Hindawi Publishing Corporation
Advances in Mathematical Physics
Volume 2016, Article ID 7528625, 7 pages
http://dx.doi.org/10.1155/2016/7528625

Research Article

Adaptive Finite Volume Method for the Shallow Water Equations on Triangular Grids

Sudi Mungkasi

Department of Mathematics, Faculty of Science and Technology, Sanata Dharma University, Mrican, Tromol Pos 29, Yogyakarta 55002, Indonesia

Correspondence should be addressed to Sudi Mungkasi; sudi@usd.ac.id

Received 31 March 2016; Revised 21 August 2016; Accepted 5 September 2016

Academic Editor: Soheil Salahshour

This paper presents a numerical entropy production (NEP) scheme for two-dimensional shallow water equations on unstructured triangular grids. We implement NEP as the error indicator for adaptive mesh refinement or coarsening in solving the shallow water equations using a finite volume method. Numerical simulations show that NEP is successful to be a refinement/coarsening indicator in the adaptive mesh finite volume method, as the method refines the mesh or grids around nonsmooth regions and coarsens them around smooth regions.

1. Introduction

The numerical entropy production (NEP) has been implemented successfully as smoothness indicator (error indicator) in adaptive finite volume methods used to solve compressible fluid problems [1–4]. In the work of Mungkasi and Roberts [5, 6], NEP was used for solving the one-dimensional shallow water equations. We note that two-dimensional problems are more complex and more applicable in real situations but challenging to solve.

The goal of this paper is to implement the NEP into an adaptive mesh finite volume method used to solve the two-dimensional shallow water equations on unstructured triangular grids. Recall that NEP diverges under grid refinement on nonsmooth regions, while it has the same order of magnitude of the local error on smooth ones [4]. The difference between the degrees of accuracy between smooth and nonsmooth regions makes the NEP able to detect the smoothness of the numerical solution. Therefore, we are confident here to implement the NEP as smoothness indicator into the adaptive mesh finite volume method.

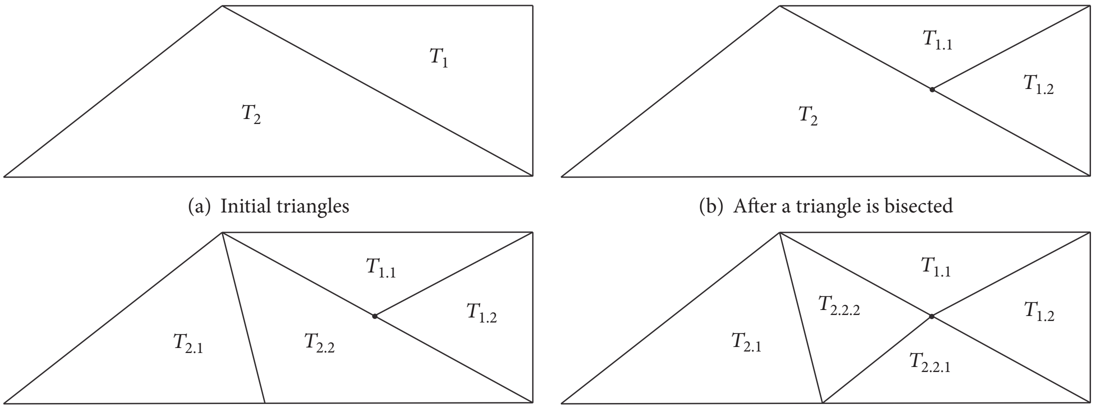
Our adaptive mesh finite volume method uses iFEM as the underlying data structure, an integrated finite element methods package in MATLAB. We note that iFEM and its data structure are not of our work, but they were developed by Chen [7]. Even though iFEM was originally developed for finite element methods, we can still use it to achieve our goal for finite volume methods. This is because what we need is actually the data structure of iFEM. Therefore, we can adapt the original iFEM into our finite volume problems.

