Journal of Scientific Computing (2021) 87:88
https://doi.org/10.1007/s10915-021-01490-3

A High-Order Well-Balanced Positivity-Preserving Moving Mesh DG Method for the Shallow Water Equations With Non-Flat Bottom Topography

Min Zhang[1] · Weizhang Huang[2] · Jianxian Qiu[3]

Received: 26 June 2020 / Revised: 1 December 2020 / Accepted: 11 April 2021

Abstract

A rezoning-type adaptive moving mesh discontinuous Galerkin method is proposed for the numerical solution of the shallow water equations with non-flat bottom topography. The well-balance property is crucial to the simulation of perturbation waves over the lake-at-rest steady state such as waves on a lake or tsunami waves in the deep ocean. To ensure the well-balance and positivity-preserving properties, strategies are discussed in the use of slope limiting, positivity-preservation limiting, and data transferring between meshes. Particularly, it is suggested that a DG-interpolation scheme be used for the interpolation of both the flow variables and bottom topography from the old mesh to the new one and after each application of the positivity-preservation limiting on the water depth, a high-order correction be made to the approximation of the bottom topography according to the modifications in the water depth. Mesh adaptivity is realized using a moving mesh partial differential equation and a metric tensor based on the equilibrium variable and water depth. A motivation for the latter is to adapt the mesh according to both the perturbations of the lake-at-rest steady state and the water depth distribution. Numerical examples in one and two spatial dimensions are presented to demonstrate the well-balance and positivity-preserving properties of the method and its ability to capture small perturbations of the lake-at-rest steady state. They also show that

M. Zhang and J. Qiu were supported partly by National Natural Science Foundation (China) grant 12071392 and Science Challenge Project (China), No. TZ 2016002 . This work was carried out while M. Zhang was visiting the Department of Mathematics, the University of Kansas under the support by the China Scholarship Council (CSC: 201806310065).

✉ Jianxian Qiu
jxqiu@xmu.edu.cn

Min Zhang
minzhang2015@stu.xmu.edu.cn

Weizhang Huang
whuang@ku.edu

[1] School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, China

[2] Department of Mathematics, University of Kansas, Lawrence, Kansas 66045, USA

[3] School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, China

Published online: 05 May 2021

the mesh adaptation based on the equilibrium variable and water depth give more desirable results than that based on the commonly used entropy function.

Keywords Well-balance · DG-interpolation · High-order · Positivity preservation · Moving mesh DG method · Shallow water equations

Mathematics Subject Classification 65M50 · 65M60 · 76B15 · 35Q35

1 Introduction

We are interested in the numerical solution of the shallow water equations (SWEs) modeling the water flow over a surface that plays an important role in the ocean and hydraulic engineering, such as hydraulic jumps/shocks, open-channel flows, bore wave propagation, tidal flows in the estuary and coastal zones. The SWEs can be derived by integrating the Navier-Stokes equations in depth under the hydrostatic assumption when the depth of the flow is small compared to its horizontal dimensions. In non-dimensional form they read as

$$U_t + \nabla \cdot \mathbf{F}(U) = S(h, B), \tag{1.1}$$

where

$$\begin{aligned}
&U = (h, hu, hv)^T = (h, m, w)^T,\\
&\mathbf{F}(U) = \big(F(U), G(U)\big),\\
&F(U) = (hu, hu^2 + \frac{1}{2}gh^2, huv)^T = (m, \frac{m^2}{h} + \frac{1}{2}gh^2, \frac{mw}{h})^T,\\
&G(U) = (hv, huv, hv^2 + \frac{1}{2}gh^2)^T = (w, \frac{mw}{h}, \frac{w^2}{h} + \frac{1}{2}gh^2)^T,\\
&S(h, B) = \big(0, -hgB_x, -hgB_y\big)^T.
\end{aligned}$$

Here, h is the depth of water, (u, v) and $(m, w) = (hu, hv)$ are the velocities and discharges (in x and y directions), respectively, g is the gravitational acceleration constant, and B is the bottom topography which is a given time-independent function.

A distinct feature of the SWEs is that they admit steady-state solutions in which the flux gradients are balanced by the source term exactly. An important example is the so-called "lake at rest" solution,

$$u = 0, \quad v = 0, \quad h + B = C, \tag{1.2}$$

where C is a constant. It is crucial that this solution is preserved by numerical methods for the SWEs. Indeed, many physical phenomena, such as waves on a lake or tsunami waves in the deep ocean, can be described as small perturbations of this lake-at-rest steady state, and they are difficult to capture by a numerical method on an unrefined mesh unless the method can preserve the steady-state solution. This property is known as the C-property or well-balance.

The "exact C-property" concept was first introduced in 1994 by Bermudez and Vazquez [4]. Since then, a number of well-balanced numerical methods have been developed for the SWEs, for example, (Godunov-type) finite volume methods [3,4,25,48], finite difference/volume WENO methods [38–40], and discontinuous Galerkin (DG) methods [12,13,27,37,39–42]. DG methods have the advantages of high-order accuracy, high parallel efficiency, flexibility for hp-adaptivity and arbitrary geometry and meshes, and these make them particularly suited for the SWEs.

It should be pointed out that the SWEs form a nonlinear hyperbolic system and its solution can develop discontinuities such as hydraulic jumps/shock waves, rarefaction waves, and stationary state transitions. Small mesh spacings are required in the regions of these structures in order to resolve them, and mesh adaptation becomes a necessary tool to improve the computational accuracy and efficiency in the numerical solution of the SWEs. In the past decades, various work has been done in this direction. For example, Remacle et al. [33] studied an h-adaptive meshing procedure for the transient computation of the SWEs. Lamby et al. [24] proposed an adaptive multi-scale finite volume method for the SWEs with source terms, combining a quadtree grid generation strategy based on B-splines and fully adaptive multi-resolution methods. Tang [36] developed an adaptive moving structured mesh kinetic flux-vector splitting (KFVS) scheme for the SWEs without treating the bed slope source terms in order to balance the source terms and flux gradients. Although the well-balance is not considered in the work, the numerical results demonstrate that the adaptive moving mesh method leads to more accurate solutions than methods based on fixed meshes. Donat et al. [11] presented a well-balanced shock capturing adaptive mesh refinement (AMR) scheme for shallow water flows. They showed that the use of well-balanced interpolation operators is essential in order to maintain the well-balance property in the numerical solution computed with the AMR code. Zhou et al. [47] proposed a well-balanced adaptive moving mesh generalized Riemann problem (GRP)-based finite volume scheme for the SWEs with irregular bottom topography. A geometrical conservative interpolation scheme was used to update the solutions from the old mesh to the new one. Recently, Arpaia and Ricchiuto [1] considered several arbitrary Lagrangian-Eulerian (ALE) formulations of the SWEs on moving meshes and provided a discrete analog in the well-balanced finite volume and residual distribution framework.

In this work we study a moving mesh DG (MM-DG) method for the numerical solution of SWEs with non-flat bottom topography in one and two spatial dimensions. Our main objective is to show that the MM-DG method maintains high-order accuracy of DG discretization while preserving lake-at-rest steady state solutions and nonnegativity of the water depth. The method is based on the rezoning approach of moving mesh methods and contains three basic components at each time step, the adaptive mesh movement, the interpolation of the solution from the old mesh to the new one, and the numerical solution of the SWEs on the new mesh that is fixed for the time step. The adaptive mesh movement is based on the moving-mesh-partial-differential-equation (MMPDE) method [15–21] which is shown analytically and numerically in [22] to produce meshes free of tangling for any domain in any dimension. With the MMPDE method, the size, shape, and orientation of the mesh elements are controlled through a metric tensor, a symmetric and uniformly positive definite matrix-valued function defined on the physical domain and computed typically using the recovered Hessian of a DG solution. Instead of the commonly used entropy/total energy in the context of adaptive mesh shock wave simulation, we propose to use the equilibrium variable $\mathcal{E} = \frac{1}{2}(u^2+v^2)+g(h+B)$ and the water depth h to construct the metric tensor. A motivation for this is that the mesh is adapted to both the perturbations of the lake-at-rest steady state through $\mathcal{E}$ and the water depth distribution through h.

We use a fixed mesh well-balanced Runge-Kutta DG method [40–42] for the numerical solution of the SWEs on the new mesh, a DG-interpolation scheme [45] for the interpolation of the solution and bottom topography from the old mesh to the new one, the TVB slope limiter [8] to avoid spurious oscillations, and a linear scaling limiter [23,49,50] for positivity preservation. Although these are existing schemes/techniques, great caution is needed in their use to warrant the well-balance and positivity-preserving (PP) properties of the overall MM-DG method.

The interpolation of the solution from the old mesh to the new one is a key component for the MM-DG method to maintain high-order accuracy, preserve the lake-at-rest steady state solution, and conserve the mass. Several conservative interpolation schemes between deforming meshes have been investigated; see, e.g. [26,35]. We use the recently developed DG-interpolation scheme [45] for the purpose. This scheme is known to work for arbitrary bounded mesh deformation, has high-order accuracy, conserves the mass, and preserves constant solution, and, with a linear scaling PP limiter [23,49,50], can preserve the nonnegativity of the function to be interpolated. We also use the same scheme to update the bottom topography on the new mesh. This update is necessary due to the movement of the mesh and must be done using a same scheme as for U to ensure the well-balance property.

Note that we need to ensure the nonnegativity of the water depth in the interpolation for the dry situation. We use the PP-DG-interpolation scheme (the DG-interpolation scheme with the linear scaling PP limiter [45]) for the water depth for this purpose. However, the PP limiter will destroy the well-balance property. In order to restore the property, we apply the same DG-interpolation scheme to the total surface level and obtain a new approximation for the bottom topography by subtracting the water depth approximation from the total surface level approximation. This new approximation of the bottom topography differs from the old one by a high-order correction that corresponds to the modifications in the water depth due to the PP limiting.

Since spurious oscillations and even nonlinear instability can occur in numerical solutions for the SWEs, a nonlinear slope limiter is needed after each Runge-Kutta stage. We apply the TVB limiter [6–8] to the local characteristic variables based on the variables $\big((h+B), m, w\big)$. Note that $\big((h+B), m, w\big)$, rather the original variables $\big(h, m, w\big)$, is used to ensure the well-balance property. Moreover, we need to apply the PP limiter to ensure the nonnegativity of the water depth every time after we use the TVB limiter. Since the TVB procedure preserves the cell averages of the water depth, we can use the linear scaling limiter [23,49,50] for this purpose. Once again, the PP limiter destroys the well-balance property. To recover the property, we propose to make a high-order correction to the approximation of the bottom topography according to the modifications in the water depth due to the PP limiting. Numerical examples show that this strategy works out well.

The remainder of the paper is organized as follows. sect. 2 is devoted to the description of the overall procedure of the well-balanced MM-DG method and its DG and Runge-Kutta discretization for the SWEs. The DG-interpolation scheme and its properties are described in sect. 3. In sect. 4, the MMPDE moving mesh method for the generation of the new mesh is discussed. Numerical results obtained with the MM-DG method for a selection of one- and two-dimensional examples are presented in sect. 5. Finally, sect. 6 contains conclusions and further comments.

2 The Well-Balanced MM-DG Scheme for the SWEs

In this section we describe the well-balanced MM-DG method for the numerical solution of the SWEs on moving simplicial meshes. We use here a rezoning approach where the unknown variables are interpolated from the old mesh to the new one and the SWEs are solved on the new mesh. In this section we focus on the overall description of the method in two dimensions (the method is similar in one dimension), the discretization of the SWEs using DG in space and a Runge-Kutta scheme in time, and the well-balance property. A

DG-interpolation scheme is described in sect. 3 and the generation of the new mesh using the MMPDE moving mesh method is discussed in sect. 4.

We assume that the numerical solutions $U_h^n = (h_h^n, m_h^n, w_h^n)^T$ at physical time t_n on the mesh $\mathcal{T}_h^n$ are known. (Notice that, in h_h, h is the water depth while the subscript h is used to indicate that h_h is a numerical approximation to the water depth.) We also assume that a new mesh $\mathcal{T}_h^{n+1}$, which has the same number of vertices and elements and the same connectivity as $\mathcal{T}_h^n$, has been generated based on U_h^n and $\mathcal{T}_h^n$ (cf. sect. 4). We denote an interpolant of $U_h^n = (h_h^n, m_h^n, w_h^n)^T$ on the new mesh $\mathcal{T}_h^{n+1}$ by $\tilde{U}_h^n = (\tilde{h}_h^n, \tilde{m}_h^n, \tilde{w}_h^n)^T$. As we will see later in this section, caution is needed in choosing the interpolation scheme to make the MM-DG method to be conservative, positivity-preserving, and well-balanced.

We now describe the DG discretization of the SWEs on $\mathcal{T}_h^{n+1}$. Let

$$\mathcal{V}_h^{k,n+1} = \{q \in L^2(\Omega) : \ q|_K \in \mathbb{P}^k(K), \ \forall K \in \mathcal{T}_h^{n+1}\}, \tag{2.1}$$

where

$$\mathbb{P}^k(K) = \text{span}\{\phi_K^1, \phi_K^2, ..., \phi_K^{n_b}\}$$

is the set of polynomials of degree at most k ($k \ge 1$) on element K and $\phi_K^j = \phi_K^j(\boldsymbol{x}), \ j = 1, ..., n_b \equiv (k+1)(k+2)/2$, denote the basis functions of $\mathbb{P}^k(K)$. Multiplying (1.1) by a test function $\phi \in \mathcal{V}_h^{k,n+1}$, integrating the resulting equation over $K \in \mathcal{T}_h^{n+1}$, and using the divergence theorem, we get

$$\frac{d}{dt}\int_K U\phi d\boldsymbol{x} - \int_K \mathbf{F}(U)\cdot\nabla\phi d\boldsymbol{x} + \int_{\partial K}\mathbf{F}(U)\cdot\boldsymbol{n}_K\phi ds = \int_K S(h, B)\phi d\boldsymbol{x}, \tag{2.2}$$

where $\boldsymbol{n}_K = (n_x, n_y)^T$ is the outward unit normal to the boundary ∂K.

Recall that each interior edge is shared by two triangular elements. Thus, on any (interior) edge of K, U_h can be defined using its value in K or in the other element sharing the common edge with K. These values are denoted by $U_{h,K}^{int}$ and $U_{h,K}^{ext}$, respectively. Moreover, we define the global Lax-Friedrichs numerical flux to approximate the flux function $\mathbf{F}(U)\cdot\boldsymbol{n}_K$ on the edge $e_K \in \partial K$ as

$$\hat{\mathbf{F}}|_{e_K} = \hat{\mathbf{F}}(U_{h,K}^{int}, U_{h,K}^{ext}, \boldsymbol{n}_K^e) = \frac{1}{2}\Big(\big(\mathbf{F}(U_{h,K}^{int}) + \mathbf{F}(U_{h,K}^{ext})\big)\cdot\boldsymbol{n}_K^e - \alpha(U_{h,K}^{ext} - U_{h,K}^{int})\Big), \tag{2.3}$$

where $\alpha = \max\limits_{K,e}\Big(\max\limits_{s}(|\lambda^s(U_{h,K}^{int})|, |\lambda^s(U_{h,K}^{ext})|)\Big)$. Here,

$$\lambda^1(U) = un_x + vn_y - c, \quad \lambda^2(U) = un_x + vn_y, \quad \lambda^3(U) = un_x + vn_y + c, \tag{2.4}$$

are the eigenvalues of the Jacobian matrix

$$\mathbf{F}'(U)\cdot\boldsymbol{n}_K = \begin{pmatrix} 0 & n_x & n_y \\ (c^2 - u^2)n_x - uvn_y & 2un_x + vn_y & un_y \\ -uvn_x + (c^2 - v^2)n_y & vn_x & un_x + 2vn_y \end{pmatrix}, \tag{2.5}$$

where $c = \sqrt{gh}$ is the sound speed. The semi-discrete DG scheme is then to find $U_h \in \mathcal{V}_h^{k,n+1}$ such that, $\forall\phi \in \mathcal{V}_h^{k,n+1}$,

$$\frac{d}{dt}\int_K U_h\phi d\boldsymbol{x} + \sum_{e_K\in\partial K}\int_{e_K}\phi\hat{\mathbf{F}}|_{e_K}ds - \int_K \mathbf{F}(U_h)\cdot\nabla\phi d\boldsymbol{x} = \int_K S(h_h, B_h)\phi d\boldsymbol{x}, \tag{2.6}$$

where $\hat{\mathbf{F}}|_{e_K}$ is the numerical flux defined (2.3). Define the residual as

$$\begin{aligned} R_{h,K}(U_h, \phi, B_h) = & \int_K S(h_h, B_h)\phi d\boldsymbol{x} + \int_K \mathbf{F}(U_h) \cdot \nabla \phi d\boldsymbol{x} \\ & - \sum_{e_K \in \partial K} \int_{e_K} \phi \hat{\mathbf{F}}|_{e_K} ds, \quad \forall K \in \mathcal{T}_h^{n+1}. \end{aligned} \tag{2.7}$$

Generally speaking, the residual does not vanish for the lake-at-rest steady state, i.e., the scheme (2.6) is not well-balanced. Our objective is to find a special numerical flux $\hat{\mathbf{F}}^*|_{e_K}$ by modifying $\hat{\mathbf{F}}|_{e_K}$ so that the residual becomes zero for the lake-at-rest steady state. We employ the hydrostatic reconstruction technique [3,40–42] to modify $\hat{\mathbf{F}}|_{e_K}$. It is noted that the bottom topography function $B(\boldsymbol{x})$ needs to be projected into the finite element space $\mathcal{V}_h^{k,n+1}$. We denote this approximation by B_h^{n+1} or B_h without confusion. We will discuss the projection/interpolation of $B(\boldsymbol{x})$ later in this section and a DG-interpolation scheme in sect. 3. Let

$$\begin{aligned} h_{h,K}^{*,int}|_{e_K} &= \max\Big(0, h_{h,K}^{int}|_{e_K} + B_{h,K}^{int}|_{e_K} - \max\big(B_{h,K}^{int}|_{e_K}, B_{h,K}^{ext}|_{e_K}\big)\Big), \\ h_{h,K}^{*,ext}|_{e_K} &= \max\Big(0, h_{h,K}^{ext}|_{e_K} + B_{h,K}^{ext}|_{e_K} - \max\big(B_{h,K}^{int}|_{e_K}, B_{h,K}^{ext}|_{e_K}\big)\Big). \end{aligned} \tag{2.8}$$

Recall that B_h is the DG approximation of B and is discontinuous on element edges. In (2.8), the value of B_h on e_K is taken as $\max\big(B_{h,K}^{int}|_{e_K}, B_{h,K}^{ext}|_{e_K}\big)$, which was first proposed by Audusse et al. [3] and is now considered as the standard choice. Some other researchers suggest to use $\min\big(B_{h,K}^{int}|_{e_K}, B_{h,K}^{ext}|_{e_K}\big)$ [32,43]. The optimal choice of the value of B_h on e_K remains an open problem, which seems to be closely related to the non-uniqueness of the Riemann problem [10]. With (2.8), we force $h_{h,K}^{*,int}|_{e_K} \geq 0$ and $h_{h,K}^{*,ext}|_{e_K} \geq 0$ while trying to satisfy

$$\begin{aligned} h_{h,K}^{*,int} + \max\big(B_{h,K}^{int}|_{e_K}, B_{h,K}^{ext}|_{e_K}\big) &= h_{h,K}^{int}|_{e_K} + B_{h,K}^{int}|_{e_K}, \\ h_{h,K}^{*,ext} + \max\big(B_{h,K}^{int}|_{e_K}, B_{h,K}^{ext}|_{e_K}\big) &= h_{h,K}^{ext}|_{e_K} + B_{h,K}^{ext}|_{e_K}. \end{aligned}$$

Then, we redefine the interior and exterior values of U_h as

$$U_{h,K}^{*,int}|_{e_K} = \begin{pmatrix} h_{h,K}^{*,int}|_{e_K} \\ \frac{h_{h,K}^{*,int}|_{e_K}}{h_{h,K}^{int}|_{e_K}} m_{h,K}^{int}|_{e_K} \\ \frac{h_{h,K}^{*,int}|_{e_K}}{h_{h,K}^{int}|_{e_K}} w_{h,K}^{int}|_{e_K} \end{pmatrix} \quad \text{and} \quad U_{h,K}^{*,ext}|_{e_K} = \begin{pmatrix} h_{h,K}^{*,ext}|_{e_K} \\ \frac{h_{h,K}^{*,ext}|_{e_K}}{h_{h,K}^{ext}|_{e_K}} m_{h,K}^{ext}|_{e_K} \\ \frac{h_{h,K}^{*,ext}|_{e_K}}{h_{h,K}^{ext}|_{e_K}} w_{h,K}^{ext}|_{e_K} \end{pmatrix}. \tag{2.9}$$

The numerical flux on the edge $e_K \in \partial K$ is modified as

$$\hat{\mathbf{F}}^*|_{e_K} = \hat{\mathbf{F}}(U_{h,K}^{*,int}, U_{h,K}^{*,ext}, \boldsymbol{n}_K^e) + \Delta_{e_K}^* \cdot \boldsymbol{n}_K^e, \tag{2.10}$$

where

$$\Delta_{e_K}^* = \begin{pmatrix} 0 & 0 \\ \frac{g}{2}(h_{h,K}^{int}|_{e_K})^2 - \frac{g}{2}(h_{h,K}^{*,int}|_{e_K})^2 & 0 \\ 0 & \frac{g}{2}(h_{h,K}^{int}|_{e_K})^2 - \frac{g}{2}(h_{h,K}^{*,int}|_{e_K})^2 \end{pmatrix}. \tag{2.11}$$

The correction term is added to ensure

$$\hat{\mathbf{F}}^*|_{e_K} = \mathbf{F}\big(U_{h,K}^{int}\big) \cdot \boldsymbol{n}_K^e \tag{2.12}$$

when the lake-at-rest steady state is reached. Replacing $\hat{\mathbf{F}}|_{e_K}$ by $\hat{\mathbf{F}}^*|_{e_K}$ in (2.6), we have

$$\begin{aligned}\frac{d}{dt}\int_K U_h\phi d\boldsymbol{x} + \sum_{e_K\in\partial K}\int_{e_K}\phi\hat{\mathbf{F}}^*|_{e_K}ds - \int_K \mathbf{F}(U_h)\cdot\nabla\phi d\boldsymbol{x}\\ = \int_K S(h_h, B_h)\phi d\boldsymbol{x}, \quad \forall\phi\in\mathcal{V}_h^{k,n+1}.\end{aligned} \tag{2.13}$$

Denote the residual for this scheme as

$$\begin{aligned}R^*_{h,K}(U_h,\phi,B_h) = \int_K S(h_h,B_h)\phi d\boldsymbol{x} + \int_K \mathbf{F}(U_h)\cdot\nabla\phi d\boldsymbol{x}\\ - \sum_{e_K\in\partial K}\int_{e_K}\phi\hat{\mathbf{F}}^*|_{e_K}ds, \quad \forall K\in\mathcal{T}_h^{n+1}.\end{aligned} \tag{2.14}$$

From (2.12), it can be shown that this residual vanishes for the lake-at-rest steady state if Gaussian quadrature rules with a suitable degree of precision are used to calculate all integrals in the (2.14) exactly. Since h_h, B_h and ϕ are polynomials of degree at most k in each element K and $m_h = 0$ and $w_h = 0$ for the lake-at-rest steady state, $S(h_h, B_h)\phi$ and $\mathbf{F}(U_h)\cdot\nabla\phi$ are polynomials of degree at most $(3k-1)$ while $\phi\hat{\mathbf{F}}^*$ is a polynomial of degree at most $3k$. Thus, to ensure that all integrals in (2.14) are calculated exactly for the lake-at-rest steady state, we should use a Gaussian quadrature rule of degree of precision at least $(3k-1)$ for the area integrals and a Gaussian quadrature rule of degree of precision at least $3k$ for the line integrals.

