A graphics processing unit-based robust numerical model for solute transport driven by torrential flow condition[*]

Jing-ming HOU[1], Bao-shan SHI[†‡1], Qiu-hua LIANG[2], Yu TONG[1], Yong-de KANG[1],

Zhao-an ZHANG[1], Gang-gang BAI[1], Xu-jun GAO[3], Xiao YANG[3]

[1]*State Key Laboratory of Eco-hydraulics in Northwest Arid Region of China, Xi'an University of Technology, Xi'an 710048, China*

[2]*School of Architecture, Building and Civil Engineering, Loughborough University, Loughborough LE11 3TU, UK*

[3]*Northwest Engineering Corporation Limited, Xi'an 710065, China*

[†]E-mail: 2180421229@stu.xaut.edu.cn

Abstract: Solute transport simulations are important in water pollution events. This paper introduces a finite volume Godunov-type model for solving a 4×4 matrix form of the hyperbolic conservation laws consisting of 2D shallow water equations and transport equations. The model adopts the Harten-Lax-van Leer-contact (HLLC)-approximate Riemann solution to calculate the cell interface fluxes. It can deal well with the changes in the dry and wet interfaces in an actual complex terrain, and it has a strong shock-wave capturing ability. Using monotonic upstream-centred scheme for conservation laws (MUSCL) linear reconstruction with finite slope and the Runge-Kutta time integration method can achieve second-order accuracy. At the same time, the introduction of graphics processing unit (GPU)-accelerated computing technology greatly increases the computing speed. The model is validated against multiple benchmarks, and the results are in good agreement with analytical solutions and other published numerical predictions. The third test case uses the GPU and central processing unit (CPU) calculation models which take 3.865 s and 13.865 s, respectively, indicating that the GPU calculation model can increase the calculation speed by 3.6 times. In the fourth test case, comparing the numerical model calculated by GPU with the traditional numerical model calculated by CPU, the calculation efficiencies of the numerical model calculated by GPU under different resolution grids are 9.8–44.6 times higher than those by CPU. Therefore, it has better potential than previous models for large-scale simulation of solute transport in water pollution incidents. It can provide a reliable theoretical basis and strong data support in the rapid assessment and early warning of water pollution accidents.

Key words: Solute transport; Shallow water equations; Godunov-type scheme; Harten-Lax-van Leer-contact (HLLC) Riemann solver; Graphics processing unit (GPU) acceleration technology; Torrential flow

https://doi.org/10.1631/jzus.A2000585   **CLC number:** TV131.2

1 Introduction

In recent years, many flash floods caused by severe rainstorms have occurred. For instance, in

March 2019, during a period of heavy rain, a severe storm occurred in Papua Province in Indonesia, and the massive flood claimed 63 lives. In July 2012, an unpredictable rainstorm killed 79 people in Beijing, China. Flash floods caused by such heavy rains occur frequently around the world. Because of the acuteness and unpredictability of heavy rains, they bring tremendous damage to human life and economy. The impact of flash floods on cities is extremely important (Barredo, 2007; Roccati et al., 2019; Bayazıt et al., 2021). In cities, such flood events can destroy

chemical plants and wastewater treatment plants. Such an event may threaten the water safety of urban and rural residents and affect the water quality of downstream rivers and lakes. Understanding the contaminant transport process is thus of fundamental and practical importance in hydraulic and environmental engineering (Wang and Cheng, 2002; Zhao et al., 2007; Zhang et al., 2010; Duarte et al., 2013). Under these circumstances, a strong and efficient model is needed to predict the track of point-source pollutants transported by flood-driven water.

Water pollution incidents have occurred frequently, causing the deterioration of water quality in rivers and lakes, leading to serious eutrophication, and threatening the survival of fish (Tao et al., 2012; Li et al., 2018; Chen et al., 2021). Because of frequent pollution incidents, mastering the process of pollutant transport in water bodies plays an important role in water conservancy projects and environmental and ecological engineering (Kachiashvili et al., 2007; Liu, 2011; Zhang, 2011; Bi et al., 2013; Tang et al., 2015). Through the establishment of high-efficiency and high-precision numerical models, the hazard range and duration of water pollution accidents can be quickly determined, and then effective measures can be taken to reduce the harm caused by the accident. The surface hydrodynamic process is mathematically described by a 2D shallow water equation, and a shallow water solver is used to establish an efficient and high-precision numerical model to study its motion (Audusse and Bristeau, 2003; Murillo et al., 2009; Hou et al., 2018a; Dong, 2020). However, the Boltzmann equation can also be used to establish numerical models for many fluid mechanics problems (Aidun and Clausen, 2010; La Rocca et al., 2015; Liu et al., 2020; Venturi et al., 2021). The advantage of this method is that it does not require the solution of large-scale nonlinear partial differential equations, which considerably simplifies the calculations. La Rocca et al. (2020), using discrete Boltzmann equations, developed a method for simulating transcritical shallow water flow, and verified the validity of the model through numerical and experimental results. Many shallow-water solute transport models have been reported in recent years and most of these adopt the Goundnov-type scheme to establish a robust numerical model. For example, Begnudelli and Sanders (2006) proposed a high-resolution unstructured grid finite volume algorithm for simulating 2D unsteady water flow and material transport, and analyzed the effects of different slope limiters on its calculation accuracy and efficiency. Benkhaldoun et al. (2007) established a self-adaptive mesh-based pollutant transport model with high computational accuracy, but the scheme did not fully satisfy total variation diminishing (TVD) characteristics. Kong et al. (2013), based on a high-precision finite volume model that solves the transport motion equation on an unstructured grid, used the operator-splitting method to calculate the material convection flux and introduced a gradient limiting factor to ensure the TVD characteristics of the calculation format and to reduce the numerical oscillation caused by the large concentration gradient of the material. Petti and Bosa (2007) developed an accurate 2D Eulerian numerical method for the prediction of pollutant transportation in a natural water body. Their model uses a non-uniform grid to discretize the computational domain, which may pose challenges to large-scale calculations. Thus, it is not easy to design an accurate and efficient numerical model for solving shallow water and solute transport equations. At the same time, the problems of a large amount of calculations and a low calculation efficiency must be faced (Song et al., 2014). Efficient and accurate numerical modelling thus provides a significant tool for environmental impact management and water project design for water pollution incidents.

In this study, the authors aim to resolve the shallow flow-driven solute transport problem by considering all the above challenging issues and presenting a graphics processing unit (GPU)-accelerated model for dry-bed simulations over a complex domain. The model solves a 4×4 matrix form of the hyperbolic conservation laws consisting of 2D shallow water and solute transport equations. The monotonic upstream-centred scheme for conservation laws (MUSCL) format, with second-order spatio-temporal precision, was used to ensure calculation accuracy and avoid numerical oscillation (Hou et al., 2014). Second-order spatial reconstruction and non-negative depth reconstruction methods improve the accuracy of the model and meet the TVD characteristics (Hou et al., 2013a). The accuracy and efficiency of the model are tested by several cases. The numerical format provides a well-balanced solution and maintains non-negative water depth and solute concentration in applications involving wet and dry complex areas. The model ensures computational

accuracy while increasing computational efficiency. Therefore, it has better potential than previous models for large-scale simulation of problems in the real world.

2 Governing equations and numerical methods

2.1 Shallow water equation and transport equation

Shallow flow hydrodynamics and passive solute concentration transport are described by the shallow water equation and the depth-averaged advection-diffusion equation, respectively. In vector form, the shallow water and advection-diffusion equations may be written as a 4×4 matrix:

$$\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = \boldsymbol{S}, \tag{1}$$

where t is the time; x and y are the Cartesian coordinates; $\boldsymbol{q}$ is the flow variable vector; $\boldsymbol{F}$ and $\boldsymbol{G}$ are the flux vectors in the x and y directions, respectively; $\boldsymbol{S}$ denotes the vector of the source terms. The vectors may be given as follows:

$$\boldsymbol{q} = \begin{bmatrix} h \\ q_x \\ q_y \\ hC \end{bmatrix}, \tag{2a}$$

$$\boldsymbol{F} = \begin{bmatrix} q_x \\ uq_x + g\,h^2/2 \\ uq_y \\ uhC \end{bmatrix}, \tag{2b}$$

$$\boldsymbol{G} = \begin{bmatrix} q_y \\ vq_x \\ vq_y + g\,h^2/2 \\ vhC \end{bmatrix}, \tag{2c}$$

$$\boldsymbol{S} = \begin{bmatrix} 0 \\ -gh\partial z_{\mathrm{b}}/\partial x - C_{\mathrm{f}} u \sqrt{u^2 + v^2} \\ -gh\partial z_{\mathrm{b}}/\partial y - C_{\mathrm{f}} v \sqrt{u^2 + v^2} \\ \frac{\partial}{\partial x}\left(D_x h \frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(D_y h \frac{\partial C}{\partial y}\right) + q_{\mathrm{in}} C_{\mathrm{in}} \end{bmatrix}, \tag{2d}$$

where h is the water level; u and v are the depth-averaged velocity components in the x and y directions, respectively; C represents the solute concentration; $q_x = uh$ and $q_y = vh$ are the unit-width discharge in the two Cartesian directions; hC is the solute volume per unit area; g represents the acceleration due to gravity; z_{b} is the bed elevation; C_{f} is the bed roughness coefficient determined by the Manning coefficients n and h, which has the form $gn^2/h^{1/3}$; D_x and D_y are the x and y components of the solute diffusion coefficient D, respectively; q_{in} is the flow intensity of the point-source solute; C_{in} is the average concentration of the solute perpendicular to the point-source. In addition, the water level η, which equals $h+z_{\mathrm{b}}$, is also used in this study.