This paper is organised as follows. Section 2 summarises the iFEM data structure for triangulations to assist mesh refinement and coarsening techniques in the finite volume method. In Section 3, we recall the two-dimensional shallow water equations and present the scheme for the numerical entropy production. Some simulation results are reported in Section 4. Finally, we draw concluding remarks in Section 5.

2. Data Structure for Triangulations

We implement two-dimensional mesh triangulations of iFEM. In this section, we summarise the iFEM data structure for triangulations to assist mesh adaptation (refinement or coarsening) techniques in our finite volume method. More detailed explanations can be found in a technical report written by Chen [7] and a paper by Chen and Zhang [8]. Note that in our work the spatial dimension is two.

Let N be the number of vertices and NT be the number of elements, where the elements are triangles. A two-dimensional triangulation is represented by the matrices $\texttt{node}(1:N, 1:2)$ and $\texttt{elem}(1:NT, 1:3)$. The ith row of



(a) Initial triangles

(b) After a triangle is bisected

(c) Hanging node still occurs

(d) Conforming triangles

Figure 1: Illustration of the refinement procedure.

the matrix `node` represents the (x, y) coordinates of the ith node. The ith row of the matrix `elem` stores the three nodes (global indices of vertices) of the ith element. The three nodes are stored counterclockwise.

Two conditions on triangulations are imposed. The first condition is that a triangulation must be conforming. That is, the intersection of any two simplexes in the triangulation is either empty or a common lower dimensional simplex. The second condition is that all triangulations must be shape-regular. A triangulation $\mathcal{T}$ is called shape-regular if there exists a constant c_0 such that

$$\max_{\tau\in T}\frac{\operatorname{diam}(\tau)^2}{|\tau|}\le c_0 \tag{1}$$

for all triangles T in $\mathcal{T}$, where $\operatorname{diam}(\tau)$ is the diameter of τ and $|\tau|$ is the measure of τ.

2.1. Refinement: The Bisection Method. A newest-vertex bisection method is implemented for grid refinement, but in the initial triangulation the longest edge is assigned as the refinement edge of each triangle. The function `bisect` refines a given triangulation by bisecting marked elements and minimal neighbouring elements to get a conforming and shape-regular triangulation. This means that a triangle is bisected based on the newest-vertex appearing on that triangle. In other words, the newest-vertex appearing on that triangle is used as the reference for the triangular bisection.

A triangulation is labeled as follows. Note that global indices of three vertices of a triangle t are given by $\texttt{elem}(t, \{1, 2, 3\})$. The newest-vertex of the triangle t is set as $\texttt{elem}(t, 1)$ and the refinement edge is $\texttt{elem}(t, \{2, 3\})$. That is, if we label vertices of the parent t as p_1, p_2, and p_3 and the new node relating to a refinement of t is p_4, then the children of t have vertices p_4, p_1, and p_2 and p_4, p_3, and p_1. The first child called the LEFT (L) child occupies the location used by t. The second child called the RIGHT (R) child is appended as a new element to `elem` matrix. To be specific, the first (L) child is stored prior to the second (R) child in `elem` matrix. We call this storing technique the (L, R) positioning.

Bisections over a collection of elements could result in a nonconforming triangulation. To enforce that we get a conforming triangulation, we do the following. Let $\texttt{cutEdge}(\tau)$ be the refinement edge of τ and $F(\tau)$ be the refinement neighbour of τ. The refinement neighbour means the neighbour sharing the refinement edge of τ. We note that a triangle τ has at most three neighbours and we may have that $\texttt{cutEdge}(\tau) \neq \texttt{cutEdge}(F(\tau))$. The rule which enforces conformity is that if $\texttt{cutEdge}(\tau)$ is bisected, then $\texttt{cutEdge}(F(\tau))$ must also be bisected. In iFEM, this rule is implemented in a loop and the loop terminates in a finite number of steps producing a conforming triangulation.