As a result, the scheme (2.13) is well-balanced. Moreover, the scheme (2.13) is known to be $(k+1)$-th order in space.

We can rewrite (2.13) into

$$\frac{d}{dt}\int_K U_h\phi d\boldsymbol{x} = R^*_{h,K}(U_h,\phi,B_h), \quad \forall\phi\in\mathcal{V}_h^{k,n+1}. \tag{2.15}$$

A third-order strong stability preserving (SSP) Runge-Kutta scheme is used to integrate (2.15) in time. We have, for any $K\in\mathcal{T}_h^{n+1}$,

$$\begin{cases}\displaystyle\int_K U_h^{(1)}\phi d\boldsymbol{x} = \int_K \tilde{U}_h^n\phi d\boldsymbol{x} + \Delta t_n R^*_{h,K}(\tilde{U}_h^n,\phi,B_h^{n+1}), \quad \forall\phi\in\mathcal{V}_h^{k,n+1}\\ \displaystyle\int_K U_h^{(2)}\phi d\boldsymbol{x} = \frac{3}{4}\int_K \tilde{U}_h^n\phi d\boldsymbol{x} + \frac{1}{4}\int_K U_h^{(1)}\phi d\boldsymbol{x}\\ \displaystyle\qquad\qquad + \frac{\Delta t_n}{4}R^*_{h,K}(U_h^{(1)},\phi,B_h^{n+1}), \quad \forall\phi\in\mathcal{V}_h^{k,n+1}\\ \displaystyle\int_K U_h^{n+1}\phi d\boldsymbol{x} = \frac{1}{3}\int_K \tilde{U}_h^n\phi d\boldsymbol{x} + \frac{2}{3}\int_K U_h^{(2)}\phi d\boldsymbol{x}\\ \displaystyle\qquad\qquad + \frac{2\Delta t_n}{3}R^*_{h,K}(U_h^{(2)},\phi,B_h^{n+1}), \quad \forall\phi\in\mathcal{V}_h^{k,n+1}\end{cases} \tag{2.16}$$

where $\Delta t_n = t_{n+1} - t_n$ and B_h^{n+1} is a polynomial approximation of the bottom topography on $\mathcal{T}_h^{n+1}$. To ensure stability, we choose Δt_n according to the CFL condition [9] as

$$\Delta t_n = \frac{C_{cfl}}{\max\limits_{K,e}\left(\max\limits_s(|\lambda^s((\tilde{U}^n_{h,K})^{int})|, |\lambda^s((\tilde{U}^n_{h,K})^{ext})|)\right)}\cdot\min\left(a^n_{min}, a^{n+1}_{min}\right), \tag{2.17}$$

where C_{cfl} is a constant and a^n_{min} and a^{n+1}_{min} are the minimum element heights of $\mathcal{T}_h^n$ and $\mathcal{T}_h^{n+1}$, respectively.

We are now ready to discuss the interpolation of $U_h^n = (h_h^n, m_h^n, w_h^n)^T$ from the old mesh $\mathcal{T}_h^n$ to the new one $\mathcal{T}_h^{n+1}$. L^2-projection is inconvenient to use here since it requires finding the intersection between elements in $\mathcal{T}_h^n$ and $\mathcal{T}_h^{n+1}$ and performing numerical integration thereon, which is known to be a daunting, if not impossible, task in programming. Moreover, for the dry situation, we need to ensure the nonnegativity of the water depth in the interpolation procedure. Thus, we use the PP-DG-interpolation scheme (the DG-interpolation scheme with the linear scaling PP limiter), see [45] or sect. 3) for the water depth and the DG-interpolation scheme (without PP limiter) for the water discharges, i.e.,

$$\tilde{h}_h^n = \text{PP-DGInterp}(h_h^n), \quad \tilde{m}_h^n = \text{DGInterp}(m_h^n), \quad \tilde{w}_h^n = \text{DGInterp}(w_h^n). \tag{2.18}$$

Note that this interpolation scheme does not maintain the well-balance property in general. To recover the property, we apply the same DG-interpolation scheme to the total surface level $h_h^n + B_h^n$ and define the new approximation to the bottom topography on the new mesh as

$$B_h^{n+1} = \text{DGInterp}(h_h^n + B_h^n) - \tilde{h}_h^n. \tag{2.19}$$

From the fact that the DG-interpolation scheme preserves constant solutions (cf. sect. 3), we know that this procedure restores the well-balance property. Moreover, from the linearity of the interpolation scheme, we can rewrite (2.19) into

$$B_h^{n+1} = \text{DGInterp}(B_h^n) - \Big(\text{PP-DGInterp}(h_h^n) - \text{DGInterp}(h_h^n) \Big),$$

which implies that B_h^{n+1} differs from DGInterp(B_h^n) by a high-order correction corresponding to the changes in the water depth due to the PP limiting.

It is interesting to point out that the well-balance property cannot be achieved if B_h^{n+1} is computed by L^2-projecting B into the DG approximation space defined on $\mathcal{T}_h^{n+1}$ (cf. numerical results in sect. 5).

We now show that the fully discrete scheme (2.16) is well-balanced. To this end, we assume that $\tilde{U}_h^n$ satisfies (1.2) (with B replaced by B_h^{n+1}). Recall that $R_{h,K}^*$ vanishes for the lake-at-rest steady state. Then, scheme (2.16) reduces to

$$\begin{cases} \displaystyle\int_K U_h^{(1)} \phi d\boldsymbol{x} = \int_K \tilde{U}_h^n \phi d\boldsymbol{x}, \quad \forall \phi \in \mathcal{V}_h^{k,n+1} \\ \displaystyle\int_K U_h^{(2)} \phi d\boldsymbol{x} = \frac{3}{4} \int_K \tilde{U}_h^n \phi d\boldsymbol{x} + \frac{1}{4} \int_K U_h^{(1)} \phi d\boldsymbol{x}, \quad \forall \phi \in \mathcal{V}_h^{k,n+1} \\ \displaystyle\int_K U_h^{n+1} \phi d\boldsymbol{x} = \frac{1}{3} \int_K \tilde{U}_h^n \phi d\boldsymbol{x} + \frac{2}{3} \int_K U_h^{(2)} \phi d\boldsymbol{x}, \quad \forall \phi \in \mathcal{V}_h^{k,n+1}. \end{cases} \tag{2.20}$$

This implies that $U_h^{n+1} \equiv U_h^{(2)} \equiv U_h^{(1)} \equiv \tilde{U}_h^n$. Thus, U_h^{n+1} also satisfies (1.2) (with B replaced by B_h^{n+1}) and (2.16) is well-balanced.

Since spurious oscillations and even nonlinear instability can occur in numerical solutions, we need to apply a nonlinear limiter after each Runge-Kutta stage. However, caution must be taken since this limiting procedure can destroy the well-balance property and the nonnegativity of the water depth. Following [3,40–42,48], we use the TVB limiter [8] for the local characteristic variables based on the variables $\big((h_h + B_h), m_h, w_h\big)^T$ (instead of $\big(h_h, m_h, w_h\big)^T$) to obtain $\big((h_h + B_h)^{mod}, m_h^{mod}, w_h^{mod}\big)^T$. Define $h_h^{mod} = (h_h + B_h)^{mod} - B_h$. It is known that this limiting maintains the well-balance property and preserves the cell averages of h_h but does not necessarily preserve the nonnegativity of h_h. Thus, after each application of the TVB limiter, we apply the linear scaling PP limiter of [23,49,50] to h_h^{mod} and denote the result by PP(h_h^{mod}). Once again, this PP limiter destroys the well-balance property. To restore the property, we make a high-order correction to the approximation of B according to the changes in the water depth due to the PP limiting, i.e.,

$$\hat{B}_h = B_h - (\mathrm{PP}(h_h^{mod}) - h_h^{mod}). \tag{2.21}$$

Notice that $\hat{B}_h$ has the same cell averages as B_h.

Remark 2.1 In practical implementation when the water depth is close to zero, large velocities $u = (hu)/h$ and $v = (hv)/h$ can result in due to small numerical error in hu and hv, which in turn can lead to very small time steps with the CFL condition. Following [34,41], we set $u = 0$ and $v = 0$ when $h < 10^{-6}$ in our computation.

To conclude this section, we summarize the procedure of the MM-DG method in Algorithm 1. From the above discussion, we have seen that the interpolation and the fully discrete scheme (2.16) are well-balanced. Hence, the MM-DG method is well-balanced.

Algorithm 1 The MM-DG method for the SWEs on moving meshes.

```
0. Initialization. Project the initial physical variables and bottom topography into the DG space $\mathcal{V}_h^{k,0}$ to
   obtain $U_h^0 = (h_h^0, m_h^0, w_h^0)^T$ and $B_h^0$. For $n = 0, 1, ...,$ do
1. Mesh adaptation. Generate the new mesh $\mathcal{T}_h^{n+1}$ using an MMPDE-based moving mesh method (cf.
   sect. 4).
2. Solution interpolation. Use the DG-interpolation procedure to update $U_h^n = (h_h^n, m_h^n, w_h^n)^T$ and $B_h^n$
   from $\mathcal{T}_h^n$ to $\mathcal{T}_h^{n+1}$ to obtain $\tilde{U}_h^n = (\tilde{h}_h^n, \tilde{m}_h^n, \tilde{w}_h^n)^T$ and $B_h^{n+1}$ (cf. (2.18) and (2.19)).
3. Solution of the SWEs on the new mesh. Integrate the SWEs from $t_n$ to $t_{n+1}$ on the new mesh $\mathcal{T}_h^{n+1}$ using
   the MM-DG scheme (2.16) to obtain $U_h^{n+1} = (h_h^{n+1}, m_h^{n+1}, w_h^{n+1})^T$. At each Runge-Kutta stage, the
   TVB limiter is applied to the local characteristic variables based on the variables $\big((h_h + B_h), m_h, w_h\big)^T$,
   followed by linear scaling PP limiter for the water depth and a corresponding high-order correction to the
   approximation of the bottom topography (cf. (2.21).
```

3 A conservative DG-interpolation scheme

In this section we briefly describe a DG-interpolation scheme [45] for the interpolation of a numerical solution q_h^n from the old mesh $\mathcal{T}_h^n$ to the new one $\mathcal{T}_h^{n+1}$. The meshes are assumed to have the same number of vertices and elements and the same connectivity and can be viewed as a deformation from each other. The scheme works for arbitrary bounded mesh deformation. It has high-order accuracy, conserves the mass, positivity-preserving, and satisfies the geometric conservation law (GCL) and thus preserves constant solutions, and, with a linear scaling PP limiter [23,49,50], can preserve the nonnegativity of the function to be interpolated. The reader is referred to [45] for the detail.

The interpolation problem between $\mathcal{T}_h^n$ and $\mathcal{T}_h^{n+1}$ is mathematically equivalent to solving the linear convection equation

$$\frac{\partial q}{\partial \varsigma}(\boldsymbol{x}, \varsigma) = 0, \quad (\boldsymbol{x}, \varsigma) \in \mathcal{D} \times (0, 1] \tag{3.1}$$

on the moving mesh $\mathcal{T}_h(\varsigma)$ that is defined as the linear interpolant of $\mathcal{T}_h^n$ and $\mathcal{T}_h^{n+1}$. Specifically, $\mathcal{T}_h(\varsigma)$ has the same number of elements and vertices and the same connectivity as $\mathcal{T}_h^n$

and $\mathcal{T}_h^{n+1}$ and its nodal positions and displacements (or deformation) are given by

$$\boldsymbol{x}_i(\varsigma) = (1-\varsigma)\boldsymbol{x}_i^n + \varsigma \boldsymbol{x}_i^{n+1}, \quad i = 1, ..., N_v \tag{3.2}$$

$$\dot{\boldsymbol{x}}_i = \boldsymbol{x}_i^{n+1} - \boldsymbol{x}_i^n, \quad i = 1, ..., N_v. \tag{3.3}$$

We define the piecewise linear mesh deformation function as

$$\dot{\boldsymbol{X}}(\boldsymbol{x}, \varsigma) = \sum_{i=1}^{N_v} \dot{\boldsymbol{x}}_i \phi_i(\boldsymbol{x}, \varsigma), \tag{3.4}$$

where ϕ_i is the linear basis function associated with the vertex $\boldsymbol{x}_i$.

We use a quasi-Lagrangian MM-DG method [28,44] to solve (3.1). The semi-discrete scheme for (3.1) is to find $q_h \in \mathcal{V}_h^k(\varsigma)$ such that

$$\frac{d}{d\varsigma}\int_K q_h \phi d\boldsymbol{x} + \sum_{e\in\partial K}\int_e \phi F_e(q_K^{int}, q_K^{ext}) ds + \int_K (q_h \dot{\boldsymbol{X}}) \cdot \nabla \phi d\boldsymbol{x} = 0, \quad \forall \phi \in \mathcal{V}_h^k(\varsigma) \tag{3.5}$$

where $F_e(q_K^{int}, q_K^{ext}) \approx -q\dot{\boldsymbol{X}} \cdot \boldsymbol{n}_K$ is a numerical flux defined on $e \in \partial K$, and q_K^{int} and q_K^{ext} denote the values of q_h on K and on the element (denoted by K') sharing the common edge e with K. We use the local Lax-Friedrichs numerical flux, i.e.,

$$F_e(q_K^{int}, q_K^{ext}) = \frac{1}{2}\Big(\big(-q_K^{int}\dot{\boldsymbol{X}}^e - q_K^{ext}\dot{\boldsymbol{X}}^e\big)\cdot \boldsymbol{n}_K^e - \alpha_e(q_K^{ext} - q_K^{int})\Big), \quad \forall e \in \partial K \tag{3.6}$$

where $\dot{\boldsymbol{X}}^e$ denotes the restriction of $\dot{\boldsymbol{X}}$ on e and $\alpha_e = \max\big(|\dot{\boldsymbol{X}}^e \cdot \boldsymbol{n}_K^e|, |\dot{\boldsymbol{X}}^e \cdot \boldsymbol{n}_{K'}^e|\big)$. Note that the numerical flux is zero on the boundary due to $\dot{\boldsymbol{X}}^e \cdot \boldsymbol{n}_K^e = 0$ if $e \in \partial\mathcal{D}$.

We consider the quasi-time instants

$$0 = \varsigma^0 < \varsigma^1 < ... < \varsigma^\nu < \varsigma^{\nu+1} < ... < \varsigma^{N_\varsigma} = 1, \qquad \Delta\varsigma^\nu = \varsigma^{\nu+1} - \varsigma^\nu.$$

To ensure the stability of the scheme, we choose $\Delta\varsigma$ according to the CFL condition

$$\Delta\varsigma = \frac{\mathcal{C}_p}{\max\limits_{e,K} |\dot{\boldsymbol{X}}^e \cdot \boldsymbol{n}_K^e|} \cdot \min\big(a_{min}^n, a_{min}^{n+1}\big). \tag{3.7}$$

where $\mathcal{C}_p$ is the CFL condition constant number and usually taken as $1/(2k+2)$ in our computation, unless stated otherwise. A third-order SSP Runge-Kutta scheme is used for time integration. Then, the fully-discrete quasi-Lagrangian MM-DG scheme for (3.1) is:

$$\begin{cases} \displaystyle\int_{K^{\nu,(1)}} q_h^{\nu,(1)}\phi^{\nu,(1)} d\boldsymbol{x} = \int_{K^\nu} q_h^\nu \phi^\nu d\boldsymbol{x} + \Delta\varsigma^\nu \mathcal{A}(q_h^\nu, \phi^\nu)|_{K^\nu}, \\ \displaystyle\int_{K^{\nu,(2)}} q_h^{\nu,(2)}\phi^{\nu,(2)} d\boldsymbol{x} = \frac{3}{4}\int_{K^\nu} q_h^\nu \phi^\nu d\boldsymbol{x} + \frac{1}{4}\int_{K^{\nu,(1)}} q_h^{\nu,(1)}\phi^{\nu,(1)} d\boldsymbol{x} \\ \qquad\qquad\qquad\qquad\displaystyle + \frac{\Delta\varsigma^\nu}{4}\mathcal{A}(q_h^{\nu,(1)}, \phi^{\nu,(1)})|_{K^{\nu,(1)}}, \\ \displaystyle\int_{K^{\nu+1}} q_h^{\nu+1}\phi^{\nu+1} d\boldsymbol{x} = \frac{1}{3}\int_{K^\nu} q_h^\nu \phi^\nu d\boldsymbol{x} + \frac{2}{3}\int_{K^{\nu,(2)}} q_h^{\nu,(2)}\phi^{\nu,(2)} d\boldsymbol{x} \\ \qquad\qquad\qquad\qquad\displaystyle + \frac{2\Delta\varsigma^\nu}{3}\mathcal{A}(q_h^{\nu,(2)}, \phi^{\nu,(2)})|_{K^{\nu,(2)}}, \end{cases} \tag{3.8}$$

where $(q_h^{\nu,(1)}, \phi^{\nu,(1)}, K^{\nu,(1)})$ are the stage values at $\varsigma = \varsigma^{\nu+1}$, $(q_h^{\nu,(2)}, \upsilon^{\nu,(2)}, K^{\nu,(2)})$ are the values at $\varsigma = \varsigma^{\nu+\frac{1}{2}}$, $(q_h^{\nu+1}, \phi^{\nu+1}, K^{\nu+1})$ are at $\varsigma = \varsigma^{\nu+1}$, and

$$\mathcal{A}(q_h, \phi)|_K = -\sum_{e\in\partial K} \int_e \phi F_e(q_K^{int}, q_K^{ext})ds - \int_K (q_h\dot{\boldsymbol{X}}) \cdot \nabla\phi d\boldsymbol{x}.$$

It is emphasized that the volume of K needs to be updated at these stages as

$$\begin{aligned}
|K^{\nu,(1)}| &= |K^\nu| + \Delta\varsigma^\nu |K^\nu| \nabla \cdot \dot{\boldsymbol{X}}|_{K^\nu},\\
|K^{\nu,(2)}| &= \frac{3}{4}|K^\nu| + \frac{1}{4}\big(|K^{\nu,(1)}| + \Delta\varsigma^\nu |K^{\nu,(1)}|\nabla \cdot \dot{\boldsymbol{X}}|_{K^{\nu,(1)}}\big),\\
|K^{\nu+1}| &= \frac{1}{3}|K^\nu| + \frac{2}{3}\big(|K^{\nu,(2)}| + \Delta\varsigma^\nu |K^{\nu,(2)}|\nabla \cdot \dot{\boldsymbol{X}}|_{K^{\nu,(2)}}\big).
\end{aligned} \tag{3.9}$$

2 This ensures that the geometric conservation law be satisfied and constant solutions be preserved. Satisfaction of the geometric conservation laws has been studied extensively; see, for example, Mavriplis and Yang [30], Cheng and Shu [5], and Mavriplis and Nastase [31] in the context of high-order methods.