2.2 Numerical method

2.2.1 Finite volume model

Using the cell-centred finite volume method, the integral form of Eq. (1) over a control cell is written as

$$\int_\Omega \frac{\partial \boldsymbol{q}}{\partial t}\mathrm{d}\Omega + \int_\Omega \left(\frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y}\right)\mathrm{d}\Omega = \int_\Omega \boldsymbol{S}\mathrm{d}\Omega, \tag{3}$$

where Ω is the volume of the control cell. Applying the divergence theorem, Eq. (3) becomes

$$\int_\Omega \frac{\partial \boldsymbol{q}}{\partial t}\mathrm{d}\Omega + \oint_\Gamma \boldsymbol{F}(\boldsymbol{q})\cdot\boldsymbol{n}\,\mathrm{d}\Gamma = \int_\Omega \boldsymbol{S}(\boldsymbol{q})\mathrm{d}\Omega, \tag{4}$$

where Γ is the boundary of the grid; $\boldsymbol{n}$ is the normal vector outside the unit perpendicular to the boundary of the grid and is defined by $[n_x, n_y]^{\mathrm{T}}$; $\boldsymbol{F}(\boldsymbol{q})\cdot\boldsymbol{n}$ is the flux at the grid boundary and is expressed as

$$\boldsymbol{F}(\boldsymbol{q})\cdot\boldsymbol{n} = \boldsymbol{F}n_x + \boldsymbol{G}n_y = \begin{bmatrix} q_x n_x + q_y n_y \\ \left(uq_x + gh^2/2\right)n_x + vq_x n_y \\ uq_y n_x + \left(vq_y + gh^2/2\right)n_y \\ uhC n_x + vhC n_y \end{bmatrix}. \tag{5}$$

The line integral for the $\boldsymbol{F}(\boldsymbol{q})\cdot\boldsymbol{n}$ total edges of the square grid is expressed by algebraic equation as

$$\oint_\Gamma \boldsymbol{F}(\boldsymbol{q}) \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = \sum_{k=1}^{n_b} \boldsymbol{F}_k(\boldsymbol{q}^n) \cdot \boldsymbol{n}_k l_k, \qquad (6)$$

where k is the index of a side of a cell; l_k is the side length of the kth side of the cell; n_b is the number of cell boundaries; $\boldsymbol{F}_k(\boldsymbol{q}^n)$ is the interface flux of the four interfaces of the unit grid–east, west, south, and north–including the mass flux, momentum flux, and solute flux; $\boldsymbol{n}_k$ is the outer normal direction of the four interfaces.

In Eq. (4), the first term on the time derivative uses the difference method. From time n to time $n+1$, the $\boldsymbol{q}$ value of the cell is updated to

$$\boldsymbol{q}_i^{n+1} = \boldsymbol{q}_i^n + \frac{\Delta t}{\Omega}\left(\int_\Omega \boldsymbol{S} \mathrm{d}\Omega - \sum_{k=1}^{n_b} \boldsymbol{F}_k(\boldsymbol{q}^n)\cdot\boldsymbol{n}_k l_k\right), \quad (7)$$

where $\boldsymbol{q}=[h, q_x, q_y, hC]^{\mathrm{T}}$ is a conservation variable, which includes water depth variables, flow variables, and solute variables; Δt is the calculation time step, which needs to meet the Courant-Friedrichs-Lewy (CFL) condition limit. The detailed calculation process is described in reference (Hou et al., 2013a); $\boldsymbol{F}$ is the hydrodynamic transport flux; $\boldsymbol{S}$ is the source term approximation, which includes the bottom slope source term and friction source term; for the specific processing method, please refer to Hou et al. (2013b). When updating the solute concentration, a minimum water depth should be set (10^{-6} m in this study). When the cell grid water depth is less than the minimum water depth, the solute concentration is 0.

To obtain a second-order numerical scheme with a time factor, the Runge-Kutta time integration method is employed, and the time-matching Eq. (7) may be rewritten as

$$\boldsymbol{q}_i^{n+1} = \frac{1}{2}\left[\left(\boldsymbol{q}_i^n + \boldsymbol{q}_i^{n*}\right) + K\left(\boldsymbol{q}_i^{n*}\right)\right], \qquad (8)$$

where $\boldsymbol{q}_i^{n*}$ is the intermediate flow variable:

$$\boldsymbol{q}_i^{n*} = \boldsymbol{q}_i^n + K\left(\boldsymbol{q}_i^n\right), \qquad (9)$$

and $K\left(\boldsymbol{q}_i^{n*}\right)$ is the Runge-Kutta coefficient, defined as

$$K\left(\boldsymbol{q}_i^{n*}\right) = \frac{\Delta t^n}{\Omega}\left[\int_\Omega \boldsymbol{S}(\boldsymbol{q}^n)\mathrm{d}\Omega - \sum_{k=1}^{n_b} \boldsymbol{F}_k(\boldsymbol{q}^n)\cdot\boldsymbol{n}_k l_k\right]. \qquad (10)$$

2.2.2 Flux calculation

The model uses the Harten-Lax-van Leer-contact (HLLC)-approximate Riemann solver to calculate the hydrodynamic force and transport flux, which has a strong ability to capture shock waves. It was implemented successfully in (Liang, 2010; Hou et al., 2013a, 2014, 2018b). This method can effectively mitigate situations of excessive numerical damping or severe numerical oscillation caused by the transport advection term. It can deal well with the changes in the dry and wet interfaces in an actual complex terrain. In this study, the coupled shallow water and advection-diffusion equations are considered, and hence, the HLLC solver should be used. Here, to calculate the grid edge fluxes, east $\boldsymbol{F}_k(\boldsymbol{q})\cdot\boldsymbol{n}_k$ can be obtained with this solver as (Kawahara and Umetsu, 1986)

$$\boldsymbol{F}_k(\boldsymbol{q})\cdot\boldsymbol{n}_k = \begin{cases} \boldsymbol{F}_{\mathrm{L}}, & 0 \le S_{\mathrm{L}}, \\ \boldsymbol{F}_{*\mathrm{L}}, & S_{\mathrm{L}} \le 0 \le S_{\mathrm{M}}, \\ \boldsymbol{F}_{*\mathrm{R}}, & S_{\mathrm{M}} \le 0 \le S_{\mathrm{R}}, \\ \boldsymbol{F}_{\mathrm{R}}, & S_{\mathrm{R}} \le 0, \end{cases} \qquad (11)$$

where $\boldsymbol{F}_{\mathrm{L}}=\boldsymbol{F}(\boldsymbol{q}_{\mathrm{L}})\cdot\boldsymbol{n}_k$ and $\boldsymbol{F}_{\mathrm{R}}=\boldsymbol{F}(\boldsymbol{q}_{\mathrm{R}})\cdot\boldsymbol{n}_k$ are calculated from the left and right Riemann states; S_{L}, S_{M}, and S_{R} are the left, middle, and right wave speeds, respectively; $\boldsymbol{F}_{*\mathrm{L}}$ and $\boldsymbol{F}_{*\mathrm{R}}$ are the fluxes in the left and right middle regions of the HLLC solution structure and are calculated as

$$\boldsymbol{F}_{*\mathrm{L}} = \begin{bmatrix} \boldsymbol{F}_{*1}(h) \\ \boldsymbol{F}_{*2}(u) \\ v_{\mathrm{L}}\boldsymbol{F}_{*1}(h) \\ C_{\mathrm{L}}\boldsymbol{F}_{*1}(h) \end{bmatrix}, \quad \boldsymbol{F}_{*\mathrm{R}} = \begin{bmatrix} \boldsymbol{F}_{*1}(h) \\ \boldsymbol{F}_{*2}(u) \\ v_{\mathrm{R}}\boldsymbol{F}_{*1}(h) \\ C_{\mathrm{R}}\boldsymbol{F}_{*1}(h) \end{bmatrix}, \qquad (12)$$

where v_{L} and v_{R} are the tangential velocities, and C_{L} and C_{R} are the substance concentrations. $\boldsymbol{F}_{*\mathrm{L}}$ and $\boldsymbol{F}_{*\mathrm{R}}$ are calculated from the Harten-Lax-van Leer (HLL) formula (Harten et al., 1983):

$$\boldsymbol{F}_* = \frac{S_{\mathrm{R}}\boldsymbol{F}_{\mathrm{L}} - S_{\mathrm{L}}\boldsymbol{F}_{\mathrm{R}} + S_{\mathrm{R}}S_{\mathrm{L}}\left(\boldsymbol{q}_{\mathrm{R}} - \boldsymbol{q}_{\mathrm{L}}\right)}{S_{\mathrm{R}} - S_{\mathrm{L}}}, \qquad (13)$$

where S_L, S_M, and S_R are the left, middle, and right wave speeds in the HLLC Riemann solution structure. Fraccarollo and Toro (1995) recommended the following formulae for estimating S_L, S_M, and S_R to facilitate applications in wetting and drying:

$$S_L = \begin{cases} u_R - 2\sqrt{gh_R}, & h_L = 0, \\ \min\left(u_L - \sqrt{gh_L}, u_* - \sqrt{gh_*}\right), & h_L > 0, \end{cases} \quad (14)$$

and

$$S_R = \begin{cases} u_L + 2\sqrt{gh_L}, & h_R = 0, \\ \max\left(u_R + \sqrt{gh_L}, u_* + \sqrt{gh_*}\right), & h_R > 0. \end{cases} \quad (15)$$