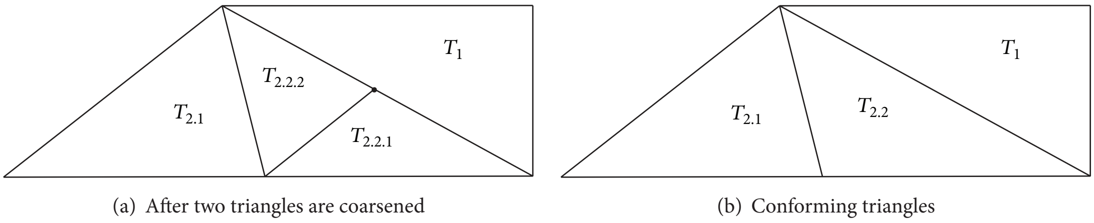
An example of the refinement procedure is illustrated in Figure 1. Let us consider an initial triangulation, as shown in Figure 1(a). Suppose that we want to refine triangle T_1. Triangle T_1 is bisected with respect to the longest edge of T_1, as shown in Figure 1(b), leading to a nonconforming triangulation because a hanging node occurs. The neighbour of T_1 having the common edge of T_1 that is bisected is triangle T_2. Then triangle T_2 is bisected with respect to the longest edge of T_2, as shown in Figure 1(c), leading to another nonconforming triangulation because the hanging node still occurs. Finally the hanging node is used as a reference for bisecting triangle $T_{2.2}$, which leads to a fine conforming triangulation.

2.2. Coarsening: A Nodal-Wise Algorithm. A nodal-wise coarsening algorithm is used for adaptive grids obtained from newest-vertex bisections, with a note that the algorithm enforces not to coarsen the grids in the initial triangulation. A coarsening process is thought as the inverse of a conforming bisection and restricted to a conforming star. A conforming bisection means a bisection which results in a conforming triangulation, and a conforming star means a star-shaped domain consisting of some conforming triangles.

A node p is called a *good-for-coarsening node* or a *good node* in short if there exists a conforming bisection b_e such that p is the middle point of e, where b_e bisects

(i) two conforming triangles into four conforming triangles if e is an interior edge or

(ii) one triangle into two conforming triangles if e is a boundary edge.



(a) After two triangles are coarsened

(b) Conforming triangles

FIGURE 2: Illustration of the coarsening procedure.

Finding good nodes can be described as follows. Let T and T_0 be conforming triangulations, where T can be found by refinement of some triangles of T_0. A node of T is a good node if and only if

(1) it is not a vertex of T_0,

(2) it is the newest-vertex of all elements in the star of the node,

(3) the number of elements in the star is four if the node is an interior node or two otherwise (a boundary node).

Therefore, if some triangles of T are marked to be coarsened, then we check the newest vertices, which satisfies the three conditions, of the marked elements.

After good nodes are identified, coarsening of conforming stars can be done by considering the (L, R) type of positioning. If a good node is an interior node, then four conforming triangles are coarsened into two conforming ones. If a good node is a boundary node, then two conforming triangles are coarsened into one triangle.

An example of the coarsening procedure is illustrated in Figure 2. Let us consider the triangulation in Figure 1(d). Suppose that we want to coarsen the triangles $T_{1.1}$ and $T_{1.2}$. Coarsening these two triangles results in a nonconforming triangulation, because a hanging node occurs, as shown in Figure 2(a). Then the two triangles $T_{2.2.1}$ and $T_{2.2.2}$ sharing the same hanging node are coarsened, which leads to a coarse conforming triangulation, as shown in Figure 2(b).

2.3. Conserved Quantities in Finite Volume Methods. The data structure for conserved quantities follows from the structure for triangulations. In addition, if a parent-cell is bisected, then we do a quantitative splitting of the amount of the conserved quantity of the parent-cell into two parts for the resulting child-cells. If two conforming child-cells are coarsened then we do a quantitative averaging of the amount of the conserved quantity of the child-cells for the resulting parent-cell. The splitting and averaging are done in a proportional weight with respect to the areas of the base child-cells.