Proposition 3.1 ([45]) *The DG-interpolation scheme* (3.8) *conserves the mass, i.e.,*

$$\sum_{K^{\nu+1}} \int_{K^{\nu+1}} q_h^{\nu+1} d\boldsymbol{x} = \sum_{K^\nu} \int_{K^\nu} q_h^\nu d\boldsymbol{x}, \quad \nu = 0, 1, ... \tag{3.10}$$

Proposition 3.2 *The DG-interpolation scheme* (3.8) *preserves constant solutions, i.e., for any arbitrary constant C, $q_h^\nu \equiv C$ implies $q_h^{\nu+1} \equiv C$ if the element volume is updated according to* (3.9).

Proof This proposition follows from the same proof for the geometric conservation law (Lamma 2.1) in [45]. □

Proposition 3.3 ([45]) *When the linear scaling PP limiter [23,49,50] is further applied at each step, the DG-interpolation scheme* (3.8) *preserves the solution positivity in the sense that, for $\nu = 0, 1, ...$, if q_h^ν has nonnegative cell averages for all elements and is nonnegative at a set of special points at each element of $\mathcal{T}_h(\varsigma^\nu)$, then $q_h^{\nu+1}$ has nonnegative cell averages for all elements and is nonnegative at the corresponding set of special points at each element of $\mathcal{T}_h(\varsigma^{\nu+1})$. The DG-interpolation with the linear scaling PP limiter we call as the PP-DG-interpolation.*

Remark 3.4 It is reasonable to assume that the mesh movement speed is finite during the course of computation. Under this assumption, the deformation between $\mathcal{T}_h^n$ and $\mathcal{T}_h^{n+1}$ is proportional to Δt_n, i.e.,

$$\dot{\boldsymbol{x}}_i = \boldsymbol{x}_i^{n+1} - \boldsymbol{x}_i^n = \mathcal{O}(\Delta t_n).$$

From (3.4), we have

$$\max_{e,K} |\dot{\boldsymbol{X}}^e \cdot \boldsymbol{n}_K^e| = \mathcal{O}(\Delta t_n).$$

Combining this with (2.17) and (3.7), we get

$$\Delta\varsigma = \mathcal{O}\left(\max_{K,e}\left(\max_s(|\lambda^s((\tilde{U}_{h,K}^n)^{int})|, |\lambda^s((\tilde{U}_{h,K}^n)^{ext})|)\right)\right),$$

which is close to being constant since the characteristic speeds (2.4) are close to c. As a result, only a constant number of steps are needed in integrating (3.1). Since each step of (3.8) requires $\mathcal{O}(N_v)$ operations, the total cost of the DG-interpolation is $\mathcal{O}(N_v)$. The detailed discussion on the cost of DG-interpolation can be seen in [45].

4 MMPDE moving mesh method

In this section we describe the generation of the new mesh $\mathcal{T}_h^{n+1}$ from the old one $\mathcal{T}_h^n$ using the MMPDE moving mesh method [15,16,20]. We use the $\boldsymbol{\xi}$-formulation of the method and its new implementation proposed in [21].

For mesh generation purpose, we introduce a computational mesh $\mathcal{T}_c = \{\boldsymbol{\xi}_1, \ ..., \ \boldsymbol{\xi}_{N_v}\}$ which can be viewed as a deformation of the mesh $\mathcal{T}_h^n$ and serves as an intermediate variable. We also assume that a reference computational mesh $\hat{\mathcal{T}}_c = \{\hat{\boldsymbol{\xi}}_1, \ ..., \ \hat{\boldsymbol{\xi}}_{N_v}\}$ has been given. This mesh is kept fixed in the computation and should be chosen as uniform as possible. Often it can be chosen as the initial physical mesh.

A key idea of the MMPDE method is to view any nonuniform mesh as a uniform one in some Riemannian metric [19,20]. The metric tensor $\mathbb{M}$, a symmetric and uniformly positive definite matrix-valued function defined on $\mathcal{D}$, provides the information needed for determining the size, shape, and orientation of the mesh elements throughout the domain. Various metric tensors have been proposed; e.g., see [18,20]. We use here an optimal metric tensor based on the L^2-norm of piece linear interpolation error. To be specific, we consider a physical variable q and its finite element approximation q_h. Let H_K be a recovered Hessian of q_h on $K \in \mathcal{T}_h$ obtained using the least squares fitting Hessian recovery technique [46]. Assuming that the eigen-decomposition of H_K is given by

$$H_K = Q\text{diag}(\lambda_1, \cdots, \lambda_d) Q^T,$$

where Q is an orthogonal matrix, we define

$$|H_K| = Q\text{diag}(|\lambda_1|, ..., |\lambda_d|) Q^T.$$

The metric tensor is defined as

$$\mathbb{M}_K = \det\left(\alpha_h \mathbb{I} + |H_K|\right)^{-\frac{1}{d+4}} \left(\alpha_h \mathbb{I} + |H_K|\right), \quad \forall K \in \mathcal{T}_h \tag{4.1}$$

where $\mathbb{I}$ is the identity matrix, $\det(\cdot)$ is the determinant of a matrix, d is the dimension of the spatial domain, and α_h is a regularization parameter defined through the algebraic equation

$$\sum_{K\in\mathcal{T}_h} |K| \det(\alpha_h \mathbb{I} + |H_K|)^{\frac{2}{d+4}} = 2 \sum_{K\in\mathcal{T}_h} |K| \det(|H_K|)^{\frac{2}{d+4}}.$$

It is interesting to note that the entropy has been commonly used for adaptive mesh simulation of shock waves [26,28]. For the SWEs, the total energy $E = \frac{1}{2}(hu^2 + hv^2) + \frac{1}{2}gh^2 + ghB$ plays the role of the entropy function. However, we have found through numerical experiment that the metric tensor based on the equilibrium variable $\mathcal{E} = \frac{1}{2}(u^2+v^2) + g(h+B)$ and the water depth h leads to more desirable mesh concentration than that based on the total energy/entropy alone (cf. Figs. 3, 4, and 25 – 28). A motivation for this is that the mesh is adapted to both the perturbations of the lake-at-rest steady state through $\mathcal{E}$ and the water depth distribution through h. Specifically, we first compute $\mathbb{M}_K^{\mathcal{E}}$ and $\mathbb{M}_K^h$ using (4.1) with $q = \mathcal{E}$ and h, respectively. Then, a new metric tensor is obtained through matrix intersection as

$$\tilde{\mathbb{M}}_K = \frac{\mathbb{M}_K^{\mathcal{E}}}{|||\mathbb{M}_K^{\mathcal{E}}|||} \cap \frac{\delta \cdot \mathbb{M}_K^h}{|||\mathbb{M}_K^h|||}, \tag{4.2}$$

where $|||\cdot|||$ denotes the maximum absolute value of the entries of a matrix and "$\cap$" stands for matrix intersection. The reader is referred to [44] for the definition and geometric interpretation of matrix intersection. We take $\delta = 0.1$ in our numerical simulations.

The next step is to make sure that the metric tensor is bounded. It is known [22] that the moving mesh generated by the MMPDE method stays nonsingular (free from tangling) if the metric tensor is bounded and the initial mesh is nonsingular. More specifically, a positive lower bound on the minimum element height is obtained in [22], which is inversely proportional to some powers of N and $\lambda_{max}(\mathbb{M})$. We define

$$\hat{\mathbb{M}}_K = \frac{\tilde{\mathbb{M}}_K}{\sqrt{1 + \left(\frac{\mathrm{tr}(\tilde{\mathbb{M}}_K)}{\beta}\right)^2}}, \tag{4.3}$$

where β is a positive number and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. It is not difficult to show $\lambda_{\max}(\hat{\mathbb{M}}_K) \leq \beta$. The greater β is, the smaller the minimum element height may result. In our computation, we take $\beta = 1000$.

The metric tensor is generally non-smooth since the received Hessian can contain the discontinuous. A common practice in the moving mesh context is to smooth the metric tensor for smoother meshes. Here we simply average the metric tensor at a vertex over its neighboring vertices, i.e.,

$$\mathbb{M}_i \longleftarrow \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \hat{\mathbb{M}}_j, \quad \forall \boldsymbol{x}_i \in \mathcal{T}_h \tag{4.4}$$

where $\hat{\mathbb{M}}_i$ is the nodal value at vertex $\boldsymbol{x}_i$ obtained by area-averaging the values of the metric tensor on the neighboring elements, the $\mathcal{N}_i$ denotes the set of the immediate neighboring vertices (including itself) of $\boldsymbol{x}_i$, and $|\mathcal{N}_i|$ is the length of the set $\mathcal{N}_i$. This process can be repeated several times every time in the computation.

Having defined the metric tensor, we are ready to describe the MMPDE method. Recalling that $\mathcal{T}_c$ and $\mathcal{T}_h$ are a deformation of each other, for any element $K \in \mathcal{T}_h$, there exists an element $K_c \in \mathcal{T}_c$ corresponding to K. Denote the affine mapping from K_c to K as F_K and its Jacobian matrix as F_K'. It is known [19,20] that any mesh $\mathcal{T}_h$ which is uniform in the metric $\mathbb{M}$ in reference to the computational mesh $\mathcal{T}_c$, satisfies

$$|K|\sqrt{\det(\mathbb{M}_K)} = \frac{\sigma_h |K_c|}{|\mathcal{D}_c|}, \qquad \forall K \in \mathcal{T}_h \tag{4.5}$$

$$\frac{1}{d}\mathrm{tr}\big((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\big) = \det\big((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T}\big)^{\frac{1}{d}}, \qquad \forall K \in \mathcal{T}_h \tag{4.6}$$

where $\mathbb{M}_K$ is the average of $\mathbb{M}$ over K and

$$|\mathcal{D}_c| = \sum_{K_c \in \mathcal{T}_c} |K_c|, \quad \sigma_h = \sum_{K \in \mathcal{T}_h} |K| \det(\mathbb{M}_K)^{\frac{1}{2}}.$$

The equidistribution condition (4.5) determines the size of elements through the metric tensor $\mathbb{M}$ and implies that all the elements have the same size in the metric tensor $\mathbb{M}$. The alignment condition (4.6) determines the shape and orientation of K through $\mathbb{M}$ and shape of K_c and implies that any element K when measured in the metric $\mathbb{M}_K$ is similar to K_c measured in the Euclidean metric.

A mesh energy function associated with the equidistribution and alignment conditions is given by

$$\begin{aligned} I_h(\mathcal{T}_h; \mathcal{T}_c) =& \frac{1}{3} \sum_{K \in \mathcal{T}_h} |K| \det(\mathbb{M}_K)^{\frac{1}{2}} \left(\mathrm{tr}((F_K')^{-1} \mathbb{M}_K^{-1} (F_K')^{-T}) \right)^{\frac{3d}{4}} \\ &+ \frac{1}{3} d^{\frac{3d}{4}} \sum_{K \in \mathcal{T}_h} |K| \det(\mathbb{M}_K)^{\frac{1}{2}} \left(\det(F_K') \det(\mathbb{M}_K)^{\frac{1}{2}} \right)^{-\frac{3}{2}}, \end{aligned} \tag{4.7}$$

which is a Riemann sum of a continuous functional developed in [17].

For a given mesh $\mathcal{T}_h^n$, we want to find a new mesh $\mathcal{T}_h^{n+1}$ by minimizing the mesh energy function I_h. Note that $I_h(\mathcal{T}_h, \mathcal{T}_c)$ is a function of the vertices $\boldsymbol{\xi}_i,\ i = 1, ..., N_v$ of $\mathcal{T}_c$ and the vertices $\boldsymbol{x}_i,\ i = 1, ..., N_v$ of $\mathcal{T}_h$. We take $\mathcal{T}_h$ as $\mathcal{T}_h^n$, minimize $I_h(\mathcal{T}_h^n, \mathcal{T}_c)$ with respect to $\mathcal{T}_c$ (and denote the final mesh as $\mathcal{T}_c^{n+1}$), and obtain $\mathcal{T}_h^{n+1}$ through the relation between $\mathcal{T}_h^n$ and $\mathcal{T}_c^{n+1}$. The minimization of $I_h(\mathcal{T}_h^n, \mathcal{T}_c)$ is carried out by solving the mesh equation defined as the gradient system of the energy function (the MMPDE approach), i.e.,

$$\frac{d\boldsymbol{\xi}_i}{dt} = -\frac{\det(\mathbb{M}(\boldsymbol{x_i}))^{\frac{1}{2}}}{\tau} \left(\frac{\partial I_h(\mathcal{T}_h^n, \mathcal{T}_c)}{\partial \boldsymbol{\xi}_i} \right)^T, \quad i = 1, ..., N_v \tag{4.8}$$

where $\partial I_h / \partial \boldsymbol{\xi}_i$ is considered as a row vector and $\tau > 0$ is a parameter used to adjust the response time of mesh movement to the changes in $\mathbb{M}$.

Define the function G associated with the energy (4.7) as

$$G(\mathbb{J}, \det(\mathbb{J})) = \frac{1}{3} \det(\mathbb{M}_K)^{\frac{1}{2}} (\mathrm{tr}(\mathbb{J} \mathbb{M}_K^{-1} \mathbb{J}^T))^{\frac{3d}{4}} + \frac{1}{3} d^{\frac{3d}{4}} \det(\mathbb{M}_K)^{\frac{1}{2}} \left(\frac{\det(\mathbb{J})}{\det(\mathbb{M}_K)^{\frac{1}{2}}} \right)^{\frac{3}{2}}, \tag{4.9}$$

where $\mathbb{J} = (F_K')^{-1} = E_{K_c} E_K^{-1}$, and $E_K = [\boldsymbol{x}_1^K - \boldsymbol{x}_0^K, \ ..., \ \boldsymbol{x}_d^K - \boldsymbol{x}_0^K]$ and $E_{K_c} = [\boldsymbol{\xi}_1^K - \boldsymbol{\xi}_0^K, \ ..., \ \boldsymbol{\xi}_d^K - \boldsymbol{\xi}_0^K]$ are the edge matrices of K and K_c, respectively. Using the notion of scalar-by-matrix differentiation, the derivatives of G with respect to $\mathbb{J}$ and $\det(\mathbb{J})$ can be obtained [21] as

$$\frac{\partial G}{\partial \mathbb{J}} = \frac{d}{2} \det(\mathbb{M}_K)^{\frac{1}{2}} (\mathrm{tr}(\mathbb{J} \mathbb{M}_K^{-1} \mathbb{J}^T))^{\frac{3d}{4}-1} \mathbb{M}_K^{-1} \mathbb{J}^T, \tag{4.10}$$

$$\frac{\partial G}{\partial \det(\mathbb{J})} = \frac{1}{2} d^{\frac{3d}{4}} \det(\mathbb{M}_K)^{-\frac{1}{4}} \det(\mathbb{J})^{\frac{1}{2}}. \tag{4.11}$$

With these formulas, we can rewrite (4.8) as (cf. [21])

$$\frac{d\boldsymbol{\xi}_i}{dt} = \frac{\det(\mathbb{M}(\boldsymbol{x_i}))^{\frac{1}{2}}}{\tau} \sum_{K \in \omega_i} |K| \boldsymbol{v}_{i_K}^K, \quad i = 1, ..., N_v \tag{4.12}$$

where ω_i is the element patch associated with the vertex $\boldsymbol{x}_i$, i_K is the local index of $\boldsymbol{x}_i$ on K, and $\boldsymbol{v}_{i_K}^K$ is the local velocity contributed by the element K to the vertex i_K. The local velocities $\boldsymbol{v}_{i_K}^K,\ i_K = 0, ..., d$ are given by

$$\begin{bmatrix} (\boldsymbol{v}_1^K)^T \\ (\boldsymbol{v}_2^K)^T \\ \vdots \\ (\boldsymbol{v}_d^K)^T \end{bmatrix} = -E_K^{-1}\frac{\partial G}{\partial \mathbb{J}} - \frac{\partial G}{\partial \det(\mathbb{J})}\frac{\det(E_{K_c})}{\det(E_K)}E_{K_c}^{-1}, \quad \boldsymbol{v}_0^K = -\sum_{i_K=1}^{d} \boldsymbol{v}_{i_K}^K. \tag{4.13}$$

Note that the velocities for the boundary nodes need to be modified properly. For example, they should be set to be zero for the corner vertices. For other boundary vertices, the velocities should be modified such that their normal component along the domain boundary are zeros so they slide only along the boundary and do not move out of the domain.

Starting with the reference computational mesh $\hat{\mathcal{T}}_c$ as the initial mesh, the mesh equation (4.12) is integrated over a physical time step. The obtained new mesh is denoted by $\mathcal{T}_c^{n+1}$. Note that $\mathcal{T}_h^n$ is kept fixed during the integration and forms a correspondence with $\mathcal{T}_c^{n+1}$, i.e., $\mathcal{T}_h^n = \Phi_h(\mathcal{T}_c^{n+1})$. Then the new physical mesh $\mathcal{T}_h^{n+1}$ is defined as $\mathcal{T}_h^{n+1} = \Phi_h(\hat{\mathcal{T}}_c)$, which can be computed using linear interpolation.

5 Numerical Results

In this section we present numerical results obtained with the MM-DG method described in the previous sections for the one- and two-dimensional SWEs. We take the CFL number C_{cfl} as 0.3 for P^1-DG and 0.18 for P^2-DG in one dimension, and 0.2 for P^1-DG and 0.1 for P^2-DG in two dimensions, unless otherwise stated. For the TVB limiter implemented in the RKDG scheme, the constant M_{tvb} is taken as zero except for accuracy test Example 5.1 (to avoid the accuracy order reduction near the extrema). For mesh movement, we take $\tau = 0.1N^{-1/d}$. The gravitation constant is taken as $g = 9.812$ in the computation. For examples where the analytical exact solution is unavailable, we take the numerical solution obtained by the P^2-DG method with a fixed mesh of $N = 10000$ as a reference solution, unless otherwise stated. The error is computed with 21 points used on each element. Except for the accuracy test (Example 5.1) and the lake-at-rest steady-state flow tests (Example 5.2 and Example 5.6), to save space we omit the results for the P^1-DG method since they are similar to those for the P^2-DG method.

Example 5.1 (The accuracy test for the 1D SWEs over a sinusoidal hump.)