For the middle wave speed S_M, Toro (2001) suggested the following choice:

$$S_M = \frac{S_L h_R (u_R - S_R) - S_R h_L (u_L - S_L)}{h_R (u_R - S_R) - h_L (u_L - S_L)}. \quad (16)$$

In Eqs. (14)–(16), u_L, u_R, h_L, and h_R are the velocity and depth components of the left and right Riemann states, respectively, and u_* and h_* can be evaluated from (Toro, 2001)

$$u_* = 0.5(u_L + u_R) + \left(\sqrt{gh_L} - \sqrt{gh_R}\right), \quad (17)$$

and

$$h_* = 1/g\left[0.5(gh_L + gh_R) + 0.25(u_L - u_R)\right]^2. \quad (18)$$

In summary, based on the finite volume method in the Godunov format, the HLLC-approximate Riemann solver is used to obtain the coupled solution of the water flow and solute transport flux. In all the test cases considered in this study, two types of boundary conditions, i.e. an open boundary and a closed boundary, are used. The boundary conditions are controlled by flux calculations at the boundary (Hou et al., 2013b). The solute advection process simulation adopts the second-order explicit TVD scheme with limiter to control the numerical error, which reduces the numerical dissipation and numerical oscillation caused by the advection term of the transport equation; the implementation is detailed in our previous studies (Hou et al., 2012, 2015).

2.2.3 GPU-accelerated computing

The dynamic wave methods and Godunov-type schemes have been successfully applied in many studies, but the available computing power limits their application in many areas of high-resolution representation. The GPU parallel computing technique offers a new approach to such simulations. The GPU's highly parallel structure makes it more efficient than central processing units (CPUs) for algorithms. The compute unified device architecture (CUDA) programming language is used to implement the GPU parallel computing technique. The CUDA programming model uses the CPU as the host and the GPU as the coprocessor or device. The CPU is responsible for processing logical transactions and serial calculations. The GPU is responsible for performing parallel processing tasks. When the model starts to run calculations, data is first read into the host (main memory), and then the data of the grid information, initial boundary conditions, calculation parameters, etc. are initialized in each cell. The corresponding GPU device space is allocated to copy the data to the device (video memory). After the computation is completed, the result is synchronously copied back to the main memory. The implementation is detailed in the authors' previous studies (Smith and Liang, 2013; Liang et al., 2016; Hou et al., 2020). The successful and efficient implementation of the GPU computing technique is shown in Fig. 1. If the data exchange between CPU memory and GPU memory is too frequent, the data exchange delay will reduce program performance. Therefore, to avoid frequent data exchange in the output process of the results, the calculation time is extended. In the calculation of efficiency comparison in this study, the results will be output only after the simulation ends.

3 Results and discussion

The proposed numerical scheme for shallow flow-driven solute transport is validated against several benchmark tests and results and alternative numerical predictions, using different types of GPU computers for large-scale calculations, and
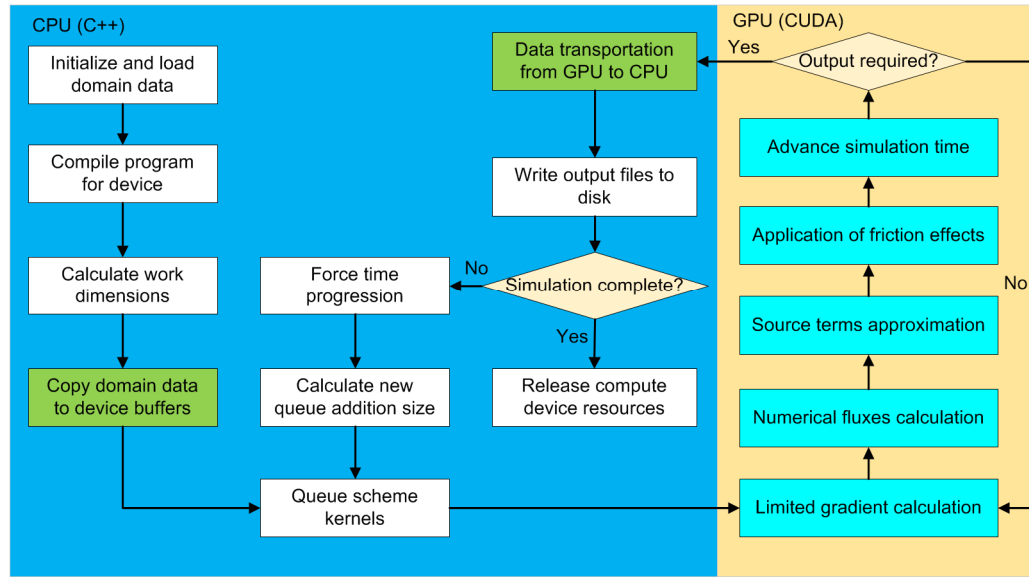
Fig. 1 Flowchart of the computation using GPUs

comparing the results with those of CPU calcula-tions. The Courant number for the CFL condition is set to 0.5, and g=9.81 m/s^2 and ρ=1000 kg/m^3. The solute is assumed to maintain the same density as water.

3.1 Pure solute advection verification

The first test case is characterized by advection of solute clouds in a stable uniform flow. The model is verified by advection in a uniform flow field using a square solute cloud. The assumed computing domain is an open channel with a slope of 10°, a length of 1000 m, and a width of 200 m. The computational domain covering the channel is divided into 1000×200 cells with a size of 1 m×1 m. The left and right boundaries are the inflow and free outflow conditions. The north and south boundaries are set to closed. In the whole domain, a constant flow field is achieved by setting the hydrodynamic parameters. That is, a depth of 0.1 m is imposed in the computa-tional domain. The 40 m×40 m solute cloud is placed in the channel with a central coordinate x=100 m. The value of the concentration for the solute cloud is as-sumed to be 3 mg/L. If the transport solute performs only advection motion on the water flow, the advec-tion motion control equation is given as

$$\frac{\partial(hC)}{\partial t}+\frac{\partial(huC)}{\partial x}+\frac{\partial(hvC)}{\partial y}=0. \qquad (19)$$

These conditions are also shown in Fig. 2a. A simulation time step of 1 s and a total time of 2000 s are adopted in this case. The contour and position distributions of the simulated solute cloud transport results at t=600 s and t=1000 s are shown in Figs. 2b and 2c, respectively. Theoretically, due to the neglect of the diffusion term, the solute cloud maintains its initial profile and concentration at all times in the flow field. However, numerical dissipation is inevi-table. Although numerical dissipation still exists, Fig. 2 shows that the dissipation is very small and that the concentration field deformation is relatively small. Therefore, the advection term processing method used in this model has a high accuracy and can achieve a high accuracy in practical applications.

3.2 Simulation of solute diffusion under hydro-static conditions

3.2.1 Setting the initial conditions of the model

The characteristic of this test case is the diffusion of solute clouds under hydrostatic conditions. The harmony of the model and the appropriateness of the solute diffusion term are verified. With h=1 m, u=0, and v=0, the 2D diffusion equation for this idealised case is given as

$$\frac{\partial(hC)}{\partial t}=\frac{\partial}{\partial x}\left(D_xh\frac{\partial C}{\partial x}\right)+\frac{\partial}{\partial y}\left(D_yh\frac{\partial C}{\partial y}\right). \qquad (20)$$
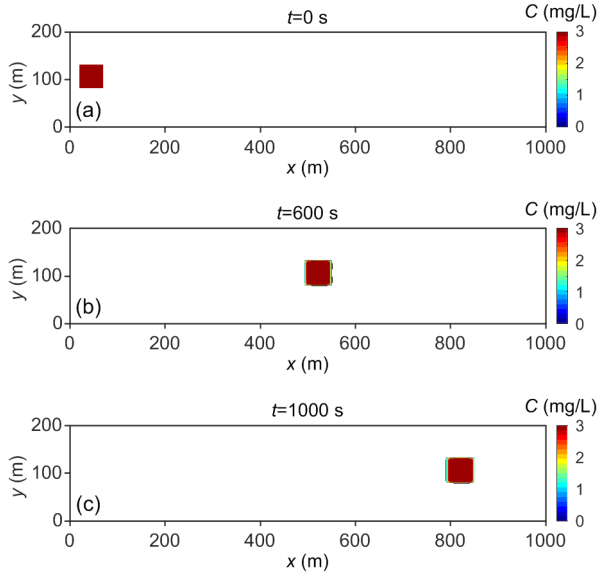
Fig. 2 Initial computational domain (a), and the solute cloud locations with a constant concentration profile at *t*=600 s (b) and *t*=1000 s (c)

This test case is similar to the transport problem with Gaussian distribution of concentration peaks (Shao et al., 2012). The length and width of the calculation domain are 75 m and 30 m, respectively. The computational domain is divided into 150×60 cells with a size of 0.5 m×0.5 m and a frictionless domain with a horizontal bed. The simulation is calculated with the time step Δ*t*=1 s, and the total calculation time is 300 s. In the whole calculation process, the four boundaries are closed boundaries, with the initial condition of concentration given as

$$C(x, y, 0) = \exp\left(-\frac{(x-37.5)^2}{2\sigma_0^2}\right), \qquad (21)$$

where σ_0 is the standard deviation of the Gaussian distribution and is set to 1 m.