Remark 1. In the refinement and coarsening processes of iFEM, the longest edge is assigned as the refinement edge of each triangle in the initial triangulation $\mathcal{T}_0$. In iFEM, the subroutine `label` is needed only for the initial triangulation. Therefore, the subroutine `label` is called outside of the function `bisect`. This procedure leads to a shape-regular triangulation and so prevents the formation of triangles with very small angles (see Chen [7] for more details on this data structure of iFEM).

3. Numerical Entropy Production

Recall the two-dimensional shallow water (wave) equations

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x + \mathbf{g}(\mathbf{q})_y = \mathbf{s}, \qquad (2)$$

where $\mathbf{q} = \begin{bmatrix} h & hu & hv \end{bmatrix}^T$ is the vector of conserved quantities consisting of water depth h, x-momentum hu, and y-momentum hv. Here, u and v are velocities in the x- and y-direction; $\mathbf{f}(\mathbf{q})$ and $\mathbf{g}(\mathbf{q})$ are flux functions in the x- and y-direction given by

$$\mathbf{f}(\mathbf{q}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \qquad \mathbf{g}(\mathbf{q}) = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}; \qquad (3)$$

the source term is

$$\mathbf{s} = \begin{bmatrix} 0 \\ -ghz_x \\ -ghz_y \end{bmatrix}, \qquad (4)$$

where $z(x, y)$ is the bed topography. The stage (absolute water level) is given by $w := z + h$. The constant g is the acceleration due to gravity. The free variables are the spaces x and y as well as time t. More explanation about the shallow water equations can be found in the literature, such as Akyildiz [9], Peng [10], and Yue et al. [11].

The entropy inequality is [12]

$$\eta_t + \psi_x + \phi_y \leq 0, \qquad (5)$$

where the entropy is

$$\eta(\mathbf{q}(x, y, t), z(x, y)) = \frac{1}{2}h\left(u^2 + v^2\right) + \frac{1}{2}gh^2 + ghz, \qquad (6)$$

and entropy fluxes are

$$\begin{aligned}\psi\left(\mathbf{q}\left(x,y,t\right),z\left(x,y\right)\right) &= \left(\frac{1}{2}h\left(u^2+v^2\right)+gh^2+ghz\right)u,\\ \phi\left(\mathbf{q}\left(x,y,t\right),z\left(x,y\right)\right) &= \left(\frac{1}{2}h\left(u^2+v^2\right)+gh^2+ghz\right)v.\end{aligned} \tag{7}$$

We solve the two-dimensional shallow water equations using a finite volume method described by Mungkasi and Roberts [13], but in the present paper, we improve the method adaptively. The finite volume method itself has been used to solve one-dimensional problems successfully [14–16].

For a smooth solution, (5) becomes an equality and is called entropy equation. To compute the NEP we need to evolve numerically the entropy based on the entropy residual, given by the discretisation of the left hand side of (5). This evolution is done together with the evolution of the conserved quantities. We may discretise the spatial domain into arbitrary polygonal cells, but in this paper we focus our work on triangular grids. Then for an arbitrary cell i, we have a semidiscrete scheme for the entropy evolution

$$\frac{d\Theta_i}{dt}+\frac{1}{A_i}\sum_{j\in\mathcal{N}(i)}\Psi_{ij}l_{ij}\lesssim 0. \tag{8}$$

The notation ij is the edge (interface) between the i and j cells. Here Θ_i is an approximation of the averaged entropy over the i cell, Ψ_{ij} is the outward normal entropy flux across the edge ij, and l_{ij} is the length of the edge ij. In addition, $\mathcal{N}(i)$ is the set of three neighbouring cells of the i cell and A_i is the area of the i cell. The flux Ψ_{ij} is evaluated using a consistent numerical flux function $\Psi(\cdot,\cdot)$ such that for all values of Θ

$$\Psi\left(\Theta,\Theta\right)=\psi\left(\Theta\right). \tag{9}$$

Furthermore,

$$\Psi_{ij}=\Psi\left(\Theta_i\left(m_{ij}\right),\Theta_j\left(m_{ij}\right)\right), \tag{10}$$

where m_{ij} is the midpoint of the ij edge.

