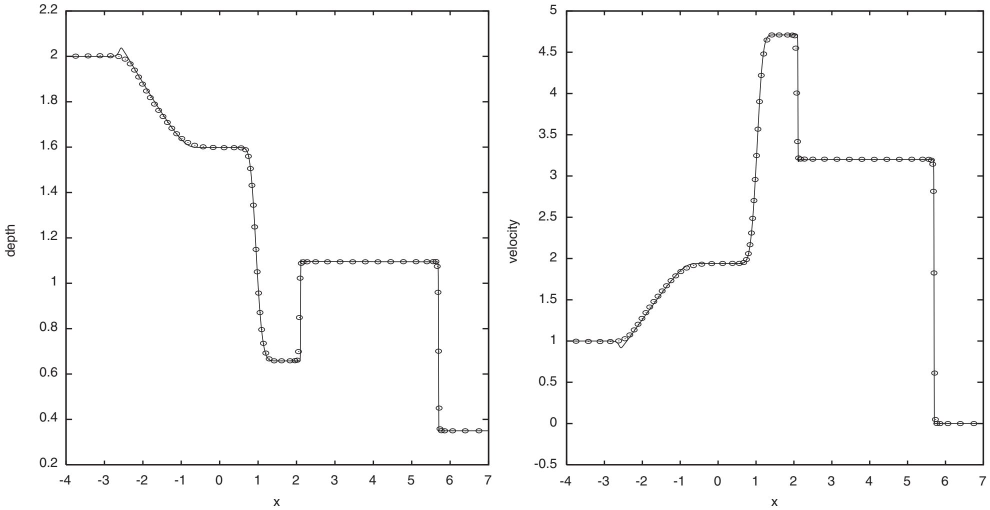

Figure 3. Same as Figure 2, except for 101 grid points.

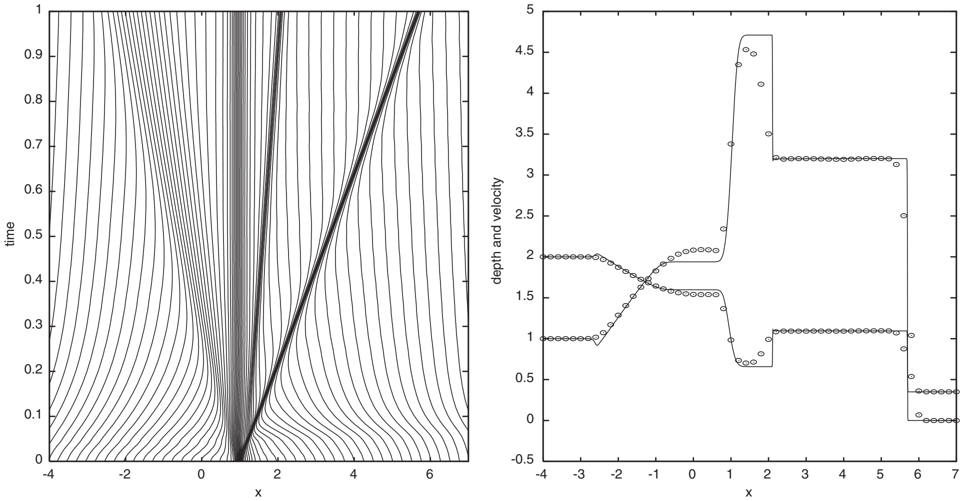

Figure 4. The trajectory of the grid points and the solutions obtained on a uniform mesh with 101 grid points.

The CFL condition is

$$\frac{\Delta t_n}{\min\{\Delta x, \Delta y\}} \max\left\{|u| + \sqrt{gh}, |v| + \sqrt{gh}\right\}_{J,K}^{n} = \sigma \tag{36}$$

where the constant σ is taken as 0.48 (or 0.24) for 1D (or 2D) case in our computations. We also take $g = 9.8$ and use the van Leer's limiter for the initial data reconstruction. The slip condition is enforced at the solid wall, i.e. the velocity normal to the wall is set to zero. The open boundary conditions will be implemented according to the local Froude number $Fr = u_{\mathbf{n}}/\sqrt{gh}$. We refer the readers to Reference [28] for the details.