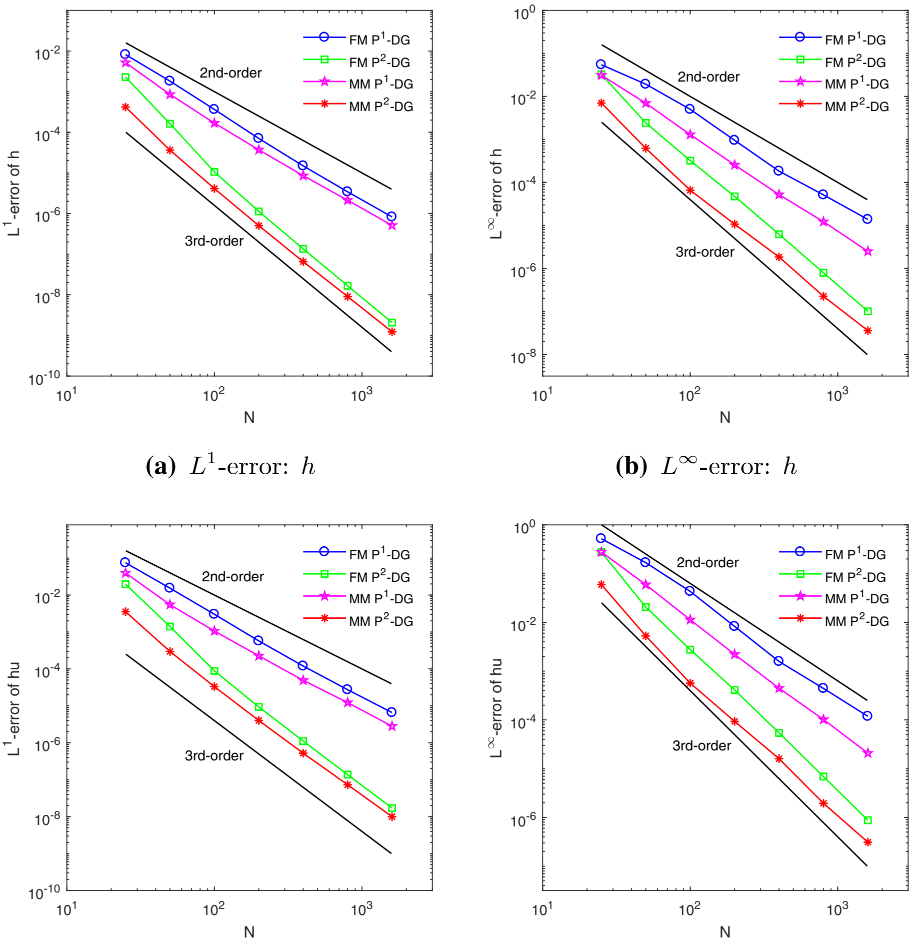
This example is used to verify the high order accuracy of the MM-DG method. The bottom topography is

$$B(x) = \sin^2(\pi x), \quad x \in (0, 1).$$

We use periodic boundary conditions for all unknown variables. The initial conditions are given as

$$h(x, 0) = 5 + e^{\cos(2\pi x)}, \quad hu(x, 0) = \sin\big(\cos(2\pi x)\big).$$

This example has been used as an accuracy-test in literature, e.g. see [27,39,40]. The final simulation time is $T = 0.1$ when the solutions remain smooth. A reference solution is obtained using the P^2-DG method with a fixed mesh of $N = 20000$. The minmod constant M_{tvb} in the TVB limiter is taken as 40 in this example to avoid the accuracy order reduction near the extrema. The L^1 and L^∞ norm of the error with moving and fixed meshes for the water depth h and water discharge hu are plotted in Fig. 1. It can be seen that the MM-DG



(a) L^1-error: h

(b) L^∞-error: h

(c) L^1-error: hu

(d) L^∞-error: hu

Fig. 1 Example 5.1. The L^1 and L^∞ norm of the error with moving and fixed meshes for the water depth h and the water discharge hu

method has the expected second-order convergence for P^1-DG and third-order for P^2-DG in both L^1 and L^∞ norm. Moreover, the figures show that a moving mesh produces a slightly smaller but otherwise comparable error than a fixed mesh of the same number of elements. The error is much smaller for P^2-DG method than the P^1-DG method, as expected for smooth problems.

Example 5.2 (The lake-at-rest steady-state flow test for the 1D SWEs.)

In this example we consider a lake-at-rest steady-state flow over three different bottom topographies to verify the well-balance property of the MM-DG method. Two of these topographies are smooth and the other is discontinuous, i.e.,

$$B(x) = 5e^{-\frac{2}{5}(x-5)^2}, \quad x \in (0, 10) \tag{5.1}$$

Table 1 Example 5.2. Well-balance test for the moving mesh P^1-DG method over the smooth bottom topography (5.1)

N	$h+B$		hu	
	L^1-error	L^∞-error	L^1-error	L^∞-error
B updated with DG-interpolation				
50	1.970E-14	2.551E-14	2.810E-14	7.858E-14
100	4.066E-14	4.958E-14	4.955E-14	1.164E-13
200	7.874E-14	9.286E-14	7.765E-14	1.911E-13
B updated with L^2 -projection				
50	3.393E-04	9.016E-04	2.827E-03	7.166E-03
100	9.550E-05	2.852E-04	8.202E-04	2.338E-03
200	2.381E-05	8.143E-05	2.039E-04	6.823E-04

$$B(x) = \begin{cases} 4, & \text{for } x \in (4,8) \\ 0, & \text{for } x \in (0,4) \cup (8,10) \end{cases} \tag{5.2}$$

$$B(x) = 10e^{-\frac{2}{5}(x-5)^2}, \quad x \in (0,10). \tag{5.3}$$

The initial data is at the lake-at-rest steady state,

$$u = 0, \quad h + B = 10.$$

This still water state solution should be preserved exactly if the MM-DG method is well-balanced.

We computed the solution up to $t = 0.5$ on moving meshes. To show that the well-balance property is attained up to the level of round-off error (double precision in MATLAB), we present the L^1 and L^∞ error for $h + B$ and hu in Tables 1 and 2 for P^1-DG and P^2-DG on moving meshes, respectively, for the smooth bottom topography (5.1). Similar results for the discontinuous bottom topography (5.2) are shown in Tables 3 and 4. For comparison purpose, we also list the results obtained with the L^2-projection of B. From the results one can see that the MM-DG method with B updated with DG-interpolation preserves the lake-at-rest solution. On the other hand, when B is updated with L^2-projection, the lake-at-rest solution is not preserved by the MM-DG method. The results show that the error, more precisely, the deviation from the lake-at-rest steady state, is about second-order for the smooth topography for both P^1-DG and P^2-DG. For the discontinuous topography, the error is about second-order in L^1 and first-order in L^∞. To explain this, we notice that in the current situation the deviation from the lake-at-rest steady state comes from the error of the scheme which in turn is dominated by the error of the TVB limiter. For this example, we have not made effort in optimizing the choice of the TVB minmod constant M_{tvb}. Instead, we have simply chosen $M_{tvb} = 0$, for which the TVB limiter becomes the TVD limiter that is second-order at best in one dimension [6,14].

The bottom topography (5.3) (with a dry region) is used to verify the well-balance and PP properties of the MM-DG method. This topography has a similar shape as (5.1) but its height touches the water level at $x = 5$ where $h = 0$ initially. The computed water depth can have negative values during the computation and the application of the PP limiter is necessary. We computed the solution up to $t = 0.5$ on moving meshes. To ensure positivity preservation (cf. [41]), we take smaller CFL numbers as 0.3 and 0.15 for P^1-DG and P^2-DG, respectively, for this test. The L^1 and L^∞ error for $h + B$ and hu is listed in Table 5 for P^1-DG and P^2-DG. The results clearly show that the MM-DG method is well-balanced.

Table 2 Example 5.2. Well-balance test for the moving mesh P^2-DG method over the smooth bottom topography (5.1)

N	$h+B$		hu	
	L^1-error	L^∞-error	L^1-error	L^∞-error
B updated with DG-interpolation				
50	3.051E-14	3.813E-14	3.951E-14	9.699E-14
100	6.282E-14	7.583E-14	7.464E-14	1.769E-13
200	1.276E-13	1.503E-13	1.449E-13	3.481E-13
B updated with L^2 -projection				
50	4.922E-06	1.251E-05	4.995E-05	1.393E-04
100	1.160E-06	3.775E-06	1.139E-05	3.573E-05
200	2.395E-07	8.878E-07	2.299E-06	8.875E-06

Table 3 Example 5.2. Well-balance test for the moving mesh P^1-DG method over the discontinuous bottom topography (5.2)

N	$h+B$		hu	
	L^1-error	L^∞-error	L^1-error	L^∞-error
B updated with DG-interpolation				
50	1.175E-14	2.014E-14	4.114E-14	1.001E-13
100	1.912E-14	3.143E-14	6.747E-14	1.501E-13
200	3.108E-14	5.032E-14	1.136E-13	2.576E-13
B updated with L^2 -projection				
50	6.001E-03	5.651E-02	4.542E-02	3.688E-01
100	1.823E-03	2.629E-02	1.637E-02	1.713E-01
200	5.499E-04	1.237E-02	5.166E-03	9.294E-02

Table 4 Example 5.2. Well-balance test for the moving mesh P^2-DG method over the discontinuous bottom topography (5.2)

N	$h+B$		hu	
	L^1-error	L^∞-error	L^1-error	L^∞-error
B updated with DG-interpolation				
50	1.518E-14	2.635E-14	5.499E-14	1.562E-13
100	2.166E-14	4.096E-14	1.003E-13	2.367E-13
200	3.419E-14	6.230E-14	1.679E-13	3.781E-13
B updated with L^2 -projection				
50	3.548E-03	2.796E-02	3.090E-02	2.522E-01
100	8.669E-04	1.073E-02	7.585E-03	9.252E-02
200	1.903E-04	3.710E-03	1.624E-03	2.714E-02

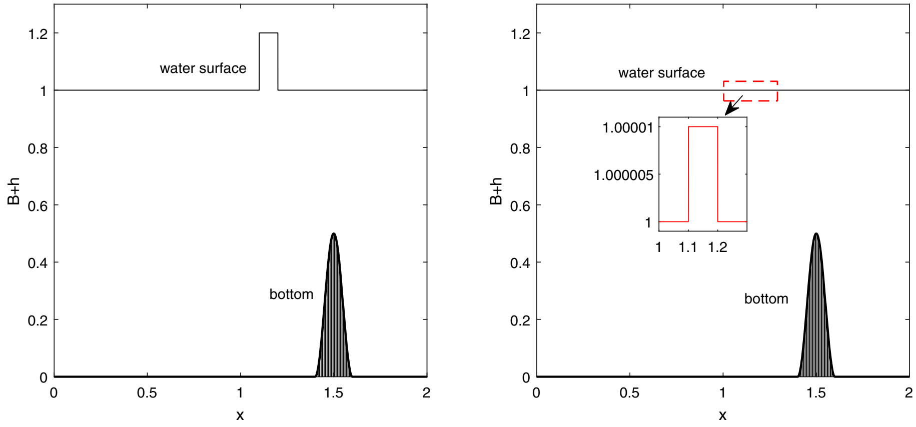
Example 5.3 (The perturbed lake-at-rest steady-state flow test for the 1D SWEs.)

This example has been used by a number of researchers [11,25,27,39,40] to demonstrate the capability of a numerical method to accurately compute small perturbations of a lake-at-rest steady-state flow over non-flat bottom topographies. The bottom topography in this example is taken as

$$B(x) = \begin{cases} 0.25(\cos(10\pi(x-1.5))+1), & \text{for } x \in (1.4, 1.6) \\ 0, & \text{for } x \in (0, 1.4) \cup (1.6, 2) \end{cases} \tag{5.4}$$

Table 5 Example 5.2. Well-balance test for the MM-DG method over the bottom topography (5.3) (with a dry region)

N	$h+B$		hu	
	L^1-error	L^∞-error	L^1-error	L^∞-error
P^1 MM-DG method with PP limiter				
50	1.362E-14	2.131E-14	3.802E-14	8.519E-14
100	2.426E-14	3.709E-14	6.166E-14	1.356E-13
200	5.079E-14	7.733E-14	1.225E-13	2.779E-13
P^2 MM-DG method with PP limiter				
50	2.404E-14	4.065E-14	7.451E-14	1.781E-13
100	4.197E-14	6.853E-14	1.355E-13	2.919E-13
200	8.531E-14	1.322E-13	2.496E-13	5.637E-13



(a) Large pulse $\varepsilon = 0.2$ **(b)** Small pulse $\varepsilon = 10^{-5}$

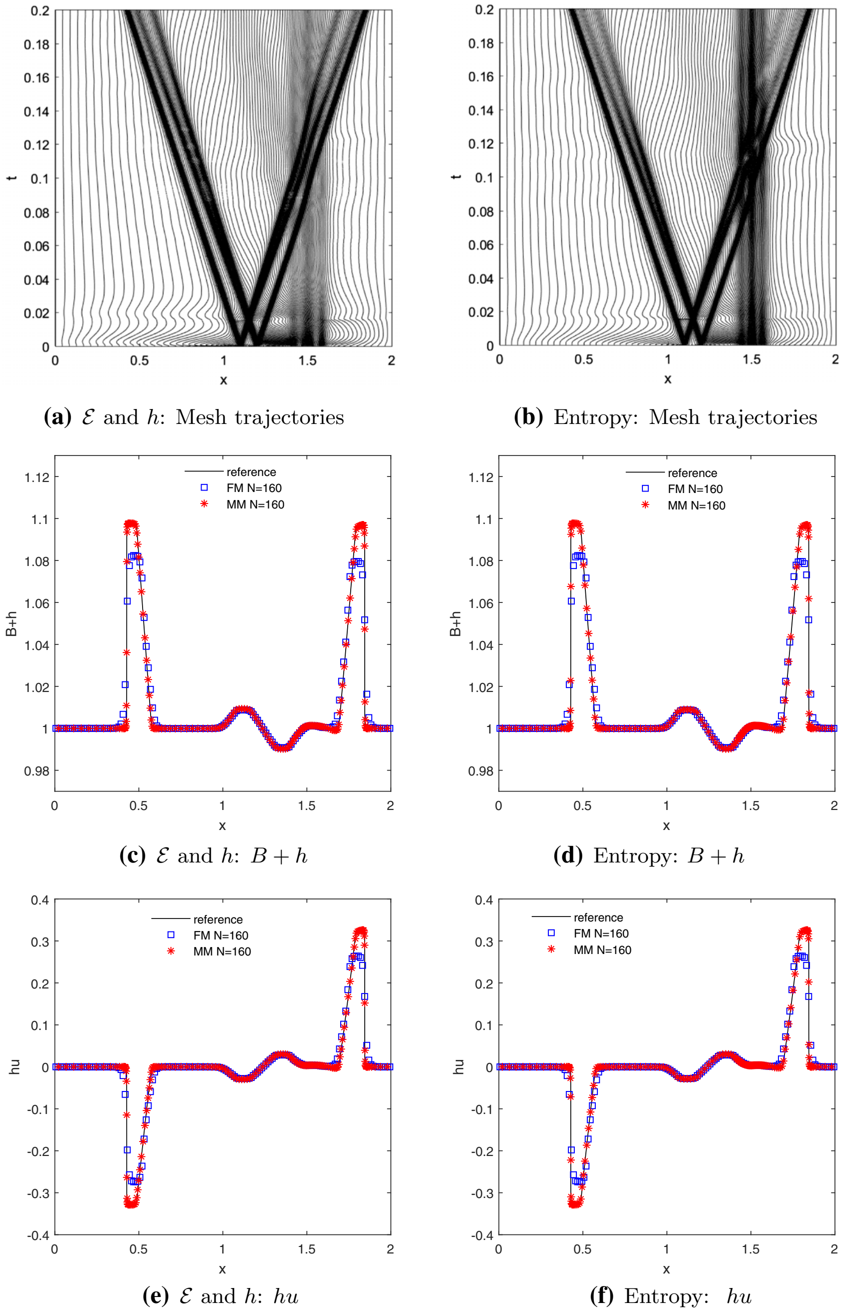
Fig. 2 Example 5.3. The initial water surface level $B+h$ and bottom topography B for the pulse of $\varepsilon = 0.2$ and $\varepsilon = 10^{-5}$

which has a bump in the middle of the physical interval. The initial conditions are given by

$$h(x,0) = \begin{cases} 1 - B(x) + \varepsilon, & \text{for } 1.1 \le x \le 1.2 \\ 1 - B(x), & \text{otherwise} \end{cases} \quad \text{and} \quad u(x,0) = 0,$$

where ε is a constant for the perturbation magnitude. We consider two cases, $\varepsilon = 0.2$ (large pulse) and $\varepsilon = 10^{-5}$ (small pulse). The initial conditions for both cases are plotted in Fig. 2. The initial wave splits into two waves propagating at the characteristic speeds $\pm\sqrt{gh}$. We use the transmissive boundary conditions and compute the solution up to $t = 0.2$ when the right wave has already passed the bottom bump.

The mesh trajectories, water surface $B+h$, and discharge hu at $t = 0.2$ obtained with P^2-DG and moving and fixed meshes of $N = 160$ are shown in Figs. 3 (for $\varepsilon = 0.2$) and 4 (for $\varepsilon = 10^{-5}$). Recall that these results are obtained with the metric tensor (4.3) based on the equilibrium variable $\mathcal{E} = \frac{1}{2}u^2 + g(h+B)$ and the water depth h. For comparison purpose, we also include the results obtained with the metric tensor based on the entropy/total energy $E = \frac{1}{2}hu^2 + \frac{1}{2}gh^2 + ghB$. Interestingly, for the large pulse $\varepsilon = 0.2$ (Fig. 3), both ways of computing the metric tensor lead to almost indistinguishable solutions although they have



(a) $\mathcal{E}$ and h: Mesh trajectories

(b) Entropy: Mesh trajectories

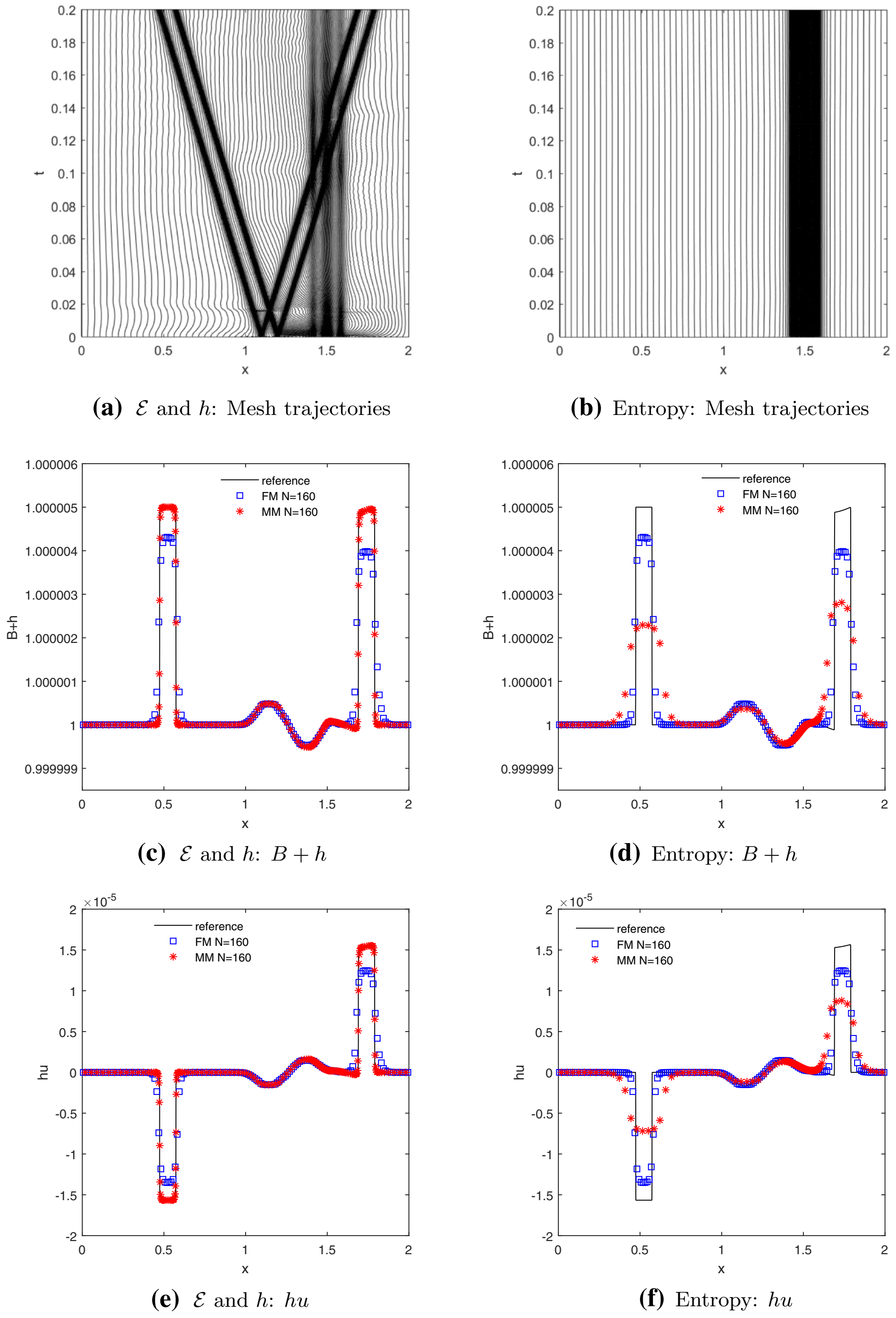
(c) $\mathcal{E}$ and h: $B+h$

(d) Entropy: $B+h$

(e) $\mathcal{E}$ and h: hu

(f) Entropy: hu

Fig. 3 Example 5.3. The mesh trajectories and the water surface $B+h$ and discharge hu at $t=0.2$ obtained with P^2-DG of a moving mesh of $N=160$ and a fixed mesh of $N=160$ for a large pulse $\varepsilon=0.2$. The metric tensor is computed based on the equilibrium variable $\mathcal{E}$ and the water depth h (left column) or the entropy/total energy (right column)