Note that in (8) we have used the sign "$\lesssim$." With this sign, we mean that it is not guaranteed that inequality "<" in (8) holds. Note also that it has been proven that the entropy production, which is the left hand side of (8), is negative definite only for some cases. In general, the entropy production may have positive overshoots. More details can be found in the work of Puppo and Semplice [4].

Recall that, in evolving the conserved quantities [13], we employ rotations from global coordinates into local coordinates in order to compute the conserved quantity fluxes. The rotations are needed because the conserved quantities are vector-valued. However, unlike the conserved quantities, the entropy is scalar-valued. Therefore, in evolving the entropy numerically we do not need to do any rotation but directly compute the edge-flux at each interface between cells.

The NEP is formulated as

$$E_i^n=\frac{1}{\Delta t}\left(\eta\left(\mathbf{Q}_i^n\right)-\Theta_i^n\right). \tag{11}$$

In (11), Θ_i^n is found from the numerical entropy scheme (8) with $\eta(\mathbf{Q}^{n-1})$ as the input; $\mathbf{Q}_i^n$ is found from the finite volume scheme with $\mathbf{Q}^{n-1}$ as the input; and Δt is the time step used in those two schemes. Following Puppo and Semplice [4], the NEP formula is expressed as

$$E_i^n=\frac{1}{\Delta t}\left(\eta\left(\mathbf{Q}_i^n\right)-\eta\left(\mathbf{Q}_i^{n-1}\right)\right)+\frac{1}{A_i}\sum_{j\in\mathcal{N}(i)}\Psi_{ij}l_{ij}. \tag{12}$$

Remark 2. The left hand side of (8) is not zero. It defines the entropy residual (in semidiscrete form), from which we then derive E_i^n in (11) and (12). The left hand side of (8) in fact is not exactly zero, even on smooth flows, because we have discretised the entropy relation in space. More discussions about entropy and related schemes are given comprehensively by a number of researchers, such as Tadmor et al. [17–21].

4. Numerical Simulations

In this section, we present two simulations to show the performance of the NEP as smoothness indicator in the adaptive mesh finite volume methods. The two simulations are planar and radial dam-break problems. A first-order method is used. Quantities are measured in SI units. The numerical flux due to Kurganov, Noelle, and Petrova (KNP) [22] is used. The acceleration due to gravity is taken as $g = 9.81$.

4.1. Planar Dam-Break Problem. We consider a spatial domain defined by $(x,y)\in[-1,1]\times[-1,1]$. The domain is discretised into 2048 triangular cells initially. The initial condition is

$$\begin{aligned}u\left(x,y,0\right)&=0,\\ v\left(x,y,0\right)&=0,\\ w\left(x,y,0\right)&=\begin{cases}0.5, & \text{if } x<0,\\ 0.2, & \text{if otherwise.}\end{cases}\end{aligned} \tag{13}$$

The bottom topography is $z(x,y)=0$ for all x and y. We take fixed time step $\Delta t = 0.002$ with 100 iterations. The maximum level of refinement/coarsening is 2 at each iteration. That is, a triangular cell can be refined or coarsened up to two levels finer or two levels coarser at each iteration. However, we restrict our refinement/coarsening only up to 8 levels of triangles from the smallest to the largest. The NEP tolerance is $\text{NEP}_{\text{tol}} = 0.25\max|\text{NEP}|$. That is, if the NEP is larger than the tolerance, then the corresponding cell is refined in order to get more accurate results. After the refinement procedure if the NEP is lower than the tolerance, then the corresponding cell is coarsened in order to get faster computation.