Example 4.1
We solve one-dimensional SWEs (1) with the wavy bottom function $B(x)$:

$$B(x) = \begin{cases} 0.3(\cos(\pi(x-1)/2))^{30}, & |x-1| \leqslant 1 \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

The initial condition is taken as [5]

$$(h_0(x), u_0(x)) = \begin{cases} (2 - B(x), 1), & -10 \leqslant x < 1 \\ (0.35 - B(x), 0), & 1 \leqslant < x \leqslant 10 \end{cases} \tag{38}$$

The physical domain is taken as $[-10, 10]$. Figures 2 and 3 show the water depth and the velocity at $t = 1$ obtained by the adaptive grid method with the parameters $(\alpha, \beta) = (0, 1)$. The solid lines denote numerical solutions computed on a uniform mesh with 4001 grid points. The trajectory of the grid points is shown in Figure 4. For comparison, we also give the solutions obtained on a uniform mesh with 101grid points. In the region $[-4, 7]$, we can have

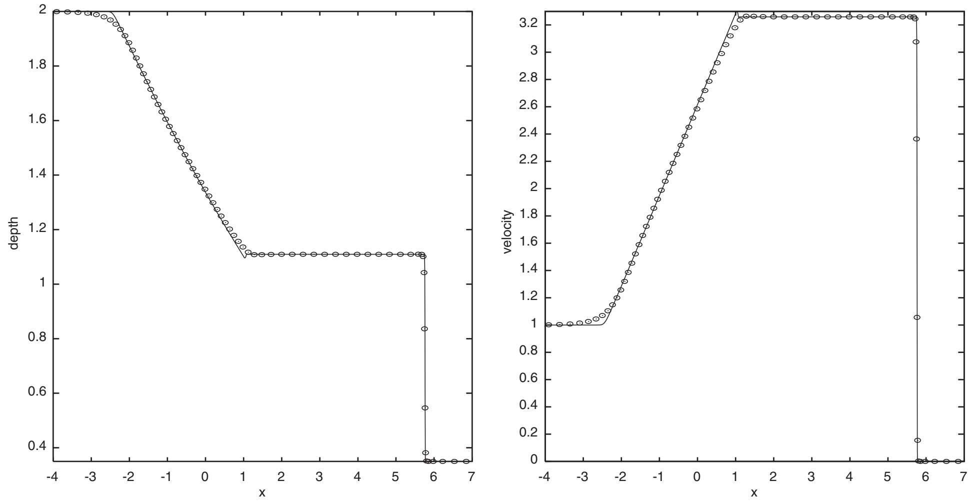

Figure 5. The water depth h and the fluid velocity u at $t=1$ of Example 4.1 with a flat bed obtained by using the adaptive grid method with 101 grid points.

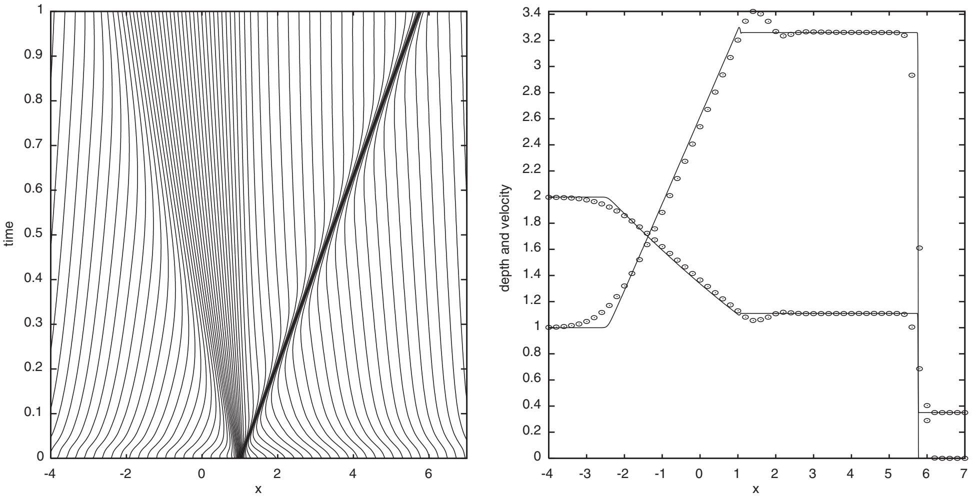

Figure 6. The trajectory of the grid points and the solutions obtained on a uniform mesh with 101 grid points.

a close look at the solution around the wavy bottom. The adaptive grid method gives high resolution results, even when the number of grid points is small. In contrast, if the numerical scheme is used without any well-balanced treatment, it will be difficult to resolve the shock near the point $x=2$ on a uniformly fine mesh, due mainly to non-uniform bed topography. The hump appearing in the uniform solutions at $x=-2.6$ is possibly due to over-compression of the present schemes with van Leer's limiter. This phenomenon has also been observed in