(a) $\mathcal{E}$ and h: Mesh trajectories

(b) Entropy: Mesh trajectories

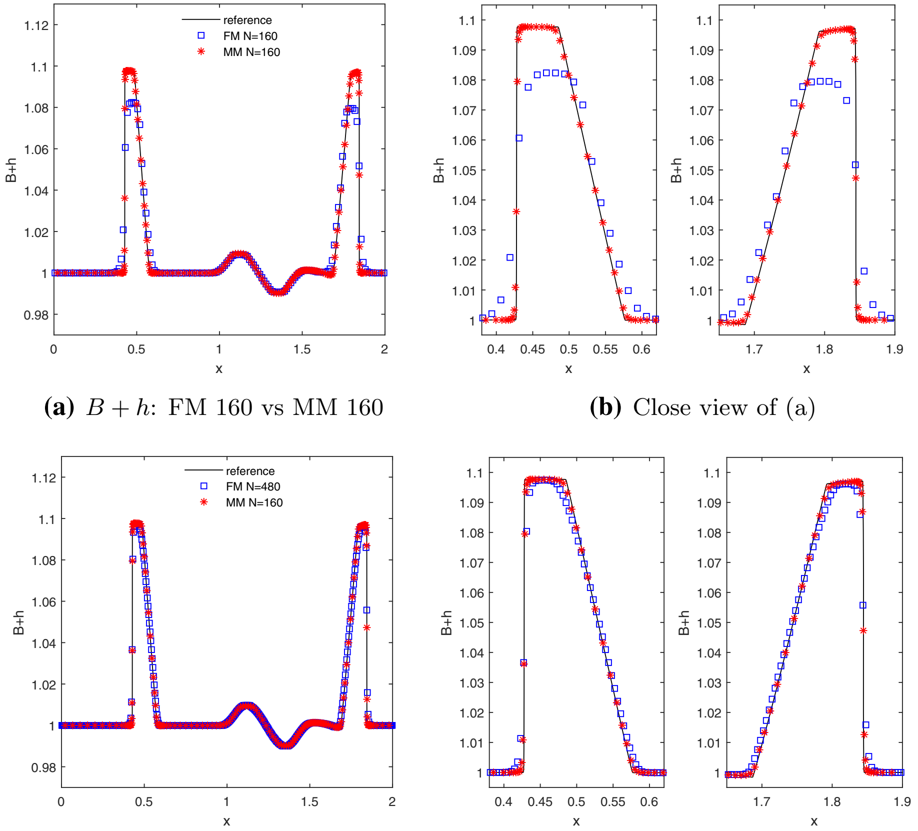
(c) $\mathcal{E}$ and h: $B + h$

(d) Entropy: $B + h$

(e) $\mathcal{E}$ and h: hu

(f) Entropy: hu

Fig. 4 Example 5.3. The mesh trajectories and the water surface $B + h$ and discharge hu at $t = 0.2$ obtained with P^2-DG of a moving mesh of $N = 160$ and a fixed mesh of $N = 160$ for a small pulse $\varepsilon = 10^{-5}$. The metric tensor is computed based on the equilibrium variable $\mathcal{E}$ and the water depth h (left column) or the entropy/total energy (right column)



(a) $B + h$: FM 160 vs MM 160

(b) Close view of (a)

(c) $B + h$: FM 480 vs MM 160

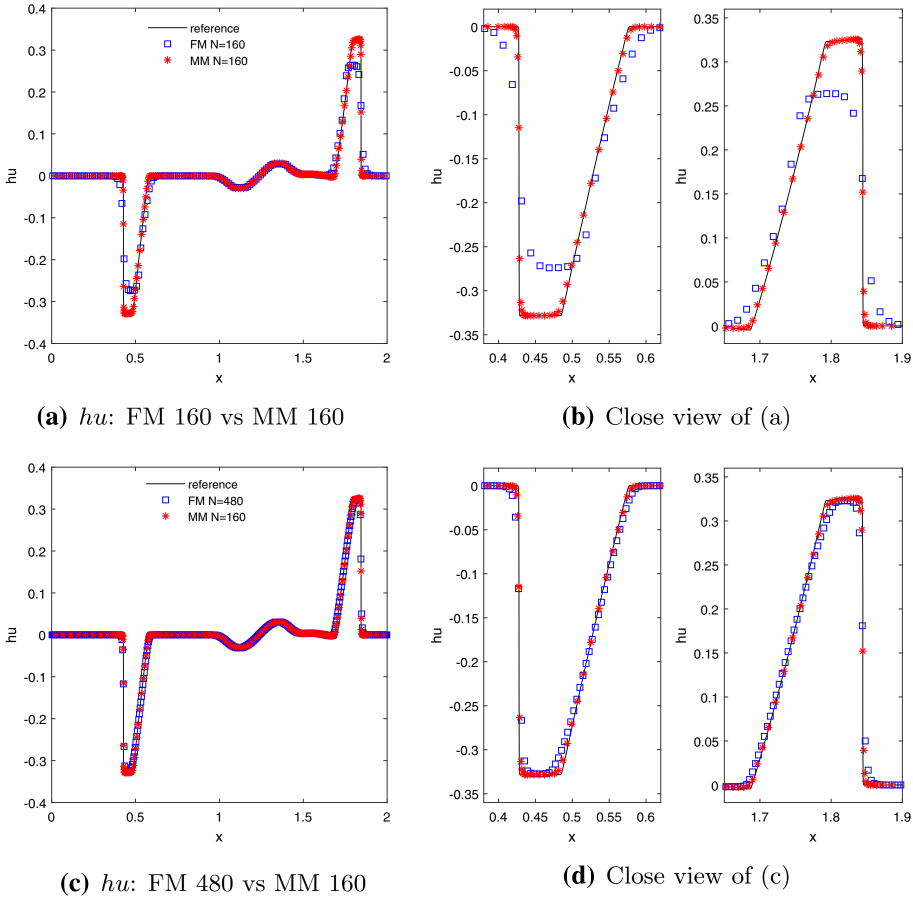
(d) Close view of (c)

Fig. 5 Example 5.3. The water surface $B + h$ at $t = 0.2$ obtained with P^2-DG and a moving mesh of $N = 160$ are compared with those obtained with the fixed meshes of $N = 160$ and $N = 480$ for a large pulse $\varepsilon = 0.2$

slightly different but correct mesh concentration: the former concentrates more points in the sharp wave regions whereas the latter leads to more points in the non-flat bottom region. For the case with a small pulse $\varepsilon = 10^{-5}$ (Fig. 4), the situation is very different. The metric tensor based on E misses the wave regions and concentrates points only in the non-flat bottom region. On the other hand, the metric tensor based on $\mathcal{E}$ and h works well for this case too and concentrates mesh points in both the wave regions and the non-flat bottom region. The advantages of using $\mathcal{E}$ and h over E in mesh adaptation are clear when the perturbation is smaller.

It is interesting to see that the small perturbation splits into two waves at about $t = 0.0165$ and the mesh elements are concentrated properly near the waves before and after the split.

The water surface $B + h$ and discharge hu obtained with P^2-DG and a moving mesh of $N = 160$ and fixed meshes of $N = 160$ and $N = 480$ are plotted in Figs. 5 and 6 (for $\varepsilon = 0.2$) and Figs. 7 and 8 (for $\varepsilon = 10^{-5}$). The results show that the moving mesh solutions with $N = 160$ are more accurate than those with the fixed mesh of $N = 160$ and $N = 480$ and contain no visible spurious numerical oscillations. They also show that the MM-DG method with moving or fixed meshes is able to capture the waves of large or small pulses.



(a) hu: FM 160 vs MM 160

(b) Close view of (a)

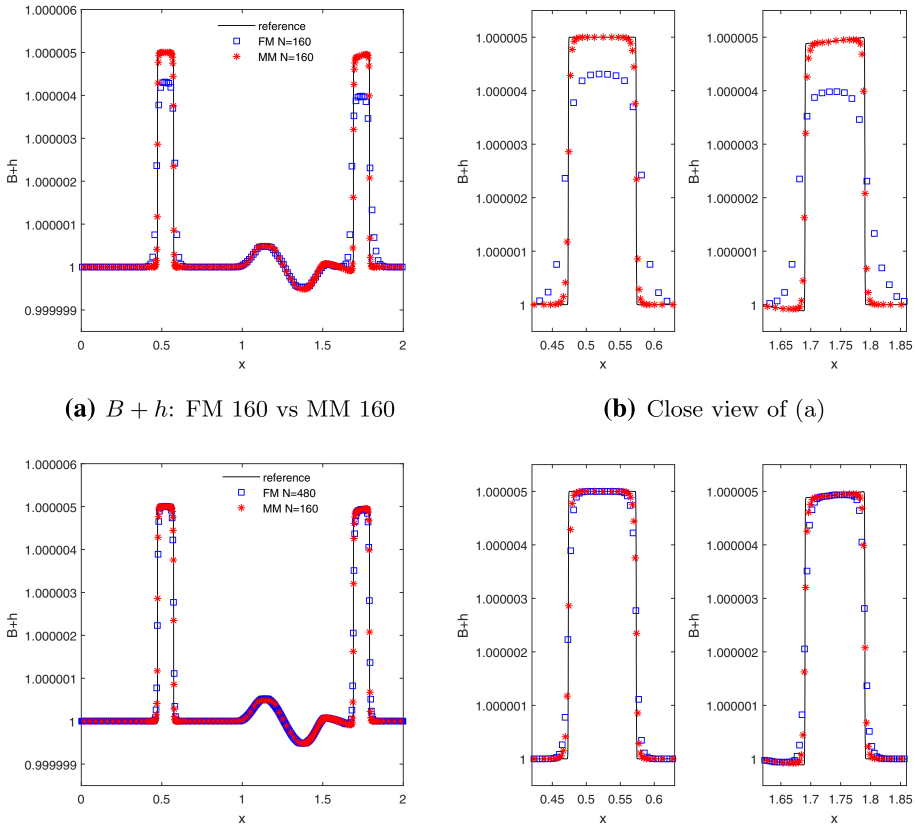
(c) hu: FM 480 vs MM 160

(d) Close view of (c)

Fig. 6 Example 5.3. The water discharge hu at $t = 0.2$ obtained with P^2-DG and a moving mesh of $N = 160$ are compared with those obtained with the fixed meshes of $N = 160$ and $N = 480$ for a large pulse $\varepsilon = 0.2$

Comparison results are shown in Figs. 9 ($\varepsilon = 0.2$) and 10 ($\varepsilon = 10^{-5}$) by using DG-interpolation or L^2-projection in updating the bottom topography B. The computed solutions are almost indistinguishable for the large pulse case with $\varepsilon = 0.2$. On the other hand, for the small pulse case with $\varepsilon = 10^{-5}$ (Fig. 10), the MM-DG method with DG-interpolation captures the waves well whereas the method with L^2-projection misses them badly. Recall that the former is well-balanced but the latter is not. For the latter, the deviation from the lake-at-rest steady state comes from the error of the scheme that is second-order at best since the TVB limiter with $M_{tvb} = 0$ is used. When this deviation is smaller than the size of the perturbation, the scheme is able to capture the waves as shown in Fig. 9. However, when the deviation is larger than the size of the perturbation, the scheme will miss the waves, as seen in Fig. 10, although the situation can be improved with finer meshes. In contrast, well-balanced schemes are able to capture small perturbations even on a reasonably coarse mesh.

We now comment on the cost of the DG-interpolation scheme. Recall from Remark 3.4 that for a given example, the integration of (3.1) requires only a constant number of time steps, which implies that each interpolation costs $\mathcal{O}(N_v)$ operations. This has been observed



(a) $B + h$: FM 160 vs MM 160

(b) Close view of (a)

(c) $B + h$: FM 480 vs MM 160

(d) Close view of (c)

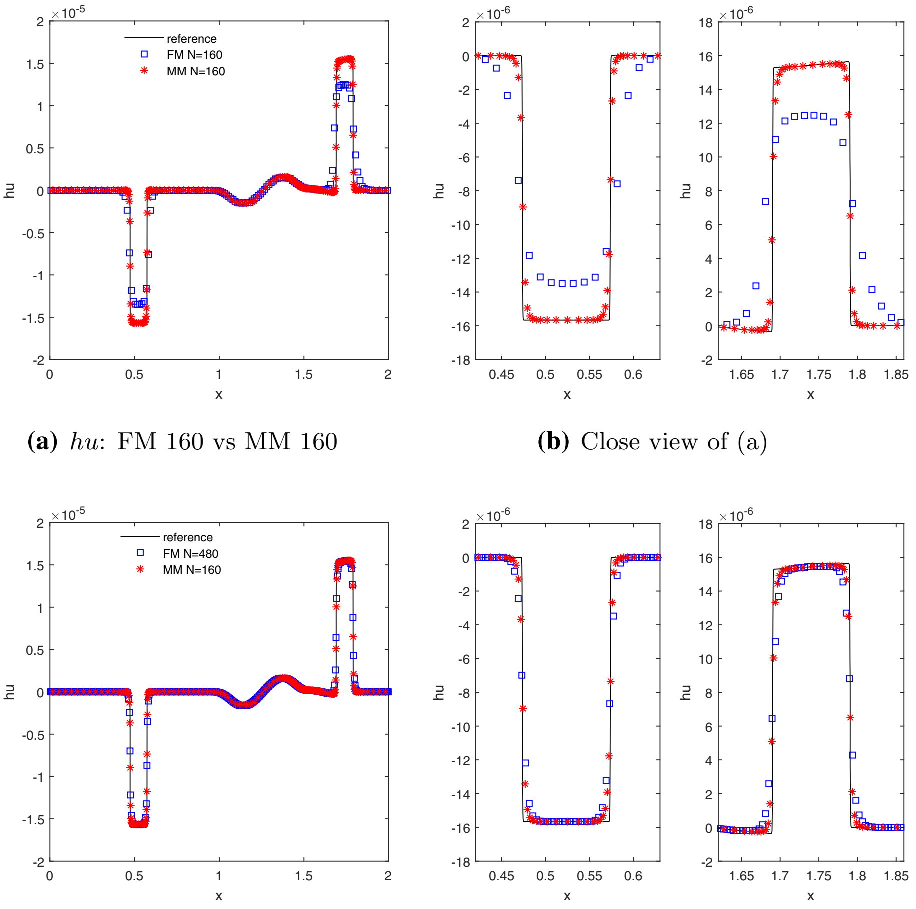
Fig. 7 Example 5.3. The water surface $B+h$ at $t = 0.2$ obtained with P^2-DG and a moving mesh of $N = 160$ are compared with those obtained with the fixed meshes of $N = 160$ and $N = 480$ for a small pulse $\varepsilon = 10^{-5}$

in all the numerical examples we have tested. For example, the average number of time steps is around 3 ($\varepsilon = 0.2$) and 4 ($\varepsilon = 10^{-5}$) for P^2-DG applied to the current example. Since the cost of the DG-interpolation scheme has been studied extensively in [45], we will not discuss this further in this work.

To verify the well-balance and positivity-preserving properties of the MM-DG method we modify the bottom topography (5.4) to contain a dry region (near $x = 1.5$),

$$B(x) = \begin{cases} 0.5(\cos(10\pi(x-1.5))+1), & \text{for } x \in (1.4, 1.6) \\ 0, & \text{for } x \in (0, 1.4) \cup (1.6, 2) \end{cases} \tag{5.5}$$

We repeat the computation with $\varepsilon = 10^{-5}$. The bottom topography, the initial water level, and the mesh trajectories of $N = 160$ obtained with P^2 MM-DG method are plotted in Fig. 11. The mesh points are concentrated around the shock waves and the non-flat topography region. Interestingly, the mesh trajectories show that the right moving shock stops after it hits the dry region.



(a) hu: FM 160 vs MM 160

(b) Close view of (a)

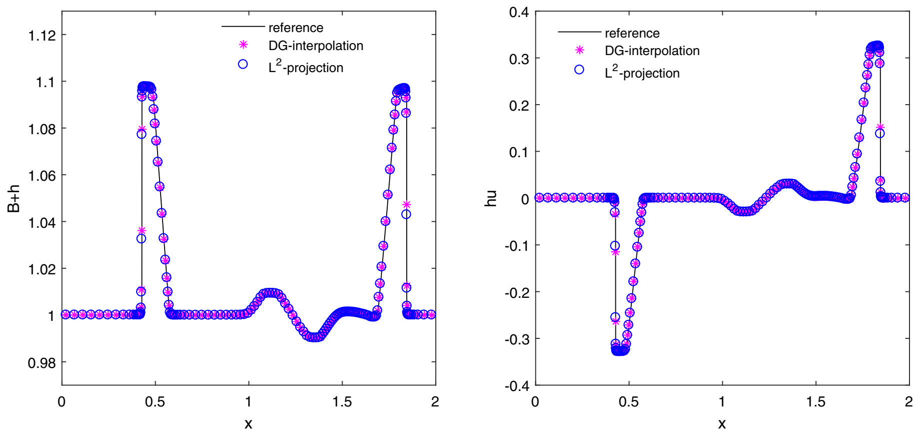
(c) hu: FM 480 vs MM 160

(d) Close view of (c)

Fig. 8 Example 5.3. The water discharge hu at $t = 0.2$ obtained with P^2-DG and a moving mesh of $N = 160$ are compared with those obtained with the fixed meshes of $N = 160$ and $N = 480$ for a small pulse $\varepsilon = 10^{-5}$

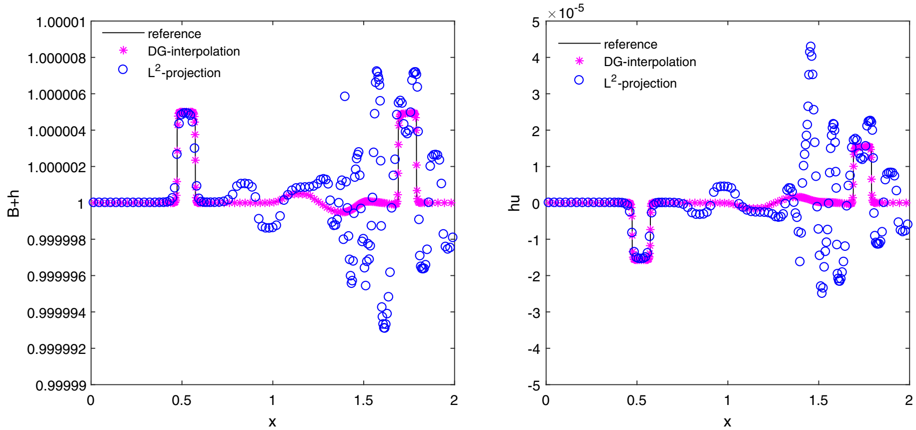
The water surface $B + h$ and discharge hu obtained with P^2-DG and a moving mesh of $N = 160$ and fixed meshes of $N = 160$ and $N = 640$ are plotted in Figs. 12 and 13. The results show that the MM-DG method with moving or fixed meshes is able to capture the waves of small perturbation for the situation with dry regions. Moreover, the moving mesh solutions with $N = 160$ are more accurate than those with fixed mesh of $N = 160$ and $N = 640$ and contain no visible spurious numerical oscillations.

The L^1-error of the approximation in the bottom topography B (5.5) (see Fig. 11a) at the final time of the simulation is plotted in Fig. 14. The result shows the second- and third-order of convergence, which are expected for P^1- and P^2-DG methods, respectively. This indicates that the DG interpolation scheme plus a high-order correction works well for updating B.



(a) $B+h$ **(b)** hu

Fig. 9 Example 5.3. The water surface $B+h$ and discharge hu at $t=0.2$ are obtained with P^2-DG and a moving mesh of $N=160$ for a large pulse $\varepsilon=0.2$. Comparison is made for the use of DG-interpolation or L^2-projection in updating B



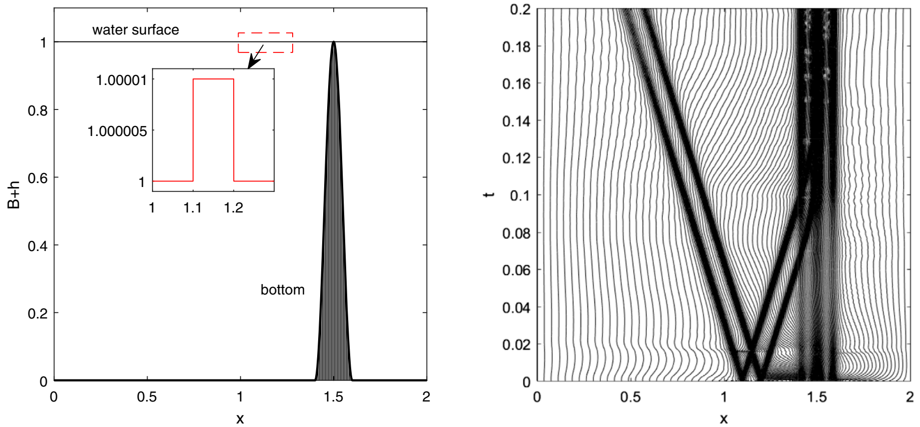
(a) $B+h$ **(b)** hu

Fig. 10 Example 5.3. The water surface $B+h$ and discharge hu at $t=0.2$ are obtained with P^2-DG and a moving mesh of $N=160$ for a small pulse $\varepsilon=10^{-5}$. Comparison is made for the use of DG-interpolation or L^2-projection in updating B

Example 5.4 (The Riemann problem for the 1D SWEs with a step bottom topography.)

