Computers and Fluids 134–135 (2016) 90–110

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Solving two-phase shallow granular flow equations with a well-balanced NOC scheme on multiple GPUs

Jian Zhai, Wei Liu, Li Yuan*

LSEC and NCMIS, Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 11 November 2015
Revised 8 April 2016
Accepted 28 April 2016
Available online 7 May 2016

Keywords:
Two-phase
Shallow granular flow
NOC
Well-balanced property
Multi-GPU

ABSTRACT

A two-phase shallow granular flow model consists of mass and momentum equations for the solid and fluid phases, coupled together by conservative and non-conservative momentum exchange terms. Development of classic Godunov methods based on Riemann problem solutions for such a model is difficult because of complexity in building appropriate wave structures. Non-oscillatory central (NOC) differencing schemes are attractive as they do not need to solve Riemann problems. In this paper, a staggered NOC scheme is amended for numerical solution of the two-phase shallow granular flow equations due to Pitman and Le. Simple discretization schemes for the non-conservative and bed slope terms and a simple correction procedure for the updating of the depth variables are proposed to ensure the well-balanced property. The scheme is further corrected with a numerical relaxation term mimicking the interphase drag force so as to overcome the difficulty associated with complex eigenvalues in some flow conditions. The resultant NOC scheme is implemented on multiple graphics processing units (GPUs) in a server by using both OpenMP-CUDA and multistream-CUDA parallelization strategies. Numerical tests in several typical two-phase shallow granular flow problems show that the NOC scheme can model wet/dry fronts and vacuum appearance robustly, and can treat some flow conditions associated with complex eigenvalues. Comparison of parallel efficiencies shows that the multistream-CUDA strategy can be slightly faster or slower than the OpenMP-CUDA strategy depending on the grid sizes.

1. Introduction

Landslides, rock avalanches, and debris flows are dangerous geological disasters that may cause great loss of properties and lives. Reliable prediction of geophysical mass movements is thus of fundamental importance for planning strategies for hazard risk mitigation. Nevertheless, the processes behind these phenomena are very complicated as they include continuous and discontinuous motion regimes and multiphase, polydisperse, multiscale, erosive and rheologically complicated materials. They have drawn great attention from numerous researchers in various fields. It is recognized that the basic ingredients in real geological disasters are granular materials composed of solid particles and interstice fluids, thus, study of the dynamics of granular materials can provide the scientific underpinnings for modeling diverse geological mass motions [1].

Savage and Hutter [2] first made their pioneering work to model dry granular avalanche flows by using depth-integrated Saint Venant like equations obeying Coulomb-type yield. Their model (now called the Savage–Hutter (SH) equations) was elaborated on and generalized by many researchers [1,3–6]. Fluids are found to play an important role in the mobility of granular avalanche flows. To take into account effects of interstice fluids in granular materials, Iverson and Denlinger [7,8] came up with a mixture model. Later, Pitman and Le [9] proposed the original two-phase (two-fluid) shallow flow model from depth averaging of the two-phase Navier–Stokes equations for the mixture of Coulomb materials and Newtonian fluids. The Pitman–Le model was subsequently reformulated and studied for its mathematical properties [10] and numerical solutions [11,12].

In this article we focus on efficient numerical solution for a popular variant of the original Pitman–Le two-phase shallow granular flow model. In this variant [10,13], the fluid momentum equations are recast into a semi-conservative form similar to the solid momentum equations. Compared with the single phase model, the major difficulties for the two-phase shallow granular flow model are the lack of explicit expressions for the eigenvalues of the quasilinear system, and hence the lack of knowledge of the exact or full-wave approximate Riemann solution, the presence of non-conservative terms, and the occurrence of complex roots in certain

* Corresponding author.
E-mail address: lyuan@Lsec.cc.ac.cn (L. Yuan).

http://dx.doi.org/10.1016/j.compfluid.2016.04.032
0045-7930/

flow conditions, all of which make some popular Riemann problem solution-based Godunov methods hard to apply.

Several researchers have developed numerical methods for the Pitman–Le two-phase flow model. Pelanti et al. [10,11] constructed a Roe approximate Riemann solver with a relaxation technique for simulating two-phase shallow granular flow over variable topography. While possessing high resolution and several other merits, this method is complicated. Dumbser et al. [14] developed a family of well-balanced path-conservative one-step ADER (Arbitrary DERivative in space and time) finite volume and discontinuous Galerkin finite element schemes for hyperbolic partial differential equations with non-conservative products and stiff source terms, but solution of an approximate Riemann problem is still a complicated thing. More recently, Ref. [15] developed a simpler HLLEM Riemann solver that works for general conservative and non-conservative systems of hyperbolic equations, and applied to the Pitman–Le two-phase flow model. Ref. [16] presented a class of first order finite volume solvers called PVM (polynomial viscosity matrix) and applied the solvers to 2D Pitman–Le model. Later on, a second order numerical scheme which was constructed by using a suitable decomposition of a Roe matrix by means of the PVM was presented in [17] and proved to be well-balanced with respect to the water-at-rest solution. Ref. [12] applied the space-time conservation element and solution element (CE/SE) method to simulating single and two-phase shallow granular flows with bottom topography, and shown higher resolution results compared with the kinetic flux-vector splitting method [18].

Non-Oscillatory Central Differencing (NOC) schemes [19] were first introduced for one-dimensional case and later elaborated by many others (e.g., [20,21]). The original NOC scheme [19] is time-space staggered. Ref. [22] translated the staggered scheme into non-staggered one for convenience of dealing with complex geometries and boundary conditions. NOC schemes belong to the class of Riemann-problem-solver-free Godunov methods. In spite of fewer applications in three dimensions due to complicated 3D space-time staggered grids and many quadrature points, NOC schemes are relatively simple and widely used for two-dimensional hyperbolic systems of conservation laws. The schemes do not need to solve Riemann problems, and this makes them attractive for complicated systems like the Pitman–Le two-phase model. Both staggered NOC [5,23] and non-staggered NOC schemes [24] have been applied to single-phase shallow granular flow equations. Extension to 1D two-layer shallow water equations was made in [25]. However, there seems to be a lack of application of NOC schemes to the Pitman–Le two-phase model up to now.

Over the last decade or so, graphics processing units (GPUs) have been increasingly used in high-performance computing. Compared to CPU, GPU has many more lightweight compute cores, and enables execution of many similar arithmetic operations. Although GPU programming is still somewhat cumbersome and time-consuming, the research community utilizing GPUs is continuously growing [26], and GPU hardware and programming environment have been steadily improved. Nowadays popular GPU programming languages are CUDA (Compute Unified Device Architecture) and OpenCL. CUDA is based on extension to C/C++ languages, and has well established tools like debuggers and profilers. There are a couple of papers on shallow water simulation using multiple GPUs in both single nodes [27] and clusters [28].

In this article, we extend the staggered NOC scheme developed for single phase SH model [5] to the Pitman–Le two-phase shallow granular flow model. In the extension, we have to specify a way the nonconservative terms and the bed slope source terms are discretized. This is very important for the numerical scheme to solve *exactly* the stationary solutions corresponding to water at rest, i.e., the C-property or well-balanced property (e.g., [29,30]). An obstacle to designing well-balanced staggered NOC scheme stems from the additional quadrature terms due to the staggered grid arrangement of the scheme. We have to design a special correction procedure to achieve the well-balanced property.

Implementation of the NOC scheme on multiple graphics processing units (GPUs) is another concern of this article. Here we focus on single node system with several GPUs connected through the PCI Express bus to the motherboard. There are several strategies to implement multi-GPU computing in a single node, and different strategies may have different efficiencies depending on the specific device and optimization technique as well as grid scale used. We compare two frequently used strategies for running multiple GPUs in a node, one is Open Multiprocessing interface (OpenMP) which spawns several CPU threads with each thread managing a GPU, another is CUDA's multistream multi-GPU capability. The comparison may be useful in gaining experience for choosing a better strategy for running multiple GPUs in a node.

The paper is organized as follows. In Section 2, we give formulation of the Pitman–Le two-phase model. In Section 3, we derive a well-balanced NOC scheme for this model and analyze its well-balanced and positive-preserving properties. In Section 4, the detail of implementing the NOC scheme with OpenMP and multistream strategies is given. In Section 5, some numerical examples, including one- and two-dimensional two-phase flows for a wide range of flow conditions, are presented, and concluding remarks are given in Section 6.

2. Two-phase shallow granular flow equations

2.1. One-dimensional equations

The Pitman–Le two-fluid shallow granular flow model we considered here is the variant [10,13] in which the fluid momentum equation is rewritten as a form similar to the solid momentum equation and the basal friction is neglected, which describes the dynamics of a shallow layer of mixture of solid granular material and fluid over a nearly horizontal surface. The solid and fluid components are assumed to be incompressible with densities ρ_s and ρ_f, where $\rho_f < \rho_s$. Let h denote the flow height and ϕ denote the solid volume fraction. The solid and fluid heights can be defined as follows,

$$h_s = \phi h, \quad h_f = (1-\phi)h. \tag{1}$$

In a Cartesian coordinate system with x being horizontal, the one-dimensional two-phase shallow flow model consisting of mass and momentum equations for the two constituents can be written as [11,12]

$$\begin{cases} \partial_t h_s + \partial_x(h_s u_s) = 0, \\ \partial_t (h_s u_s) + \partial_x\left(h_s u_s^2 + \frac{g}{2}h_s^2 + \frac{g}{2}(1-\gamma)h_s h_f\right) + \gamma g h_s \partial_x h_f \\ \quad = -g h_s \partial_x b + \gamma F^D, \\ \partial_t h_f + \partial_x(h_f u_f) = 0, \\ \partial_t h_f u_f + \partial_x\left(h_f u_f^2 + \frac{g}{2}h_f^2\right) + g h_f \partial_x h_s = -g h_f \partial_x b - F^D, \end{cases} \tag{2}$$

where u_s and u_f are the solid and fluid velocities in the x direction, respectively, $\gamma = \rho_f/\rho_s$ is the fluid/solid density ratio and g is the gravitational constant, $b := b(x)$ is the basal topography, and F^D is the inter-phase drag force which can be expressed as $F^D = D(h_s + h_f)(u_f - u_s)$ where D is the drag function. In this paper, the inter-phase drag force is neglected in designing the NOC scheme. However, since the inter-phase drag is important for maintaining flow conditions in the hyperbolic regime [11,31], it will be accounted for as a numerical remedy to make the NOC scheme behave normally when the loss of hyperbolicity of system (2) may happen (Section 3.5).

2.2. Quiescent steady states

The quiescent steady state for Newtonian fluid has an important feature that the free surface level is horizontal. This feature should be maintained by a good numerical scheme (so called well-balanced scheme in literature). For the two-fluid system (2) without Coulomb friction, the quiescent steady states are solution of lake at rest and constant ratio of volume fractions everywhere [10],

$$u_s = u_f = 0, \quad h_s + h_f + b = \text{Const}, \quad \phi \equiv \frac{h_s}{h_s + h_f} = \text{Const}. \tag{3}$$

2.3. Quasi-linear form and eigenstructure

Rewrite system (2) without inter-phase drag terms into compact form, it reads

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \mathbf{H}\left(\mathbf{U}, \frac{\partial \mathbf{U}}{\partial x}\right) = \mathbf{S}(\mathbf{U}), \tag{4}$$

where

$$\mathbf{U} = \begin{pmatrix} h_s \\ h_s u_s \\ h_f \\ h_f u_f \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} h_s u_s \\ h_s u_s^2 + \frac{g}{2}h_s^2 + \frac{g}{2}(1-\gamma)h_s h_f \\ h_f u_f \\ h_f u_f^2 + \frac{g}{2}h_f^2 \end{pmatrix},$$

$$\mathbf{H}\left(\mathbf{U}, \frac{\partial \mathbf{U}}{\partial x}\right) = \begin{pmatrix} 0 \\ \gamma g h_s \frac{\partial h_f}{\partial x} \\ 0 \\ g h_f \frac{\partial h_s}{\partial x} \end{pmatrix}, \quad \mathbf{S}(\mathbf{U}) = \begin{pmatrix} 0 \\ -g h_s \frac{\partial b}{\partial x} \\ 0 \\ -g h_f \frac{\partial b}{\partial x} \end{pmatrix}. \tag{5}$$

Here, $\mathbf{F}$ is the conservative flux vector, $\mathbf{H}$ is the non-conservative terms which have non-conservative products that couple the dynamics of the solid and fluid phases in the momentum equations, and $\mathbf{S}$ is the contribution from the varying basal topography.

Let us further rewrite the compact form (4) in quasi-linear form for later reference:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x} = \mathbf{S}(\mathbf{U}), \tag{6}$$

where matrix $\mathbf{A}$ is

$$\mathbf{A}(\mathbf{U}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -u_s^2 + g h_s + g\frac{1-\gamma}{2}h_f & 2u_s & g\frac{1+\gamma}{2}h_s & 0 \\ 0 & 0 & 0 & 1 \\ g h_f & 0 & -u_f^2 + g h_f & 2u_f \end{pmatrix}. \tag{7}$$

In general cases, there are no simple explicit expressions of the eigenvalues of the matrix $\mathbf{A}(\mathbf{U})$, but there is a theoretical result proved in [10] as follows.

Proposition 2.1. *Matrix $\mathbf{A}(\mathbf{U})$ has always at least two real eigenvalues $\lambda_{1,4}$, and moreover, the eigenvalues λ_k of $\mathbf{A}(\mathbf{U})$, satisfy the following inequalities:*

$$u_{\min} - a \le \lambda_1 \le \mathcal{R}(\lambda_2) \le \mathcal{R}(\lambda_3) \le \lambda_4 \le u_{\max} + a \tag{8}$$

where $a = \sqrt{gh} = \sqrt{g(h_s + h_f)}$, $u_{\min} = \min(u_f, u_s)$, $u_{\max} = \max(u_f, u_s)$, and $\mathcal{R}(\cdot)$ denotes the real part. Furthermore:

(i) *If $|u_s - u_f| \le 2a\beta$ or $|u_s - u_f| \ge 2a$ (where $\beta = \sqrt{\frac{1}{2}(1-\phi)(1-\gamma)} < 1$) then all the eigenvalues are real. If one of these inequalities are strictly satisfied, and the eigenvalues are also distinct, then system (6) is strictly hyperbolic.*

(ii) *If $2a\beta < |u_s - u_f| < 2a$ then the internal eigenvalues $\lambda_{2,3}$ may be complex.*

Pelanti et al. [10,11] used Newton's iteration method to compute all eigenvalues of matrix $\mathbf{A}(\mathbf{U})$ as needed by their Roe type solver. We remark that one can also calculate these eigenvalues according to Vieta's formula for polynomials. However, for the NOC scheme presented in this article, there is no need to compute the eigenvalues at all, and only the lower bound, $u_{\min} - a$, and the upper bound, $u_{\max} + a$, are used to calculate the time step based on a CFL condition.

2.4. Two-dimensional equations

Similarly, the two-dimensional two-phase shallow flow model without inter-phase drag force terms can be written as

$$\begin{cases} \partial_t h_s + \partial_x(h_s u_s) + \partial_y(h_s v_s) = 0, \\ \partial_t(h_s u_s) + \partial_x\left(h_s u_s^2 + \frac{g}{2}h_s^2 + \frac{g}{2}(1-\gamma)h_s h_f\right) + \partial_y(h_s u_s v_s) \\ \quad +\gamma g h_s \partial_x h_f = -g h_s \partial_x b, \\ \partial_t(h_s v_s) + \partial_x(h_s u_s v_s) + \partial_y(h_s v_s^2 + \frac{g}{2}h_s^2 + \frac{g}{2}(1-\gamma)h_s h_f) \\ \quad +\gamma g h_s \partial_y h_f = -g h_s \partial_y b, \\ \partial_t h_f + \partial_x(h_f u_f) + \partial_y(h_f v_f) = 0, \\ \partial_t h_f u_f + \partial_x\left(h_f u_f^2 + \frac{g}{2}h_f^2\right) \\ \quad +\partial_y(h_f u_f v_f) + g h_f \partial_x h_s = -g h_f \partial_x b, \\ \partial_t(h_f v_f) + \partial_x(h_f u_f v_f) + \partial_y\left(h_f v_f^2 + \frac{g}{2}h_f^2\right) + g h_f \partial_y h_s \\ \quad = -g h_f \partial_y b, \end{cases} \tag{9}$$

where v_s and v_f is the solid and fluid velocities in the y direction, respectively, and basal topography $b(x, y)$ is a function of both x and y coordinates. Write system (9) into compact form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} + \mathbf{H}\left(\mathbf{U}, \frac{\partial \mathbf{U}}{\partial x}, \frac{\partial \mathbf{U}}{\partial y}\right) = \mathbf{S}(\mathbf{U}), \tag{10}$$

where

$$\mathbf{U} = \begin{pmatrix} h_s \\ h_s u_s \\ h_s v_s \\ h_f \\ h_f u_f \\ h_f v_f \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} h_s u_s \\ h_s u_s^2 + \frac{g}{2}h_s^2 + \frac{g}{2}(1-\gamma)h_s h_f \\ h_s u_s v_s \\ h_f u_f \\ h_f u_f^2 + \frac{g}{2}h_f^2 \\ h_f u_f v_f \end{pmatrix},$$

$$\mathbf{G}(\mathbf{U}) = \begin{pmatrix} h_s v_s \\ h_s u_s v_s \\ h_s v_s^2 + \frac{g}{2}h_s^2 + \frac{g}{2}(1-\gamma)h_s h_f \\ h_f v_f \\ h_f u_f v_f \\ h_f v_f^2 + \frac{g}{2}h_f^2 \end{pmatrix},$$

$$\mathbf{H}\left(\mathbf{U}, \frac{\partial \mathbf{U}}{\partial x}, \frac{\partial \mathbf{U}}{\partial y}\right) = \begin{pmatrix} 0 \\ \gamma g h_s \partial_x h_f \\ \gamma g h_s \partial_y h_f \\ 0 \\ g h_f \partial_x h_s \\ g h_f \partial_y h_s \end{pmatrix}, \mathbf{S}(\mathbf{U}) = \begin{pmatrix} 0 \\ -g h_s \partial_x b \\ -g h_s \partial_y b \\ 0 \\ -g h_f \partial_x b \\ -g h_f \partial_y b \end{pmatrix}. \tag{11}$$