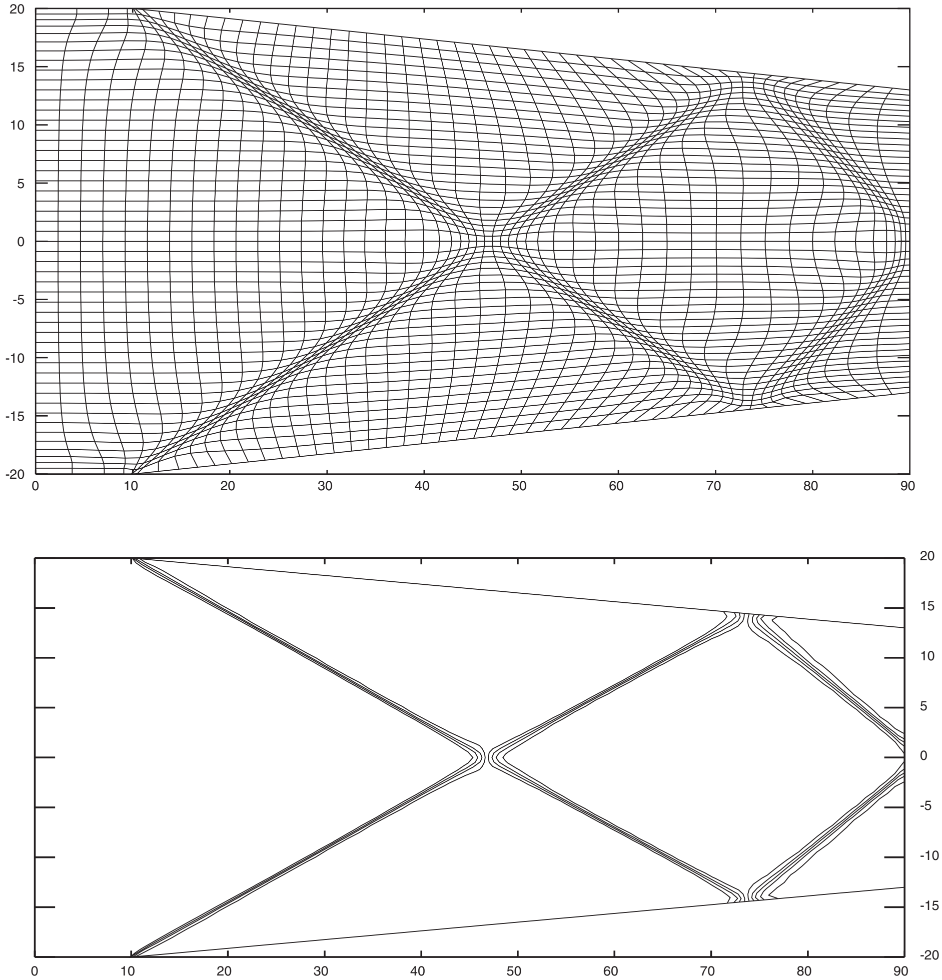

Figure 7. The adaptive mesh (top) and the water depth h (bottom) with 13 equally spaced contour lines at $t = 30$ of Example 4.2 obtained by using the adaptive grid method with 51×51 grid points.

computations of the compressible Euler equations. From Figures 2–4, we can see clearly the main wave structure in the solutions and its evolution.

Figures 5 and 6 also show the numerical solutions at $t = 1$ for the same problem, except for a flat bottom, i.e. $B(x) = 0$. The same parameters have been used as ones in the previous case. For this case, the solutions have only two waves: a left rarefaction wave and a right shock wave, which are simpler than the former.

Example 4.2

The second problem concerns 2D liquid flow in a symmetric channel with a flat bottom constricted from both side in the y-direction. We consider two cases: *channel I* with a constriction angle $\alpha = 5°$ started at $x = 10$, and *channel II* with a constriction angle $\alpha = 15°$ started