The analytical solution of the solute concentration distribution at any time is

$$C(x, y, t) = \frac{\sigma_0}{\sigma} \exp\left(-\frac{(x-x')^2}{2\sigma^2}\right), \qquad (22)$$

where $\sigma^2 = \sigma_0^2 + 2D_t$, $x' = 37.5 + \int_0^t u(\tau)\mathrm{d}\tau$, and the diffusion coefficient D_t in the x direction is set to 0 m²/s, 0.1 m²/s, 0.3 m²/s, and 0.5 m²/s.

To quantitatively analyze the simulation results, error method diagnostics are employed to measure the accuracy. The lower the ERR is, the closer the result is to the exact solution. C_i and C_{exact} are defined as the numerical and exact solutions, respectively, and ERR is defined as

$$\mathrm{ERR} = \frac{1}{k_\mathrm{c}} \sum_i^{k_\mathrm{c}} |C_i - C_{\mathrm{exact}}|, \qquad (23)$$

where i is the cell number, and k_c denotes the total number of cells.

3.2.2 Discussion of the simulation results and accuracy analysis

Fig. 3 shows the results at t=300 s. During the long calculation period, the calculated water level of the model does not change and remains at 1 m, which is consistent with the analytical solution; the flow velocity is always maintained at 0. Therefore, the model has good harmony, and no false flow occurs.
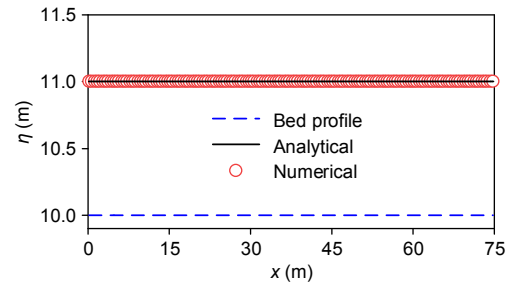

Fig. 3 Solute diffusion under hydrostatic conditions: comparison of the numerical and analytical water surface elevations along the x direction at the centre of the computed domain line at t=300 s

The concentration peak transport process is calculated for diffusion coefficients D of 0 m²/s, 0.1 m²/s, 0.3 m²/s, and 0.5 m²/s. A comparison of the numerical solution and the analytical solution at t=10 s is shown in Fig. 4. Fig. 4a shows the results of D=0 m²/s (the initial concentration distribution). The numerical method in this study can also obtain stable and reasonable results, and the calculated results before and after the solute cloud concentration peak are in good agreement with the analytical solution. Figs. 4b–4d show that as the diffusion coefficient increases, the concentration peak decreases, the distribution width increases, the concentration peak

becomes increasingly flat, and the numerical solution and analytical solution are better fitted. Table 1 shows that the error is small at different times under different diffusion conditions. It is clear that the numerical method adopted maintains the highest accuracy by holding the closest shape to the analytical result. This indicates that the model can effectively reduce numerical damping and has high precision and good stability.

3.3 Simulation of dam-break flow and solute transport with uniform concentration

Proposed by Kawahara and Umetsu (1986), this case is often used to check the calculation accuracy of the model and the effectiveness of the dynamic dry and wet boundary treatment. The calculation domain is a rectangular area of 75 m×30 m, and the bottom topography is calculated by

$$b(x, y) = \max \begin{bmatrix} 0 \\ 1 - 0.125\sqrt{(x-30)^2 + (x-6)^2} \\ 1 - 0.125\sqrt{(x-30)^2 + (x-24)^2} \\ 3 - 0.3\sqrt{(x-47.5)^2 + (x-15)^2} \end{bmatrix}.$$
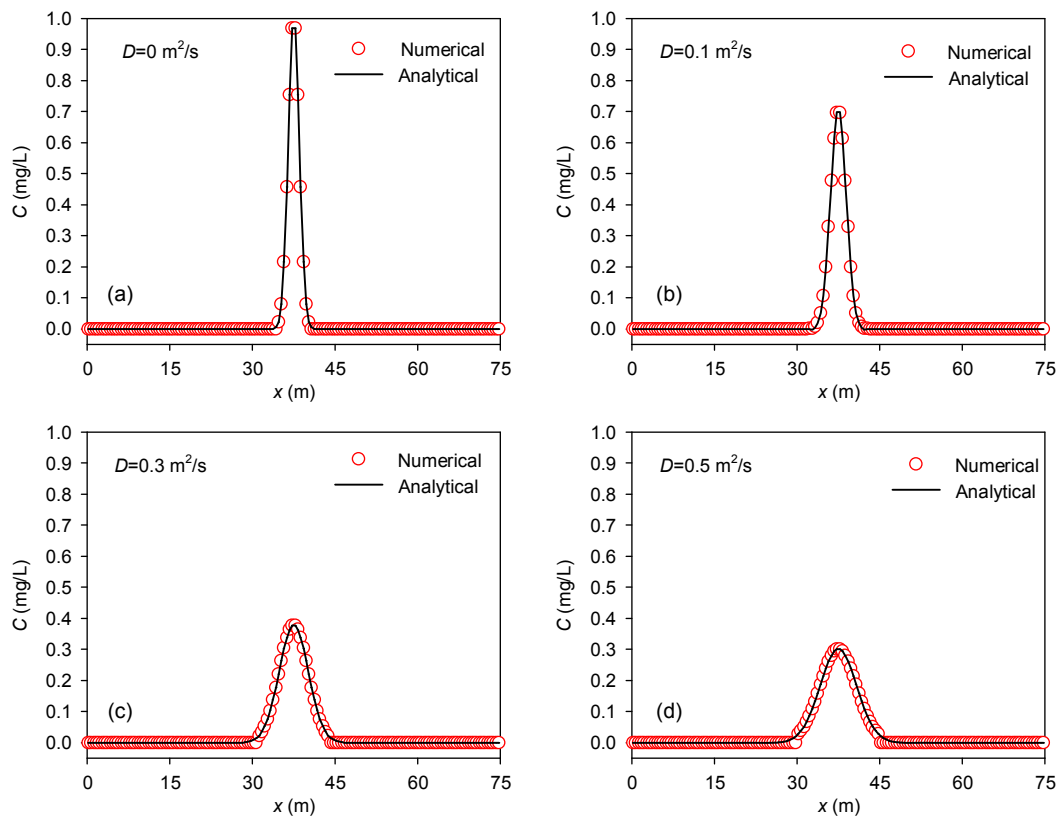
$$(24)$$

Fig. 4 Solute diffusion under hydrostatic conditions: the concentration curve along the *x* direction, comparing the numerical and analytical solutions
(a) $D=0$ m^2/s; (b) $D=0.1$ m^2/s; (c) $D=0.3$ m^2/s; (d) $D=0.5$ m^2/s

Table 1 ERRs of 2D diffusion equations

Diffusion coefficient	ERR (×10^{-3})						
	$t=0$ s	$t=5$ s	$t=10$ s	$t=15$ s	$t=20$ s	$t=25$ s	$t=30$ s
$D=0$ m^2/s	0.079	0.079	0.079	0.079	0.079	0.079	0.079
$D=0.1$ m^2/s	0.018	0.098	0.472	0.550	0.578	0.576	0.578
$D=0.3$ m^2/s	0.018	0.500	0.052	0.631	0.645	0.842	1.000
$D=0.5$ m^2/s	0.080	0.476	0.078	0.816	0.976	1.050	1.330

An infinitely thin dam is located at x=16 m, and divides the area into an upstream reservoir and a downstream dry flood area. The still water depth upstream of the dam is 1.75 m. It is contaminated by a solute mixture with a concentration of 1 mg/L. The Manning roughness coefficient n=0.018 s/m$^{1/3}$ is assumed throughout the domain. The computational domain is discretized into 150×60 regular square grids. The four side walls are closed boundaries. The movement of the solute does not consider material diffusion or degradation during transport. We assume that the dam is removed immediately at t=0, and that the total duration of the entire simulation calculation is 300 s.