In this test we solve the 1D SWEs with a step bottom topography [2,29],

$$B(x)=\begin{cases}0, & \text{for } x\in(-10,0)\\ 1, & \text{for } x\in(0,10)\end{cases}$$



(a) Initial surface and bottom **(b)** Mesh trajectories

Fig. 11 Example 5.3. (a) The initial water surface $B + h$ and the bottom B (5.5) for the small perturbation test with a dry region. (b) The mesh trajectories obtained with P^2 MM-DG method of $N = 160$

The initial conditions are given by

$$h(x,0) = \begin{cases} 4, & \text{for } x \le 0 \\ 1, & \text{otherwise} \end{cases} \qquad u(x,0) = \begin{cases} 5, & \text{for } x \le 0 \\ -0.9, & \text{otherwise.} \end{cases}$$

The solution contains two hydraulic jumps/shocks and a stationary step subcritical transition. The first shock moves to the left while the second one to the right. The stationary step transition of the flow occurs at $x = 0$.

The mesh trajectories for the P^2-DG method with a moving mesh of $N = 100$ are shown in Fig. 15. We can see that the mesh has high element concentration at the location of two hydraulic jumps/shocks and the stationary step transition. The moving mesh solutions of $N = 100$ at $t = 1$ and the fixed mesh solutions obtained with $N = 100$ and $N = 800$ are shown in Figs. 16 and 17 for water surface level and water discharge, respectively. The results show that the moving mesh solution ($N = 100$) provides a better resolution of the shocks than that with the fixed mesh of $N = 100$ and $N = 800$.

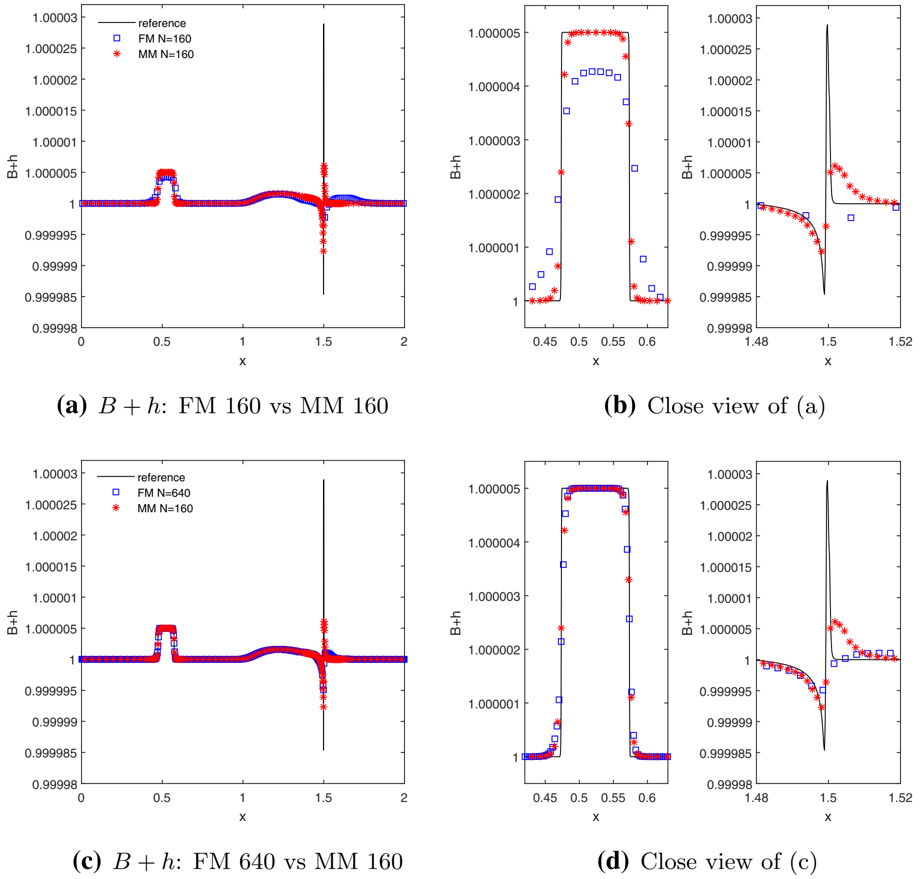
Example 5.5 (The rarefaction and shock waves test for the 1D SWEs with flat and wavy bottom topographies.)

In this test we compute the 1D SWEs with two sets of bottom topographies $B(x)$. The initial conditions are given by

$$h(x,0) = \begin{cases} 2 - B(x), & \text{for } x \in (-10, 1) \\ 0.35 - B(x), & \text{for } x \in (1, 10) \end{cases} \qquad u(x,0) = \begin{cases} 1, & \text{for } x \in (-10, 1) \\ 0, & \text{for } x \in (1, 10). \end{cases}$$

We choose the transmissive boundary conditions and compute the solution up to $t = 1$.

We first consider $B(x) = 0$. The solution contains a rarefaction wave moving to the left and a shock wave (hydraulic jump) traveling to the right. The mesh trajectories with a moving mesh of $N = 100$ are shown in Fig. 18. One can see that the mesh has high element concentration around the rarefaction and shock. In Figs. 19 and 20, we plot the water surface $B+h$ and water discharge hu at $t = 1$ obtained with P^2-DG and a moving mesh of $N = 100$



(a) $B + h$: FM 160 vs MM 160

(b) Close view of (a)

(c) $B + h$: FM 640 vs MM 160

(d) Close view of (c)

Fig. 12 Example 5.3 with the bottom topography (5.5) with a dry region. The water surface $B + h$ at $t = 0.2$ obtained with P^2-DG and a moving mesh of $N = 160$ are compared with those obtained with the fixed meshes of $N = 160$ and $N = 640$

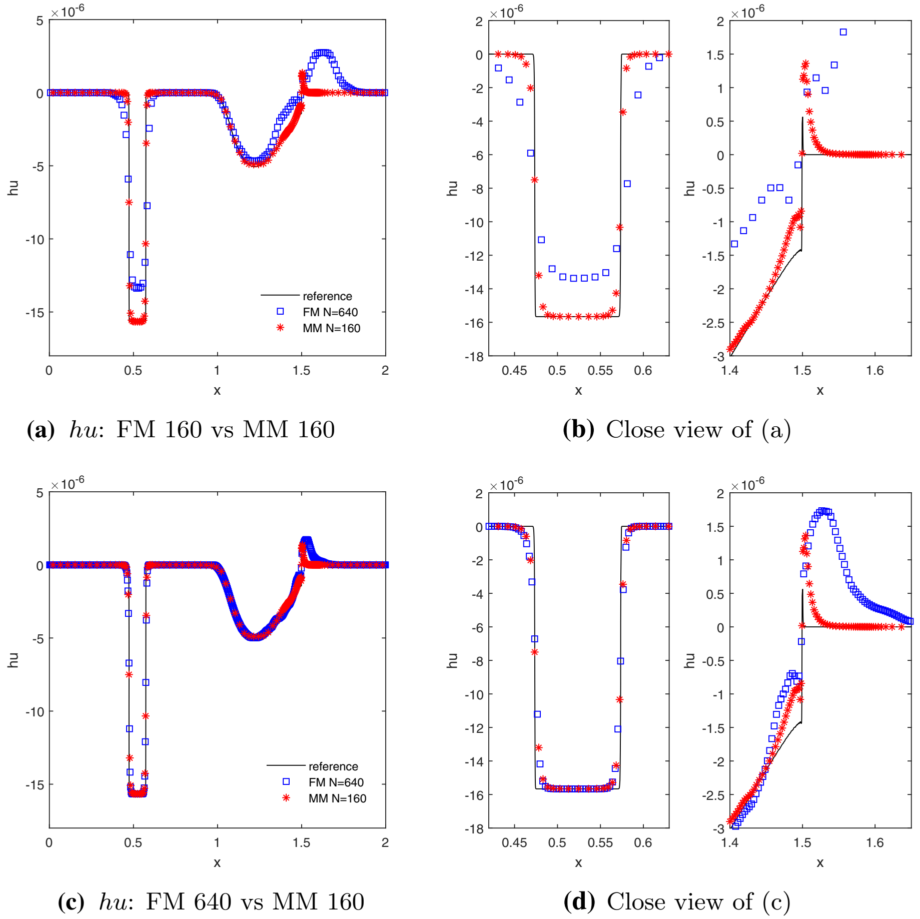
and fixed meshes of $N = 100$ and $N = 1280$. It can be seen that the moving mesh solutions of $N = 100$ are more accurate than those with a fixed mesh of $N = 100$ and comparable with those with a fixed mesh of $N = 1280$.

To show the effects of non-flat bottom topographies on water flow, we repeat this example with a wavy bottom topography [36],

$$B(x) = \begin{cases} 0.3\cos^{30}(\frac{\pi}{2}(x-1)), & \text{for } 0 \le x \le 2 \\ 0, & \text{otherwise.} \end{cases} \tag{5.6}$$

The solution shows more complex features, made up of a left rarefaction wave and two hydraulic jumps/shocks. One shock near the location of $x = 2$ occurs in the flow over the non-flat bed topography.

The moving mesh trajectories ($N = 160$) are plotted in Fig. 21. One can see that the mesh elements concentrate around the rarefaction and hydraulic jumps/shocks. In Figs. 22 and 23, we plot the water surface level $B + h$ and water discharge hu at $t = 1$ obtained with P^2-DG and a moving mesh of $N = 160$ and fixed meshes of $N = 160$ and $N = 1280$. These results



(a) hu: FM 160 vs MM 160

(b) Close view of (a)

(c) hu: FM 640 vs MM 160

(d) Close view of (c)

Fig. 13 Example 5.3 with the bottom topography (5.5) with a dry region. The water discharge hu at $t = 0.2$ obtained with P^2-DG and a moving mesh of $N = 160$ are compared with those obtained with the fixed meshes of $N = 160$ and $N = 640$

Fig. 14 Example 5.3 with the bottom topography (5.5) with a dry region. The L^1-error of the approximation to the bottom topography B at the final time of the simulation is plotted as a function of N

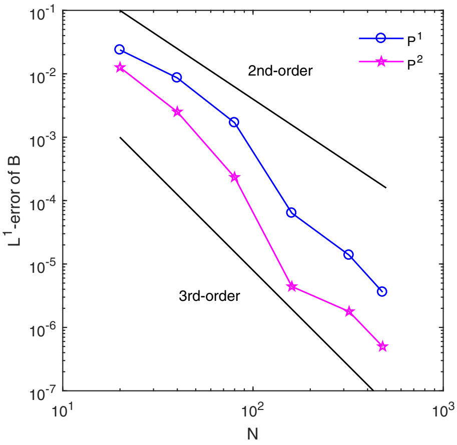
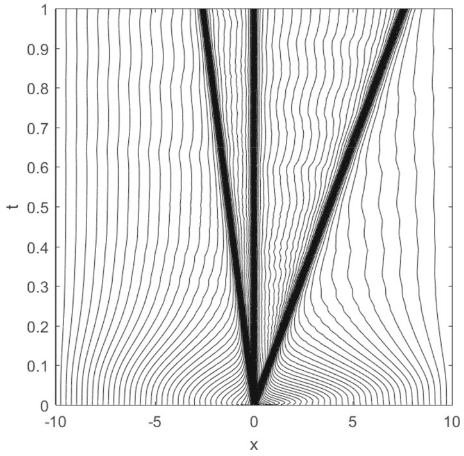
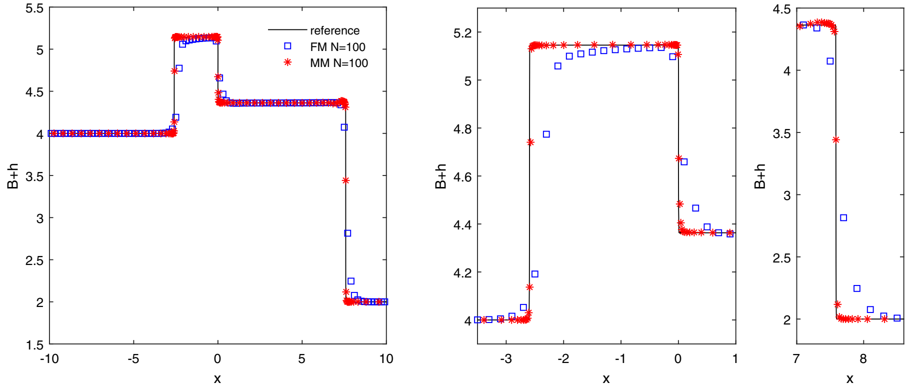


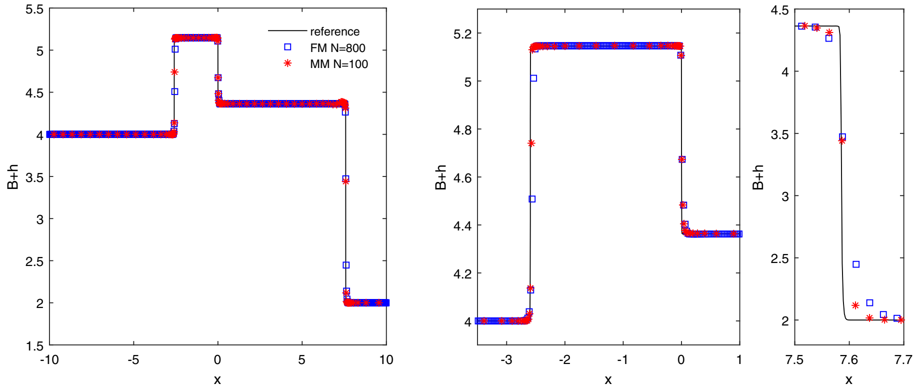
Fig. 15 Example 5.4. The mesh trajectories are obtained with P^2-DG method with a moving mesh of $N = 100$





(a) FM 100 vs MM 100

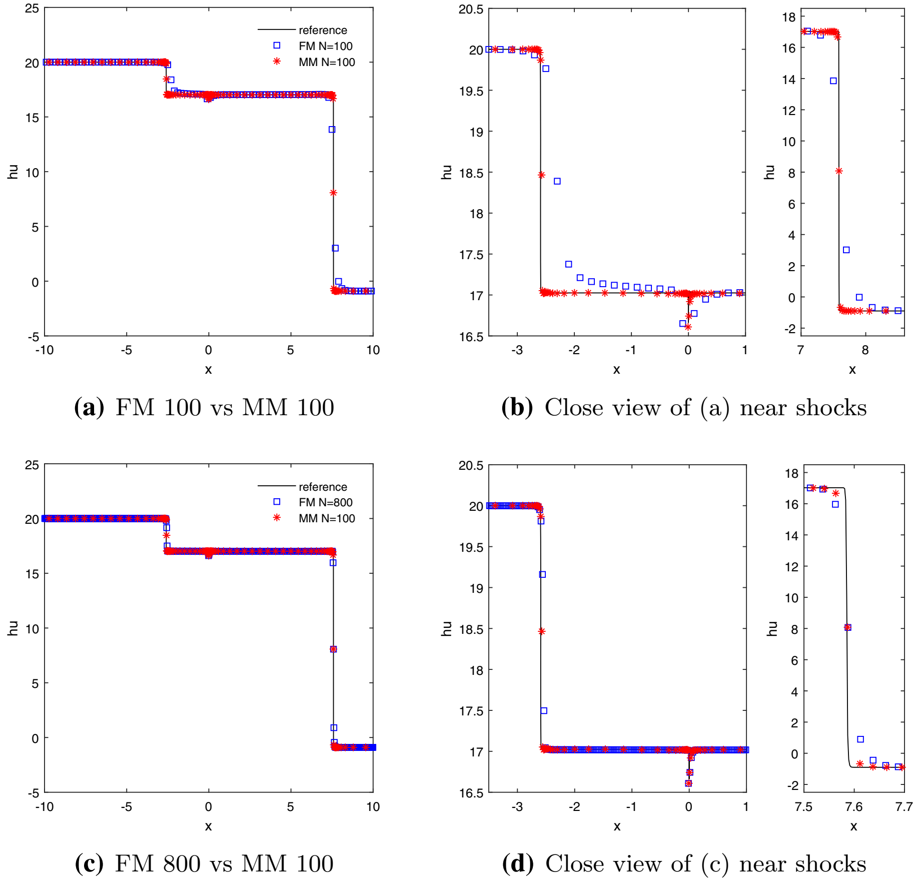
(b) Close view of (a) near shocks



(c) FM 800 vs MM 100

(d) Close view of (c) near shocks

Fig. 16 Example 5.4. The water surface $B + h$ at $t = 1$ obtained with P^2-DG method with a moving mesh of $N = 100$ is compared with those obtained with the fixed meshes of $N = 100$ and $N = 800$



(a) FM 100 vs MM 100

(b) Close view of (a) near shocks

(c) FM 800 vs MM 100

(d) Close view of (c) near shocks

Fig. 17 Example 5.4. The water discharge hu at $t = 1$ obtained with P^2-DG method with a moving mesh of $N = 100$ is compared with those obtained with the fixed meshes of $N = 100$ and $N = 800$

show that the moving mesh solutions of $N = 160$ are more accurate than those with a fixed mesh of $N = 160$ and comparable with that with the fixed mesh of $N = 1280$. Moreover, the MM-DG method does a good job in resolving the shock near $x = 2$ which seems to be a difficult structure for a fixed mesh to resolve.

Example 5.6 (The lake-at-rest steady-state flow test for the 2D SWEs.)

We choose this example to verify the well-balance property of the MM-DG scheme over non-flat bottom topographies. We take a bottom topography with an isolated elliptical-shaped bump [25] as

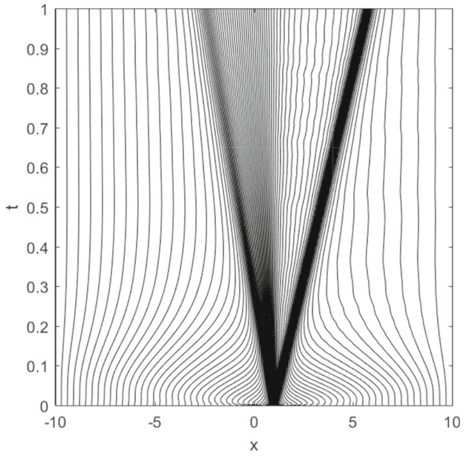
$$B(x, y) = 0.8e^{-50\left((x-0.5)^2+(y-0.5)^2\right)}, \quad (x, y) \in (0, 1) \times (0, 1). \tag{5.7}$$

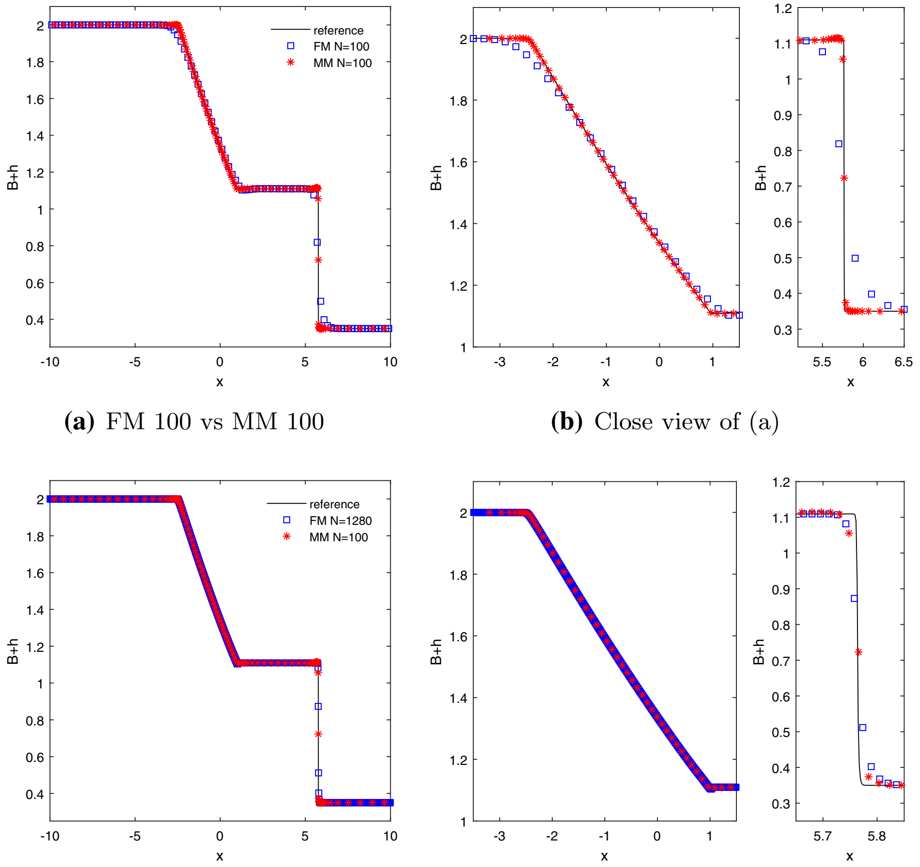
The initial depth of water and velocities are given by

$$h(x, y, 0) = 1 - B(x, y), \quad u(x, y, 0) = 0, \quad v(x, y, 0) = 0.$$

We use periodic boundary conditions and compute the solution up to $t = 0.1$ on moving meshes. The flow surface should remain steady. An initial triangular mesh, shown in Fig. 24,

Fig. 18 Example 5.5. The mesh trajectories for $B = 0$ are obtained with P^2-DG and a moving mesh of $N = 100$