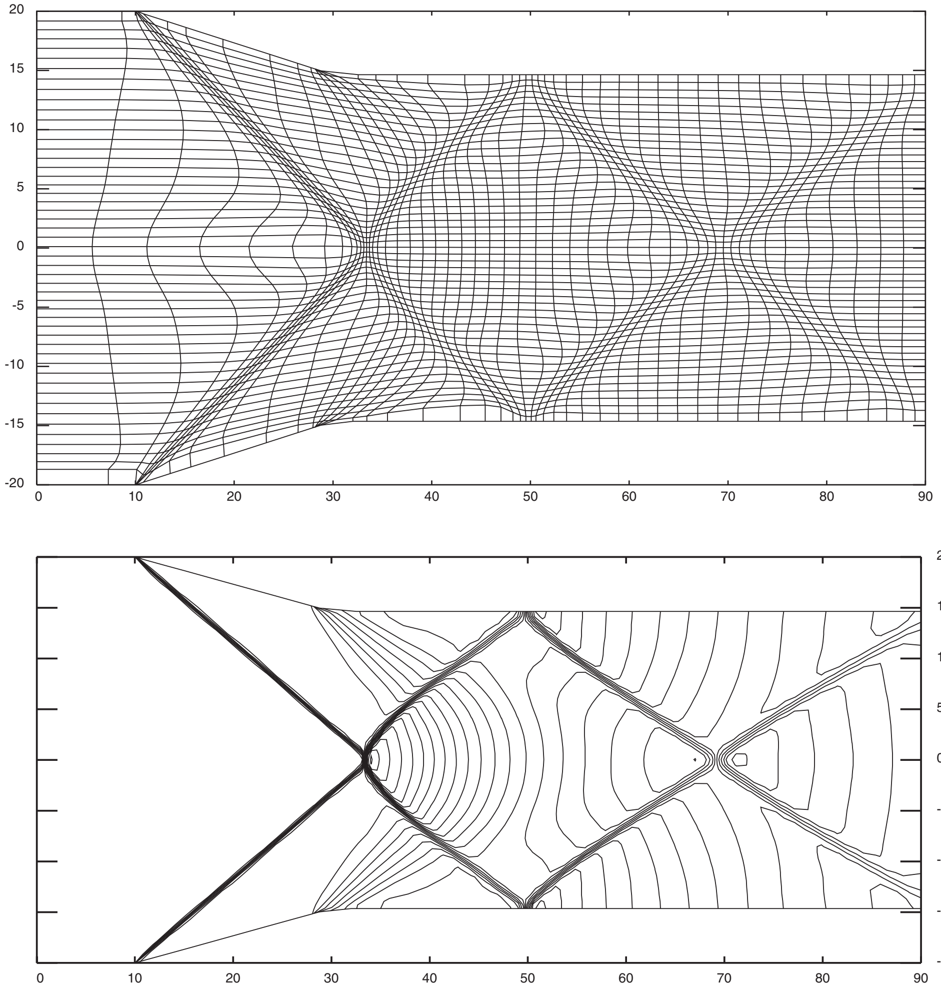

Figure 8. The adaptive mesh (top) and the water depth h (bottom) with 20 equally spaced contour lines at $t=30$ of Example 4.2 for Channel II obtained by using the adaptive grid method with 51×51 grid points.

at $x=10$ and ended at $x=30$, see Figures 7 and 8 for the detailed geometry, respectively. This problem has been computed by other authors, e.g. Reference [29]. The inflow condition at $x=0$ is the depth $h=1$ and the Froude number $Fr=|\mathbf{u}|/\sqrt{gh}=2.5$. Slip boundary conditions are used in the y-direction, and an outflow boundary condition is specified at $x=90$. Figures 7 and 8 show the adaptive meshes and the water depth h computed by the adaptive grid method with 51×51 grid points and $(\alpha, \beta)=(0,2)$. At steady state, reflected bore waves and their interactions are evident due to the constriction. In this case we can compare numerical results with analytical ones, see Reference [29]. Numerical values of the heights $h_2=1.25$, $h_3=1.527$ of the first and second plateau agree well with analytical results $h_2=1.254$, $h_3=1.55$. For flows in *channel II*, two expansion waves are generated by shock waves interacting with the