Fig. 5 shows the results at t=2 s in terms of a 3D surface plot and contours for both water depth and solute distribution. After the dam break, the polluted water rushes into the downstream area, and at t=2 s, the flood front reaches two small humps and begins to climb. Theoretically, because diffusion is neglected, the concentration of the solute is always 1 mg/L in the wet region and 0 in the dry region. The simulation results are consistent with the facts, as shown in Figs. 5c and 5d. Therefore, the solute concentrations at the dry and wet edges are correctly simulated, and no obvious numerical diffusion is observed. Due to the large momentum generated by the dam break, the water continues to flow downstream rapidly. The

dam-break water infiltrates and passes through the two small humps after t=6 s, as shown in Fig. 6. After hitting the large hump, the water begins to climb up the large hump. Due to the blocking effect of the large hump, only part of the water can pass through the boundary wall near the southern and northern ends of the large hump and continue to flow downstream; this flow direction is shown in Fig. 6e. The leading edge of the solute concentration coincides with the dry-wet interface. A reliable and effective numerical model is established. As shown in Fig. 7, at t=12 s, the dam-break water flows downstream through the front portion of the large hump to reach the downstream end of the calculation area. Due to the interaction between the dam-break water and the hump, a shock wave in the opposite direction is generated, and the water flow propagates both downstream and to the upstream boundary, as shown in Fig. 7e. Fig. 8 (p.845) shows the results at t=30 s. It can be seen from the figure that the current reaches the eastern boundary. Since the dam-break wave collides with the east boundary to generate waves in the opposite direction, the water flow moves upstream, as shown in Fig. 8e. This wave-topography-boundary interaction continues until the momentum of the dam break is dissipated by the bed friction. Eventually, the flow becomes static once more, where the tops of the three humps are dry again. The Godunov-type scheme is
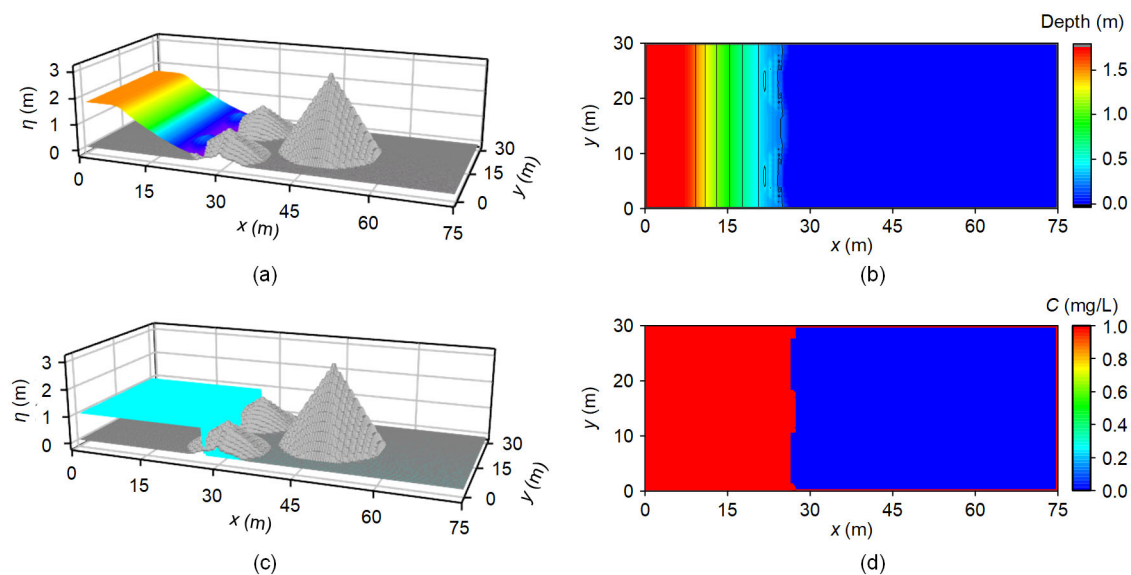

Fig. 5 Dam-break flood evolution: numerical results at t=2 s

(a) 3D plot of water surface; (b) Water depth contours; (c) 3D surface plot of solute concentration; (d) Contours of solute concentration
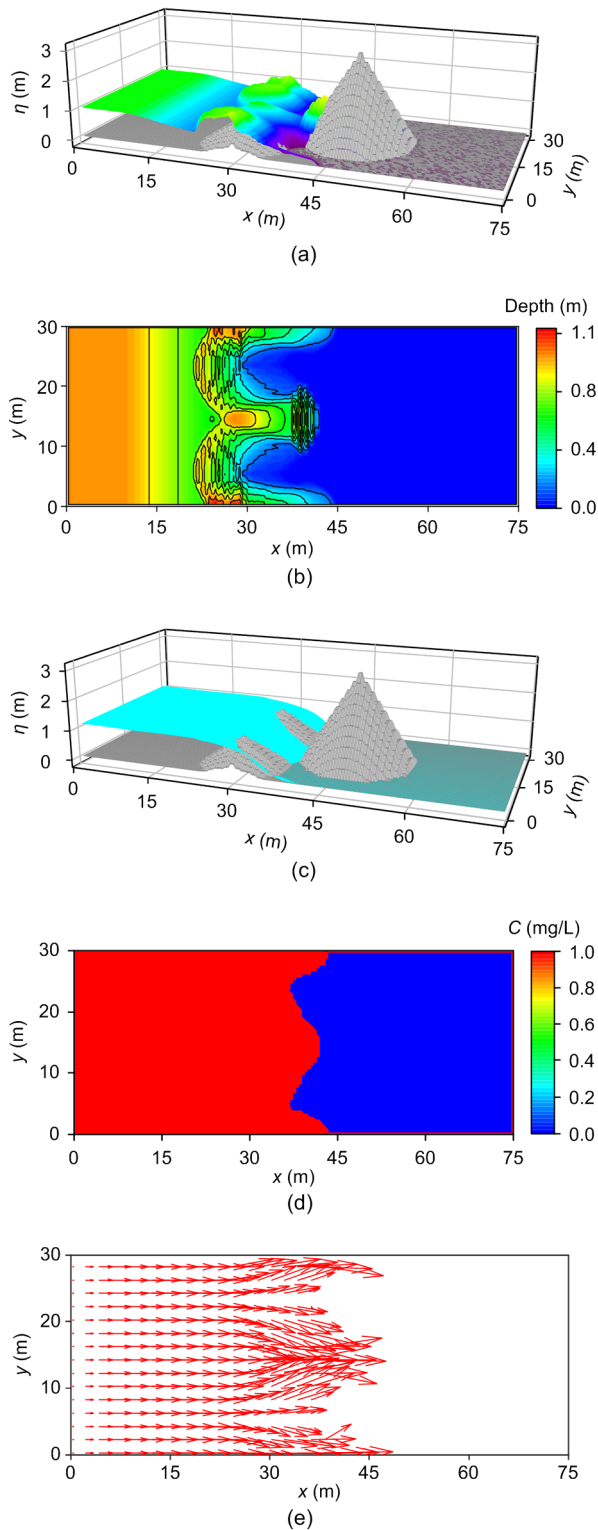
Fig. 6 Dam-break flood evolution: numerical results at *t*=6 s

(a) 3D plot of water surface; (b) Water depth contours; (c) 3D surface plot of solute concentration; (d) Contours of solute concentration; (e) Water flow velocity distribution
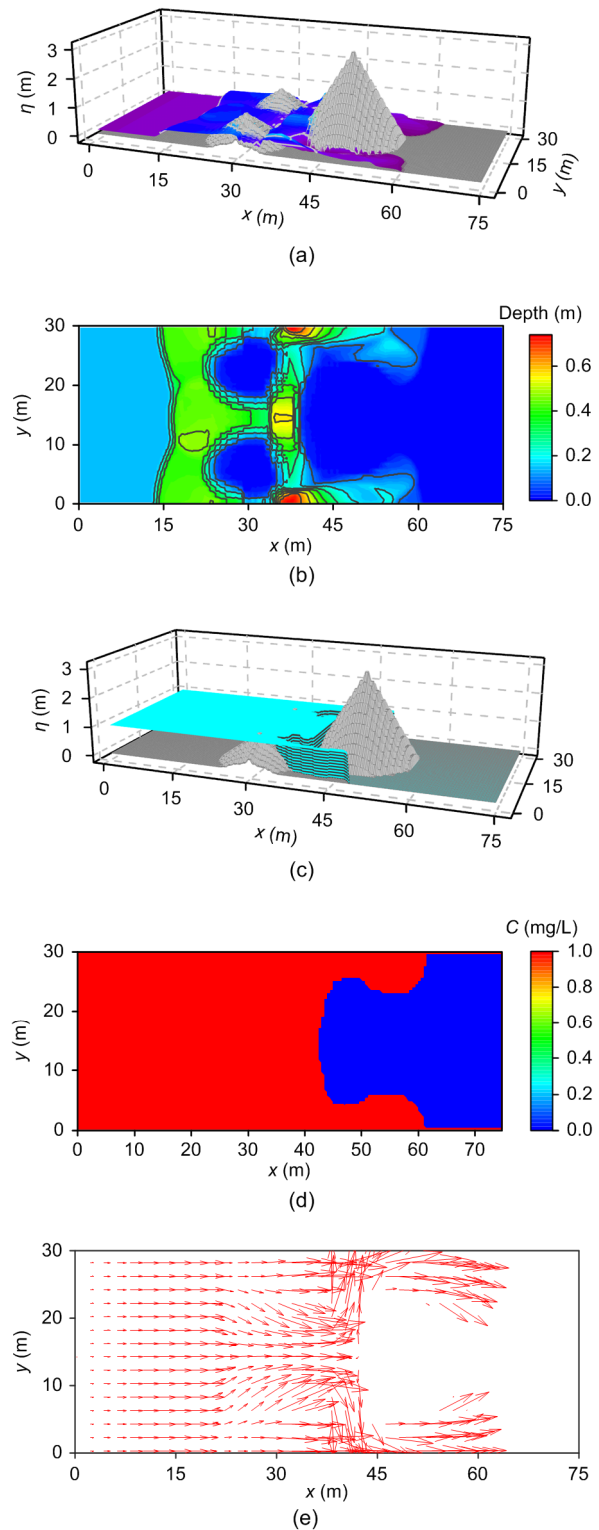
Fig. 7 Dam-break flood evolution: numerical results at *t*=12 s