Rewrite system (10) in quasi-linear form

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}(\mathbf{U})\frac{\partial \mathbf{U}}{\partial y} = \mathbf{S}(\mathbf{U}), \tag{12}$$

where matrices $\mathbf{A}$ and $\mathbf{B}$ are

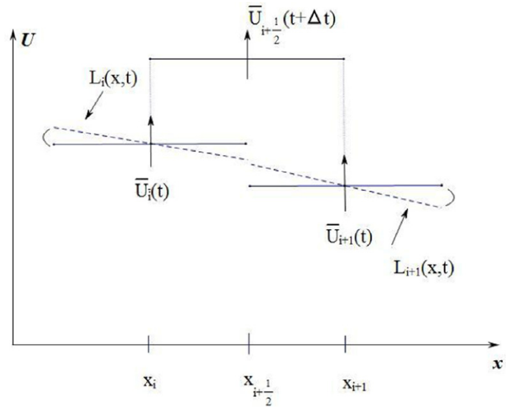


Fig. 1. Sketch of staggered grid for NOC scheme.

$$\mathbf{A}(\mathbf{U}) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -u_s^2 + gh_s + \frac{g}{2}(1-\gamma)h_f & 2u_s & 0 & \frac{g}{2}(1+\gamma)h_s & 0 & 0 \\ -u_s v_s & v_s & u_s & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ gh_f & 0 & 0 & -u_f^2 + gh_f & 2u_f & 0 \\ 0 & 0 & 0 & -u_f v_f & v_f & u_f \end{pmatrix},$$

$$\mathbf{B}(\mathbf{U}) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ -u_s v_s & v_s & u_s & 0 & 0 & 0 \\ -v_s^2 + gh_s + \frac{g}{2}(1-\gamma)h_f & 0 & 2v_s & \frac{g}{2}(1+\gamma)h_s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -u_f v_f & v_f & u_f \\ gh_f & 0 & 0 & -v_f^2 + gh_f & 0 & 2v_f \end{pmatrix}. \tag{13}$$

From (13), it is easy to verify that matrix **A** has two eigenvalues u_s and u_f (see 3rd row and 3rd column, and 6th row and 6th column), and the remaining eigenvalues are exactly the same as those in the 1d case. Similarly, matrix **B** has two eigenvalues v_s and v_f, and the remaining eigenvalues are analogous to those of matrix **A**. Hence, we can continue to make use of Proposition 2.1 for each matrix. Again, only the lower and upper bounds of the eigenvalues, $u_{\min} - a$, $u_{\max} + a$, $v_{\min} - a$, and $v_{\max} + a$, are utilized to calculate the time step in the present NOC scheme.

3. Non-oscillatory Central Differencing scheme (NOC)

Tai et al. [23,32] and Wang et al. [5] applied the staggered NOC scheme to numerical simulations of single-phase shallow granular flows. In this work, we apply the staggered NOC scheme to two-phase shallow granular flow equations. The well-balanced property of the resulting NOC scheme is ensured with a modification procedure, and the scheme is proven to be positivity-preserving. A numerical treatment [31] using the relaxation term to recover the hyperbolicity of the system is adopted to make the scheme work fine for flow conditions in which the loss of hyperbolicity happens.

3.1. One-dimensional NOC scheme

We first illustrate one-dimensional time-space staggered NOC scheme. As a Godunov method, the solution variables **U** are the cell averages on interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ at $t = t^n$ and on staggered interval $[x_i, x_{i+1}]$ at $t = t^{n+1}$ (Fig. 1). With MUSCL reconstruction, one can reconstruct the piecewise linear distribution as

$$\mathbf{L}_i(x, t^n) = \overline{\mathbf{U}}_i(t^n) + (x - x_i)\frac{\boldsymbol{\sigma}_i^n}{\Delta x}, \qquad x_{i-\frac{1}{2}} \le x \le x_{i+\frac{1}{2}}, \tag{14}$$

where cell average at time t^n is defined as $\overline{\mathbf{U}}_i(t^n) = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{U}(x, t^n)\mathrm{d}x$, and $\boldsymbol{\sigma}_i^n/\Delta x$ is the slope and $\boldsymbol{\sigma}_i^n$ could be some limiter for undivided differences of $\overline{\mathbf{U}}_i$. Integrate system (4) over control volume $[x_i, x_{i+1}] \times [t^n, t^{n+1}]$ as shown in Fig. 2,

$$\begin{aligned} &\int_{x_i}^{x_{i+1}} \mathbf{U}(x, t^{n+1})\mathrm{d}x - \int_{x_i}^{x_{i+1}} \mathbf{U}(x, t^n)\mathrm{d}x \\ &\quad + \int_{t^n}^{t^{n+1}} \mathbf{F}(x_{i+1}, t) - \int_{t^n}^{t^{n+1}} \mathbf{F}(x_i, t)\mathrm{d}t \\ &= -\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} \mathbf{H}\mathrm{d}x\mathrm{d}t + \int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} \mathbf{S}\mathrm{d}x\mathrm{d}t, \end{aligned} \tag{15}$$

Divide Eq. (15) by Δx, and define cell average at time t^{n+1} which is the unknown solution as

$$\overline{\mathbf{U}}_{i+\frac{1}{2}}^{n+1} = \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} \mathbf{U}(x, t^{n+1})\mathrm{d}x, \tag{16}$$

then

$$\begin{aligned} \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} \mathbf{U}(x, t^n)\mathrm{d}x &= \frac{1}{\Delta x}\left[\int_{x_i}^{x_{i+\frac{1}{2}}} \mathbf{U}(x, t^n)\mathrm{d}x + \int_{x_{i+\frac{1}{2}}}^{x_{i+1}} \mathbf{U}(x, t^n)\mathrm{d}x\right] \\ &= \frac{1}{\Delta x}\left[\int_{x_i}^{x_{i+\frac{1}{2}}} \mathbf{L}_i(x, t^n)\mathrm{d}x + \int_{x_{i+\frac{1}{2}}}^{x_{i+1}} \mathbf{L}_{i+1}(x, t^n)\mathrm{d}x\right] \\ &= \frac{1}{2}\left(\overline{\mathbf{U}}_i^n + \overline{\mathbf{U}}_{i+1}^n\right) + \frac{1}{8}\left(\boldsymbol{\sigma}_i^n - \boldsymbol{\sigma}_{i+1}^n\right). \end{aligned} \tag{17}$$

The time-integration of flux **F** can be approximated by midpoint rule of integral (∘ points in the right frame of Fig. 2),

$$\begin{aligned} &\frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \mathbf{F}(x_{i+1}, t)\mathrm{d}t \approx \frac{\Delta t}{\Delta x}\mathbf{F}\left(\mathbf{U}_{i+1}^{n+\frac{1}{2}}\right), \\ &\frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \mathbf{F}(x_i, t)\mathrm{d}t \approx \frac{\Delta t}{\Delta x}\mathbf{F}\left(\mathbf{U}_i^{n+\frac{1}{2}}\right). \end{aligned} \tag{18}$$

To proceed, we need to evaluate the integration of non-conservative term $\mathbf{H}(\mathbf{U}, \mathbf{U}_x)$ in Eq. (15). Because the second and fourth components of vector **H** are similar as seen from Eq. (5), we take the fourth component as example. For clarity, we omit constant g. If it is approximated with

$$\begin{aligned} \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} h_f\frac{\partial h_s}{\partial x}\mathrm{d}x\mathrm{d}t &\approx \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \left(\widetilde{h}_f\widetilde{\delta h}_s\right)_{i+\frac{1}{2}}\mathrm{d}t \\ &\approx \frac{\Delta t}{\Delta x}\left(\widetilde{h}_f\widetilde{\delta h}_s\right)_{i+\frac{1}{2}}^{n+\frac{1}{2}} \\ &= \frac{\Delta t}{\Delta x}\mathcal{H}_{i+\frac{1}{2}}^{n+\frac{1}{2}}, \end{aligned} \tag{19}$$

where $(\widetilde{h}_f)_{i+1/2}^{n+1/2}$ and $(\widetilde{\delta h}_s)_{i+1/2}^{n+1/2}$ are suitable approximations to $(h_f)_{i+1/2}^{n+1/2}$ and $(\delta h_s)_{i+1/2}^{n+1/2}$, and if the last bed slope term in Eq. (15) is also discretized properly, then (15) becomes NOC scheme

$$\begin{aligned} \overline{\mathbf{U}}_{i+\frac{1}{2}}^{n+1} &= \frac{1}{2}\left(\overline{\mathbf{U}}_i^n + \overline{\mathbf{U}}_{i+1}^n\right) + \frac{1}{8}\left(\boldsymbol{\sigma}_i^n - \boldsymbol{\sigma}_{i+1}^n\right) \\ &\quad - \frac{\Delta t}{\Delta x}\left[\mathbf{F}\left(\mathbf{U}_{i+1}^{n+\frac{1}{2}}\right) - \mathbf{F}\left(\mathbf{U}_i^{n+\frac{1}{2}}\right)\right] \\ &\quad - \frac{\Delta t}{\Delta x}\mathcal{H}_{i+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta t}{\Delta x}\mathcal{S}_{i+\frac{1}{2}}^{n+\frac{1}{2}}. \end{aligned} \tag{20}$$

The RHS of (20) consists of *four* parts: the quadrature terms from the reconstruction (first two terms), the flux terms (third term), the nonconservative terms (fourth term), and the bed slope source terms (fifth term). As will be shown later, suitable second-order discretization schemes for the nonconservative and bed slope

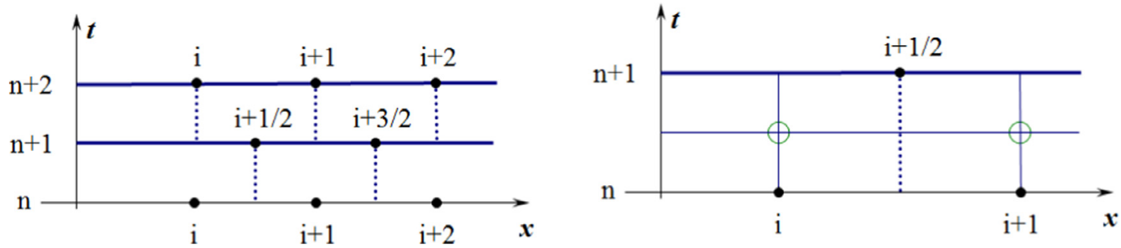


Fig. 2. Diagram of a NOC scheme in the $x-t$ plane. The left frame shows grid points in the NOC scheme. The right frame shows the NOC integration stencil, where • indicate the grid points at time level n and $n+1$, ∘ represent the quadrature points for fluxes **F** across the cell boundaries and the data points for discretizing non-conservative terms **H** and bed slope source terms **S**.

terms for ensuring the well-balanced property are

$$\mathcal{H}^{n+\frac{1}{2}}_{i+\frac{1}{2}} = \begin{cases} 0 \\ \frac{\gamma g}{2}\left(h^{n+\frac{1}{2}}_{s,i}+h^{n+\frac{1}{2}}_{s,i+1}\right)\left(h^{n+\frac{1}{2}}_{f,i+1}-h^{n+\frac{1}{2}}_{f,i}\right) \\ 0 \\ \frac{g}{2}\left(h^{n+\frac{1}{2}}_{f,i}+h^{n+\frac{1}{2}}_{f,i+1}\right)\left(h^{n+\frac{1}{2}}_{s,i+1}-h^{n+\frac{1}{2}}_{s,i}\right) \end{cases},$$

$$\mathcal{S}^{n+\frac{1}{2}}_{i+\frac{1}{2}} = \begin{cases} 0 \\ -g\dfrac{h^{n+\frac{1}{2}}_{s,i+1}+h^{n+\frac{1}{2}}_{s,i}}{2}(b_{i+1}-b_i) \\ 0 \\ -g\dfrac{h^{n+\frac{1}{2}}_{f,i+1}+h^{n+\frac{1}{2}}_{f,i}}{2}(b_{i+1}-b_i) \end{cases}. \tag{21}$$

By letting cell average $\overline{\mathbf{U}}^n_i$ coincide with point $(i,\ n)$ (second-order approximation in space), the required middle time values of $\mathbf{U}^{n+\frac{1}{2}}_i$ and $\mathbf{U}^{n+\frac{1}{2}}_{i+1}$ are obtained by the first-order Taylor expansions,

$$\mathbf{U}^{n+\frac{1}{2}}_{i+q} = \overline{\mathbf{U}}^n_{i+q} + \frac{\Delta t}{2}\left(\frac{\partial \mathbf{U}}{\partial t}\right)^n_{i+q}, \quad q = 0, 1, \tag{22}$$

where $\left(\frac{\partial \mathbf{U}}{\partial t}\right)^n_i$ are computed according to discretization of quasi-linear system (6) at $(i,\ n)$,

$$\left(\frac{\partial \mathbf{U}}{\partial t}\right)^n_i = \mathbf{S}(\overline{\mathbf{U}}^n_i) - \mathbf{A}(\overline{\mathbf{U}}^n_i)\frac{\boldsymbol{\sigma}^n_i}{\Delta x}. \tag{23}$$

The limiter we used for the undivided slope $\boldsymbol{\sigma}$ in (23) is

$$\boldsymbol{\sigma}_i = \bar{\delta}\mathbf{U}_i = \text{minmod}(\overline{\mathbf{U}}_i - \overline{\mathbf{U}}_{i-1},\ \overline{\mathbf{U}}_{i+1} - \overline{\mathbf{U}}_i). \tag{24}$$

It is remarked that the bed slope term $\mathbf{S}(\overline{\mathbf{U}}_i)$ in (23) should be approximated with a finite difference to be given in Eq. (34) in Section 3.3 so as to ensure the well-balanced property. It is different from $\mathcal{S}^{n+1/2}_{i+1/2}$ in Eq. (21).

3.2. Two-dimensional NOC scheme

The 2D NOC scheme is similar to 1D NOC scheme (20) but is more involved due to the 2D staggered grids (see Fig. 3). We follow [5] to give the 2D staggered NOC scheme for system (10). The final 2D NOC scheme also consists of four parts and is written as

$$\begin{aligned} \overline{\mathbf{U}}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = & \frac{1}{4}\left(\overline{\mathbf{U}}^n_{i,j}+\overline{\mathbf{U}}^n_{i+1,j}+\overline{\mathbf{U}}^n_{i,j+1}+\overline{\mathbf{U}}^n_{i+1,j+1}\right) \\ & + \frac{1}{16}\left(\boldsymbol{\sigma}^{x,n}_{i,j}-\boldsymbol{\sigma}^{x,n}_{i+1,j}+\boldsymbol{\sigma}^{x,n}_{i,j+1}-\boldsymbol{\sigma}^{x,n}_{i+1,j+1}\right) \\ & + \frac{1}{16}\left(\boldsymbol{\sigma}^{y,n}_{i,j}-\boldsymbol{\sigma}^{y,n}_{i,j+1}+\boldsymbol{\sigma}^{y,n}_{i+1,j}-\boldsymbol{\sigma}^{y,n}_{i+1,j+1}\right) \\ & - \frac{\Delta t}{2\Delta x}\left[\mathbf{F}\left(\mathbf{U}^{n+\frac{1}{2}}_{i+1,j}\right)-\mathbf{F}\left(\mathbf{U}^{n+\frac{1}{2}}_{i,j}\right)\right. \\ & \left. + \mathbf{F}\left(\mathbf{U}^{n+\frac{1}{2}}_{i+1,j+1}\right)-\mathbf{F}\left(\mathbf{U}^{n+\frac{1}{2}}_{i,j+1}\right)\right] \\ & - \frac{\Delta t}{2\Delta y}\left[\mathbf{G}\left(\mathbf{U}^{n+\frac{1}{2}}_{i,j+1}\right)-\mathbf{G}\left(\mathbf{U}^{n+\frac{1}{2}}_{i,j}\right)\right. \\ & \left. + \mathbf{G}\left(\mathbf{U}^{n+\frac{1}{2}}_{i+1,j+1}\right)-\mathbf{G}\left(\mathbf{U}^{n+\frac{1}{2}}_{i+1,j}\right)\right] \\ & - \Delta t\mathcal{H}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta t\mathcal{S}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}. \end{aligned} \tag{25}$$

where $\boldsymbol{\sigma}^{x,n}_{i,j}$ and $\boldsymbol{\sigma}^{y,n}_{i,j}$ represent limited undivided "slopes" in the x and y directions, respectively. Suitable second-order discretizations for the nonconservative terms and the bed slope source terms are similar to previous 1d case, i.e., the same quadrature points ⊕ as for fluxes at the middle time level $n+\frac{1}{2}$ are used:

$$\mathcal{H}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} = \begin{cases} 0, \\ \frac{\gamma g}{4\Delta x}\left[\left(h^{n+\frac{1}{2}}_{s,i,j}+h^{n+\frac{1}{2}}_{s,i+1,j}\right)\left(h^{n+\frac{1}{2}}_{f,i+1,j}-h^{n+\frac{1}{2}}_{f,i,j}\right)\right. \\ \left.+\left(h^{n+\frac{1}{2}}_{s,i,j+1}+h^{n+\frac{1}{2}}_{s,i+1,j+1}\right)\left(h^{n+\frac{1}{2}}_{f,i+1,j+1}-h^{n+\frac{1}{2}}_{f,i,j+1}\right)\right], \\ \frac{\gamma g}{4\Delta y}\left[\left(h^{n+\frac{1}{2}}_{s,i,j}+h^{n+\frac{1}{2}}_{s,i,j+1}\right)\left(h^{n+\frac{1}{2}}_{f,i,j+1}-h^{n+\frac{1}{2}}_{f,i,j}\right)\right. \\ \left.+\left(h^{n+\frac{1}{2}}_{s,i+1,j}+h^{n+\frac{1}{2}}_{s,i+1,j+1}\right)\left(h^{n+\frac{1}{2}}_{f,i+1,j+1}-h^{n+\frac{1}{2}}_{f,i+1,j}\right)\right], \\ 0, \\ \frac{g}{4\Delta x}\left[\left(h^{n+\frac{1}{2}}_{f,i,j}+h^{n+\frac{1}{2}}_{f,i+1,j}\right)\left(h^{n+\frac{1}{2}}_{s,i+1,j}-h^{n+\frac{1}{2}}_{s,i,j}\right)\right. \\ \left.+\left(h^{n+\frac{1}{2}}_{f,i,j+1}+h^{n+\frac{1}{2}}_{f,i+1,j+1}\right)\left(h^{n+\frac{1}{2}}_{s,i+1,j+1}-h^{n+\frac{1}{2}}_{s,i,j+1}\right)\right], \\ \frac{g}{4\Delta y}\left[\left(h^{n+\frac{1}{2}}_{f,i,j}+h^{n+\frac{1}{2}}_{f,i,j+1}\right)\left(h^{n+\frac{1}{2}}_{s,i,j+1}-h^{n+\frac{1}{2}}_{s,i,j}\right)\right. \\ \left.+\left(h^{n+\frac{1}{2}}_{f,i+1,j}+h^{n+\frac{1}{2}}_{f,i+1,j+1}\right)\left(h^{n+\frac{1}{2}}_{s,i+1,j+1}-h^{n+\frac{1}{2}}_{s,i+1,j}\right)\right], \end{cases} \tag{26}$$

$$\mathcal{S}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} = \begin{cases} 0, \\ -\frac{g}{4\Delta x}\left[\left(h^{n+\frac{1}{2}}_{s,i+1,j+1}+h^{n+\frac{1}{2}}_{s,i,j+1}\right)(b_{i+1,j+1}-b_{i,j+1})\right. \\ \left.\quad+\left(h^{n+\frac{1}{2}}_{s,i+1,j}+h^{n+\frac{1}{2}}_{s,i,j}\right)(b_{i+1,j}-b_{i,j})\right], \\ -\frac{g}{4\Delta y}\left[\left(h^{n+\frac{1}{2}}_{s,i+1,j+1}+h^{n+\frac{1}{2}}_{s,i+1,j}\right)(b_{i+1,j+1}-b_{i+1,j})\right. \\ \left.\quad+\left(h^{n+\frac{1}{2}}_{s,i,j+1}+h^{n+\frac{1}{2}}_{s,i,j}\right)(b_{i,j+1}-b_{i,j})\right], \\ 0, \\ -\frac{g}{4\Delta x}\left[\left(h^{n+\frac{1}{2}}_{f,i+1,j+1}+h^{n+\frac{1}{2}}_{f,i,j+1}\right)(b_{i+1,j+1}-b_{i,j+1})\right. \\ \left.\quad+\left(h^{n+\frac{1}{2}}_{f,i+1,j}+h^{n+\frac{1}{2}}_{f,i,j}\right)(b_{i+1,j}-b_{i,j})\right], \\ -\frac{g}{4\Delta y}\left[\left(h^{n+\frac{1}{2}}_{f,i+1,j+1}+h^{n+\frac{1}{2}}_{f,i+1,j}\right)(b_{i+1,j+1}-b_{i+1,j})\right. \\ \left.\quad+\left(h^{n+\frac{1}{2}}_{f,i,j+1}+h^{n+\frac{1}{2}}_{f,i,j}\right)(b_{i,j+1}-b_{i,j})\right], \end{cases} \tag{27}$$

The required middle time values of $\mathbf{U}^{n+\frac{1}{2}}_{i,j}$, $\mathbf{U}^{n+\frac{1}{2}}_{i+1,j}$, $\mathbf{U}^{n+\frac{1}{2}}_{i,j+1}$, $\mathbf{U}^{n+\frac{1}{2}}_{i+1,j+1}$ for evaluating fluxes, nonconservative terms, and bed slope terms, are calculated according to the first-order Taylor expansions with

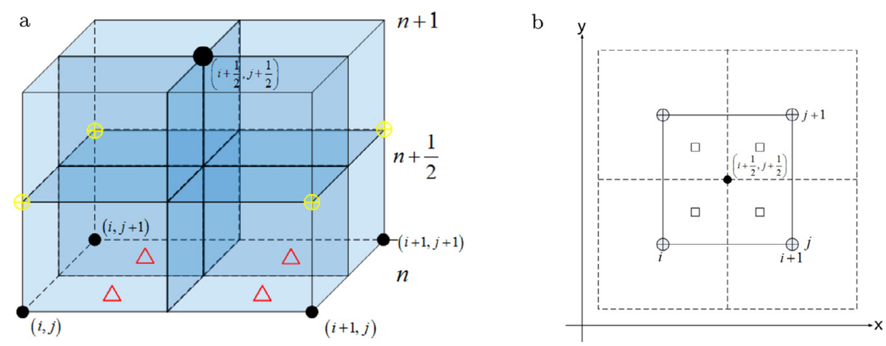


Fig. 3. Diagram of a 2D NOC scheme in the $(x, y) - t$ space. (a) Perspective view, where • indicate the computational grid points at time levels n and $n+1$, red triangles △ are quadrature points for the piecewise linear reconstruction at time level n, and ⊕ are quadrature points for fluxes **F** and **G** across cell boundaries of the control volume, and data points for discretizing nonconservative terms **H** and source terms **S**. (b) Top view of (a) onto the (x, y) plane, where • is the projected grid point for $(i+\frac{1}{2}, j+\frac{1}{2})$ at time $n+1$, □ are the quadrature points for the piecewise linear reconstruction at time level n, and ⊕ are quadrature points for the flux terms and data points for discretizing the nonconservative terms and source terms, which coincide with the computational points at time n. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

respect to piece-wise integer grid points at time level n,

$$\mathbf{U}_{i+q,j+r}^{n+\frac{1}{2}} = \overline{\mathbf{U}}_{i+q,j+r}^{n} + \frac{\Delta t}{2}\left(\frac{\partial \mathbf{U}}{\partial t}\right)_{i+q,j+r}^{n}, \qquad q, r = 0, 1. \tag{28}$$

The time partial derivative $\partial_t \mathbf{U}$ is similarly determined from 2D quasi-linear system (12) as

$$\left(\frac{\partial \mathbf{U}}{\partial t}\right)_{i,j} = \mathbf{S}(\overline{\mathbf{U}}_{i,j}) - \mathbf{A}(\overline{\mathbf{U}}_{i,j})\frac{\boldsymbol{\sigma}_{i,j}^{x}}{\Delta x} - \mathbf{B}(\overline{\mathbf{U}}_{i,j})\frac{\boldsymbol{\sigma}_{i,j}^{y}}{\Delta y}. \tag{29}$$

Again, the bed slopes in $\mathbf{S}(\overline{\mathbf{U}}_{i,j})$ are computed with a special finite difference to be given in Eq. (34) in Section 3.3 in order to ensure the well-balanced property. The limited undivided slopes $\boldsymbol{\sigma}_{i,j}^{x}$ and $\boldsymbol{\sigma}_{i,j}^{y}$ are computed in dimension-by-dimension way. Computational effects using different limiter functions have been compared in [5]. Based on their study, the *minmod* limiter is used in this work. The Courant–Friedrichs–Levy (CFL) condition is

$$\Delta t \ \max\left(\frac{|c_x^{\max}|}{\Delta x}, \frac{|c_y^{\max}|}{\Delta y}\right) \le \frac{1}{2}, \tag{30}$$

where $c_x^{\max}$, $c_y^{\max}$ are the maximum wave speeds in the x- and y-directions, respectively, which are estimated from Pelanti's preposition 2.1. The NOC scheme is formally second-order accurate.

3.3. Modify the NOC scheme for well-balancedness

Compared with conventional shallow water equations, the two-phase shallow water model poses additional difficulties to developing well-balanced scheme because nonconservative terms need to be considered and the volume fraction ϕ needs to preserve a constant for quiescent steady states. Further, the staggered-grid NOC scheme contains quadrature terms of the linear reconstructions, e.g., the first two terms in (20), and they also have to be considered. In this work, we show analysis of 1D NOC scheme and the well-balanced modification in detail. Extension to 2D NOC scheme is straightforward.

Consider 1D NOC scheme (20). The well-balanced property states that if the numerical solution $\mathbf{U}_i^n$ satisfies quiescent steady states (3) in the sense that $h_{s,i}^n + h_{f,i}^n + b_i = \text{Const}$, $h_{s,i}^n/(h_{s,i}^n + h_{f,i}^n) = \text{Const}$, and $\mathbf{u}_i^n = 0, \forall i$, then the numerical solution $\mathbf{U}_{i+\frac{1}{2}}^{n+1}$ preserves the same quiescent steady states, i.e., $h_{s,i+\frac{1}{2}}^{n+1} + h_{f,i+\frac{1}{2}}^{n+1} + b_{i+\frac{1}{2}} = \text{Const}$, $h_{s,i+\frac{1}{2}}^{n+1}/(h_{s,i+\frac{1}{2}}^{n+1} + h_{f,i+\frac{1}{2}}^{n+1}) = \text{Const}$, and $\mathbf{u}_{i+\frac{1}{2}}^{n+1} = 0, \forall i$.

For scheme (20) with corresponding discretizations (21)-(24), we can prove that the conservative flux terms, the nonconservative product terms, and the bed slope source terms cancel *exactly* under quiescent steady state conditions, leading to the following Proposition 3.1. But the first two terms in scheme (20) do not necessarily satisfy the well-balanced property. A special correction procedure to ensure the well-balancedness will be introduced later. We first prove Proposition 3.1.

Proposition 3.1. *For scheme (20), the discrete flux gradients, nonconservative products, and bed slope source terms cancel exactly at quiescent steady states.*

Proof. For the first component in Eq. (20), the sum of the flux gradient, nonconservative product, and bed slope source terms is

$$(\Delta h_s)_{sum} = -\frac{\Delta t}{\Delta x}\left[(h_s u_s)_{i+1}^{n+\frac{1}{2}} - (h_s u_s)_{i}^{n+\frac{1}{2}}\right]. \tag{31}$$

The numerical flux $(h_s u_s)_i^{n+\frac{1}{2}}$ is obtained from the 1st-order Taylor expansion (22) with respect to the cell center value $(h_s u_s)_i^n$ together with (23). Hence,

$$\begin{aligned}(h_s u_s)_i^{n+\frac{1}{2}} &= (h_s u_s)_i^n + \frac{\Delta t}{2}\left(\frac{\partial (h_s u_s)}{\partial t}\right)_i^n \\ &= (h_s u_s)_i^n + \frac{\Delta t}{2}\left[\mathbf{S}_i^{n,(2)} - \mathbf{A}_i^{n,(2)}\frac{\boldsymbol{\sigma}_i^n}{\Delta x}\right].\end{aligned} \tag{32}$$

where superscript 2 denotes the second row of a vector or matrix. Inserting $\mathbf{S}_i^{n,(2)}$ as in (5) and $\mathbf{A}_i^{n,(2)}$ as in (7) evaluated at quiescent steady states into (32), and replacing $\boldsymbol{\sigma}_i^n$ by limited finite difference $\bar{\delta}\mathbf{U}_i^n$ as in (24), Eq. (32) becomes

$$\begin{aligned}(h_s u_s)_i^{n+\frac{1}{2}} &= \frac{\Delta t}{2}\left\{-g h_{s,i}^n\left[\left(\frac{\delta b}{\delta x}\right)_i + \left(\frac{\bar{\delta} h_s}{\Delta x}\right)_i^n + \left(\frac{\bar{\delta} h_f}{\Delta x}\right)_i^n\right]\right. \\ &\quad \left. - g\frac{1-\gamma}{2}\left[\left(h_f\frac{\bar{\delta} h_s}{\Delta x}\right)_i^n - \left(h_s\frac{\bar{\delta} h_f}{\Delta x}\right)_i^n\right]\right\},\end{aligned} \tag{33}$$

Define $\eta_i^n := b_i + h_{s,i}^n + h_{f,i}^n$. If $(\delta b/\delta x)_i$ in (33) is approximated with

$$\left(\frac{\delta b}{\delta x}\right)_i = \left(\frac{\bar{\delta}\eta}{\Delta x} - \frac{\bar{\delta} h_s}{\Delta x} - \frac{\bar{\delta} h_f}{\Delta x}\right)_i^n, \tag{34}$$

then the first term in (33) becomes zero at quiescent steady states because $\eta_i = \text{Const}, \forall i$. Note that quiescent steady states of two-phase shallow flows also imply $\left(h_s/h_f\right)_i = \text{Const}, \forall i$, thus $\left(h_f\bar{\delta} h_s/\Delta x\right)_i^n - \left(h_s\bar{\delta} h_f/\Delta x\right)_i^n = 0$ is easily satisfied. Therefore, numerical flux (33) is zero at quiescent steady states. It immediately follows that the result of (31) is zero.

Consequently, the conclusion for the first component of scheme (20) is proved. In a similar way, it is easy to prove the third component of (20).

Before we prove the more involved second and fourth components of (20), we note that for quiescent steady states, Taylor expansion (22) together with (23) gives

$$\begin{aligned}(h_s)_i^{n+\frac{1}{2}} &= (h_s)_i^n + \frac{1}{2}\Delta t\left(\frac{\partial h_s}{\partial t}\right)_i^n \\ &= (h_s)_i^n + \frac{1}{2}\Delta t\left(-\frac{\bar{\delta}(hu_s)}{\Delta x}\right)_i^n \\ &= (h_s)_i^n,\end{aligned} \tag{35}$$

and similarly, $(h_f)_i^{n+\frac{1}{2}} = (h_f)_i^n$. As the fourth component is simpler than the second component, we start from the fourth component. Using quiescent steady states and discretization (21), the sum of the discrete flux gradient, nonconservative product, and bed slope source terms for the fourth component is

$$\begin{aligned}&\Delta(h_f u_f)_{sum} \\ &= -\Delta t\left\{\frac{\left(\frac{g}{2}h_f^2\right)_{i+1}^{n+\frac{1}{2}} - \left(\frac{g}{2}h_f^2\right)_i^{n+\frac{1}{2}}}{\Delta x} + g\left(h_{f,i+1}^{n+\frac{1}{2}} + h_{f,i}^{n+\frac{1}{2}}\right)\frac{h_{s,i+1}^{n+\frac{1}{2}} - h_{s,i}^{n+\frac{1}{2}}}{2\Delta x}\right. \\ &\quad \left. + g\left(h_{f,i+1}^{n+\frac{1}{2}} + h_{f,i}^{n+\frac{1}{2}}\right)\frac{b_{s,i+1}^{n+\frac{1}{2}} - b_{s,i}^{n+\frac{1}{2}}}{2\Delta x}\right\}.\end{aligned} \tag{36}$$

Notice that since $h_{s,f}^{n+\frac{1}{2}} = h_{s,f}^n$, we omit sup-script $n+\frac{1}{2}$ and extract the factor $g/2\Delta x$ out of the big bracket. Then the remaining part becomes

$$\begin{aligned}&\left(h_{f,i+1}^2 - h_{f,i}^2\right) + (h_{f,i+1} + h_{f,i})(h_{s,i+1} - h_{s,i}) \\ &\quad + (h_{f,i+1} + h_{f,i})(b_{i+1} - b_i) \\ &= \left(h_{f,i+1}^2 - h_{f,i}^2\right) + (h_{f,i+1} + h_{f,i})(h_{s,i+1} - h_{s,i} + b_{i+1} - b_i) \\ &= \left(h_{f,i+1}^2 - h_{f,i}^2\right) + (h_{f,i+1} + h_{f,i})(\text{const} - h_{f,i+1} - \text{const} + h_{f,i}) \\ &= 0.\end{aligned} \tag{37}$$

Similarly, for the second component in (20), the sum can be written as

$$\begin{aligned}&\frac{\Delta(h_s u_s)_{sum}}{\Delta t} \\ &= -\frac{\left(\frac{g}{2}h_s^2 + \frac{g}{2}(1-\gamma)h_s h_f\right)_{i+1}^{n+\frac{1}{2}} - \left(\frac{g}{2}h_s^2 + \frac{g}{2}(1-\gamma)h_s h_f\right)_i^{n+\frac{1}{2}}}{\Delta x} \\ &\quad -\gamma g(h_{s,i+1}^{n+\frac{1}{2}} + h_{s,i}^{n+\frac{1}{2}})\frac{h_{f,i+1}^{n+\frac{1}{2}} - h_{f,i}^{n+\frac{1}{2}}}{2\Delta x} - g(h_{s,i+1}^{n+\frac{1}{2}} + h_{s,i}^{n+\frac{1}{2}})\frac{b_{i+1} - b_i}{2\Delta x}\end{aligned} \tag{38}$$

Again, omit superscript $n+\frac{1}{2}$ and extract the factor $-g/2\Delta x$, the RHS of (38) reduces to

$$\begin{aligned}&\left(h_{s,i+1}^2 - h_{s,i}^2\right) + (1-\gamma)\left(h_{s,i+1}h_{f,i+1} - h_{s,i}h_{f,i}\right) \\ &\quad + \gamma(h_{s,i+1} + h_{s,i})\left(h_{f,i+1} - h_{f,i}\right) + (h_{s,i} + h_{s,i+1})(b_{i+1} - b_i) \\ &= \left(h_{s,i+1}^2 - h_{s,i}^2\right) + (1-\gamma)\left(h_{s,i+1}h_{f,i+1} - h_{s,i}h_{f,i}\right) \\ &\quad + (\gamma - 1)(h_{s,i+1} + h_{s,i})\left(h_{f,i+1} - h_{f,i}\right) \\ &\quad + (h_{s,i} + h_{s,i+1})\left(h_{f,i+1} - h_{f,i} + b_{i+1} - b_i\right) \\ &= \left(h_{s,i+1}^2 - h_{s,i}^2\right) + (1-\gamma)\left(h_{s,i+1}h_{f,i} - h_{s,i}h_{f,i+1}\right) \\ &\quad + (h_{s,i} + h_{s,i+1})(\text{const} - h_{s,i+1} - \text{const} + h_{s,i}) \\ &= (1-\gamma)\left(h_{s,i+1}h_{f,i} - h_{s,i}h_{f,i+1}\right) \\ &= 0,\end{aligned} \tag{39}$$

where the last equality results from the condition $(h_s/h_f)_i = \text{const}, \forall i$.