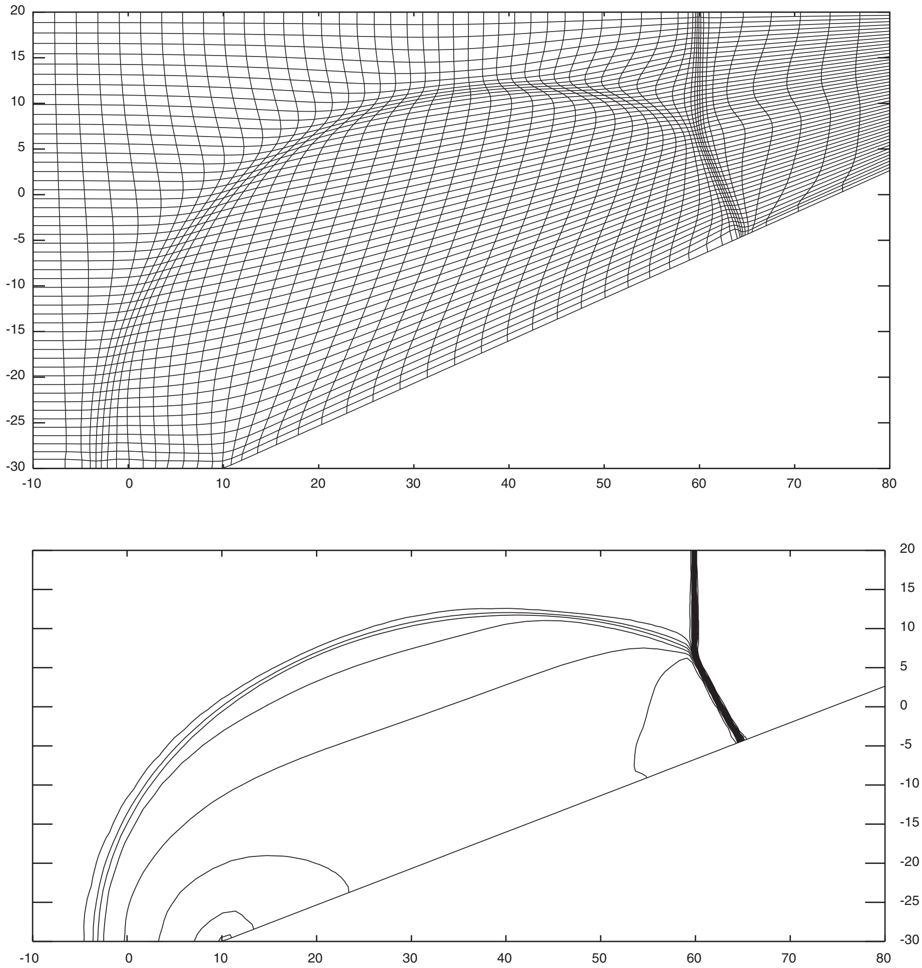

Figure 9. The adaptive mesh (top) and the water depth h (bottom) at $t=8$ of Example 4.3, obtained by using the adaptive grid method with 51×51 grid points.

wall corners. The results are similar to those in Reference [29], and show that the bore waves of high resolution are obtained, the solutions are non-oscillatory and symmetry is preserved.

Example 4.3
This 2D validation case considers unsteady bore reflection at a sea wall. The lower wall is contracted with angle 25° started at $x=10$. The bore Froude number and the initial condition on the left and right states of the bore wave are [8]:

$$\begin{aligned} &Fr=2, \quad h_R=1, \quad u_R=v_R=v_L=0 \\ &h_L=\frac{h_R}{2}\left(\sqrt{1+8Fr^2}-1\right), \quad u_L=Fr\left(1-\frac{h_R}{h_L}\right)\sqrt{gh_R} \end{aligned} \tag{39}$$