(a) 3D plot of water surface; (b) Water depth contours; (c) 3D surface plot of solute concentration; (d) Contours of solute concentration; (e) Water flow velocity distribution
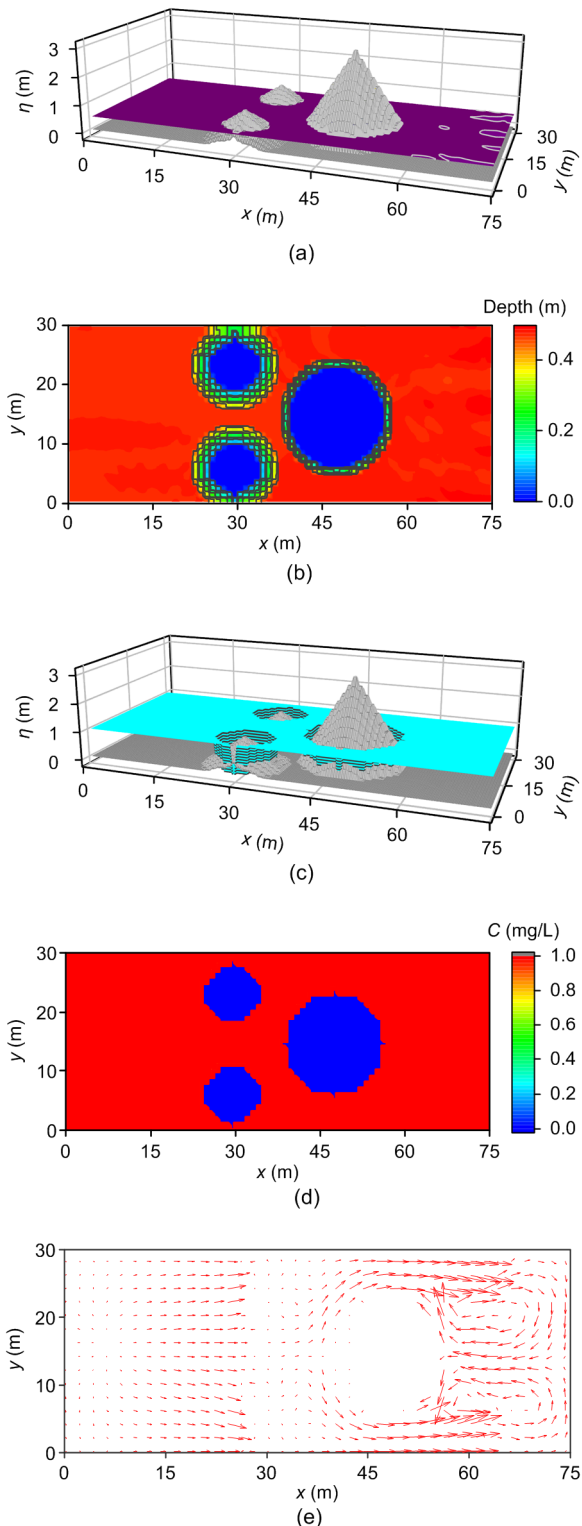
Fig. 8 Dam-break flood evolution: numerical results at
t=30 s
(a) 3D plot of water surface; (b) Water depth contours; (c) 3D
surface plot of solute concentration; (d) Contours of solute
concentration; (e) Water flow velocity distribution

observed to correctly capture the complex flow patterns induced by the dam-break wave (shock) interacting with the three humps and the domain boundaries. The numerical results are the same as those reported by Liang (2010) on a uniform grid and those reported on the adaptive grid by Zhang et al. (2015). For the GPU and CPU high-resolution numerical models on an NVIDIA Tesla M 1080 GPU computer, it took 3.875 s and 13.875 s, respectively, to complete the 300-s dam water flow evolution process on 9000 computing grids. Liang (2010) and Zhang et al. (2015) reported that the time required to complete the simulation on 16 000 uniform grids or adaptive grids was 277.3 s and 101 s, respectively. Therefore, for this more practical simulation calculation, the GPU's parallel computing efficiency is more obvious and it maintains a similar accuracy.

3.4 Point-source solute transport driven by the Malpasset dam break

The French Malpasset dam was built in 1956, had a maximum storage capacity of 5.5×10^7 m^3, and was mainly used for agricultural irrigation and urban water use. Due to continuous heavy rain at the end of November, 1959, the Malpasset dam water level rose rapidly in a short period of time. The dam finally broke at 21:14 on December 2, and the dam break was rapid. It can be considered a transient dam separating the Malpasset reservoir and the downstream terrain, as shown in Fig. 9. The dam-break water flows rapidly downstream and eventually enters the sea. The sea surface elevation is 0 m, the water level upstream of the dam before the dam break is 100 m, and the downstream water depth is 0 m. Hypothetical
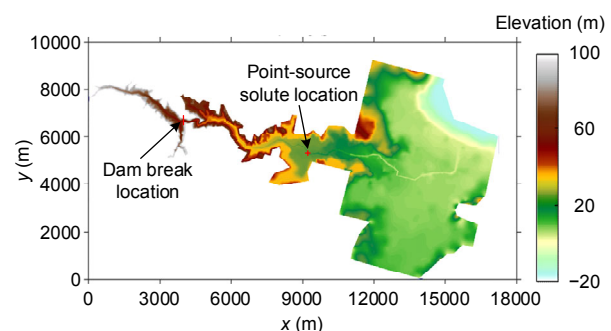

Fig. 9 Malpasset dam break: the topographic numerical
elevation, dam break position, and point-source solute
position

point source given by q_{in}=1 m/s and C_{in}=1 mg/L is set at (9281 m, 5244 m). The solute diffusion coefficient is 0.5 m^2/s, and the river Manning coefficient is taken as 0.033 s/m$^{1/3}$. The terrain data use digital elevation model (DEM) data with a resolution of 10 m (1.5 million square grid cells). The GPU-accelerated high-resolution model is used to simulate the dam-break water flow evolution that drives the solute transport process, and the simulation time is 3000 s.

Fig. 10 shows the results at t=1000 s and t=1800 s in terms of water depth. After the dam break, the water flow rapidly evolves downstream and then reaches the downstream plain, as shown in Fig. 10b. The numerical simulation results are the same as those reported by Hou et al. (2013b) for the dam-break water flow evolution on unstructured grids. The Godunov scheme can correctly capture the complex flow caused by the interaction of dam-break waves with the complex terrain. Fig. 11 shows the results of the solute concentration distribution at different times. After the simulation starts, the flood reaches the solute release source at t=900 s, as shown in Fig. 11a. After t=900 s, the solute is transported to low-lying places with the flood. During the simulation, it was observed that the water depth profile matches the solute distribution profile, which indicates that drastically changing floods can cause large-scale water pollution and rapidly affect downstream water bodies after destroying solute-releasing facilities. At all output times, the solute transport process of the flood-driven point source is very similar to that obtained by Cao et al. (2019).

## 3.5 Simulations for different grid resolutions and their efficiencies

Different resolutions of the Malpasset terrain were used to simulate the dam-break flood-driven point-source solute transport based on the test case in Section 3.4. The elaborate computational grid leads to an increase in the number of grid squares, which seriously affects the calculation efficiency of the model and may even make it impossible to perform large-scale computing. To obtain GPU acceleration technology that can achieve large-scale and high-efficiency calculations, uniform cell size grids of 2 m×2 m, 5 m×5 m, 10 m×10 m, and 15 m×15 m were

used to simulate the flood-driven point-source solute transport processes. When the computational domain adopts a square grid with a resolution of 2 m, 5 m, 10 m, and 15 m, the computational domain is discretized into 39 542 850 cells, 6 326 856 cells, 1 581 714 cells, and 703 188 cells, respectively. The grid resolution affects the number of computational domain grids and thus the total GPU time required to complete the entire simulation.

To demonstrate the performance of GPU code calculation efficiency, a model of CPU calculation was used to simulate the same test case, and a CPU with four cores was used (Intel Xeon E5-2609/CPU (4-core)). Table 2 shows the runtime using different grid resolutions on different hardwares. The calculation time results show that using the NVIDIA Tesla M 1080 GPU calculations, it only takes 136.19 s to complete this flood evolution process event in the 15-m resolution computing domain, while it takes 2684.29 s in the 5-m computing domain. Using the NVIDIA Tesla P 100 GPU to calculate the 15-m grid and 5-m grid to simulate the same event took 59.59 s and 1202.33 s, respectively, and it even performed calculations on a total of 39 542 850 grids with a resolution of 2 m. Therefore, the computation of the graphics on the NVIDIA Tesla P 100 GPU was more efficient than that of the NVIDIA Tesla M 1080 GPU. Simultaneously, this flood evolution event uses terrain data with different grid resolutions, and the GPU simulation time was 9.8–44.6 times the CPU time. Therefore, in flood-driven point-source solute transport events, GPU acceleration can achieve large-scale simulation calculations, and play a large role in water evaluation of pollution incidents.