In summary, the sum of discrete flux terms, nonconservative terms and topography source terms in scheme (20) is zero at quiescent steady states. This ends the proof. □

Next, we consider how to make the results from the first two terms in scheme (20), i.e., the quadrature terms for the linear reconstructions, satisfy the well-balanced property. The second and fourth components of the two terms at quiescent steady states give the solid and fluid momentums,

$$\begin{aligned}(h_s u_s)_{i+\frac{1}{2}}^{n+1} &= \frac{1}{2}\left[(h_s u_s)_i^n + (h_s u_s)_{i+1}^n\right] + \frac{1}{8}\left(\boldsymbol{\sigma}_i^{n,(2)} - \boldsymbol{\sigma}_{i+1}^{n,(2)}\right), \\ (h_f u_f)_{i+\frac{1}{2}}^{n+1} &= \frac{1}{2}\left[(h_f u_f)_i^n + (h_f u_f)_{i+1}^n\right] + \frac{1}{8}\left(\boldsymbol{\sigma}_i^{n,(4)} - \boldsymbol{\sigma}_{i+1}^{n,(4)}\right).\end{aligned} \tag{40}$$

since $\mathbf{u}_i^n = 0, \forall i$, all terms in the RHS of (40) are equal to zero, giving $h\mathbf{u}_{i+\frac{1}{2}}^{n+1} = 0$ so that the first item of well-balanced condition (3) is satisfied. But the first and third components of the first two terms in (20) need special treatment. The first and third components of the two terms at quiescent steady states give the solid and fluid heights,

$$\begin{aligned}h_{s,i+\frac{1}{2}}^{n+1} &= \frac{1}{2}\left(h_{s,i}^n + h_{s,i+1}^n\right) + \frac{1}{8}\left(\boldsymbol{\sigma}_i^{n,(1)} - \boldsymbol{\sigma}_{i+1}^{n,(1)}\right), \\ h_{f,i+\frac{1}{2}}^{n+1} &= \frac{1}{2}\left(h_{f,i}^n + h_{f,i+1}^n\right) + \frac{1}{8}\left(\boldsymbol{\sigma}_i^{n,(3)} - \boldsymbol{\sigma}_{i+1}^{n,(3)}\right).\end{aligned} \tag{41}$$

With $\left(h_s/h_f\right)_i^n = \text{Const}$, it is easy to show the results from (41) satisfy $\left(h_s/h_f\right)_{i+\frac{1}{2}}^{n+1} = \text{Const}$, thus the last item of well-balanced condition (3) is satisfied. However, the new free surface level at a staggered gird point as calculated from (41) is

$$\begin{aligned}&h_{s,i+\frac{1}{2}}^{n+1} + h_{f,i+\frac{1}{2}}^{n+1} + b_{i+\frac{1}{2}} \\ &= \frac{1}{2}\left[\left(h_{s,i}^n + h_{f,i}^n + b_i\right) + \left(h_{s,i+1}^n + h_{f,i+1}^n + b_{i+1}\right)\right] \\ &\quad + \frac{1}{8}\left(\boldsymbol{\sigma}_i^{n,(1)} + \boldsymbol{\sigma}_i^{n,(3)} - \boldsymbol{\sigma}_{i+1}^{n,(1)} - \boldsymbol{\sigma}_{i+1}^{n,(3)}\right) + b_{i+\frac{1}{2}} - \frac{1}{2}(b_i + b_{i+1}) \\ &= \text{Const} + \frac{1}{8}\left(\boldsymbol{\sigma}_i^{n,(1)} + \boldsymbol{\sigma}_i^{n,(3)} - \boldsymbol{\sigma}_{i+1}^{n,(1)} - \boldsymbol{\sigma}_{i+1}^{n,(3)}\right) \\ &\quad - \frac{1}{2}(b_i + b_{i+1}) + b_{i+\frac{1}{2}}.\end{aligned} \tag{42}$$

Since $h_{s,i}^n$ and $h_{f,i}^n$ may vary from point to point, the second term in the second equality of (42) are not zero in general, and this may cause $h_{s,i+\frac{1}{2}}^{n+1} + h_{f,i+\frac{1}{2}}^{n+1} + b_{i+\frac{1}{2}} \neq \text{Const}$ if b_i, b_{i+1} and $b_{i+\frac{1}{2}}$ are predefined irrespective of the flow solution. We give a modification procedure to cure this problem.

Modification procedure for well-balancedness. This modification makes use of the fact that the total mass equation using the free surface level $\eta = h_s + h_f + b$ as solution variable can automatically ensure the solution of lake at rest [33], where basal topography b is assumed to be time-independent. It only modifies the solid and fluid heights as computed from the first and third equations of scheme (20).

To avoid confusion, the following variables are not quiescent steady states. By adding the two mass equations in Eq. (2), we obtain the total mass equation

$$\frac{\partial \eta}{\partial t} + \frac{\partial\left(h_s u_s + h_f u_f\right)}{\partial x} = 0. \tag{43}$$

By applying NOC scheme (20) to (43), we additionally compute the free surface level

$$\eta_{i+\frac{1}{2}}^{n+1} = \frac{1}{2}\left(\eta_i^n + \eta_{i+1}^n\right) + \frac{1}{8}\left(\boldsymbol{\sigma}_i^{n,(\eta)} - \boldsymbol{\sigma}_{i+1}^{n,(\eta)}\right) - \frac{\Delta t}{\Delta x}\left[\left(h_s u_s + h_f u_f\right)_{i+1}^{n+\frac{1}{2}} - \left(h_s u_s + h_f u_f\right)_i^{n+\frac{1}{2}}\right], \quad (44)$$

where $\boldsymbol{\sigma}_i^{n,(\eta)}$ is limited difference of $\eta_i^n = h_{s,i}^n + h_{f,i}^n + b_i$. According to previous analysis in Proposition 3.1, the mass fluxes in (44) are zero for quiescent steady states, thus (44) satisfies $\eta_{i+\frac{1}{2}}^{n+1} = \text{Const}$ at quiescent steady states. In general cases, variable $\eta_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ computed from (44) is used to modify the solid and fluid heights $h_{s,i+\frac{1}{2}}^{n+1}$ and $h_{f,i+\frac{1}{2}}^{n+1}$ computed from scheme (20) in the following way:

if $(h_{s,i+\frac{1}{2}}^{n+1} + h_{f,i+\frac{1}{2}}^{n+1} + b_{i+\frac{1}{2}} \neq \eta_{i+\frac{1}{2}}^{n+1})$ *then*

$$h_{s,i+\frac{1}{2}}^{n+1,\mathrm{mod}} = \frac{h_{s,i+\frac{1}{2}}^{n+1}}{h_{s,i+\frac{1}{2}}^{n+1} + h_{f,i+\frac{1}{2}}^{n+1}} \times \left(\eta_{i+\frac{1}{2}}^{n+1} - b_{i+\frac{1}{2}}\right),$$
$$h_{f,i+\frac{1}{2}}^{n+1,\mathrm{mod}} = \frac{h_{f,i+\frac{1}{2}}^{n+1}}{h_{s,i+\frac{1}{2}}^{n+1} + h_{f,i+\frac{1}{2}}^{n+1}} \times \left(\eta_{i+\frac{1}{2}}^{n+1} - b_{i+\frac{1}{2}}\right), \quad (45)$$

else no modification is made to $h_{s,i+\frac{1}{2}}^{n+\frac{1}{2}}$ and $h_{f,i+\frac{1}{2}}^{n+\frac{1}{2}}$ as computed from scheme (20).

The topography $b_{i+\frac{1}{2}}$ in (45) can be defined as $b_{i+\frac{1}{2}} = \frac{1}{2}(b_i + b_{i+1})$ or $z_b(x_{i+1/2})$ and never changes with time stepping. The solid and fluid heights after modification (45) are then used as the final solution for $(n+1)$th time step. It is seen that (45) leads to $h_{s,i+\frac{1}{2}}^{n+1,\mathrm{mod}} + h_{f,i+\frac{1}{2}}^{n+1,\mathrm{mod}} + b_{i+\frac{1}{2}} = \eta_{i+\frac{1}{2}}^{n+1}$, which can keep the constant at quiescent steady states. And the ratio $h_{s,i+\frac{1}{2}}^{n+1,\mathrm{mod}}/h_{f,i+\frac{1}{2}}^{n+1,\mathrm{mod}} = h_{s,i+\frac{1}{2}}^{n+1}/h_{f,i+\frac{1}{2}}^{n+1} = \text{Const}$ is reserved.

3.4. Positivity preserving

We can prove that NOC scheme (20) is positivity-preserving under a suitable CFL condition.

Proposition 3.2. *Assume that system (2) is solved with NOC scheme (20) and that* $h_{s,i}^n \geq 0, h_{f,i}^n \geq 0, \ \forall i$. *Then* $h_{s,i+\frac{1}{2}}^{n+1} \geq 0, \ h_{f,i+\frac{1}{2}}^{n+1} \geq 0, \forall i$, *provided that*

$$\Delta t \leq \Delta x \min\left(\frac{\text{CFL}}{\mathcal{S}}, \frac{1}{4\mathcal{U}}\right) \quad (46)$$

where

$$\mathcal{U} = \max_i\left(\max(|u_{s,i}^n|, |u_{f,i}^n|)\right),$$
$$\mathcal{S} = \max_i\left(\left|\max(u_{s,i}^n, u_{f,i}^n) + \sqrt{g(h_{s,i}^n + h_{f,i}^n)}\right|, \left|\min(u_{s,i}^n, u_{f,i}^n) - \sqrt{g(h_{s,i}^n + h_{f,i}^n)}\right|\right). \quad (47)$$

Proof. Since the mass equations of fluid and solid are similar, we take the fluid mass equation as example. The third component of NOC scheme (20) gives

$$h_{f,i+\frac{1}{2}}^{n+1} = \frac{1}{2}\left(h_{f,i}^n + \frac{1}{4}\bar{\delta} h_{f,i}^n + h_{f,i+1}^n - \frac{1}{4}\bar{\delta} h_{f,i+1}^n\right) - \frac{\Delta t}{\Delta x}\left[\left(h_f u_f\right)_{i+1}^{n+\frac{1}{2}} - \left(h_f u_f\right)_i^{n+\frac{1}{2}}\right], \quad (48)$$

where $\bar{\delta} h_{f,i}^n$ represents the limited difference. The numerical flux $(h_f u_f)_i^{n+1/2}$ is computed according Taylor expansion (22),

$$(h_f u_f)_i^{n+\frac{1}{2}} = (h_f u_f)_i^n + \frac{\Delta t}{2}\left.\frac{\partial(h_f u_f)}{\partial t}\right|_i^n. \quad (49)$$

The second term can be ignored as it is $\mathcal{O}(\Delta t)$ higher. Hence, (48) can be rewritten as

$$\begin{aligned} h_{f,i+\frac{1}{2}}^{n+1} &= \frac{1}{2}\left[h_{f,i}^n + \frac{1}{4}\bar{\delta} h_{f,i}^n + \frac{2\Delta t}{\Delta x}\left(h_f u_f\right)_i^n\right] \\ &\quad + \frac{1}{2}\left[h_{f,i+1}^n - \frac{1}{4}\bar{\delta} h_{f,i+1}^n - \frac{2\Delta t}{\Delta x}\left(h_f u_f\right)_{i+1}^n\right] \\ &= \frac{1}{2}\left[\frac{1}{2}\left(h_{f,i}^n + \frac{1}{2}\bar{\delta} h_{f,i}^n\right) + h_{f,i}^n\left(\frac{1}{2} + \frac{2\Delta t u_{f,i}^n}{\Delta x}\right)\right] \\ &\quad + \frac{1}{2}\left[\frac{1}{2}\left(h_{f,i+1}^n - \frac{1}{2}\bar{\delta} h_{f,i+1}^n\right) + h_{f,i+1}^n\left(\frac{1}{2} - \frac{2\Delta t u_{f,i+1}^n}{\Delta x}\right)\right] \end{aligned} \quad (50)$$

Based on the monotonicity of the minmod-MUSCL interpolation, we have $h_{f,i}^n + \frac{1}{2}\bar{\delta} h_{f,i}^n \geq 0$ and $h_{f,i+1}^n - \frac{1}{2}\bar{\delta} h_{f,i+1}^n \geq 0$ if $h_{f,i}^n \geq 0, \forall i$. In order to make the second terms in the square brackets in (50) be non-negative, time step Δt must satisfy

$$\frac{\Delta t |u_{f,i}^n|}{\Delta x} \leq \frac{1}{4}, \quad \forall i. \quad (51)$$

With (51) holding, the result of (50) is non-negative. A similar time step constraint can also be obtained for the solid phase.

It is evident that (46) is the minimal time step constrained by the positivity preserving condition (51) for fluid and solid phases and by the original CFL condition of NOC scheme,

$$\Delta t = \text{CFL}\frac{\Delta x}{\mathcal{S}}. \quad (52)$$

Therefore, if (46) is satisfied, then $h_{s,i+\frac{1}{2}}^{n+1} \geq 0, h_{f,i+\frac{1}{2}}^{n+1} \geq 0, \ \forall i.$ □

3.5. Numerical treatment for the loss of hyperbolicity

Although NOC scheme (20) with well-balanced correction (45) is stable even if the matrices of the system have complex eigenvalues, strong unphysical oscillations may appear in the numerical solutions when the modulus of the complex eigenvalues are big enough. Therefore, we adopt a similar predictor/corrector strategy [31] to recover the hyperbolic nature of the two-phase shallow-water system once the necessary condition (ii) of Proposition 2.1 is present. The predictor step is scheme (20) with well-balanced correction (45), which gives the first set of approximations at time t^{n+1}, $\mathbf{U}^{n+1,*} = [h_{s,i}^{n+1,*}, h_{f,i}^{n+1,*}, (hu)_{s,i}^{n+1,*}, (hu)_{f,i}^{n+1,*}]^T$. In the corrector step, the state $\mathbf{U}^{n+1}$ is computed by a numerical relaxation procedure

$$\begin{aligned} h_{s,i}^{n+1} &= h_{s,i}^{n+1,*}, \\ h_{f,i}^{n+1} &= h_{f,i}^{n+1,*}, \\ (hu)_{s,i}^{n+1} &= (hu)_{s,i}^{n+1,*} + \gamma\,\Delta t D_i^{n+1}\left(h_{s,i}^{n+1} + h_{f,i}^{n+1}\right)\left(u_{f,i}^{n+1} - u_{s,i}^{n+1}\right), \\ (hu)_{f,i}^{n+1} &= (hu)_{f,i}^{n+1,*} - \Delta t D_i^{n+1}\left(h_{s,i}^{n+1} + h_{f,i}^{n+1}\right)\left(u_{f,i}^{n+1} - u_{s,i}^{n+1}\right). \end{aligned} \quad (53)$$

The last two equations in (53) are rewritten as follows

$$\begin{aligned} u_{s,i}^{n+1} &= u_{s,i}^{n+1,*} + \frac{\gamma\,\Delta t D_i^{n+1}\left(h_{s,i}^{n+1} + h_{f,i}^{n+1}\right)}{h_{s,i}^{n+1}}\left(u_{f,i}^{n+1} - u_{s,i}^{n+1}\right), \\ u_{f,i}^{n+1} &= u_{f,i}^{n+1,*} - \frac{\Delta t D_i^{n+1}\left(h_{s,i}^{n+1} + h_{f,i}^{n+1}\right)}{h_{f,i}^{n+1}}\left(u_{f,i}^{n+1} - u_{s,i}^{n+1}\right). \end{aligned} \quad (54)$$

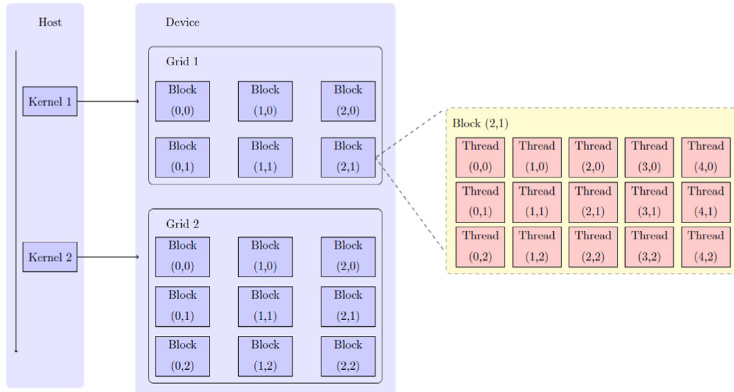


Fig. 4. Sketch of organization of a CUDA program.