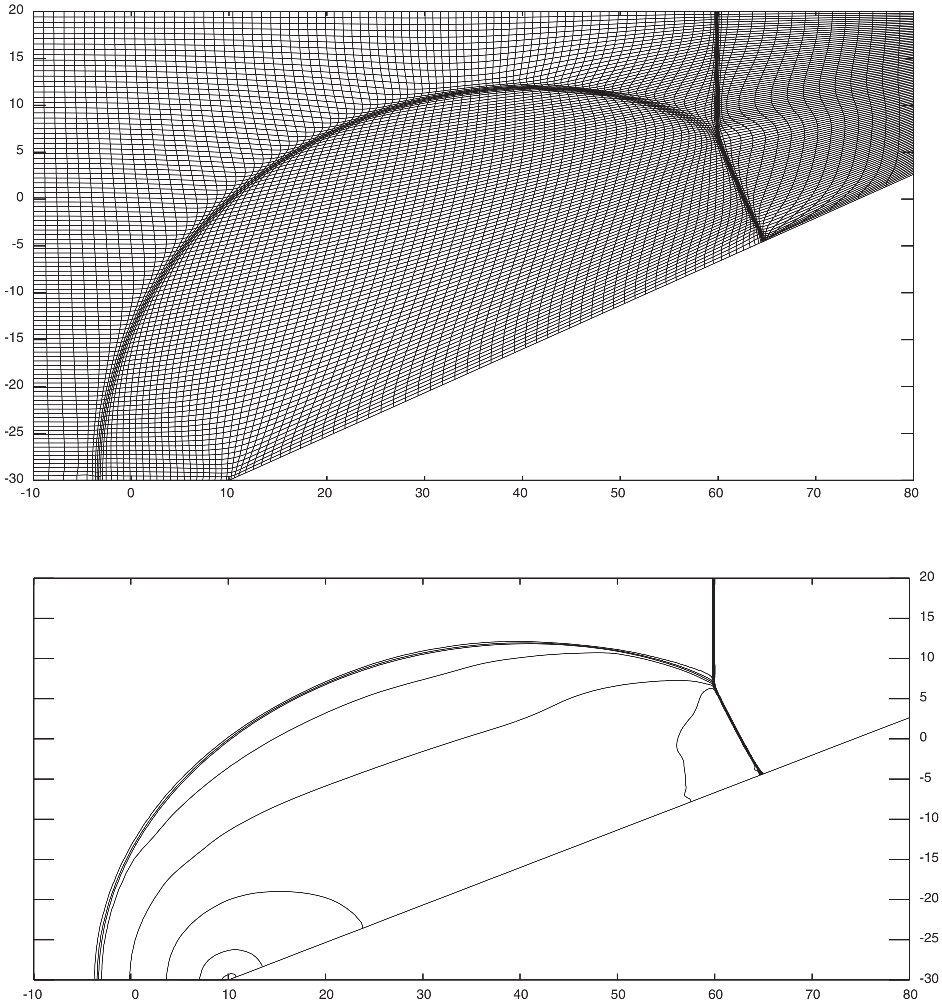

Figure 10. Same as Figure 9, except for 101 × 101 grid points.

The left state of the bore wave is also used as the inflow boundary condition at the left boundary, i.e. $x=-10$; solid wall boundary conditions are applied at the lower wall, and the outflow boundary condition is specified at the right-hand side of the channel, i.e. $x=80$ and the upper boundary, i.e. $y=20$. In Figures 9 and 10, we give the adaptive meshes and the water depth h with 22 equally spaced contour lines at $t=8$ obtained by our adaptive grid method with 51×51 and 101×101 grid points, respectively. They have shown that the bore waves can be captured with a high resolution by our adaptive grid method. The results are similar to those in Reference [30]. To increase the concentration of the grid points near the lune wave, we have used the following monitor function:

$$w_{J,K}=\sqrt{1+\tilde{\phi}_{J,K}} \tag{40}$$

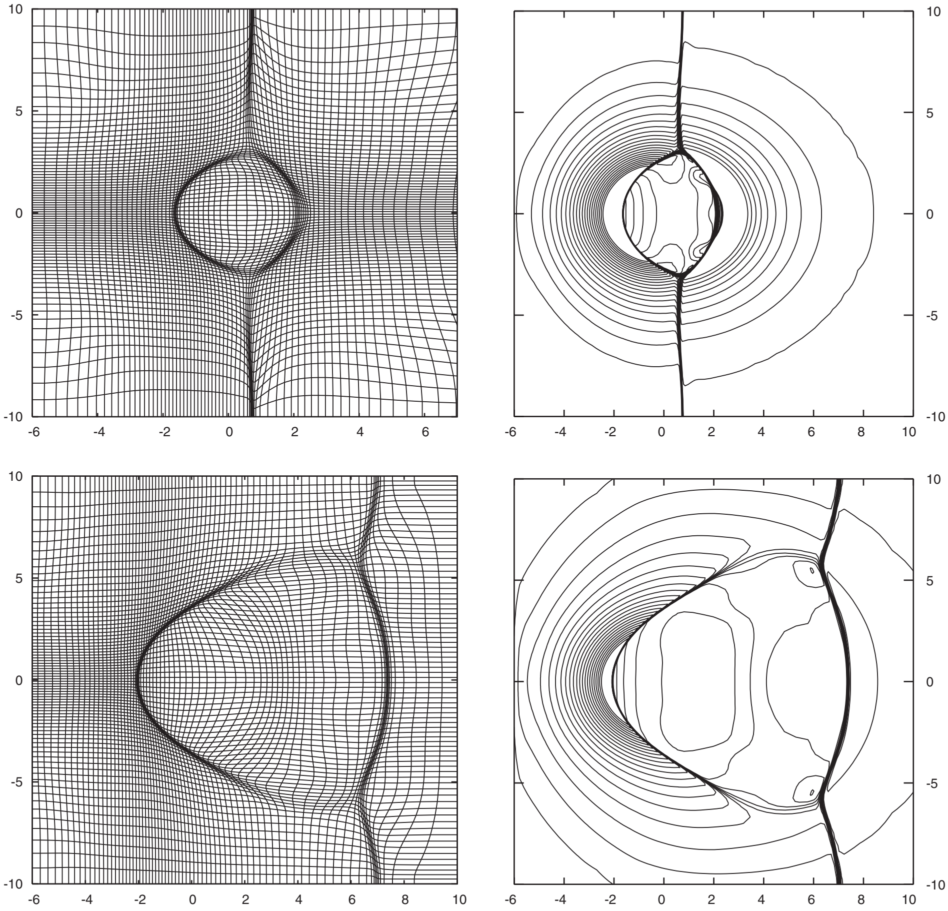

Figure 11. The adaptive solutions of Example 4.4 with 75 × 75 grid points. Top: $t=1$; bottom: $t=2$.

with

$$\tilde{\phi}_{J,K}=\begin{cases}\phi_{J,K} & \text{if } \phi_{J,K}<\theta\Phi\\ \theta\Phi & \text{otherwise}\end{cases} \tag{41}$$

Here $\phi=(\partial h/\partial\xi)^2+(\partial h/\partial\eta)^2$, $\Phi=\max\{\phi_{J,K}\}$, and $\theta=0.03$.