4 Conclusions

This paper introduces a robust model based on GPU acceleration for simulating shallow flow-driven solute transport processes. The second-order Godunov-type finite volume method is used to numerically solve the integrals of shallow water and transport equations. The current model has been verified against test cases, and the results show that the numerical dissipation of solute in transport is very small and it can effectively simulate the dry and wet
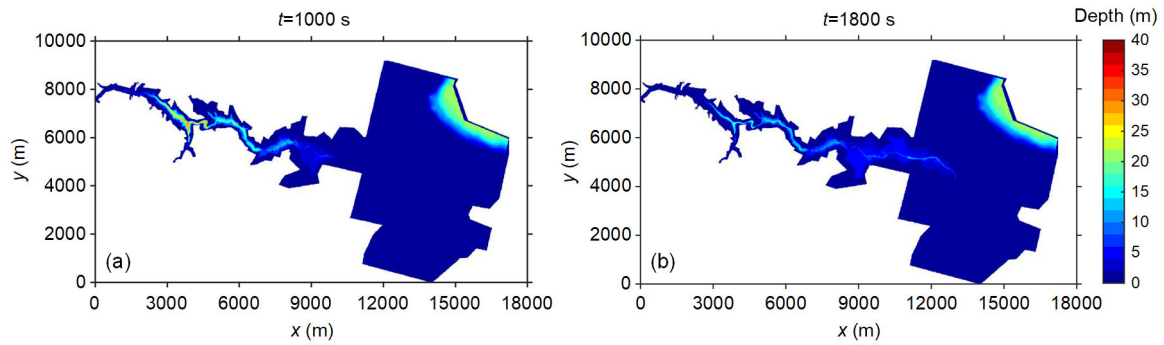
Fig. 10 Malpasset dam break: the flood evolution process
(a) *t*=1000 s; *t*=1800 s
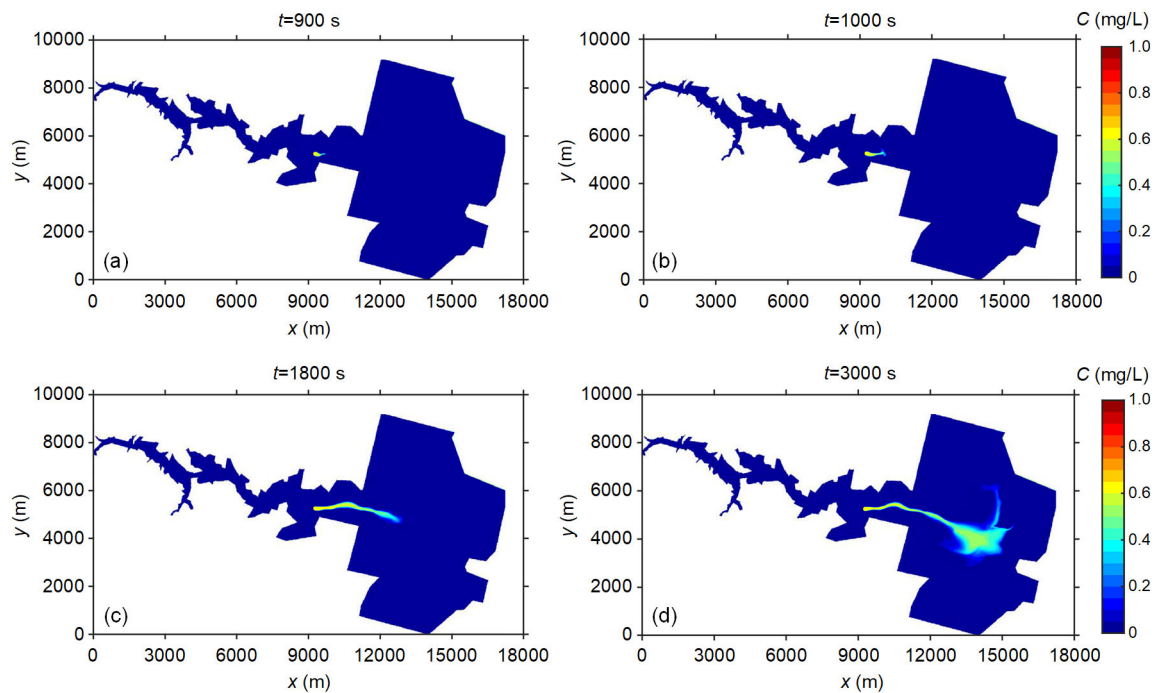
Fig. 11 Results of the solute transport and distribution driven by the flood
(a) *t*=900 s; (b) *t*=1000 s; (c) *t*=1800 s; (d) *t*=3000 s

Table 2 Total execution time and speedup ratio by CPU and GPU

Resolution	Cells number	Intel Xeon E5-2609/CPU (4-core) (s)	NVIDIA Tesla M 1080/Singe GPU (s)	NVIDIA Tesla P 100/Singe GPU (s)	CPU/GPU speedup ratio/1080	CPU/GPU speedup ratio/P100
2 m	39 542 850	–	–	17 941.5	–	–
5 m	6 326 856	34 494	2684.29	1202.33	12.8	28.7
10 m	1 581 714	7661.78	382.49	171.79	20.0	44.6
15 m	703 188	1334.75	136.19	59.59	9.8	23.4

'–' indicates that the number of grids exceeds the calculation limit of the device

boundary areas. The numerical prediction and analytical solution are better than those in previously obtained results.

The third test example is the advection of solute water driven over three humps by dam-break flood waves. It takes 3.865 s and 13.865 s to complete the

300-s flood event using GPU and CPU computing models, respectively. The fourth case uses different grid resolutions and different hardwares to calculate the flood evolution event. The results show that different GPU models require different times to complete the same event. For different grid resolutions, compared with traditional CPU calculations, GPU computing speed can be increased by 9.8–44.6 times.

The current model is also directly applicable to the practical application of solute transport driven by shallow flows, e.g. the transmission of pollutants in sudden water pollution accidents. This method can be further developed in future work to combine chemical and biological reactions involving the migration of reactive solutes in natural water.

Contributors

Jing-ming HOU and Qiu-hua LIANG developed the numerical model of the GPU-accelerated surface water flow and associated transport. Bao-shan SHI perfected the model and wrote the manuscript. Yu TONG, Yong-de KANG, Zhao-an ZHANG, and Gang-gang BAI helped to organize the manuscript. Xu-jun GAO and Xiao YANG revised and edited the final version. Jing-ming HOU and Bao-shan SHI contributed equally to this work.

Conflict of interest

Jing-ming HOU, Bao-shan SHI, Qiu-hua LIANG, Yu TONG, Yong-de KANG, Zhao-an ZHANG, Gang-gang BAI, Xu-jun GAO, and Xiao YANG declare that they have no conflict of interest.

References

Aidun CK, Clausen JR, 2010. Lattice-Boltzmann method for complex flows. *Annual Review of Fluid Mechanics*, 42(1):439-472.
https://doi.org/10.1146/annurev-fluid-121108-145519

Audusse E, Bristeau MO, 2003. Transport of pollutant in shallow water a two time steps kinetic method. *ESAIM: M2AN*, 37(2):389-416.
https://doi.org/10.1051/m2an:2003034

Barredo JI, 2007. Major flood disasters in Europe: 1950-2005. *Natural Hazards*, 42(1):125-148.
https://doi.org/10.1007/s11069-006-9065-2

Bayazıt Y, Koç C, Bakış R, 2021. Urbanization impacts on flash urban floods in Bodrum Province, Turkey. *Hydrological Sciences Journal*, 66(1):118-133.
https://doi.org/10.1080/02626667.2020.1851031

Begnudelli L, Sanders BF, 2006. Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying. *Journal of Hydraulic Engineering*, 132(4):371-384.

https://doi.org/10.1061/(asce)0733-9429(2006)132:4(371)

Benkhaldoun F, Elmahi I, Seaïd M, 2007. Well-balanced finite volume schemes for pollutant transport by shallow water equations on unstructured meshes. *Journal of Computational Physics*, 226(1):180-203.
https://doi.org/10.1016/j.jcp.2007.04.005

Bi S, Zhou JZ, Cheng SS, et al., 2013. A high-precision two-dimensional flow-transport coupled model based on Godunov's schemes. *Advances in Water Science*, 24(5): 706-714 (in Chinese).
https://doi.org/10.14042/j.cnki.32.1309.2013.05.003

Cao Y, Ye YT, Liang LL, et al., 2019. High efficient and accurate simulation of pollutant transport in torrential flow based on adaptive grid method. *Journal of Hydraulic Engineering*, 50(3):388-398 (in Chinese).
https://doi.org/10.13243/j.cnki.slxb.20180891

Chen CY, Lu TH, Yang YF, et al., 2021. Toxicokinetic/ toxicodynamic-based risk assessment of freshwater fish health posed by microplastics at environmentally relevant concentrations. *Science of the Total Environment*, 756: 144013.
https://doi.org/10.1016/j.scitotenv.2020.144013