By subtracting and deducing the common factor we obtain

$$u_{f,i}^{n+1} - u_{s,i}^{n+1} = \frac{u_{f,i}^{n+1,*} - u_{s,i}^{n+1,*}}{1 + \Delta t D_i^{n+1}\left(h_{f,i}^{n+1} + h_{s,i}^{n+1}\right)\left(\dfrac{1}{h_{f,i}^{n+1}} + \dfrac{\gamma}{h_{s,i}^{n+1}}\right)} \tag{55}$$

In order to ensure hyperbolicity, the new velocity difference from (55) must satisfy the sufficient condition (i) of Proposition 2.1, which leads to

$$\begin{aligned}
&\frac{|u_{f,i}^{n+1,*} - u_{s,i}^{n+1,*}|}{\left|1 + \Delta t D_i^{n+1}\left(h_{f,i}^{n+1} + h_{s,i}^{n+1}\right)\left(\dfrac{1}{h_{f,i}^{n+1}} + \dfrac{\gamma}{h_{s,i}^{n+1}}\right)\right|} \le 2a\beta, \quad \text{or} \\
&\frac{|u_{f,i}^{n+1,*} - u_{s,i}^{n+1,*}|}{\left|1 + \Delta t D_i^{n+1}\left(h_{f,i}^{n+1} + h_{s,i}^{n+1}\right)\left(\dfrac{1}{h_{f,i}^{n+1}} + \dfrac{\gamma}{h_{s,i}^{n+1}}\right)\right|} \ge 2a, \\
&\text{with} \quad a = \sqrt{g(h_{f,i}^{n+1} + h_{s,i}^{n+1})},
\end{aligned} \tag{56}$$

For the first inequality, $D_i^{n+1} \ge 0$, and for the second inequality, $D_i^{n+1} \le 0$. So D_i^{n+1} must satisfy

$$\begin{aligned}
&\max\left(0, \frac{\dfrac{|\Delta u^*|}{2a\beta} - 1}{\Delta t\left(\dfrac{1}{1-\phi} + \dfrac{\gamma}{\phi}\right)}\right) \le D_i^{n+1}, \\
&\text{or} \quad D_i^{n+1} \le \min\left(0, \frac{\dfrac{|\Delta u^*|}{2a} - 1}{\Delta t\left(\dfrac{1}{1-\phi} + \dfrac{\gamma}{\phi}\right)}\right),
\end{aligned} \tag{57}$$

where $\Delta u^* = |u_{f,i}^{n+1,*} - u_{s,i}^{n+1,*}|$, $\phi = h_{s,i}^{n+1}/(h_{f,i}^{n+1} + h_{s,i}^{n+1})$. The relevant hyperbolic regime for applications is the one corresponding to small $|u_f - u_s|$. It is understood that real inter-phase drag forces tend to drive phase velocities closer. Therefore, the first equality in (57) is chosen.

4. GPU implementation

In this section, we will describe GPU implementation for 2D NOC scheme (25). This scheme is explicit and therefore ideal for the parallel execution on GPUs. We start from single GPU implementation as it is the building block of multiple GPU implementation.

4.1. Single GPU

A CUDA program on single GPU includes two kinds of codes, the serial codes and the parallel codes [34,35]. The serial codes that run on the host (CPU) side are responsible for variables declaration, initialization, data transmission, and kernel invocation. The parallel codes (called "kernel functions") running on the device (GPU) side are executed in parallel by massive light-weight threads organized to match the GPU hardware feature and allow for mapping typical data structures (arrays, matrices). Many warps of threads (1 warp $=32$ in most architectures) make up a block, and many blocks stack together to make up a grid, which is the counterpart of a kernel function. The block can be organized into 1D, 2D or 3D array of threads, while the grid may also be in 1D, 2D or 3D array of blocks. In Fig. 4, a 2D block and two 2D grids are illustrated. On single GPU, the NOC scheme (25) is implemented similar to the predictor–corrector Davis method as explained in our previous paper [36]. After initialization, the CPU code loops a stepping function to advance the solution in time. The stepping function invokes several different CUDA kernels, which do all the computations of the NOC scheme including slope limiter, reconstruction, numerical flux, source term $\mathbf{S}$, nonconservative term $\mathcal{H}$, as well as solution update $\overline{\mathbf{U}}^{n+1}$. These are explicit stencil computations can be run in parallel on GPU. The global time step which involves reduction operation is also computed on GPU. We used 2D arrays rather than conventional 1D arrays in GPU global memory to enhance the readability of the code but sacrifice efficiency. For more detail, see [36].

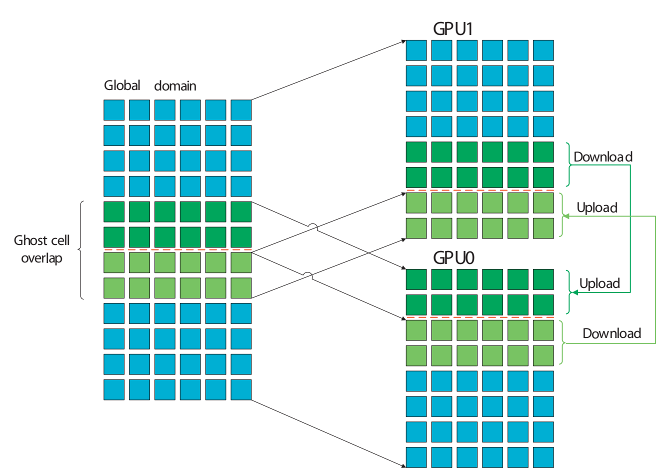


Fig. 5. Domain decomposition and each GPU takes charge of one subdomain.

Table 1
Comparison of running times using OpenMP-CUDA and multistream-CUDA for 1–3 GPUs with meshes 200 × 200 and 800 × 800 respectively. Result using a CPU with OpenMP is also shown.

Number of GPUs	$n_x \times n_y$	OpenMP-CUDA	Multistream-CUDA
1	200 × 200	2478 ms	
	800 × 800	115938 ms	
2	200 × 200	2080 ms	2190 ms
	800 × 800	68799 ms	68200 ms
3	200 × 200	2035 ms	2260 ms
	800 × 800	51523 ms	49980 ms
10 cores CPU	200 × 200	7354 ms	
	800 × 800	492586 ms	

4.2. Multiple GPUs

For using multiple GPUs in a single node, there are several choices of strategies. One can use multistream technique provided by CUDA, or multiple threads with OpenMP (or Pthread), or message passing interface (MPI), with each stream/thread/process managing a GPU executing a partitioned task. The MPI approach is mainly suited to GPU clusters, so in this work, we only implement the OpenMP-CUDA and the multistream-CUDA strategies and compare their performances. The computational domain is divided into multiple subdomains for multiple GPUs to handle. The subdomains have an overlap of four ghost cells for the purpose of communication, two from each of the neighboring subdomains, as exemplified in Fig. 5. In the OpenMP-CUDA strategy, as the code runs in the OpenMP *parallel region*, the main program spawns multiple threads equal to the number of subdomains, and each thread is attached to a GPU that initializes and computes a corresponding subdomain. In the multistream-CUDA strategy, the main program issues multiple streams in turn, and each stream attaches to a GPU. The OpenMP-CUDA strategy is shown in procedure 1. It is seen that there is no need to do initialization in CPU and copy data to GPU. But device synchronization is performed after step 6 for calculating a global time step and in step 8 for exchanging overlapped cells between GPUs (“download” and “upload” in Fig. 5). The multistream-CUDA strategy is shown in procedure 2. The only difference between the two strategies is that the starting of GPU execution is synchronous in the OpenMP strategy while it is asynchronous in the multistream strategy.

Procedure 1. OpenMP-CUDA procedure:

1: Set the number of OMP threads and the same number of GPU devices.

2: **♯pragma omp parallel shared** (Δt, ghost variables...) {
/* Start parallel region executed by OpenMP threads. Declare ghost variables as shared variables in order to transfer data between GPUs. */

3: Each OpenMP thread binds with a GPU device, allocate the shared variables, and initialize a subdomain, e.g. the current thread $k =$ *omp_get_thread_num*(*void*)will take charge of a subdomain with index (b_x, b_y) for the 2D decomposition of the computational domain being

$$(b_x, b_y) = (k \,\%\, OMPthreadnumber\_x,\\ k \,\%\, OMPthreadnumber\_y)$$

4: *InitCon*<<<*BLOCKs*, *THREADs*>>>(variable list including GPU id k)
/* This kernel function computes initial conditions; run in parallel on all GPUs. Need not copy data to GPU. */

5: **while (T<totaltime) do**

6: Execute *getdeltat*<<<*BLOCKs*, *THREADs*>>>(variable list including GPU id k)
/* this kernel returns $\Delta t(k)$ of each subdomain. Subsequently, synchronize all threads and then apply OpenMP *reduce* operation to all threads' $\Delta t(k)$ to get a minimum Δt of the whole domain. */

7: Execute kernels for the NOC formula (25) to compute solution variables.

8: Synchronize all threads, and then exchange the ghost variables betweenGPUs. Can use the peer-to-peer function to exchange data in new device like K40 GPU used in this work.

9: **enddo while**

10: Copy data back to CPU and write the data file.

11: } // end of OMP parallel region

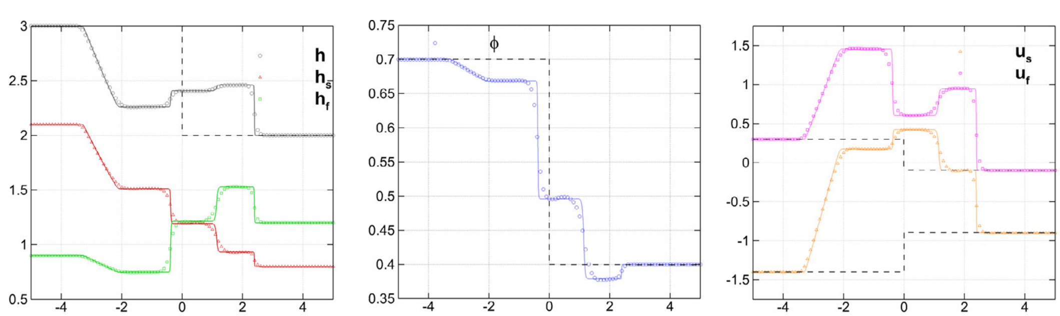
MultiStream-CUDA procedure:

1: Set the number of streams and the same number of GPU devices.

2: Use **for loop** to create stream, and define variables and allocate corresponding global memories on each GPU device corresponding to each stream.
/* Streams are created serially but execute own contexts in parallel asynchronously */

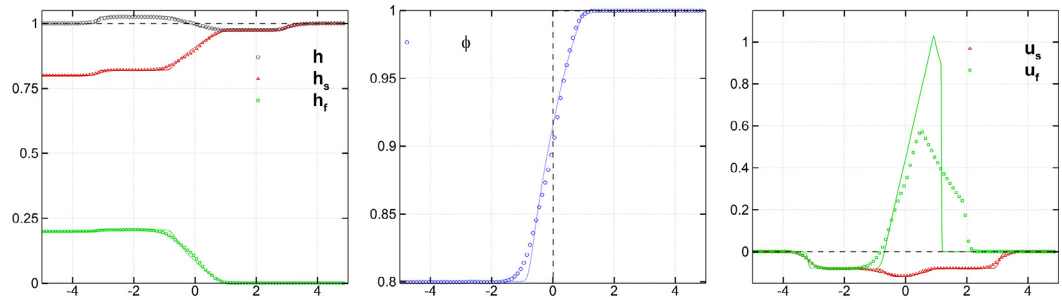
3: Each stream executes *InitCon*<<<*BLOCKs*, *THREADs*, *0*, *stream[k]*>>> (variable list)on kth GPU.
/* This is done in a **for loop**, but streams managing corresponding GPUs run in parallel and asynchronously. */

4: **while T<totltime) do**



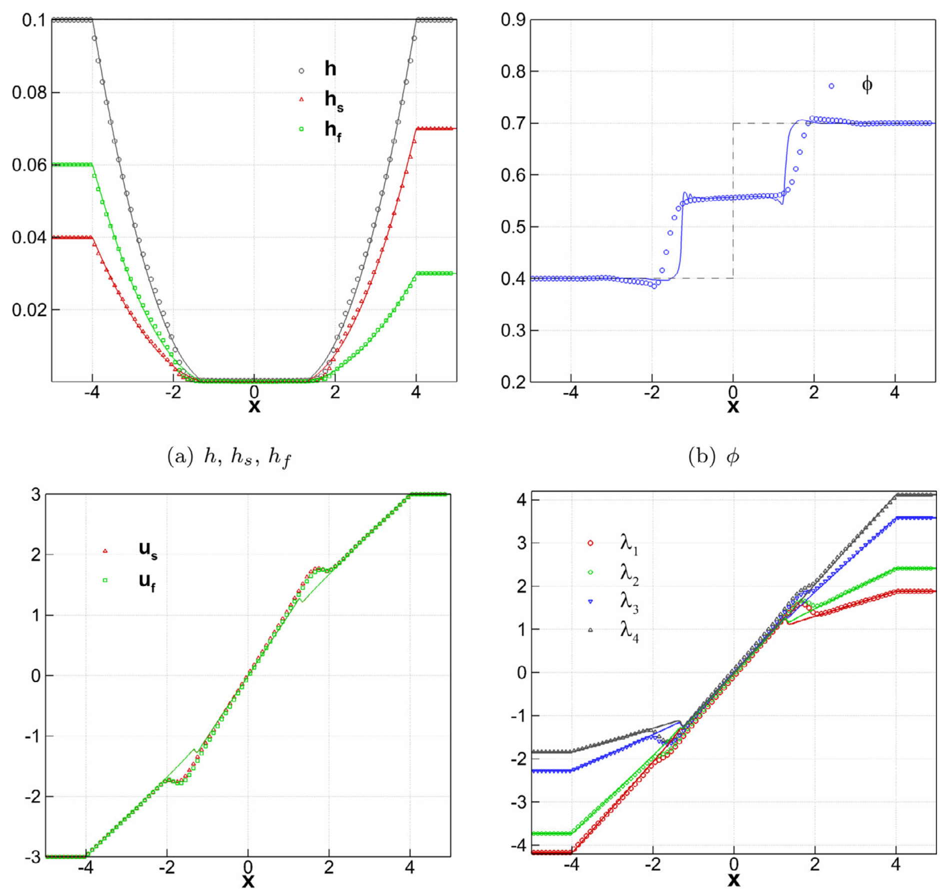
(a) h, h_s, h_f (b) ϕ (c) u_s and u_f

Fig. 6. Simple Riemann problem. Results computed with 100 grid cells (symbols) are compared with reference solution obtained with 1000 cells (solid lines). Dashed lines represent the initial conditions.



(a) h, h_s, h_f (b) ϕ (c) u_s and u_f

Fig. 7. Rarefaction into vacuum for the fluid constituent. Results computed with 100 grid cells (symbols) are compared with reference solution obtained with 1000 cells (solid lines). Dashed line is the initial conditions.



(a) h, h_s, h_f (b) ϕ

(c) u_s and u_f (d) eigenvalues

Fig. 8. Test 1. Results at $t = 1$ computed with 100 grid cells (symbols) in comparison with reference solution obtained with 4000 cells (solid lines). The eigenvalues are computed according to formulas in [11].

5: Each stream executes *getdeltat*<<<*BLOCKs*, *THREADs*, *0*, *stream[k]*>>> (variable list)
/* Each stream executes this kernel function to compute $\Delta t(k)$ for each subdomain. Subsequently, synchronize all streams and then reduce to get the min-value of Δt of the whole domain. */

6: Each stream executes kernels for NOC formula (25) in a corresponding GPU device to compute the solution variables.

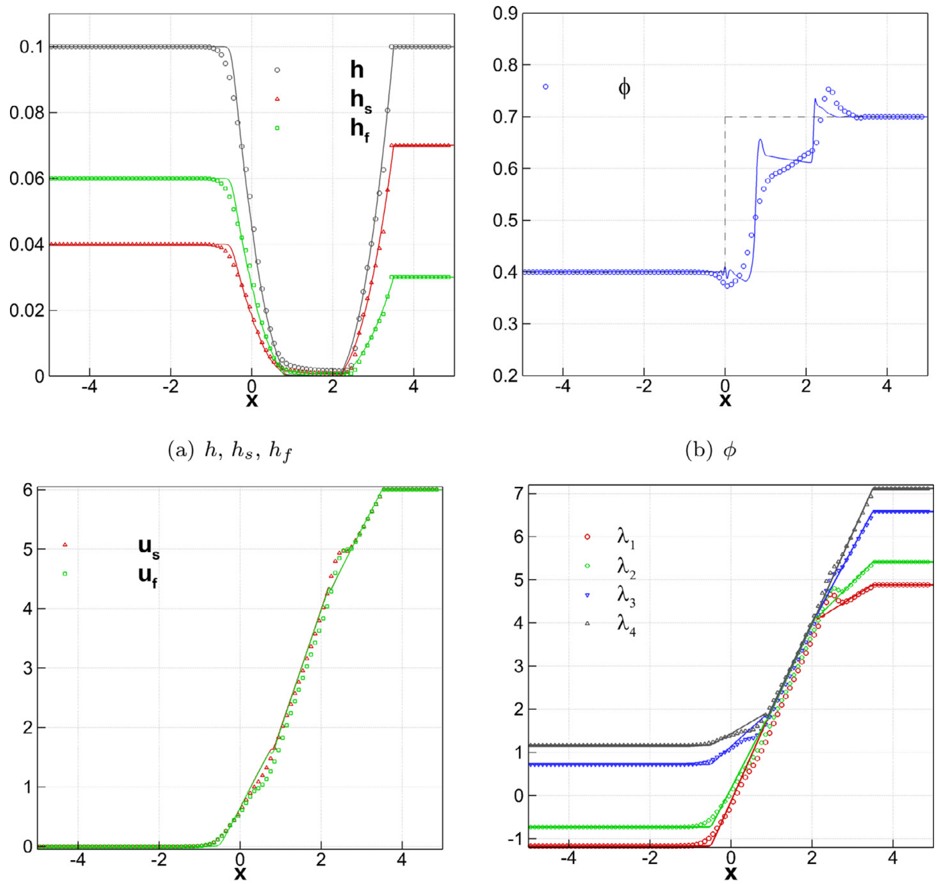
7: Synchronize all streams, and then exchange the ghost variables betweenGPUs. Can use the peer-to-peer function to do so in K40 GPU.