Example 4.4

The final example is a right-going bore wave past an isolated and downward hump centered at (0,0) in the domain $[-6,10]\times[-10,10]$. We solve 2D shallow water equations with a wavy bottom function $B(x,y)$:

$$B(x,y)=-0.4\mathrm{e}^{0.2(12.5-x^2-y^2)} \tag{42}$$

The shock wave is located initially at $x=-5.5$, whose left and right states are given in (39). The inflow boundary condition is set at the left boundary, i.e. $x=-6$, and the outflow boundary

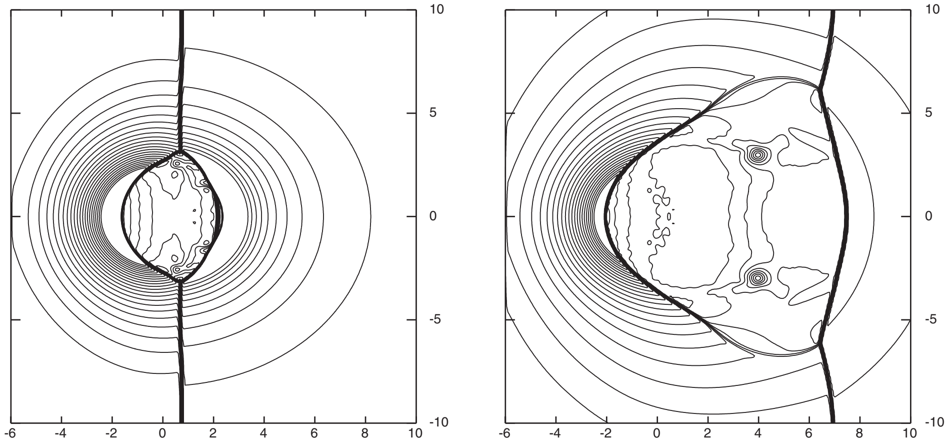

Figure 12. The solutions of Example 4.4 with 300×300 grid points. Left: $t = 1$; right: $t = 2$.

conditions are specified at the other boundaries. If the river bed is flat, the solution will be a single right-going bore wave. Figure 11 shows the adaptive meshes and the water surface with 22 equally spaced contour lines at $t = 1$ and 2 obtained by using 75×75 grid points, respectively. These adaptive solutions can be in comparison with those shown in Figure 12. The solutions in Figure 12 are obtained by a high-resolution Godunov-type method combined with the surface gradient method, see Reference [8], on a fixed uniform mesh with 300×300 grid cells. In this computation, the monitor function (27) is used with $\psi = h + B(\mathbf{x})$ and $(\alpha, \beta) = (0, 1)$.

5. CONCLUSIONS

In this paper, we have extended our adaptive moving mesh method to the 2D SWEs with source terms, which formed by two independent parts: the SWEs evolution and the mesh redistribution. In the first part, the high-resolution kinetic flux-vector splitting (KFVS) method is used in combination with the surface gradient method for initial data reconstruction, but the other well-balanced treatment is not added. An iteration procedure is used in the second part of our algorithm. In each iteration, meshes are redistributed by a variational principle and then the underlying numerical solutions are updated by a conservative-interpolation formula on the resulting new mesh. When there are some hydraulic jumps with different strength or scale in the solutions, adjustments may be made to the monitor functions to increase the concentration of the grid points near a weaker hydraulic jump, see e.g. Equations (40) and (41).

Several 1D and 2D test problems have computed using the present moving mesh algorithm. The numerical results have shown that the solutions with a higher resolution can be obtained by using a KFVS scheme for the SWEs with a much smaller number of grid points than the uniform mesh approach, even though the SWEs solver does not include an additional well-balanced treatment. They also demonstrate that the algorithm is efficient for solving problems with complex bore waves and their interactions or a general geometrical domain.

ACKNOWLEDGEMENTS

This work is supported partly by Special Funds for Major State Basic Research Projects of China, and the Natural Science Foundation of China. The author also wishes to acknowledge support from the Alexander von Humboldt foundation.