Dong J, 2020. A robust central scheme for the shallow water flows with an abrupt topography based on modified hydrostatic reconstructions. *Mathematical Methods in the Applied Sciences*, 43(15):9024-9045.
https://doi.org/10.1002/mma.6597

Duarte HDO, Droguett EL, Araújo M, et al., 2013. Quantitative ecological risk assessment of industrial accidents: the case of oil ship transportation in the coastal tropical area of northeastern Brazil. *Human and Ecological Risk Assessment: An International Journal*, 19(6):1457-1476.
https://doi.org/10.1080/10807039.2012.723187

Fraccarollo L, Toro EF, 1995. Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems. *Journal of Hydraulic Research*, 33(6):843-864.
https://doi.org/10.1080/00221689509498555

Harten A, Lax PD, van Leer B, 1983. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35-61.
https://doi.org/10.1137/1025002

Hou JM, Simons F, Hinkelmann R, 2012. Improved total variation diminishing schemes for advection simulation on arbitrary grids. *International Journal for Numerical Methods in Fluids*, 70(3):359-382.
https://doi.org/10.1002/fld.2700

Hou JM, Simons F, Hinkelmann R, 2013a. A new TVD method for advection simulation on 2D unstructured grids. *International Journal for Numerical Methods in Fluids*, 71(10):1260-1281.
https://doi.org/10.1002/fld.3709

Hou JM, Simons F, Mahgoub M, et al., 2013b. A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex

topography. *Computer Methods in Applied Mechanics and Engineering*, 257:126-149.
https://doi.org/10.1016/j.cma.2013.01.015

Hou JM, Liang QH, Zhang HB, et al., 2014. Multislope MUSCL method applied to solve shallow water equations. *Computers & Mathematics with Applications*, 68(12): 2012-2027.
https://doi.org/10.1016/j.camwa.2014.09.018

Hou JM, Liang QH, Li ZB, et al., 2015. Numerical error control for second-order explicit TVD scheme with limiters in advection simulation. *Computers & Mathematics with Applications*, 70(9):2197-2209.
https://doi.org/10.1016/j.camwa.2015.08.022

Hou JM, Wang R, Liang QH, et al., 2018a. Efficient surface water flow simulation on static Cartesian grid with local refinement according to key topographic features. *Computers & Fluids*, 176:117-134.
https://doi.org/10.1016/j.compfluid.2018.03.024

Hou JM, Wang T, Li P, et al., 2018b. An implicit friction source term treatment for overland flow simulation using shallow water flow model. *Journal of Hydrology*, 564:357-366.
https://doi.org/10.1016/j.jhydrol.2018.07.027

Hou JM, Kang YD, Hu CH, et al., 2020. A GPU-based numerical model coupling hydrodynamical and morphological processes. *International Journal of Sediment Research*, 35(4):386-394.
https://doi.org/10.1016/j.ijsrc.2020.02.005

Kachiashvili K, Gordeziani D, Lazarov R, et al., 2007. Modeling and simulation of pollutants transport in rivers. *Applied Mathematical Modelling*, 31(7):1371-1396.
https://doi.org/10.1016/j.apm.2006.02.015

Kawahara M, Umetsu T, 1986. Finite element method for moving boundary problems in river flow. *International Journal for Numerical Methods in Fluids*, 6(6):365-386.
https://doi.org/10.1002/fld.1650060605

Kong J, Xin P, Shen CJ, et al., 2013. A high-resolution method for the depth-integrated solute transport equation based on an unstructured mesh. *Environmental Modelling & Software*, 40:109-127.
https://doi.org/10.1016/j.envsoft.2012.08.009

La Rocca M, Montessori A, Prestininzi P, et al., 2015. A multispeed discrete Boltzmann model for transcritical 2D shallow water flows. *Journal of Computational Physics*, 284:117-132.
https://doi.org/10.1016/j.jcp.2014.12.029

La Rocca M, Miliani S, Prestininzi P, 2020. Discrete Boltzmann numerical simulation of simplified urban flooding configurations caused by dam break. *Frontiers in Earth Science*, 8:346.
https://doi.org/10.3389/feart.2020.00346

Li YP, Wei J, Gao XM, et al., 2018. Turbulent bursting and sediment resuspension in hyper-eutrophic Lake Taihu, China. *Journal of Hydrology*, 565:581-588.
https://doi.org/10.1016/j.jhydrol.2018.08.067

Liang QH, 2010. A well-balanced and non-negative numerical scheme for solving the integrated shallow water and solute transport equations. *Communications in Computational Physics*, 7(5):1049-1075.
https://doi.org/10.4208/cicp.2009.09.156

Liang QH, Xia XL, Hou JM, 2016. Catchment-scale high-resolution flash flood simulation using the GPU-based technology. *Procedia Engineering*, 154:975-981.
https://doi.org/10.1016/j.proeng.2016.07.585

Liu RX, 2011. Simulation of water transferring impact on the water quality in Danjiangkou reservoir of the south to north water diversion middle line project. *Journal of Basic Science and Engineering*, 19(S1):193-200 (in Chinese).
https://doi.org/10.3969/j.issn.1005-0930.2011.s1.022

Liu WH, Chen RQ, Qiu RF, et al., 2020. Generalized form of interpolation-supplemented lattice Boltzmann method for computational aeroacoustics. *Journal of Zhejiang University (Engineering Science)*, 54(8):1637-1644 (in Chinese).
https://doi.org/10.3785/j.issn.1008-973X.2020.08.024

Murillo J, García-Navarro P, Burguete J, 2009. Conservative numerical simulation of multi-component transport in two-dimensional unsteady shallow water flow. *Journal of Computational Physics*, 228(15):5539-5573.
https://doi.org/10.1016/j.jcp.2009.04.039

Petti M, Bosa S, 2007. Accurate shock-capturing finite volume method for advection-dominated flow and pollution transport. *Computers & Fluids*, 36(2):455-466.
https://doi.org/10.1016/j.compfluid.2005.11.008

Roccati A, Faccini F, Luino F, et al., 2019. Heavy rainfall triggering shallow landslides: a susceptibility assessment by a GIS-approach in a Ligurian Apennine catchment (Italy). *Water*, 11(3):605.
https://doi.org/10.3390/w11030605

Shao JR, Wu SQ, Zhou J, et al., 2012. High-accuracy numerical simulation of 2D transport problems. *Advances in Water Science*, 23(3):383-389 (in Chinese).
https://doi.org/10.14042/j.cnki.32.1309.2012.03.008

Smith LS, Liang QH, 2013. Towards a generalised GPU/CPU shallow-flow modelling tool. *Computers & Fluids*, 88: 334-343.
https://doi.org/10.1016/j.compfluid.2013.09.018

Song LX, Yang F, Hu XZ, et al., 2014. A coupled mathematical model for two-dimensional flow-transport simulation in tidal river network. *Advances in Water Science*, 25(4): 550-559 (in Chinese).
https://doi.org/10.14042/j.cnki.32.1309.2014.04.012

Tang CY, Li YP, Jiang P, et al., 2015. A coupled modeling approach to predict water quality in Lake Taihu, China: linkage to climate change projections. *Journal of Freshwater Ecology*, 30(1):59-73.
https://doi.org/10.1080/02705060.2014.999360

Tao T, Lu YJ, Fu X, et al., 2012. Identification of sources of pollution and contamination in water distribution

networks based on pattern recognition. *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*, 13(7):559-570.
  https://doi.org/10.1631/jzus.A1100286

Toro EF, 2001. Shock-capturing Methods for Free-surface Shallow Flow. John Wiley & Sons, Ltd., Chichester, UK.

Venturi S, Francesco SD, Geier M, et al., 2021. Modelling flood events with a cumulant CO lattice Boltzmann shallow water model. *Natural Hazards*, 105(2):1815-1834.
  https://doi.org/10.1007/s11069-020-04378-x

Wang ZW, Cheng WP, 2002. Analysis of ecological mechanism of urban flood and waterlog—research based mainly on Hangzhou City. *Journal of Zhejiang University* (*Engineering Science*), 36(5):582-587 (in Chinese).
  https://doi.org/10.3785/j.issn.1008-973X.2002.05.023

Zhang LL, Liang QH, Wang YL, et al., 2015. A robust coupled model for solute transport driven by severe flow condi-
tions. *Journal of Hydro-environment Research*, 9(1):49-60.
  https://doi.org/10.1016/j.jher.2014.04.005

Zhang WJ, Lin XY, Su XS, 2010. Transport and fate modeling of nitrobenzene in groundwater after the Songhua River pollution accident. *Journal of Environmental Management*, 91(11):2378-2384.
  https://doi.org/10.1016/j.jenvman.2010.06.025

Zhang WW, 2011. Measuring the value of water quality improvements in Lake Tai, China. *Journal of Zhejiang University-SCIENCE A* (*Applied Physics & Engineering*), 12(9):710-719.
  https://doi.org/10.1631/jzus.A11b0157

Zhao Y, Nan J, Cui FY, et al., 2007. Water quality forecast through application of BP neural network at Yuqiao reservoir. *Journal of Zhejiang University-SCIENCE A*, 8(9):1482-1487.
  https://doi.org/10.1631/jzus.2007.A1482