8: enddo while

9: Copy data back to CPU, write the data file and destroy all streams.

In this work, Nvidia K40 GPU is used which enables the peer-to-peer (GPU to GPU) memory copy [37], we avoid the copy of GPU → CPU → GPU for transferring the overlapped ghost cells. The function of peer-to-peer copy is [37,38]

cudaError_t cudaMemcpyPeer(void *dst, int dstDevice, const void *src, int srcDevice, size_t count)



(a) h, h_s, h_f

(b) ϕ

(c) u_s and u_f

(d) eigenvalues

Fig. 9. Test 2. Results $t = 0.5$ computed with 100 grid cells (symbols) compared with reference solution obtained with 4000 cells (solid lines). The eigenvalues are computed according to formulas in [11].

which copies **count** bytes of data from pointer **src** in source device **srcDevice** to pointer **dst** in destination device **dstDevice.**

In our numerical simulations, we used a server with 2 × Intel Xeon E5-2690v2 (3.0GHz, 10 cores) CPUs, 64 GB DDR3-1600 memory, and 3 × Tesla K40 GPUs (2880 cores). Table 1 shows comparison of wall times for two multi-GPU strategies. It is seen that the multistream-CUDA strategy can be slightly faster or slower than the OpenMP-CUDA one in some grid sizes. On the 800 × 800 cells, 3-GPU parallel efficiency is 77%, and the speedup of GPU/CPU(10-core OpenMP) is 4.2. There is space to improve single GPU performance.

5. Numerical examples

We now present numerical results for the 1D and 2D two-phase flow models, Eqs. (4) and (10), respectively. Neither inter-phase drag nor basal friction force are considered. In all the examples we set $\gamma = 0.5$.

5.1. 1D simple Riemann problem

This Riemann problem was studied in Refs. [10,12]. The initial conditions consists of two constant states separated by an interface located at $x = 0$. The left and right states are

$$(h, \phi, u_s, u_f)_l = (3, 0.7, -1.4, \quad 0.3),$$
$$(h, \phi, u_s, u_f)_r = (2, 0.4, -0.9, -0.1), \tag{58}$$

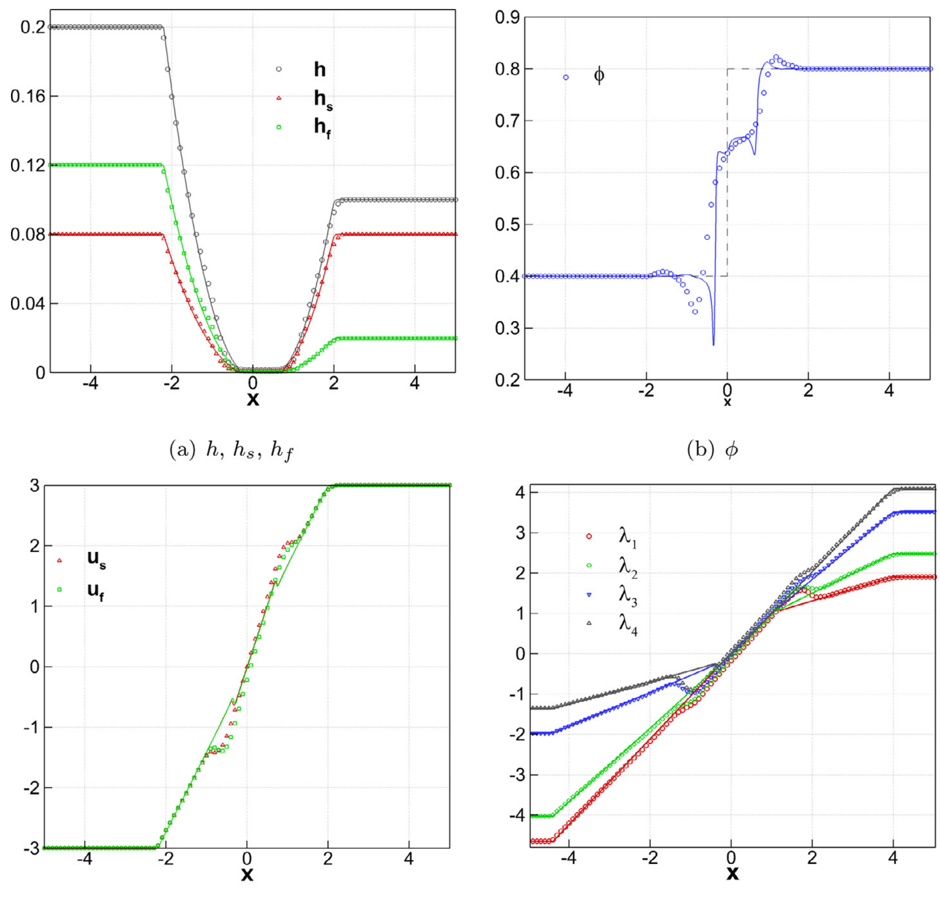
The value $g = 9.81$, the bottom is flat, i.e. $b(x) = \text{constant}$, the computational domain is $[-5, 5]$, and CFL $= 0.2$. In Fig. 6, we display the results at $t = 0.5$ obtained with 100 grid cells (symbols) and 1000 cells (solid lines) for flow height variables h, h_s, h_f, the solid volume fraction ϕ and the phase velocities u_s, u_f. It is observed that the Riemann solution of this problem consists of a 1-rarefaction wave, a 2-shock wave, a 3-rarefaction wave, and a 4-shock wave (n-wave means that the wave is associated with the nth eigenvalue in Eq. (8)).

5.2. Rarefaction into vacuum of the fluid constituent (dam break problem for liquid)

The problem in [11,12] is considered here. The fluid mixture has the same height over the whole spatial domain, and the right side states for the fluid phase is a vacuum zone. The initial data are

$$(h, \phi, u_s, u_f)_l = (1, 0.8, 0, 0), \quad \text{if} \quad x \le 0,$$
$$(h, \phi, u_s, u_f)_r = (1, \quad 1, 0, 0), \quad \text{if} \quad x > 0, \tag{59}$$

The gravity constant $g = 9.81$, and the bottom is flat with $b(x) = \text{constant}$. The numerical results obtained with 100 cells and 1000 cells at $t = 1$ are shown in Fig. 7. The Riemann solution for the mixture contains a 2-rarefaction across which h_f vanishes. Moreover, in the far left zone of $x < -2$, which is the pure solid material, a 1-shock is formed, and in the far right zone of $x > 2$, which is also pure solid material, a 4-rarefaction wave occurs. The third wave associated to eigenvalue λ_3 is not evident as explained in [11]. The present results are in fair agreement with those in [11], and the resolution is comparable to the CE/SE method [12] but lower than the VFRoe solver [11].



(a) h, h_s, h_f (b) ϕ

(c) u_s and u_f (d) eigenvalues

Fig. 10. Test 3. Results $t = 0.5$ computed with 100 grid cells (symbols) compared with reference solution obtained with 4000 cells (solid lines). The eigenvalues are computed according to formulas in [11].

Table 2
Initial data for the test cases of dry bed formation.

Test	$(h, \phi, u_s, u_f)_l$	$(h, \phi, u_s, u_f)_r$
1	(0.1, 0.4, −3, −3)	(0.1, 0.7, 3, 3)
2	(0.1, 0.4, 0, 0)	(0.1, 0.7, 6, 6)
3	(0.2, 0.4, −3, −3)	(0.1, 0.8, 3, 3)

5.3. Dry bed generation

This problem in [11,12] is concerned with the formation of a dry bed zone. Three test cases of the Riemann problems are considered. The solutions consist of two opposite moving rarefaction fans between which a dry bed region is generated. The initial data are given in Table 2:

In all test cases, the initial interface is located at $x = 0$ and the bottom is constant, i.e. $b(x) =$ constant. The computational domain is $[-5, 5]$, CFL $= 0.5$, and $g = 9.81$. Our model is the same as [12] in that no inter-phase drag is used whereas [11] used infinitely large drag force so that so that $|u_s - u_f|$ is instantaneously driven to zero. Figs. 8–10 display results at $t = 1$, $t = 0.5$, $t = 0.5$ for the three cases, respectively.

In Fig. 8 we display for test 1 the plots of the flow height h, h_s, h_f, solid volume fraction ϕ, velocities u_s, u_f, and the four eigenvalues λ_1, λ_2, λ_3, λ_4 of matrix **A** in (7), respectively. As the fluids separate outward at $x = 0$, a region of vacuum is generated. From Fig. 8(a) it is seen that both 4000 grid cells (solid lines) and 100 grid cells (symbols) capture a vacuum region (defined as where $h < 10^{-5}$) . The distribution of the solid fraction and velocity inside this vacuum region may be abnormal but we plot them in the vacuum region for completeness rather than leaving them blank as in [11]. We can see that there are overshot and undershot in ϕ near the edges of the vacuum region, but they diminish for 4000 cells.

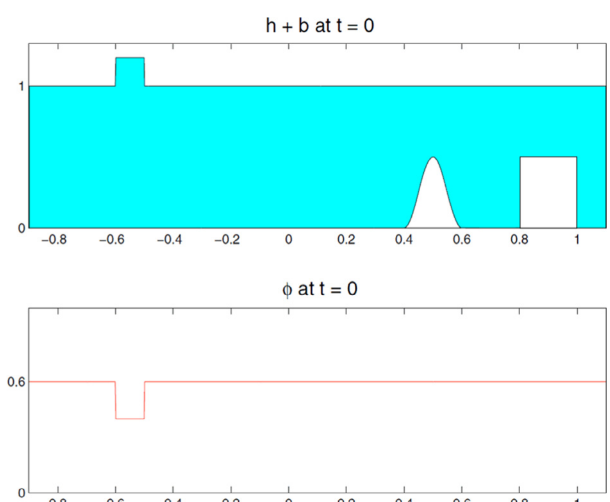
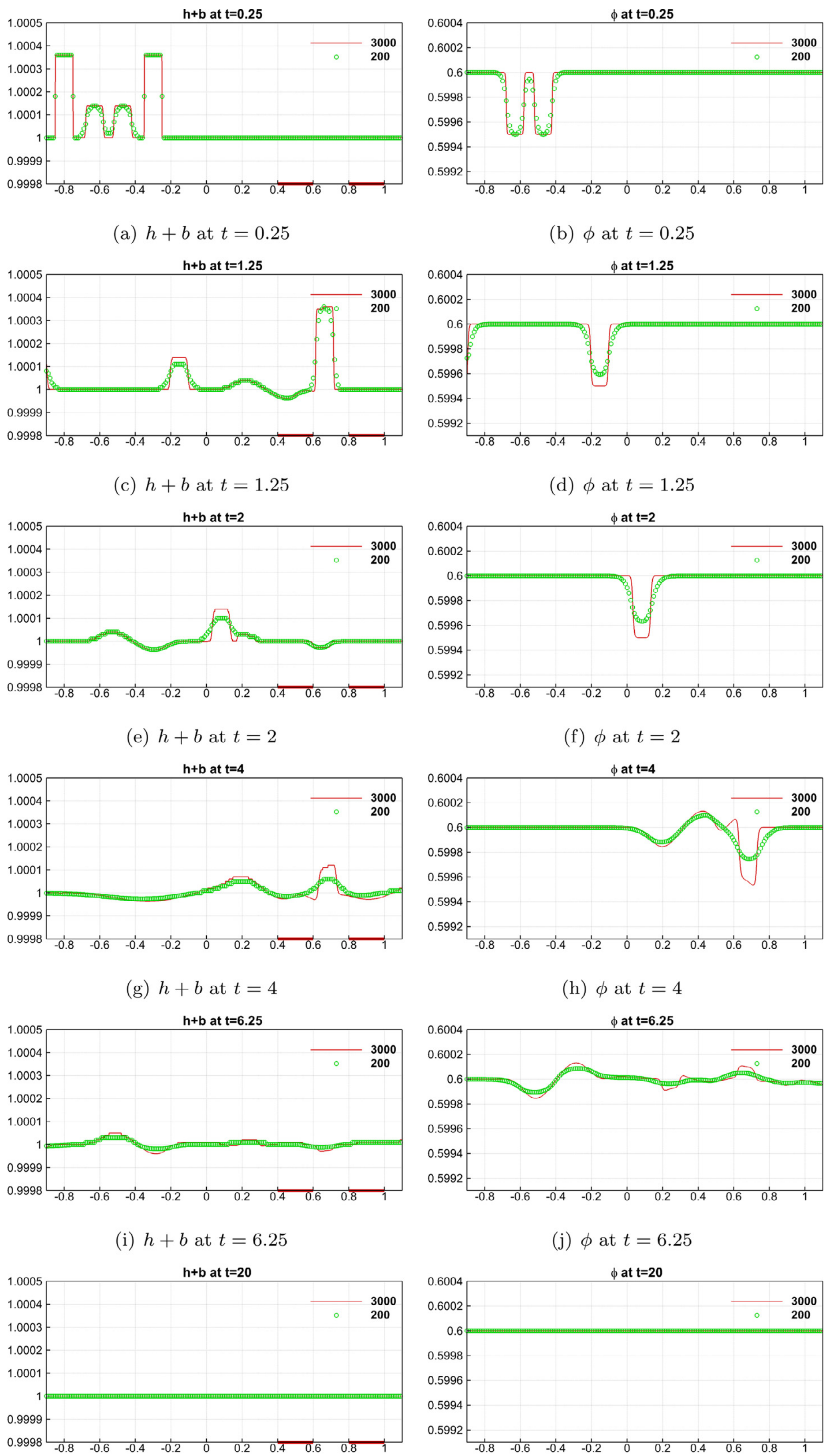


Fig. 11. Initial conditions for the numerical test of perturbation of a steady state at rest. Left: total flow height $h + b$, Right: solid volume fraction ϕ. (Here $\tilde{h}$ and $\tilde{\phi}$ are made much larger than the values used in the calculation in order to make it clear for the readers.)



(a) $h+b$ at $t=0.25$ (b) ϕ at $t=0.25$

(c) $h+b$ at $t=1.25$ (d) ϕ at $t=1.25$

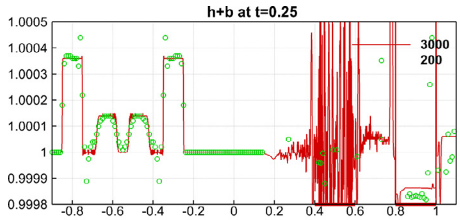
(e) $h+b$ at $t=2$ (f) ϕ at $t=2$

(g) $h+b$ at $t=4$ (h) ϕ at $t=4$

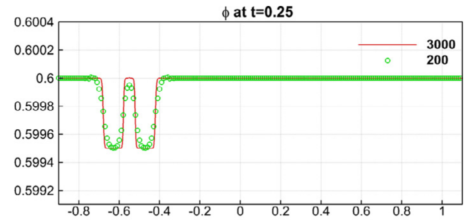
(i) $h+b$ at $t=6.25$ (j) ϕ at $t=6.25$

(k) $h+b$ at $t=20$ (l) ϕ at $t=20$

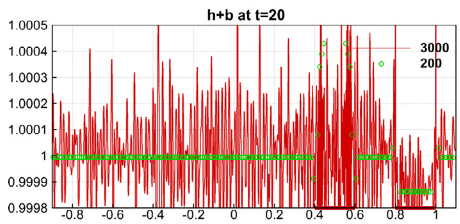
Fig. 12. Perturbation of a steady state at rest($\tilde{h} = \tilde{\phi} = 10^{-3}$). Circles: solution computed with 200 grid cells; solid line: reference solution computed with 3000 grid cells. Red bold lines over [0.4, 0.6] and [0.8, 1.0] : region where $b(x) \neq 0$.



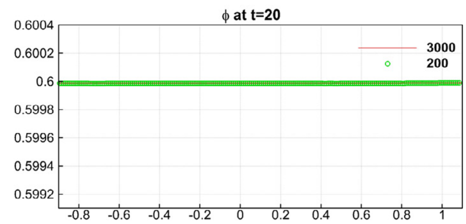
(a) $h+b$ at $t=0.25$ without modification



(b) ϕ at $t=0.25$ without modification



(c) $h+b$ at $t=20$ without modification



(d) ϕ at $t=20$ without modification

Fig. 13. The same problem as Fig. 12 but without using the correction method (45). Circles: solution computed with 200 grid cells; solid line: reference solution computed with 3000 grid cells. Red bold lines over [0.4, 0.6] and [0.8, 1.0]: region where $b(x) \neq 0$.

The eigenvalues of the solution states are real at this moment as shown in Fig. 8(d), and the solution evolves entirely in the hyperbolic regime.

Fig. 9 shows similar results for test 2, except that a right translation of initial velocities is observed, and the rarefaction in the left is transonic. The vacuum region is not well resolved with 100 grid cells (symbols) while it is captured with 4000 cells, and it is smaller than that in test 1. There are also overshot and undershot in the solid volume fraction. The eigenvalues are all real.

In Fig. 10 for test 3, initial discontinuity is present in the flow height. The results are similar to test 1 and the vacuum on 4000 cells is captured but is smaller than that in test 1. But overshot and undershot are still present in the solid volume fraction even if the eigenvalues are real this time.

5.4. Perturbation of a steady state at rest in 1D

To check the well-balanced property of the present NOC scheme, we consider a test problem of variable bottom topography. This test problem was presented in [10,12], but we add a discontinuous rectangular topography in the right side to show the capability of our well-balanced scheme.