REFERENCES

1. Berbudez A, Vazquez ME. Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids* 1994; **23**:1049–1071.
2. Greenberg JM, LeRoux AY. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM Journal on Numerical Analysis* 1996; **33**:1–16.
3. Jin S. A steady-state capturing method for hyperbolic system with geometrical source terms. *Mathematical Modelling on Numerical Analysis* 2001; **35**:631–646.
4. LeVeque RJ. Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. *Journal of Computational Physics* 1998; **146**:346–365.
5. Tang HZ, Tang T, Xu K. A gas-kinetic scheme for shallow-water equations with source terms. *Zeitschrift fuer Angewandte Mathematik und Physik* 2003, to appear; available at http://www.math.hkbu.edu.hk/~ttang.
6. Vazquez-Cendon ME. Improved treatment of source terms in upwind schemes for shallow water equations in channels with irregular geometry. *Journal of Computational Physics* 1999; **148**:497–526.
7. Xu K. A well-balanced gas-kinetic scheme for the shallow-water equations with source terms. *Journal of Computational Physics* 2002; **178**:633–562.
8. 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; **168**:1–25.
9. Azarenok BN, Ivanenko SA, Tang T. Adaptive mesh redistribution method based on Godunov's scheme. *Communications in Mathematical Science* 2002; **1**:153–180.
10. Ceniceros HD, Hou TY. An efficient dynamically adaptive mesh for potentially singular solutions. *Journal of Computational Physics* 2001; **172**:609–639.
11. Harten A, Hyman JM. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics* 1983; **50**:235–269.
12. Liu F, Ji S, Liao G. An adaptive grid method and its application to steady Euler flow calculations. *SIAM Journal on Scientific Computing* 1998; **20**:811–825.
13. Saleri K, Steinberg S. Flux-corrected transport in a moving grid. *Journal of Computational Physics* 1994; **111**: 24–32.
14. Tang HZ, Tang T. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. *SIAM Journal on Numerical Analysis* 2003; **41**:487–515.
15. Winslow A. Numerical solution of the quasi-linear Poisson equation. *Journal of Computational Physics* 1967; **1**: 149–172.
16. Brackbill JU. An adaptive grid with directional control. *Journal of Computational Physics* 1993; **108**:38–50.
17. Brackbill JU, Saltzman JS. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics* 1982; **46**:342–368.
18. Dvinsky AS. Adaptive grid generation from harmonic maps on Riemannian manifolds. *Journal of Computational Physics* 1991; **95**:450–476.
19. Li R, Tang T, Zhang PW. Moving mesh methods in multiple dimensions based on harmonic maps. *Journal of Computational Physics* 2001; **170**:562–588.
20. Miller K, Miller RN. Moving finite element, I. *SIAM Journal on Numerical Analysis* 1981; **18**:1019–1032.
21. Davis SF, Flaherty JE. An adaptive finite element method for initial-boundary value problems for partial differential equations. *SIAM Journal on Scientific and Statistical Computing* 1982; **3**:6–27.
22. Cao WM, Huang WZ, Russell RD. An r-adaptive finite element method based upon moving mesh PDEs. *Journal of Computational Physics* 1999; **149**:221–244.
23. Xu K. *Gas-Kinetic Schemes for Unsteady Compressible Flow Simulations*. Lecture Series 1998-03. von Karman Institute for Fluid Dynamics, 1998.
24. Pullin DI. Direct simulations methods for compressible inviscid ideal gas-flows. *Journal of Computational Physics* 1980; **34**:231–244.
25. van Leer B. Towards the ultimate conservative difference scheme, V. A second order sequel to Godunov's method. *Journal of Computational Physics* 1979; **32**:101–136.
26. Thompson JF, Soni BK, Weatherill NP. *Handbook of Grid Generation*. CRC Press: Boca Raton, London, New York, Washington, DC, 1999.

27. Tang HZ, Tang T, Zhang PW. An adaptive mesh redistribution method for nonlinear Hamilton–Jacobi equations in two- and three-dimensions. *Journal of Computational Physics* 2003; **188**:534–572.
28. Rogers B, Fujihara M, Borthwick AGL. Adaptive Q-tree Godunov-type scheme for shallow water equations. *International Journal for Numerical Methods in Fluids* 2001; **35**:247–280.
29. Liska R, Wendroff B. Two-dimensional shallow water equations by composite schemes. *International Journal for Numerical Methods in Fluids* 1999; **30**:831–843.
30. Causon DM, Ingram DM, Mingham CG, Yang G, Pearson RV. Calculation of shallow water flows using a Cartesian cut cell approach. *Advances in Water Resources* 2000; **23**:545–562.