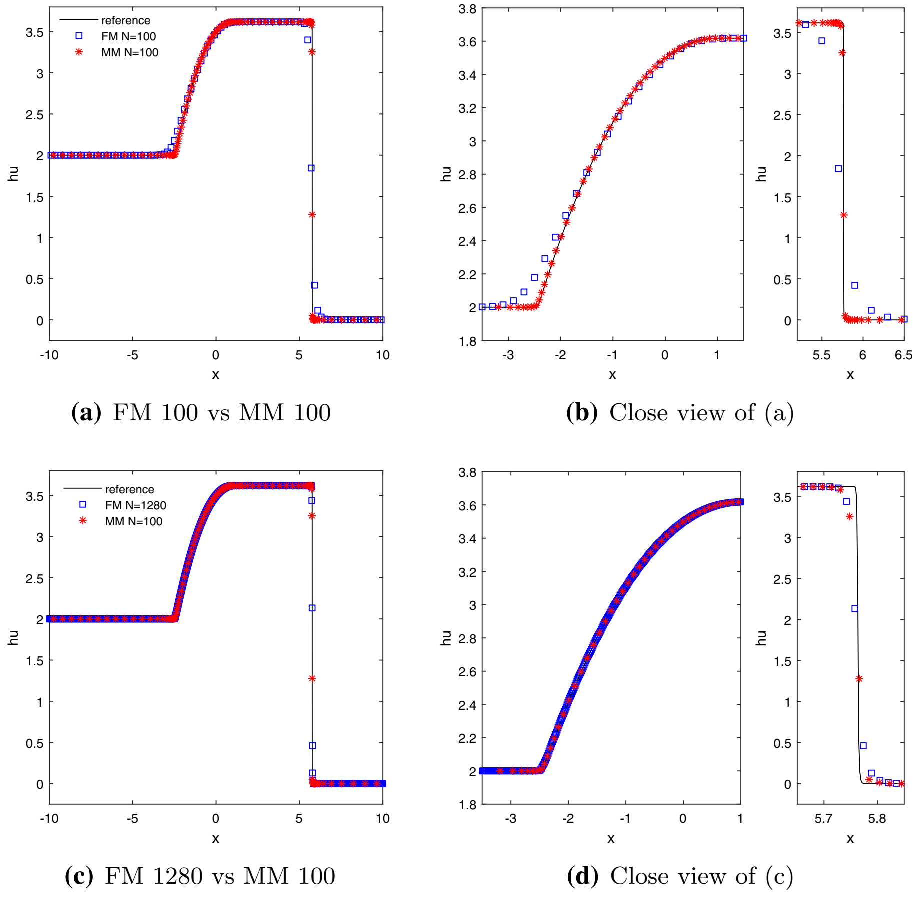
(a) FM 100 vs MM 100

(b) Close view of (a)

(c) FM 1280 vs MM 100

(d) Close view of (c)

Fig. 19 Example 5.5. The water depth h (for $B = 0$) at $t = 1$ obtained with P^2-DG and a moving mesh of $N = 100$ is compared with those obtained with the fixed meshes of $N = 100$ and $N = 1280$



(a) FM 100 vs MM 100

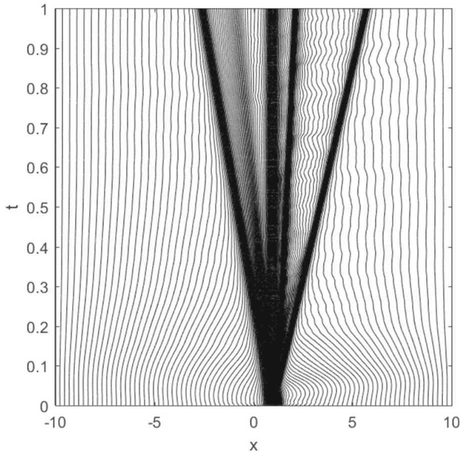
(b) Close view of (a)

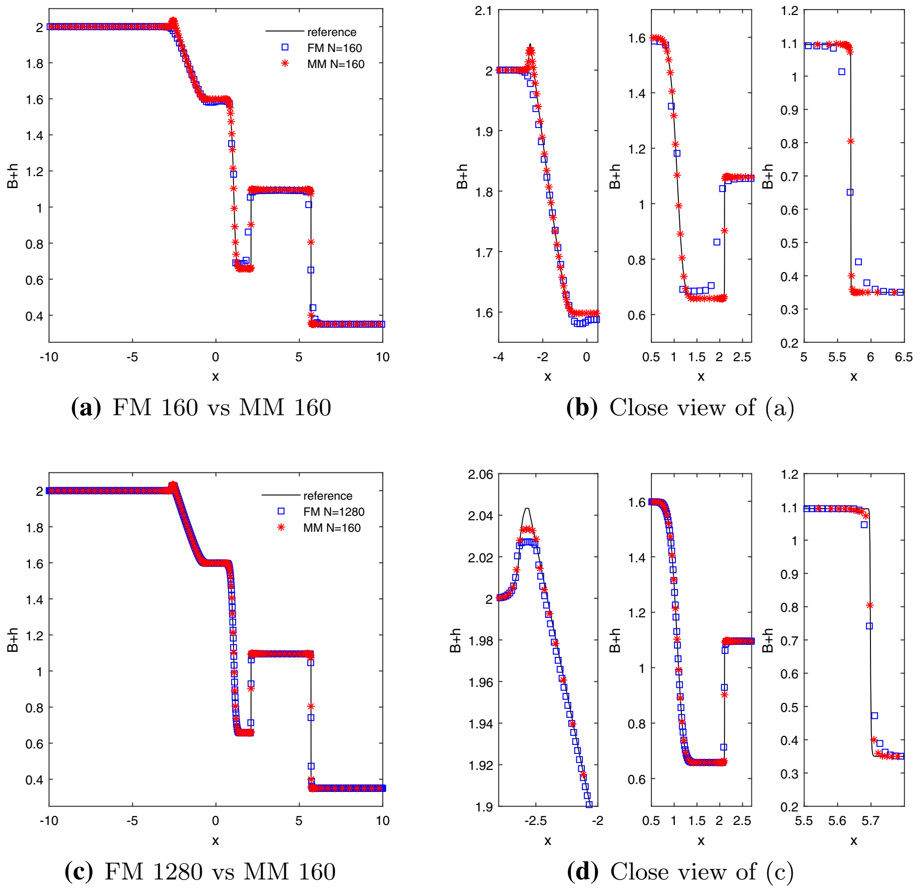
(c) FM 1280 vs MM 100

(d) Close view of (c)

Fig. 20 Example 5.5. The water discharge hu (for $B = 0$) at $t = 1$ obtained with P^2-DG and a moving mesh of $N = 100$ is compared with those obtained with the fixed meshes of $N = 100$ and $N = 1280$

Fig. 21 Example 5.5. The mesh trajectories (for B defined in (5.6)) are obtained with P^2-DG and a moving mesh of $N = 160$





(a) FM 160 vs MM 160

(b) Close view of (a)

(c) FM 1280 vs MM 160

(d) Close view of (c)

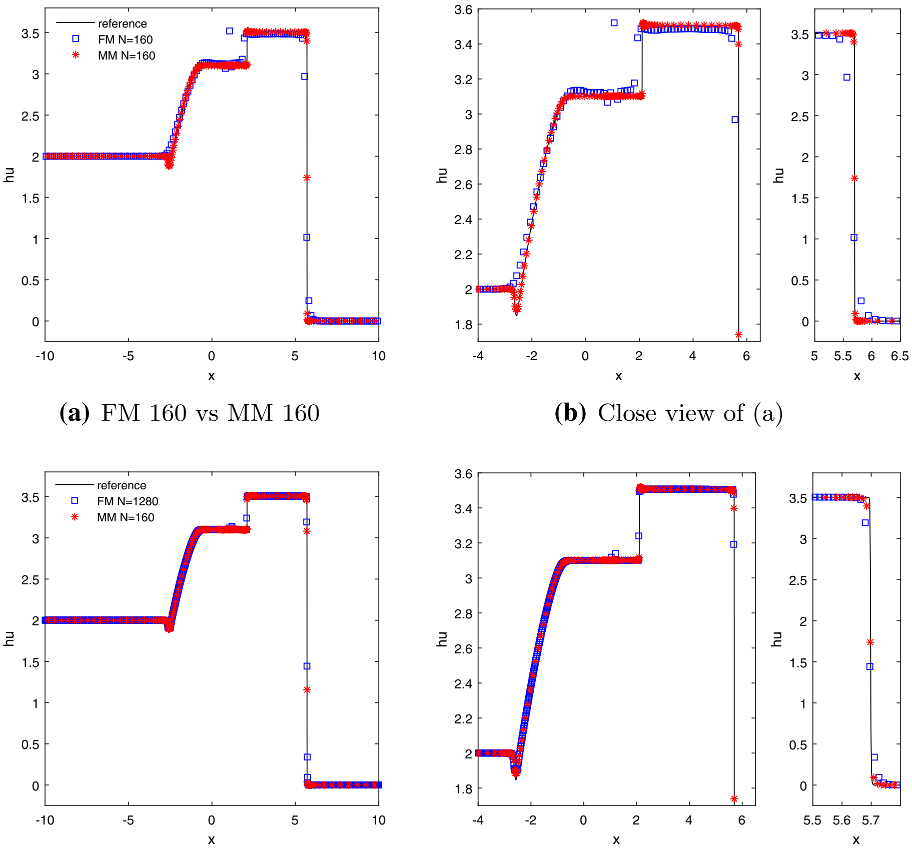
Fig. 22 Example 5.5. The water surface level $B + h$ (for B defined in (5.6)) at $t = 1$ obtained with P^2-DG and a moving mesh of $N = 160$ is compared with those obtained with the fixed meshes of $N = 160$ and $N = 1280$

is formed by dividing each cell of a rectangular mesh into four triangular elements. The L^1 and L^∞ error for $h + B$ and discharges hu and hv is listed in Table 6 for the moving mesh DG method with B updated with DG-interpolation. They show that the MM-DG method with B updated with DG-interpolation maintains the lake-at-rest steady state to the level of round-off error in both L^1 and L^∞ norm.

Example 5.7 (The perturbed lake-at-rest steady-state flow test for the 2D SWEs.)

We use this example, first proposed by LeVeque [25], to demonstrate the ability of our well-balanced MM-DG scheme to capture small perturbations over the lake-at-rest water surface. The bottom topography is an isolated elliptical shaped hump,

$$B(x, y) = 0.8e^{-5(x-0.9)^2-50(y-0.5)^2}, \quad (x, y) \in (-1, 2) \times (0, 1).$$



(a) FM 160 vs MM 160 **(b)** Close view of (a)

(c) FM 1280 vs MM 160 **(d)** Close view of (c)

Fig. 23 Example 5.5. The water discharge hu (for B defined in (5.6)) at $t = 1$ obtained with P^2-DG and a moving mesh of $N = 160$ is compared with those obtained with the fixed meshes of $N = 160$ and $N = 1280$

Fig. 24 Example 5.6. An initial triangular mesh used in the computation is formed by dividing each cell of a rectangular mesh into 4 triangular elements

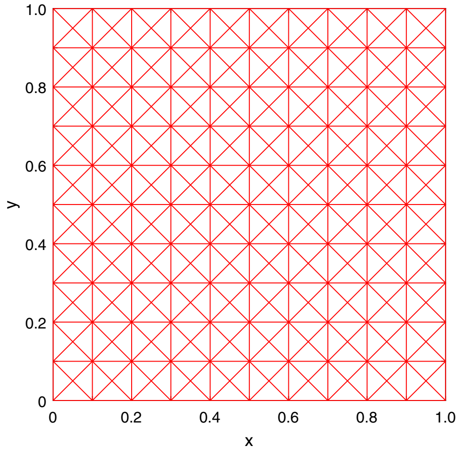
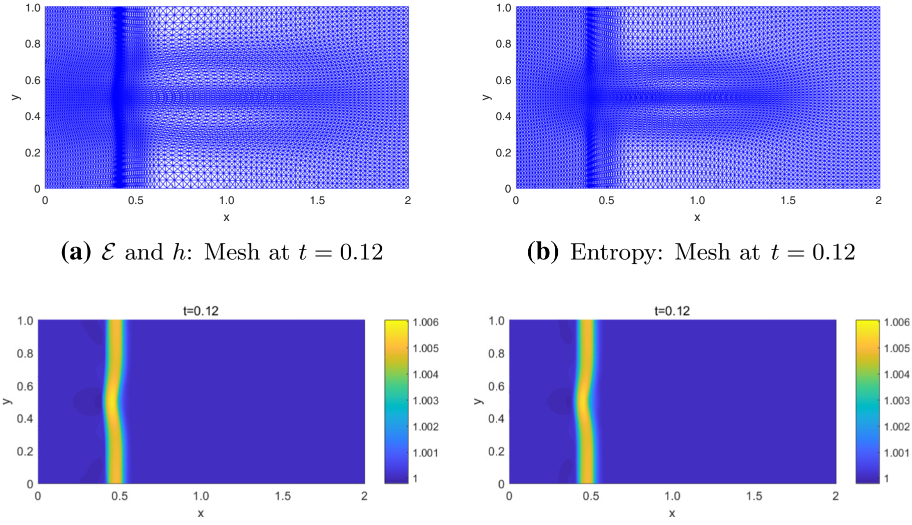


Table 6 Example 5.6. Well-balance test for the moving mesh DG method with B updated with DG-interpolation, over an isolated elliptical shaped hump bottom topography (5.7)

N	$h+B$		hu		hv	
	L^1-error	L^∞-error	L^1-error	L^∞-error	L^1-error	L^∞-error
P^1 MM-DG method						
$10 \times 10 \times 4$	1.589E-16	3.077E-16	1.359E-16	8.941E-16	1.361E-16	8.588E-16
$20 \times 20 \times 4$	2.179E-16	4.641E-16	1.775E-16	1.658E-15	1.787E-16	1.627E-15
$40 \times 40 \times 4$	3.344E-16	8.298E-16	2.793E-16	3.069E-15	2.813E-16	3.054E-15
P^2 MM-DG method						
$10 \times 10 \times 4$	4.259E-16	1.855E-15	8.005E-16	4.425E-15	8.105E-16	4.288E-15
$20 \times 20 \times 4$	3.870E-16	1.944E-15	9.724E-16	6.313E-15	9.104E-16	5.361E-15
$40 \times 40 \times 4$	4.257E-16	2.158E-15	1.467E-15	1.129E-14	1.204E-15	8.449E-15



(a) $\mathcal{E}$ and h: Mesh at $t = 0.12$

(b) Entropy: Mesh at $t = 0.12$

(c) $\mathcal{E}$ and h: $h + B$ at $t = 0.12$

(d) Entropy: $h + B$ at $t = 0.12$

Fig. 25 Example 5.7. The mesh and contours of $B + h$ at $t = 0.12$ are obtained with the P^2 MM-DG method and a moving mesh of $N = 150 \times 50 \times 4$. The metric tensor is computed based on the equilibrium variable $\mathcal{E}$ and the water depth h (left column) or the entropy/total energy (right column)

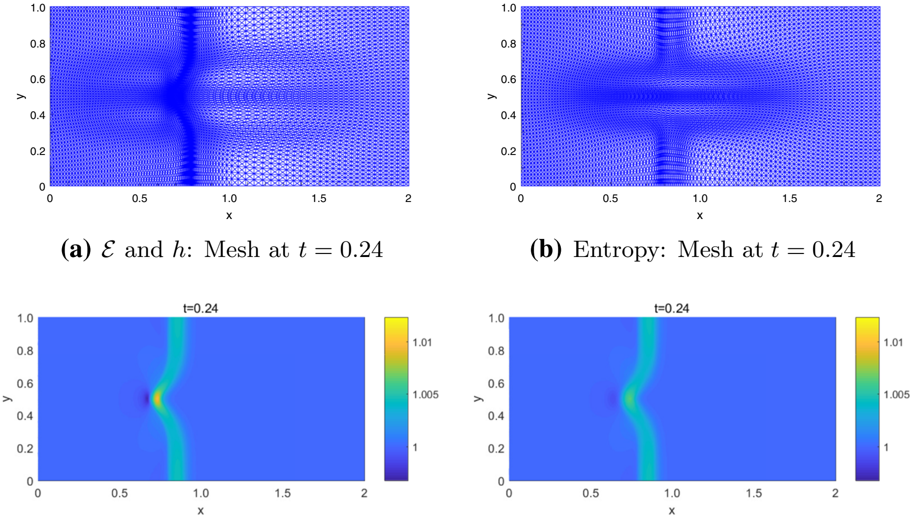
The initial depth of water and velocities are given by

$$h(x, y, 0) = \begin{cases} 1 - B(x, y) + 0.01, & \text{for } x \in (0.05, 0.15) \\ 1 - B(x, y), & \text{otherwise} \end{cases}$$

$$u(x, y, 0) = 0, \qquad v(x, y, 0) = 0.$$

As time being, the initial perturbation splits into two waves, propagating left and right at the characteristic speeds $\pm\sqrt{gh}$. The reflection boundary conditions [37] are used for all domain boundary.

The mesh and contours of $B + h$ at $t = 0.12$, 0.24, 0.36, and 0.48 obtained with the P^2 MM-DG method and a moving mesh of $N = 150 \times 50 \times 4$ are shown in Figs. 25,



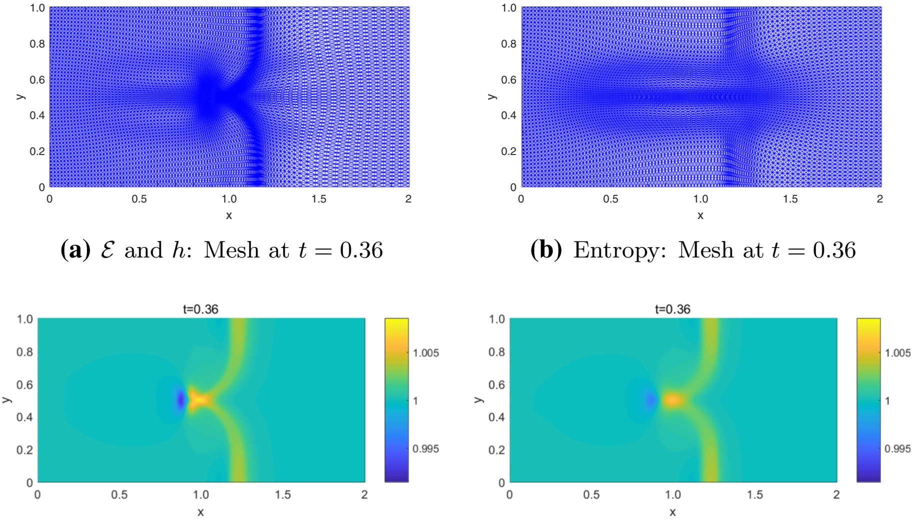
(a) $\mathcal{E}$ and h: Mesh at $t = 0.24$

(b) Entropy: Mesh at $t = 0.24$

(c) $\mathcal{E}$ and h: $h + B$ at $t = 0.24$

(d) Entropy: $h + B$ at $t = 0.24$

Fig. 26 Continuation of Fig. 25: $t = 0.24$



(a) $\mathcal{E}$ and h: Mesh at $t = 0.36$

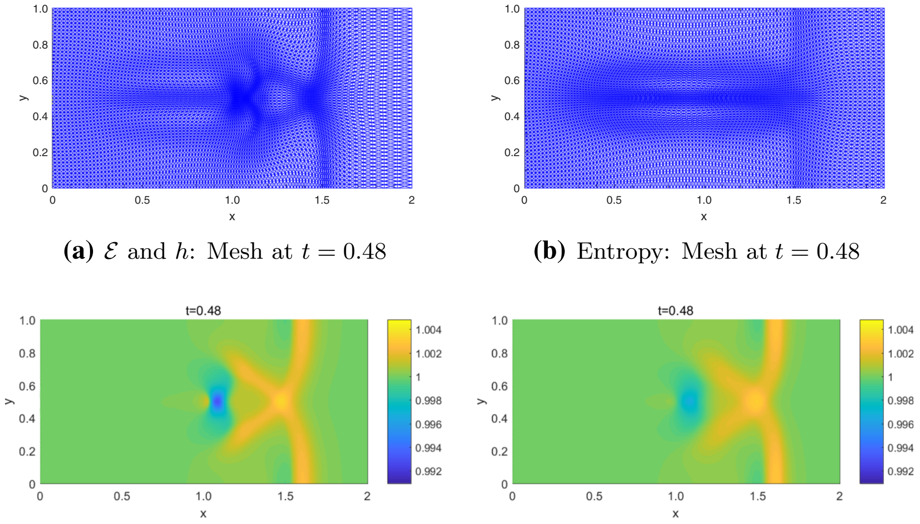
(b) Entropy: Mesh at $t = 0.36$

(c) $\mathcal{E}$ and h: $h + B$ at $t = 0.36$

(d) Entropy: $h + B$ at $t = 0.36$

Fig. 27 Continuation of Fig. 25: $t = 0.36$

26, 27, 28. Recall that these results are obtained with the metric tensor (4.3) based on the equilibrium variable $\mathcal{E} = \frac{1}{2}(u^2 + v^2) + g(h + B)$ and the water depth h. For comparison purpose, we also include the results obtained the metric tensor based on the entropy/total energy $E = \frac{1}{2}(hu^2 + hv^2) + \frac{1}{2}gh^2 + ghB$. One can see that the distribution of the mesh concentration based on $\mathcal{E}$ and h is consistent with the contours of $B + h$ while capturing the