In this problem we look at the behavior of a small perturbation of steady state conditions at rest over a bottom topography defined as

$$b(x) = \begin{cases} 0.25\left(\cos(10\pi(x-\frac{1}{2}))+1\right), & \text{if } |x-0.5| < 0.1 \\ 0.25, & \text{if } |x-0.9| < 0.1\,. \\ 0, & \text{otherwise} \end{cases} \qquad (60)$$

Initially, we take a small perturbation of the flow depth h and of the solid volume fraction ϕ:

$$h(x,0) = h_0 + \tilde{h} \quad \text{and} \quad \phi(x,0) = \phi_0 - \tilde{\phi}, \quad \text{for } -0.6 < x < -0.5, \qquad (61)$$

where $h_0 + b(x) = 1,\ \phi_0 = 0.6,$ and $\tilde{h} = \tilde{\phi} = 10^{-3}$. Fig. 11 shows the initial data for the total flow height $h+b$ and the solid volume fraction ϕ. In order to make the initial conditions more clear, we make $\tilde{h}$ and $\tilde{\phi}$ much larger. The computational domain is $[-0.9, 1.1]$, and free flowing boundary conditions are used. Moreover, we take $g=1$ and $\gamma = 0.5$. We compute the solution with 200 grid cells and compare it with a fine grid reference solution obtained with 3000 grid cells.

In Fig. 12, we display the results for $h+b$ and ϕ at six different times. The red bold line over the two intervals of $x \in [0.4, 0.6]$ and $x \in [0.8, 1.0]$ on the x-axis in the left frames for $h+b$ marks the region where the topography $b(x) \neq 0$.

Fig. 12(a,b) is almost the same as Fig. 9(a) of [10], in which the initial perturbation splits into four waves: two right-going waves and two left-going waves which leave the domain from the left edge. Fig. 12(c,d) shows at $t = 1.25$ the right-going external wave has just passed over the obstacle at the bottom, and it has been partially reflected. This wave has gone out of the right boundary of the domain in Fig. 12(e,f), and the reflected wave generated by it has passed through the incoming right-going internal wave. In Fig. 12(g,h) this right-going internal wave has moved pass the hump and has produced its own reflected wave, which can be clearly distinguished in Fig. 12(g,h). Fig. 12(i,j) shows the time at which the second internal wave goes of the right boundary. Fig. 12(k,l) shows the final situation in which all the disturbances have exited from the computational domain and the lake-at-rest equilibrium is attained.

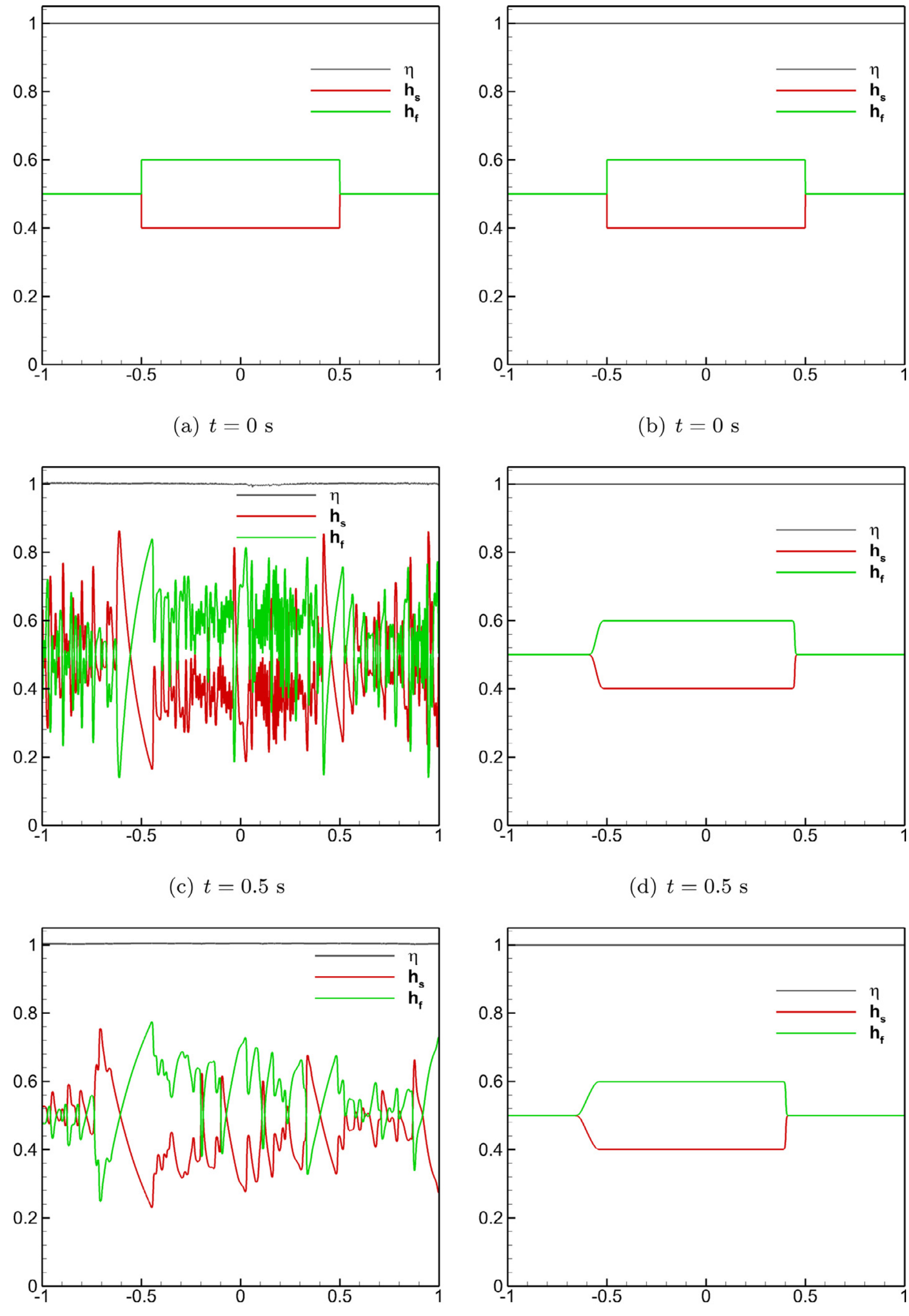
If the same example is computed without the correction method (45), the results will have nonphysical disturbances as shown in Fig. 13(a) and (c) for $t = 0.25$ and $t = 20$. This indicates that our modification is effective in preserving the solution of lake at rest.

5.5. Perturbation of a steady state at rest in 1D with complex eigenvalues

This test [31] is to assess the predictor/corrector strategy in Section 3.5 to enforce the hyperbolicity for the present two-phase shallow granular flow model. The test has an initial condition which is far beyond the hyperbolicity regime. We consider a flat channel ($b(x) = 0$) whose axis is given by the interval $[-5, 5]$. The initial condition is

$$h_s(x,0) = \begin{cases} 0.4, & \text{if } -0.5 \le x \le 0.5 \\ 0.5, & \text{otherwise} \end{cases}, \quad h_s(x,0) = 1 - h_f(x,0),$$

$$u_s(x,0) = 0.2, \quad u_f(x,0) = -0.3. \qquad (62)$$



(a) $t = 0$ s (b) $t = 0$ s

(c) $t = 0.5$ s (d) $t = 0.5$ s

(e) $t = 1$ s (f) $t = 1$ s

Fig. 14. Evolution of free surface level η, solid height h_s, and fluid height h_f. The left column is the result without the inter-phase drag force corrector step, while the right column is that with the inter-phase drag force corrector step. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Free boundary conditions are imposed. The CFL parameter is set to 0.5 and $\Delta x = 0.01$, $\gamma = 0.99$. Note that

$$|u_s - u_f| = 0.5, \quad 2a = 2\sqrt{9.8 \times 1} = 6.26, \quad \phi = 0.5 \ \text{ or } \ 0.6,$$

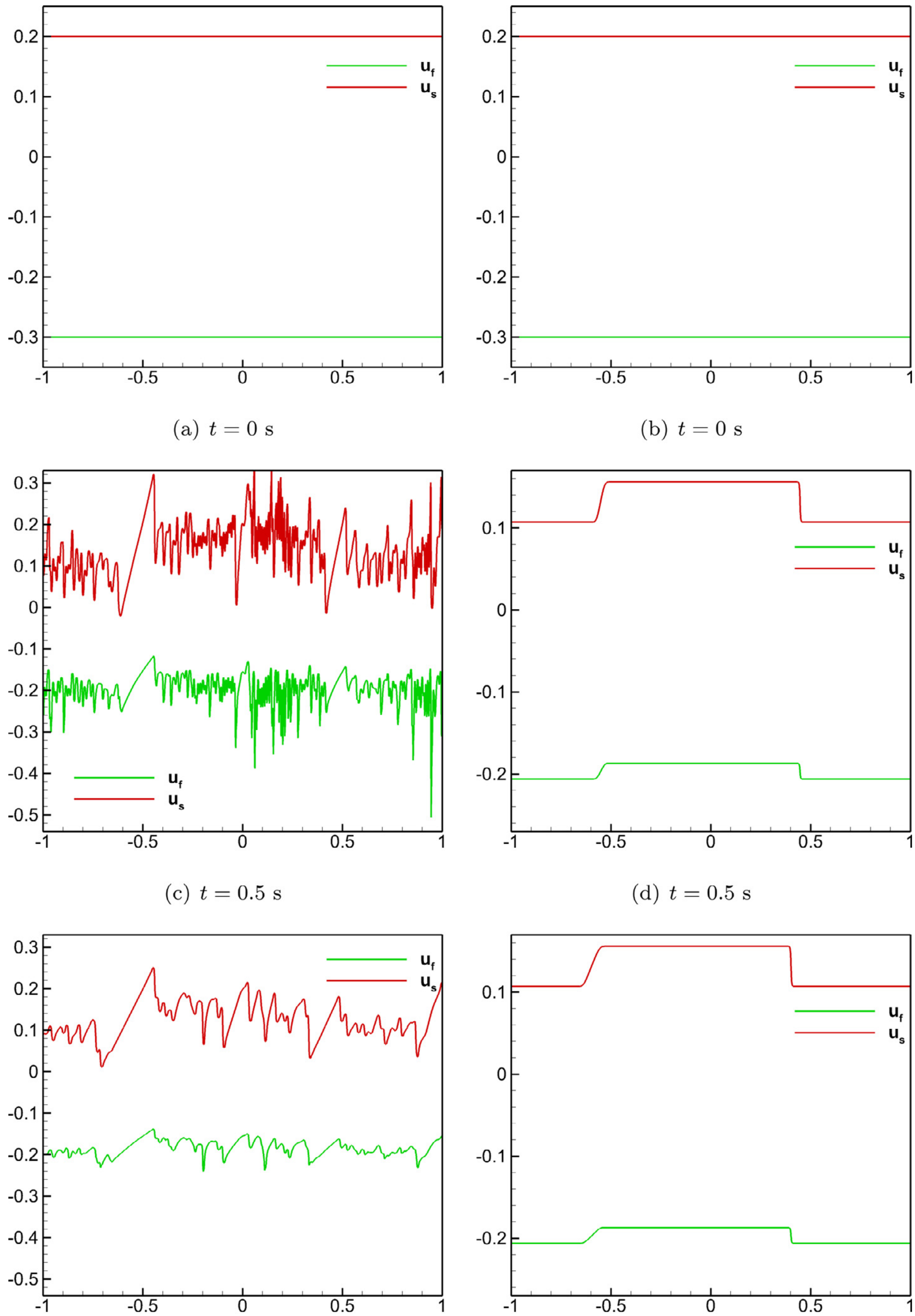
$$\beta = \sqrt{\frac{1}{2}(1-\phi)(1-0.99)} \leq 0.05, \tag{63}$$

the necessary condition (ii), $2a\beta < |u_s - u_f| < 2a$, of Proposition 2.1 is satisfied at every point, therefore, the system may lose hyperbolicity initially.

Fig. 14 shows the free surface level η and partial heights h_s and h_f at $t = 0$ s, $t = 0.5$ s and $t = 1$ s obtained with the well-balanced NOC scheme without (left) or with (right) the corrector step described in Section 3.5. It is seen that the initial perturbation grows without the corrector step while it not the case when the corrector step is performed. Similar results can be observed for the velocities (see Fig. 15).

5.6. Perturbation of a steady state at rest in 2D

This example is modified from a classical example [39] to describe the perturbation of the stationary state of water. The topography and the initial free surface level used are same as those in



(a) $t = 0$ s (b) $t = 0$ s

(c) $t = 0.5$ s (d) $t = 0.5$ s

(e) $t = 1$ s (f) $t = 1$ s

Fig. 15. Evolution of solid and fluid velocities of u_s and u_f. The left column is the result without the inter-phase drag force corrector step, while the right column is the result with the inter-phase drag force corrector step. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

[39] except that an initial steady solid fraction $\phi_0 = 0.6$ is used in the present work.

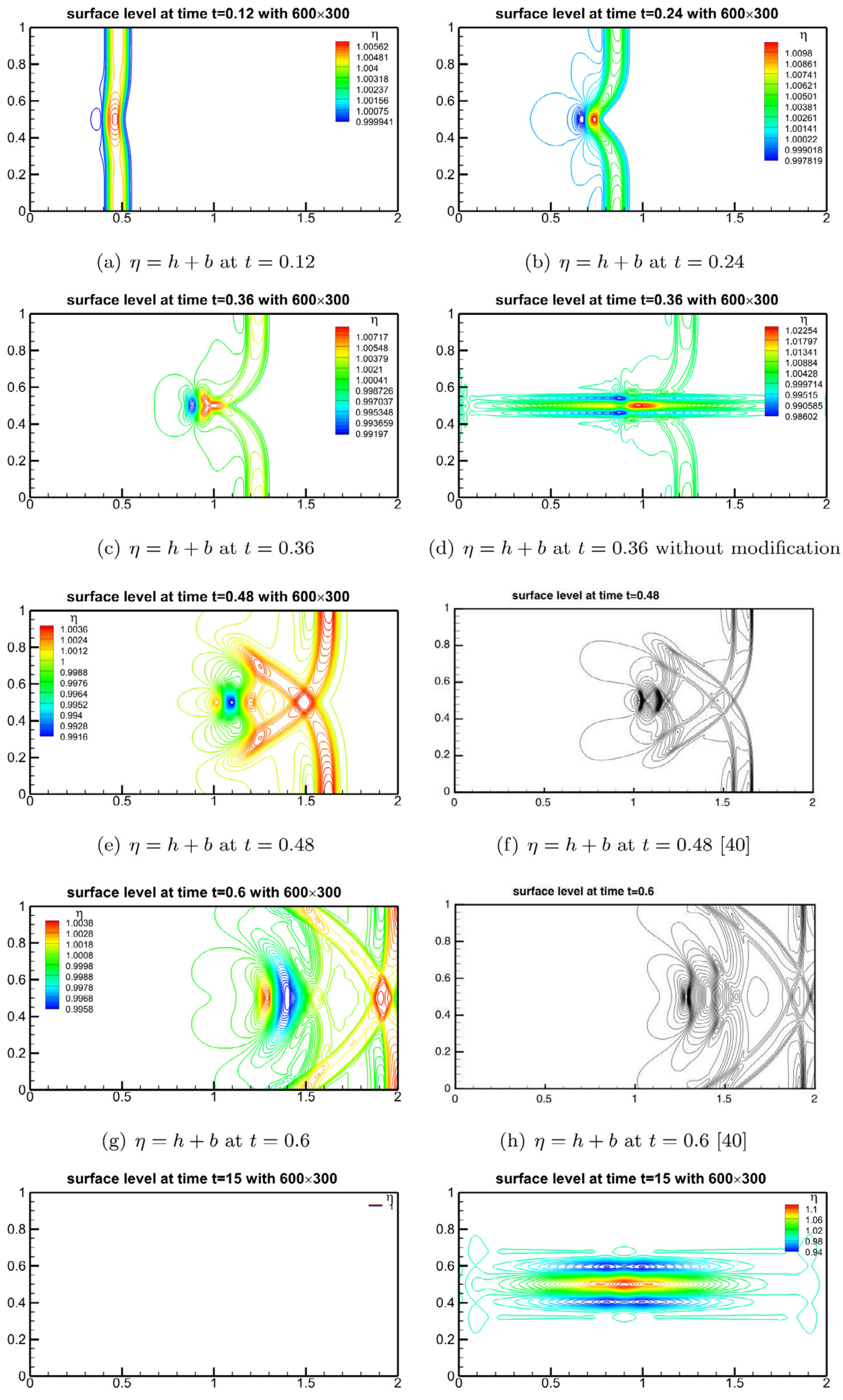
The system is simulated in a rectangular domain [0, 2] × [0, 1]. The bottom topography is an isolated elliptical shaped hump:

$$b(x, y) = 0.8e^{-5(x-0.9)^2-50(y-0.5)^2}. \tag{64}$$

The surface is initially given by

$$\begin{aligned} h(x, y, 0) &= \begin{cases} 1 - b(x, y) + 0.01 & \text{if } 0.05 \le x \le 0.15 \\ 1 - b(x, y) & \text{otherwise} \end{cases}, \\ \phi(x, y, 0) &= 0.6, \\ hu(x, y, 0) &= hv(x, y, 0) = 0. \end{aligned} \tag{65}$$

Hence, the surface is almost flat except for $0.05 \le x \le 0.15$, where h is perturbed upward by 0.01. Fig. 16 displays contours of the free surface $b + h$ at six times ($t = 0.12$, 0.24, 0.36, 0.48, 0.6 and 15) computed by using 2D scheme (25) with 600 × 300 mesh cells. For $t = 0.48$ and 0.6 we compare our results with the single-phase shallow water simulations in [40], and for $t = 0.36$ and 15 we also compare the results between with and without using well-balanced modification (45). The results indicate that our scheme can resolve the complex small features of the flow very well. Fig. 16 (i) and (j) show that at $t = 15$ the free surface level η recovers static status with the modification but has much