Representatives of simulation results are shown in Figures 3 and 4. We see that fine grids occur around shock and

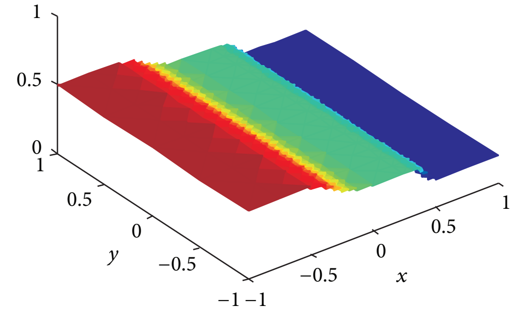


Figure 3: The water surface of the planar dam-break problem at $t = 0.2$ using the adaptive method. The base plane is the xy-plane.

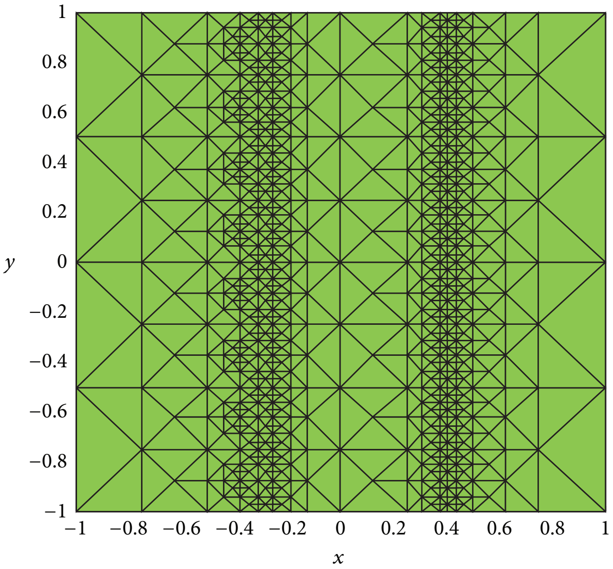


Figure 4: The mesh of the planar dam-break problem at $t = 0.2$ using the adaptive method. The x-axis is horizontal. The y-axis is vertical.

rarefaction regions in order to get accurate solutions at those regions. Coarser grids occur in smooth regions having flat water surface in this case. This means that the adaptive finite volume method works well as we expect and is consistent with the behaviour for the one-dimensional problems as in our Ph.D. thesis [23]. At this time ($t = 0.2$) the number of cells is 1544, which is a smaller number than 2048, the initial number of cells.

4.2. Radial Dam-Break Problem. We consider a spatial domain defined by $(x, y) \in [-1, 1] \times [-1, 1]$. The initial condition is

$$\begin{aligned} u(x,y,0) &= 0, \\ v(x,y,0) &= 0, \\ w(x,y,0) &= \begin{cases} 1, & \text{if } x^2 + y^2 < 0.25, \\ 0.5, & \text{if otherwise.} \end{cases} \end{aligned} \tag{14}$$

The bottom topography is $z(x, y) = 0$ for all x and y.

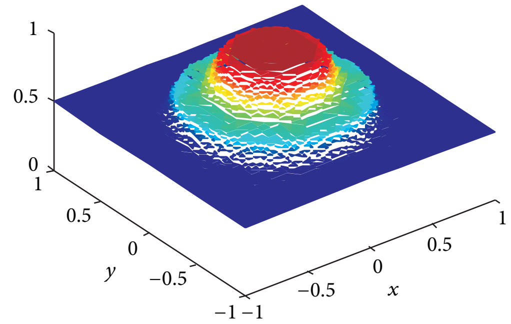


Figure 5: The water surface of the radial dam-break problem at $t = 0.05$ using the adaptive method. The base plane is the xy-plane.