(a) $\mathcal{E}$ and h: Mesh at $t = 0.48$

(b) Entropy: Mesh at $t = 0.48$

(c) $\mathcal{E}$ and h: $h + B$ at $t = 0.48$

(d) Entropy: $h + B$ at $t = 0.48$

Fig. 28 Continuation of Fig. 25: $t = 0.48$

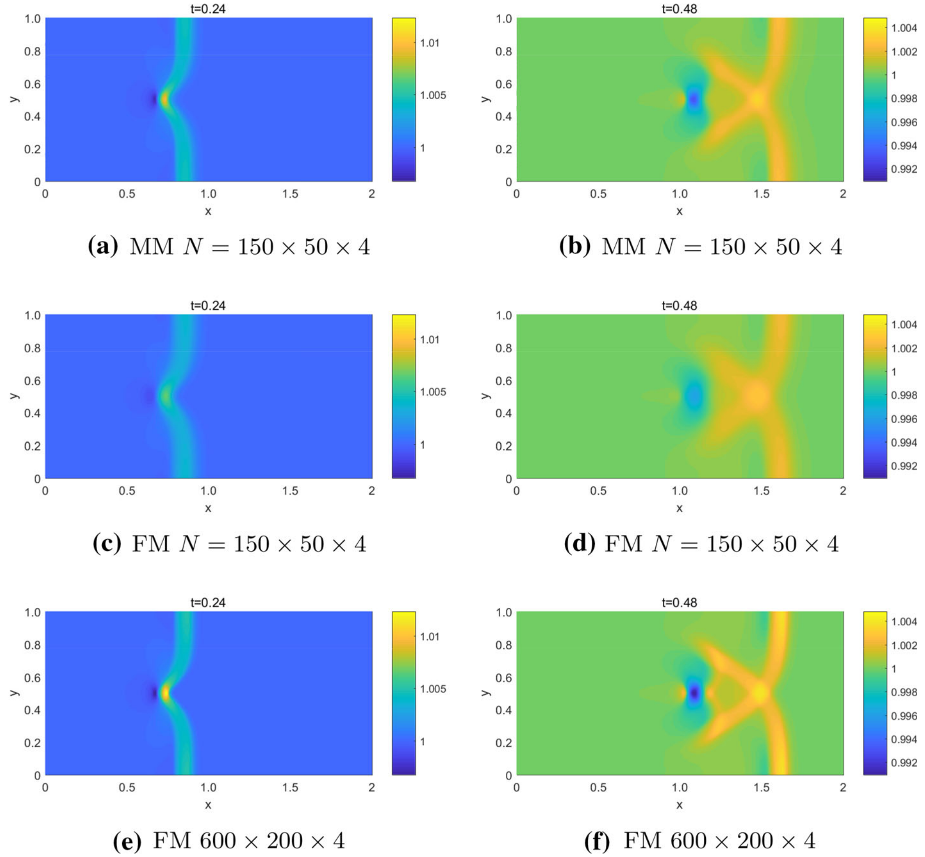
complex features in small perturbations. On the other hand, the concentration of the mesh based on E does not reflect fully the variations of $B + h$ especially at $t = 0.24$, 0.36, and 0.48. The advantages of using $\mathcal{E}$ and h over E in mesh adaptation are clear.

The contours of $h + B$ at $t = 0.24$ and 0.48 obtained using the P^2 MM-DG method with a moving mesh of $N = 150 \times 50 \times 4$ and the fixed meshes of $N = 150 \times 50 \times 4$ and $N = 600 \times 200 \times 4$ are shown in Fig. 29. In Fig. 30, the cut of the corresponding results along the line $y = 0.495$ is compared for the moving and fixed meshes. We can see that the moving mesh solution with $N = 150 \times 50 \times 4$ is more accurate than that with a fixed mesh of $N = 150 \times 50 \times 4$ and comparable with that with a fixed mesh of $N = 600 \times 200 \times 4$.

6 Conclusions and Further Comments

We have presented a high-order well-balanced positivity-preserving adaptive moving mesh DG method for the numerical solution of the SWEs with non-flat bottom topography. The method is of rezoning type and contains three main components at each time step, the generation of the new mesh by node redistribution/movement, the interpolation of the physical variables from the old mesh to the new one, and the numerical solution of the SWEs on the new mesh that is fixed over the time step. A focus of the study is the well-balance and positivity-preserving properties of the method that are crucial to its ability to simulate perturbation waves to the lake-at-rest steady state such as waves on a lake or tsunami waves in deep ocean.

We have employed the MMPDE moving mesh scheme to generate the new mesh at each time step. A key of the MMPDE scheme is to define the metric tensor that provides the information needed for controlling the size, shape, and orientation of mesh elements over the whole spatial domain. The numerical examples have shown that the metric tensor based



(a) MM $N = 150 \times 50 \times 4$ **(b)** MM $N = 150 \times 50 \times 4$

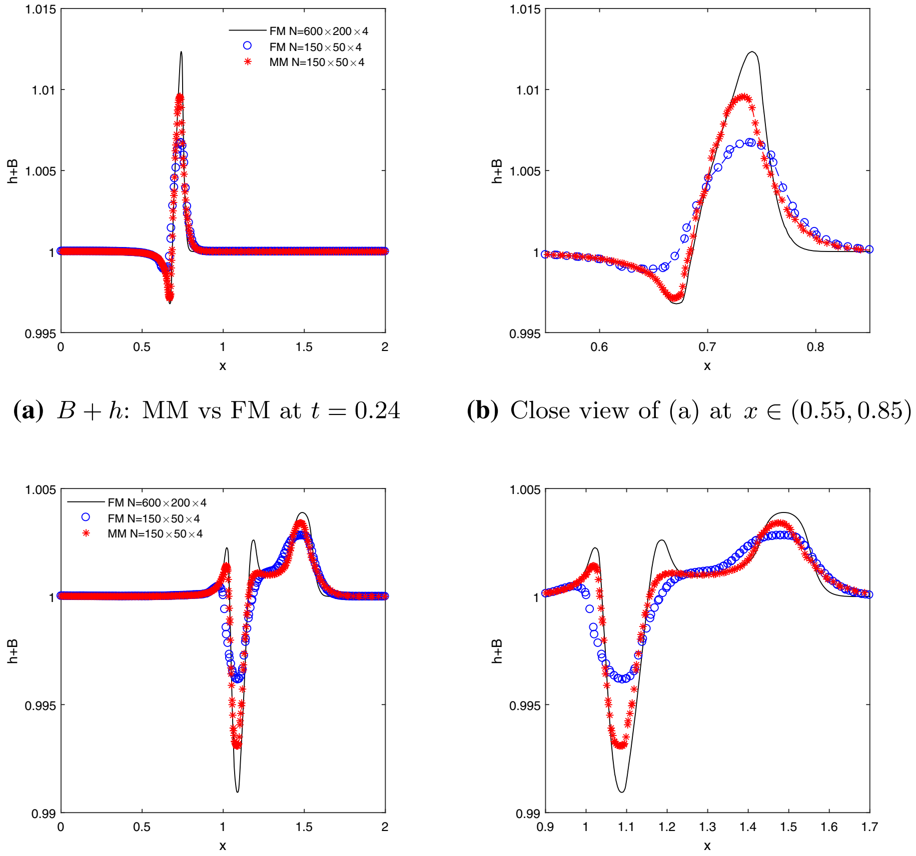
(c) FM $N = 150 \times 50 \times 4$ **(d)** FM $N = 150 \times 50 \times 4$

(e) FM $600 \times 200 \times 4$ **(f)** FM $600 \times 200 \times 4$

Fig. 29 Example 5.7. The contours of $h+B$ at $t = 0.24$ (left column) and 0.48 (right column) are obtained with the P^2 MM-DG method and a moving mesh of $N = 150 \times 50 \times 4$ and the fixed meshes of $N = 150 \times 50 \times 4$ and $N = 600 \times 200 \times 4$

on the entropy/total energy $E = \frac{1}{2}(hu^2 + hv^2) + \frac{1}{2}gh^2 + ghB$, a common choice in the adaptive mesh simulation of shock waves, does not lead to fully correct mesh concentration for waves of small magnitude. Instead, we have proposed to use the equilibrium variable $\mathcal{E} = \frac{1}{2}(u^2 + v^2) + g(h + B)$ combined with the water depth h in the computation of the metric tensor and demonstrated numerically that they are well suited for the simulation of the lake-at-rest steady state and its perturbations.

We have used a fixed mesh well-balanced DG scheme for solving the SWEs on the new mesh. However, to ensure the well-balance property of the overall MM-DG method, we need to pay special attention to the interpolation of the flow variables and bottom topography, the slope limiting, and the positivity-preserving (PP) limiting. We have proposed to use a DG-interpolation scheme (cf. sect. 3) for the purpose. The scheme has high-order accuracy, preserves constant solutions, and is conservative while being economic to implement: it requires $\mathcal{O}(N_v)$ operations in each use, where N_v is the number of vertices of the mesh. We note that although the bottom topography has been assumed to be a given time-independent function in this work, it needs to be updated at each time step from the old mesh to the new one due to the movement of the mesh. Moreover, the same scheme should be used for updating



(a) $B + h$: MM vs FM at $t = 0.24$

(b) Close view of (a) at $x \in (0.55, 0.85)$

(c) $B + h$: MM vs FM at $t = 0.48$

(d) Close view of (c) at $x \in (0.9, 1.7)$

Fig. 30 Example 5.7. The cut of the water surface $h + B$ along the line $y = 0.495$ at $t = 0.24$ and 0.48 obtained using the P^2 MM-DG method with a moving mesh of $N = 150 \times 50 \times 4$ is compared with those obtained with the fixed meshes of $N = 150 \times 50 \times 4$ and $N = 600 \times 200 \times 4$

both the bottom topography and flow variables in order to attain the well-balance property of the MM-DG method. We have used the DG-interpolation scheme for both. It is worth mentioning that the MM-DG scheme also works without modification for a time-dependent bottom topography, e.g., in tsunamis on the moving ocean floor. In this case, the update of the bottom topography from mesh to mesh is necessary for both fixed and moving mesh computation.

To ensure the nonnegativity of the water depth, we use the PP-DG-interpolation (DG-interpolation with PP limiter) to interpolate the water depth from the old mesh to the new one, and apply a scaling PP limiter to the water depth every time after we use the TVB limiter. However, the PP limiter will destroy the well-balance property. To restore the property, we have proposed to make a high-order correction to the approximation of the bottom topography according to the modifications in the water depth due to the PP limiting; see (2.19) and (2.21).

A selection of numerical examples in one and two dimensions have been presented to demonstrate the well-balance property, positivity-preserving, and high-order accuracy of the

MM-DG method. They have also shown that the method is well suited for the numerical simulation of the lake-at-rest steady state and its perturbations. Particularly, the mesh concentration correctly reflects structures in the flow variables and bottom topography and leads to more accurate numerical solutions than a fixed mesh with the same number of elements.

While our focus here is on the well-balance and positivity-preservation properties of the DG method for SWEs on moving meshes, we would like to comment on the efficiency of the method. Generally speaking, the advantage of the MM-DG method in efficiency over a fixed mesh counterpart is problem-dependent. For example, it has been found that the MM-DG method can greatly improve the computational efficiency for the radiative transfer equation (cf. [44, Figs. 5 and 16] and [45, Figs. 13c and 14]) where an implicit time scheme is employed and for higher-accuracy solutions of hyperbolic conservation laws (cf. [28, Fig. 5.23]) where an explicit time scheme is used. However, for the SWE examples presented in this work, we found that the MM-DG method typically takes a longer CPU time than a fixed mesh DG method for the same level of accuracy and therefore has no gain in efficiency although it requires less mesh elements. A major issue is that the CFL conditions (2.17) and (3.7), which have been developed for uniform meshes, are very restrictive on adaptive meshes and result in very small time steps. Investigations on developing a more practical CFL conditions for nonuniform meshes have been underway.

References

1. Arpaia, L., Ricchiuto, M.: r-adaptation for shallow water flows: conservation, well balancedness, efficiency. Comput. Fluids **160**, 175–203 (2018)
2. Alcrudo, F., Benkhaldoun, F.: Exact solutions to the Riemann problem of the shallow water equations with a bottom step. Comput. Fluids **30**, 643–671 (2001)
3. Audusse, E., Bouchut, F., Bristeau, M.-O., Klein, R., Perthame, B.: A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. SIAM J. Sci. Comput. **25**, 2050–2065 (2004)
4. Bermudez, A., Vazquez, M.E.: Upwind methods for hyperbolic conservation laws with source terms. Comput. Fluids **23**, 1049–1071 (1994)
5. Cheng, J., Shu, C.-W.: A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. J. Comput. Phys. **227**, 1567–1596 (2007)
6. Cockburn, B., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. Math. Comp. **52**, 411–435 (1989)
7. Cockburn, B., Lin, S.-Y., Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. J. Comput. Phys. **84**, 90–113 (1989)
8. Cockburn, B., Shu, C.-W.: The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J. Comput. Phys. **141**, 199–224 (1998)
9. Cockburn, B., Shu, C.-W.: Runge-Kutta discontinuous Galerkin methods for convection dominated problems. J. Sci. Comput. **16**, 173–261 (2001)
10. Chinnayya, A., LeRoux, A., Seguin, N.: A well-balanced numerical scheme for the approximation of the shallow-water equations with topography: the resonance phenomenon. Int. J. Finite **1**, 1–33 (2004)
11. Donat, R., Martí, M.C., Martínez-Gavara, A., Mulet, P.: Well-balanced adaptive mesh refinement for shallow water flows. J. Comput. Phys. **257**, 937–953 (2014)
12. Eskilsson, C., Sherwin, S.J.: A triangular spectral/hp discontinuous Galerkin method for modelling 2D shallow water equations. Int. J. Numer. Meth. Fluids **45**, 605–623 (2004)
13. Ern, A., Piperno, S., Djadel, K.: A well-balanced Runge-Kutta discontinuous Galerkin method for the shallow-water equations with flooding and drying. Int. J. Numer. Meth. Fluids **58**, 1–25 (2008)
14. Goodman, J.B., LeVeque, R.J.: On the accuracy of stable schemes for 2D scalar conservation laws. Math. Comp. **45**, 15–21 (1985)
15. Huang, W., Ren, Y., Russell, R.: Moving mesh methods based on moving mesh partial differential equations. J. Comput. Phys. **113**, 279–290 (1994)
16. Huang, W., Ren, Y., Russell, R.: Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle. SIAM J. Numer. Anal. **31**, 709–730 (1994)

17. Huang, W.: Variational mesh adaptation: isotropy and equidistribution. J. Comput. Phys. **174**, 903–924 (2001)
18. Huang, W., Sun, W.: Variational mesh adaptation II: error estimates and monitor functions. J. Comput. Phys. **184**, 619–648 (2003)
19. Huang, W.: Mathematical principles of anisotropic mesh adaptation. Comm. Comput. Phys. **1**, 276–310 (2006)
20. Huang, W., Russell, R.: Adaptive Moving Mesh Methods. Applied Mathematical Sciences Series. Springer: New York **174**, (2011)
21. Huang, W., Kamenski, L.: A geometric discretization and a simple implementation for variational mesh generation and adaptation. J. Comput. Phys. **301**, 322–337 (2015)
22. Huang, W., Kamenski, L.: On the mesh nonsingularity of the moving mesh PDE method. Math. Comp. **87**, 1887–1911 (2018)
23. Liu, X.-D., Osher, S.: Non-oscillatory high order accurate self similar maximum principle satisfying shock capturing schemes. SIAM J. Numer. Anal. **33**, 760–779 (1996)
24. Lamby, P., Müller, S., Stiriba, Y.: Solution of shallow water equations using fully adaptive multiscale schemes. Int. J. Numer. Meth. Fluids **49**, 417–437 (2005)
25. LeVeque, R.: Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. J. Comput. Phys. **146**, 346–365 (1998)
26. Li, R., Tang, T.: Moving mesh discontinuous Galerkin method for hyperbolic conservation laws. J. Sci. Comput. **27**, 347–363 (2006)
27. Li, G., Song, L., Gao, J.: High order well-balanced discontinuous Galerkin methods based on hydrostatic reconstruction for shallow water equations. J. Comput. App. Math. **340**, 546–560 (2018)
28. Luo, D., Huang, W., Qiu, J.: A quasi-Lagrangian moving mesh discontinuous Galerkin method for hyperbolic conservation laws. J. Comput. Phys. **396**, 544–578 (2019)
29. Li, G., Lu, C., Qiu, J.: Hybrid well-balanced WENO schemes with different indicators for shallow water equations. J. Sci. Comput. **51**, 527–559 (2012)
30. Mavriplis, D., Yang, Z.: Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes. J. Comput. Phys. **213**, 557–573 (2006)
31. Mavriplis, D., Nastase, C.: On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes. J. Comput. Phys. **230**, 4285–4300 (2011)
32. Noelle, S., Xing, Y., Shu, C.-W.: High-order well-balanced finite volume WENO schemes for shallow water equation with moving water. J. Comput. Phys. **226**, 29–58 (2007)
33. Remacle, J.-F., Frazao, S.S., Li, X., Shephard, M.: Adaptive discontinuous Galerkin method for the shallow water equations. Int. J. Numer. Meth. Fluids **52**, 903–923 (2006)
34. Ricchiuto, M., Bollermann, A.: Stabilized residual distribution for shallow water simulations. J. Comput. Phys. **228**, 1071–1115 (2009)
35. Tang, H., Tang, T.: Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. SIAM J. Numer. Anal. **41**, 487–515 (2003)
36. Tang, H.: Solution of the shallow-water equations using an adaptive moving mesh method. Int. J. Numer. Meth. Fluids **44**, 789–810 (2004)
37. Tumolo, G., Bonaventura, L., Restelli, M.: A semi-implicit, semi-lagrangian, p-adaptive discontinuous Galerkin method for the shallow water equations. J. Comput. Phys. **232**, 46–67 (2013)
38. Xing, Y., Shu, C.-W.: High order finite difference WENO schemes with the exact conservation property for the shallow water equations. J. Comput. Phys. **208**, 206–227 (2005)
39. Xing, Y., Shu, C.-W.: High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. J. Comput. Phys. **214**, 567–598 (2006)
40. Xing, Y., Shu, C.-W.: A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. Comm. Comput. Phys. **1**, 100–134 (2006)
41. Xing, Y., Zhang, X., Shu, C.-W.: Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations. Adv. Water Resourc. **33**, 1476–1493 (2010)
42. Xing, Y., Zhang, X.: Positivity-preserving well-balanced discontinuous Galerkin methods for the shallow water equations on unstructured triangular meshes. J. Sci. Comput. **57**, 19–41 (2013)
43. Xing, Y.: Exactly well-balanced discontinuous Galerkin methods for the shallow water equations with moving water equilibrium. J. Comput. Phys. **257**, 536–553 (2014)
44. Zhang, M., Cheng, J., Huang, W., Qiu, J.: An adaptive moving mesh discontinuous Galerkin method for the radiative transfer equation. Comm. Comput. Phys. **27**, 1140–1173 (2020)
45. Zhang, M., Huang, W., Qiu, J.: High-order conservative positivity-preserving DG-interpolation for deforming meshes and application to moving mesh DG simulation of radiative transfer. SIAM J. Sci. Comput. **42**, A3109–A3135 (2020)

46. Zhang, Z., Naga, A.: A new finite element gradient recovery method: Superconvergence property. SIAM J. Sci. Comput. **26**, 1192–1213 (2005)
47. Zhou, F., Chen, G., Noelle, S., Guo, H.: A well-balanced stable generalized Riemann problem scheme for shallow water equations using adaptive moving unstructured triangular meshes. Int. J. Numer. Meth. Fluids **73**, 266–283 (2013)
48. Zhou, J.G., Causon, D.M., Mingham, C.G., Ingram, D.M.: The surface gradient method for the treatment of source terms in the shallow-water equations. J. Comput. Phys. **168**, 1–25 (2001)
49. Zhang, X., Shu, C.-W.: On positivity preserving high order discontinuous Galerkin methods for compressible Euler equations on rectangular meshes. J. Comput. Phys. **229**, 8918–8934 (2010)
50. Zhang, X., Xia, Y., Shu, C.-W.: Maximum-principle-satisfying and positivity-preserving High Order discontinuous Galerkin schemes for conservation Laws on Triangular Meshes. J. Sci. Comput. **50**, 29–62 (2012)