(a) $\eta = h + b$ at $t = 0.12$

(b) $\eta = h + b$ at $t = 0.24$

(c) $\eta = h + b$ at $t = 0.36$

(d) $\eta = h + b$ at $t = 0.36$ without modification

(e) $\eta = h + b$ at $t = 0.48$

(f) $\eta = h + b$ at $t = 0.48$ [40]

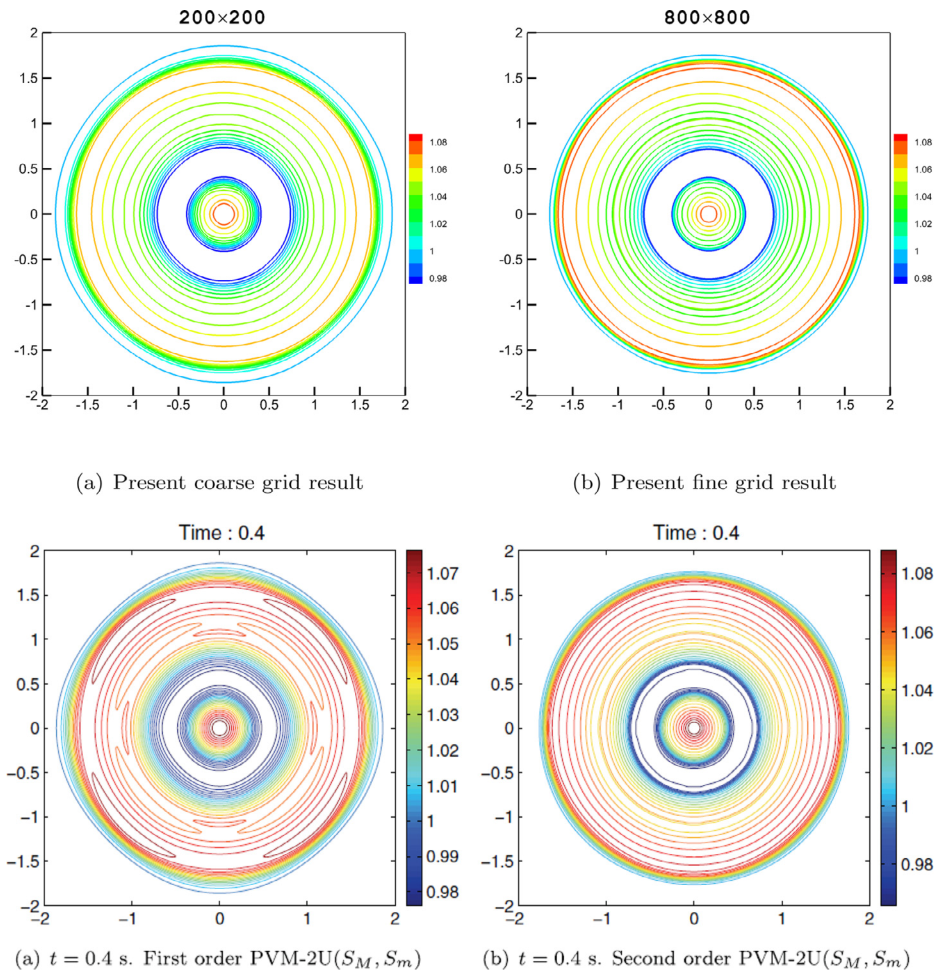
(g) $\eta = h + b$ at $t = 0.6$

(h) $\eta = h + b$ at $t = 0.6$ [40]

(i) $\eta = h + b$ at $t = 15$

(j) $\eta = h + b$ at $t = 15$ without modification

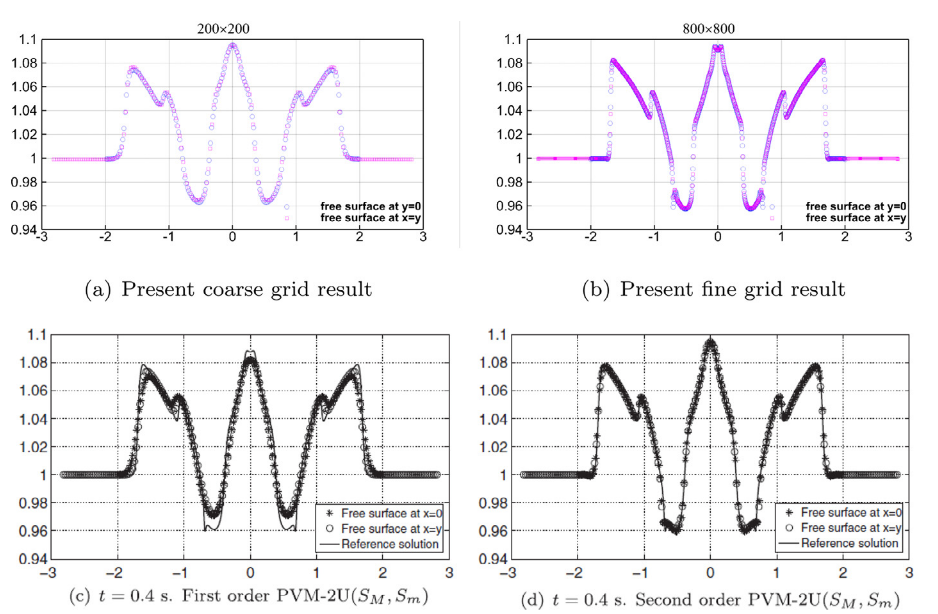
Fig. 16. Perturbation of a steady state at rest($\tilde{h} = 10^{-2}$). Solution of $h + b$ computed with 600×300 mesh cells. Shallow water solutions at $t = 0.48$ and $t = 0.6$ from [40] are also shown in (f) and (h). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Present coarse grid result (b) Present fine grid result

(a) $t = 0.4$ s. First order PVM-2U(S_M, S_m) (b) $t = 0.4$ s. Second order PVM-2U(S_M, S_m)

Fig. 17. The present counter lines of $\eta = h + b$ with meshes 200 × 200 and 800 × 800 at $t = 0.4$ in the upper two frames. The lower two frames are from [16], where the lower left is computed with the first order PVM and the right is computed with the second-order PVM schemes using 200 × 200 cells. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Present coarse grid result (b) Present fine grid result

(c) $t = 0.4$ s. First order PVM-2U(S_M, S_m) (d) $t = 0.4$ s. Second order PVM-2U(S_M, S_m)

Fig. 18. Comparison of the profiles of $\eta = h + b$ across the center at $t = 0.4$ computed with meshes 200 × 200 and 800 × 800 in the present results (upper two frames) with those computed in [16] (lower two frames). The reference solution in [16] was obtained using first order PVM method on 800 × 800 meshes.

larger perturbation without the modification, indicating that our NOC scheme can keep the well-balanced property very well.

5.7. A 2D circular dam break problem

Now we simulate the 2D example of [16] by using scheme (25) together with the well-balanced correction (45). Consider a $[-2,2]\times[-2,2]$ square domain. The bottom function is given by $b(x,y)=0.5e^{-\frac{1}{2}(x^2+y^2)}$. As initial conditions we set $(u_s,v_s)=(u_f,v_f)=(0,0)$ and

$$h(x,y,0)=\begin{cases}1.5-b(x,y) & \text{if}\sqrt{x^2+y^2}\le 0.5,\\ 1-b(x,y) & \text{otherwise},\end{cases}$$

$$\phi(x,y,0)=\begin{cases}0.1 & \text{if}\sqrt{x^2+y^2}\le 0.5,\\ 0.9 & \text{otherwise}.\end{cases} \tag{66}$$

The gravity constant $g=9.8$, CFL=0.2, and the computational time is $t=0.4$. Wall boundary conditions are set: $\mathbf{u}_s\cdot\mathbf{n}=\mathbf{u}_f\cdot\mathbf{n}=0$, where $\mathbf{n}$ is the unit normal vector to the boundary. The upper two frames in Fig. 17 display the counter line of $\eta=h+b$ computed with meshes of 200 × 200 and 800 × 800, respectively. The fine 800 × 800 meshes give a sharper front than the coarse 200 × 200 meshes. The lower two frames are taken from [16], which were computed with the first order and second order PVM (polynomial viscosity matrix) schemes using the 200 × 200 mesh. We can see that our contours on the 200 × 200 mesh (upper left) are very similar to the second-order PVM result (lower right), but the first order PVM result totally did not keep the radial symmetry of the solution (lower left). Fig. 18 shows comparison of two profiles across the center of the domain. From the lower two frames, we see that the 1st-order PVM result on the 200 × 200 mesh has smaller height at $x=0$ compared with the reference solution computed with the 1st-order PVM on 800 × 800 mesh, while the 2nd-order PVM result on the 200 × 200 mesh has a little larger height compared to the same reference solution. Our results on the 200 × 200 mesh (upper left frame) show a trend similar to the 2nd-order PVM (lower right frame). It is seen that there is some difference between profiles across $y=0$ and $y=x$ for our results on the 200 × 200 mesh, but this difference becomes small on the 800 × 800 mesh, and the results on the fine mesh are close to the reference solution of PVM.

6. Conclusions

In this article, we have applied the staggered NOC scheme to numerical solution of a two-phase shallow granular flow model. The NOC scheme does not need to solve a Riemann problem. We propose a simple and oscillation-free discretization scheme for the nonconservative terms. To obtain well-balanced property, we give a correction procedure which uses additionally evolved total surface level to correct the solid and fluid heights evolved from the NOC scheme. A predictor/corrector strategy by using a numerical inter-phase drag relaxation term is adopted to enforce hyberbolicity. The resulting numerical method is implemented on multiple GPUs using both OpenMP-CUDA and multistream-CUDA strategies to compare their performance. Numerical tests in both 1D and 2D problems demonstrate that the present NOC scheme is able to handle a wide range of flow conditions involving shock, dry bed region and vacuum formation. The calculated results are in good agreement with literature. The computational efficiencies of OpenMP-CUDA and multistream-CUDA strategies differ a little and depend on specific grid sizes used. It does not matter to use either strategy, but it is critical to optimize the performance of single GPU computing, which will be our future work.

Acknowledgments

The authors thank the support of state key program for developing basic sciences (2010CB731505), Natural Science Foundation of China (11321061, 11261160486), and Fundamental Research of Civil Aircraft (MJ-F-2012-04).

References

[1] Denlinger RP, Iverson RM. Granular avalanches across irregular three-dimensional terrain: 1. Theory and computation. J Geophys Res 2004;109(B1):F01014. doi:10.1029/2003JF000085.
[2] Savage SB, Hutter K. The motion of a finite mass of granular material down a rough incline. J Fluid Mech 1989;199(1):177–215. doi:10.1017/S0022112089000340.
[3] Gray JMNT, Wieland M, Hutter K. Gravity driven free surface flow of granular avalanches over complex basal topography. Proc R Soc London, Ser A 1999;455:1841–74. doi:10.1098/rspa.1999.0383.
[4] Wieland M, Gray JMNT, Hutter K. Channelised free surface flow of cohesionless granular avalanche in a chute with shallow lateral curvature. J Fluid Mech 1999;392:73–100. doi:10.1017/S0022112099005467.
[5] Wang YQ, Hutter K, Pudasaini SP. The Savage–Hutter theory: a system of partial differential equations for avalanche flows of snow, debris, and mud. ZAMM—J Appl Math Mech/Z Angew Math Mech 2004;84(8):507–27. doi:10.1002/zamm.200310123.
[6] McDougall S, Hungr O. A model for the analysis of rapid landslide motion across three-dimensional terrain. Can Geotech J 2004;42:1084–97. doi:10.1139/t95-063.
[7] Iverson RM, Denlinger RP. Flow of variably fluidized granular masses across three-dimensional terrain: 1. Coulomb mixture theory. J Geophys Res 2001a;106(B1):537–52. doi:10.1029/2000JB900329.
[8] Denlinger RP, Iverson RM. Flow of variably fluidized granular masses across three-dimensional terrain: 2. Numerical predictions and experimental tests. J Geophys Res 2001b;106(B1):553–66. doi:10.1029/2000JB900330.
[9] Pitman E, Le L. A two-fluid model for avalanche and debris flows. Philos Trans R Soc A: Math Phys Eng Sci 2005;363(1832):1573–601.
[10] Pelanti M, Bouchut F, Mangeney A. A Roe-type scheme for two-phase shallow granular flows over variable topography. ESAIM: Math Model Numer Anal 2008;42(05):851–85.
[11] Pelanti M, Bouchut F, Mangeney A. Riemann solver for single-phase and two-phase shallow flow models based on relaxation. Relations with Roe and VFRoe solvers. J Comput Phys 2011;230:515–50.
[12] Qamar S, Zia S, Ashraf W. The space time CE/SE method for solving single and two-phase shallow flow models. Comput Fluids 2014;96:136–51.
[13] Le LH. New models for geophysical mass flows. State University of New York at Buffalo; 2006. Ph.D. thesis.
[14] Dumbser M, Castro M, Parés C, Toro EF. ADER schemes on unstructured meshes for nonconservative hyperbolic systems: applications to geophysical flows. Comput Fluids 2009;38:1731–48. doi:10.1016/j.compfluid.2009.03.008.
[15] Dumbser M, Balsara DS. A new efficient formulation of the HLLEM Riemann solver for general conservative and non-conservative hyperbolic systems. J Comput Phys 2016;304:275–319.
[16] Castro-Díaz MJ, Fernández-Nieto ED. A class of computationally fast first order finite volume solvers: PVM methods. SIAM J Sci Comput 2012;34(4):A2173–96. doi:10.1137/100795280.
[17] Castro M.J., Fernández-Nieto E.D., González-Vida J.M.. IFCP Riemann solver for the Pitman–Le system. Series in contemporary applied mathematics: hyperbolic problems2012; Chapter 29, p. 328–336. doi: 10.1142/9789814417099_0029.
[18] Zia S, Qamar S. A kinetic flux-vector splitting method for single-phase and two-phase shallow flows. Comput Math Appl 2014;67(6):1271–88. doi:10.1016/j.camwa.2014.01.015.
[19] Nessyahu H, Tadmor E. Nonoscillatory central differencing for hyperbolic conservation laws. J Comput Phys 1990;87:408–63.
[20] Kurganov A, Noelle S, Petrova G. Semi-discrete central-upwind scheme for hyperbolic conservation laws and Hamilton–Jacobi equations. SIAM J Sci Comput 2001;23:707–40.
[21] Kurganov A, Lin C-T. On the reduction of numerical dissipation in central-upwind schemes. Commun Comput Phys 2007;2:141–63.
[22] Jiang G-S, Levy D, Lin C-T, Osher S, Tadmor E. High-resolution nonoscillatory central schemes with nonstaggered grids for hyperbolic conservation laws. SIAM J Numer Anal 1998;35:2147–68.
[23] Tai Y-C, Noelle S, Gray JMNT, Hutter K. Shock-capturing and front-tracking methods for granular avalanches. J Comput Phys 2002;175(1):269–301.
[24] Cui X. Computational and experimental studies of rapid free-surface granular flows around obstacles. Comput Fluids 2014;89(0):179–90. doi:10.1016/j.compfluid.2013.10.036.
[25] Kurganov A, Miller J. Central-upwind scheme for Savage–Hutter type model of submarine landslides and generated tsunami waves. Comput Methods Appl Math 2014:1–25. doi:10.1515/cmam-2014-0003.ahead print
[26] Brodtkorb AR, Hagen TR, Satra ML. Graphics processing unit (GPU) programming strategies and trends in GPU computing. J Parallel Distrib Comput 2013;73(1):4–13. doi:10.1016/j.jpdc.2012.04.003.

[27] Satra ML, Brodtkorb AR. Shallow water simulations on multiple GPUs. In: Jonasson K, editor. PARA 2010, Part II, LNCS 7134; 2012. p. 56–66.
[28] Vinas M, Lobeiras J, Fraguela BB, Arenaz M, Amor M, Garcia JA, et al. A multi-GPU shallow water simulation with transport of contaminants. Concurrency Comput: Pract Exp 2013;25(8):1153–69.
[29] Bermúdez A, Vázquez-Cendóon ME. Upwind methods for hyperbolic conservation laws with source terms. Comput Fluids 1994;23:1049–71.
[30] Adusse E, Bochut F, Bristeau M, Klein R, Perthame B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. SIAM J Sci Comput 2004;25:2050–65.
[31] Castro-Díaz MJ, Fernández-Nieto ED, Gonzáez-Vida JM, Parés-Madroñal C. Numerical treatment of the loss of hyperbolicity of the two-layer shallow-water system. J Sci Comput 2011;48:16–40. doi:10.1007/s10915-010-9427-5.
[32] Tai Y-C. Dynamics of granular avalanches and their simulations with shock–capturing and front-tracking numerical schemes. Darmstadt University of Technology; 2000. Ph.D. Thesis.
[33] Rogers B, Borthwick A, Taylor P. Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver. J Comput Phys 2003;192:422–51.
[34] Zhang S, Chu YL. GPU high performance computing with CUDA. Chinese Hydrology Publishing House; 2009.[in Chinese]
[35] Sanders J, Kandrot E. CUDA by example: an introduction to general-purpose GPU programming. 1st. Addison-Wesley Professional; 2010.
[36] Zhai J, Yuan L, Liu W, Zhang XT. Solving the Savage–Hutter equations for granular avalanche flows with a second-order Godunov type method on GPU. Int J Numer Methods Fluids 2015;77(7):381–99. doi:10.1002/fld.3988.
[37] NVIDIA Corporation. NVIDIA CUDA C programming guide, version 6.0; 2014.
[38] NVIDIA Corporation. CUDA API reference manual, version 4.2, 2012, http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_Toolkit_Reference_Manual.pdf.
[39] LeVeque RJ. Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. J Comput Phys 1998;146(1):346–65.
[40] Xing Y, Shu C-W. High order finite difference WENO schemes with the exact conservation property for the shallow water equations. J Comput Phys 2005;208(1):206–27.