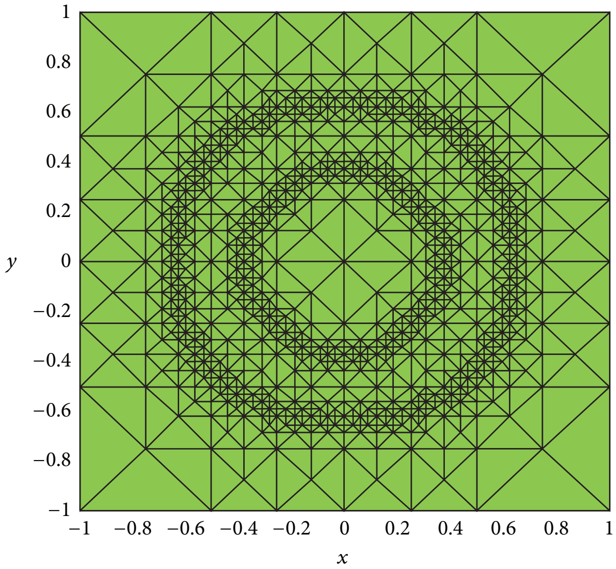


Figure 6: The mesh of the radial dam-break problem at $t = 0.05$ using the adaptive method. The x-axis is horizontal. The y-axis is vertical.

The domain is discretised into 2048 triangular cells initially. We take fixed time step $\Delta t = 0.002$ with 25 iterations. The maximum level of refinement/coarsening is 2 at each iteration, so that a triangular cell can be refined or coarsened up to two levels finer or two levels coarser at each iteration. Again, we restrict our refinement/coarsening only up to 8 levels of triangles from the smallest to the largest. The NEP tolerance is $\text{NEP}_{\text{tol}} = 0.25 \max|\text{NEP}|$. Cells having absolute NEP larger than NEP_{tol} are candidates for refinement, while otherwise they are candidates for coarsening.

Representatives of simulation results are shown in Figures 5 and 6. Similar to the previous simulation, we see that the grids are small around the shock and rarefaction in order to get accurate solutions and large at flat water surface in order to reduce the computational cost. At this time ($t = 0.05$), the number of cells is 1968.

Remark 3. In our simulations above, we have specified intuitively the criterion for choosing the NEP threshold (tolerance) used to trigger grid refinement and coarsening.

Obviously the tolerance should be between zero and the maximum value of the absolute NEP. If the tolerance is too large, then the grids will be too coarse so the wave will be too diffusive and inaccurate. If the tolerance is too small, then the grids will be too fine; that is, the computation will be very expensive even though the solution is accurate. This trade-off could be studied further but is out of the scope of this paper.

5. Conclusions

Numerical entropy production has been implemented into an adaptive mesh finite volume method used to solve the two-dimensional shallow water equations. The implementation is successful, as the method does refinement around nonsmooth regions and coarsening around smooth regions. Future work will focus on the error analysis and investigate the rate of convergence of the adaptive method.

Disclosure

Some parts of this work were conducted and written as a portion of the author's Ph.D. thesis at the Australian National University. Other parts were conducted and written at Sanata Dharma University after the author received the Ph.D. degree.

Competing Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was financially supported by the Australian National University (ANU), Canberra, Australia, and Sanata Dharma University, Yogyakarta, Indonesia. The author thanks Professor Stephen Gwyn Roberts at the Mathematical Sciences Institute of the ANU for the advice and discussions.

References

[1] F. Golay, "Numerical entropy production and error indicator for compressible flows," *Comptes Rendus Mécanique*, vol. 337, no. 4, pp. 233–237, 2009.

[2] G. Puppo, "Numerical entropy production on shocks and smooth transitions," *Journal of Scientific Computing*, vol. 17, pp. 263–271, 2002.

[3] G. Puppo, "Numerical entropy production for central schemes," *SIAM Journal on Scientific Computing*, vol. 25, no. 4, pp. 1382–1415, 2003.

[4] G. Puppo and M. Semplice, "Numerical entropy and adaptivity for finite volume schemes," *Communications in Computational Physics*, vol. 10, no. 5, pp. 1132–1160, 2011.

[5] S. Mungkasi and S. G. Roberts, "Numerical entropy production for shallow water flows," *ANZIAM Journal*, vol. 52, pp. C1–C17, 2010.

[6] S. Mungkasi and S. G. Roberts, "Behaviour of the numerical entropy production of the one-and-a-half-dimensional shallow water equations," *ANZIAM Journal*, vol. 54, pp. C18–C33, 2012.

[7] L. Chen, "iFEM: an integrated finite element methods package in MATLAB," Tech. Rep., University of California, Irvine, Calif, USA, 2009.

[8] L. Chen and C. Zhang, "A coarsening algorithm on adaptive grids by newest vertex bisection and its applications," *Journal of Computational Mathematics*, vol. 28, no. 6, pp. 767–789, 2010.

[9] Y. Akyildiz, "The shallow water equations: conservation laws and symplectic geometry," *International Journal of Mathematics and Mathematical Sciences*, vol. 10, no. 3, pp. 557–562, 1987.

[10] S.-H. Peng, "1D and 2D numerical modeling for solving dam-break flow problems using finite volume method," *Journal of Applied Mathematics*, vol. 2012, Article ID 489269, 14 pages, 2012.

[11] Z. Yue, H. Liu, Y. Li, P. Hu, and Y. Zhang, "A well-balanced and fully coupled noncapacity model for dam-break flooding," *Mathematical Problems in Engineering*, vol. 2015, Article ID 613853, 13 pages, 2015.

[12] F. Bouchut, "Efficient numerical finite volume schemes for shallow water models," in *Nonlinear Dynamics of Rotating Shallow Water: Methods and Advances*, V. Zeitlin, Ed., vol. 2 of *Edited Series on Advances in Nonlinear Science and Complexity*, chapter 4, pp. 189–256, Elsevier, 2007.

[13] S. Mungkasi and S. G. Roberts, "A finite volume method for shallow water flows on triangular computational grids," in *Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS '11)*, pp. 79–84, Jakarta, Indonesia, December 2011.

[14] S. Mungkasi and S. G. Roberts, "On the best quantity reconstructions for a well balanced finite volume method used to solve the shallow water wave equations with a wet/dry interface," *ANZIAM Journal*, vol. 51, pp. C48–C65, 2009.

[15] S. Bi, J. Zhou, Y. Liu, and L. Song, "A finite volume method for modeling shallow flows with wet-dry fronts on adaptive Cartesian grids," *Mathematical Problems in Engineering*, vol. 2014, Article ID 209562, 20 pages, 2014.

[16] H. R. Vosoughifar, A. Dolatshah, and S. K. S. Shokouhi, "Discretization of multidimensional mathematical equations of dam break phenomena using a novel approach of finite volume method," *Journal of Applied Mathematics*, vol. 2013, Article ID 642485, 12 pages, 2013.

[17] E. Tadmor, "Numerical viscosity and the entropy condition for conservative difference schemes," *Mathematics of Computation*, vol. 43, no. 168, pp. 369–381, 1984.

[18] B. Perthame and E. Tadmor, "A kinetic equation with kinetic entropy functions for scalar conservation laws," *Communications in Mathematical Physics*, vol. 136, no. 3, pp. 501–517, 1991.

[19] E. Tadmor, "Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems," *Acta Numerica*, vol. 12, pp. 451–512, 2003.

[20] U. S. Fjordholm, R. Käppeli, S. Mishra, and E. Tadmor, "Construction of approximate entropy measure-valued solutions for hyperbolic systems of conservation laws," *Foundations of Computational Mathematics*, In press.

[21] E. Tadmor, "Perfect derivatives, conservative differences and entropy stable computation of hyperbolic conservation laws," *Discrete and Continuous Dynamical Systems*, vol. 36, no. 8, pp. 4579–4598, 2016.

[22] A. Kurganov, S. Noelle, and G. Petrova, "Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations," *SIAM Journal on Scientific Computing*, vol. 23, no. 3, pp. 707–740, 2001.

[23] S. Mungkasi, “A study of well-balanced finite volume methods and refinement indicators for the shallow water equations,” *Bulletin of the Australian Mathematical Society*, vol. 88, pp. 351–352, 2013, (PhD thesis, Australian National University, Canberra, 2012).

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization