Depth-Averaged Two Dimensional Model Using Cartesian Cut-Cell Approach

Chunshui Yu, Ph.D. Candidate and Jennifer Duan, Ph.D., P.E.
Dept. of Civil Engineering and Engineering Mechanics, University of Arizona, Tucson, AZ, 85721 USA

INTRODUCTION

To simulate flood wave propagation over arbitrary topography, such as alluvial fan, depth-averaged Navier-Stokes equation can be solved accurately using the Godunov-type finite volume method (FVM) with an approximate Riemann solver. Traditionally, the body-fitted mesh is generated by using the unstructured mesh (Zhao, Shen et al. 1994; Begnudelli and Sanders 2006; Murillo, García-Navarro et al. 2009). An alternative approach is to use the Cartesian cut-cell method (Causon, Ingram et al. 2000; Zhou, Causon et al. 2004; Kim and Cho 2011). In general, the boundaries of a domain are cut out of a background uniform rectangular mesh and the cells of the domain are divided into rectangular internal cells and irregular boundary cells. Resulting cells can fluid cells, solid cells and cut cells. The cut cells can be further categorized into 16 sub-types depending upon the slope of the cutting edge (Causon et al. 2000).

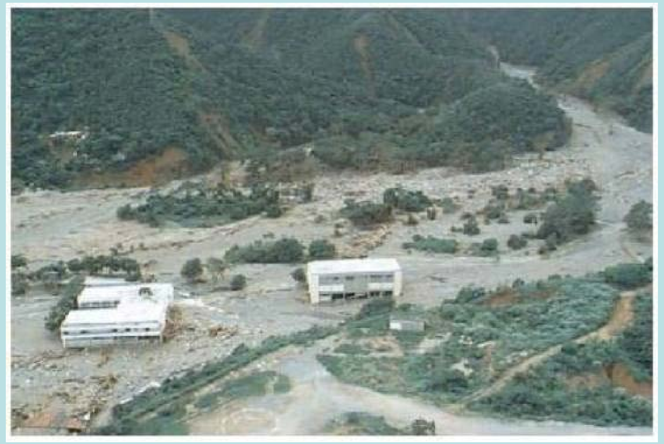


Fig.1. Flash flood Flow in Arizona, July 2006

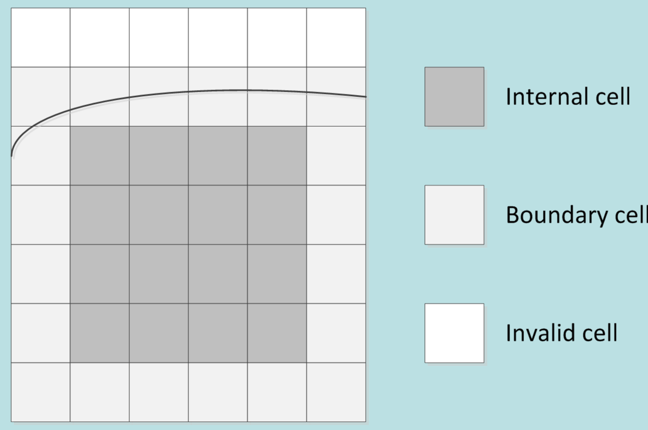


Fig. 2. Classification of cells in Cartesian Cut-cell Method

NUMERICAL METHOD

The MUSCL-Hancock finite volume method incorporated with the surface gradient method (Zhou et al., 2001) is adopted. Two steps in every time step: the predictor step and the corrector step.

$$(AU)_{ij}^{n+\frac{1}{2}} = (AU)_{ij}^{n} - \frac{\Delta t}{2}\left(\sum_{m=1}^{M} F(U_m)^n \cdot L_m - (AS)_{ij}^{n}\right)$$

where $\boldsymbol{L}_m$ is the cell side vector defined as the length of side m multiplied by the outward normal vector; M is the number of cell sides; $\boldsymbol{F}(\boldsymbol{U}_m)$ is the flux vector and is calculated at each side m by the following linear reconstruction function:

$$U_m = U_{ij}^{n} + r_m \cdot \nabla U_{ij}^{n}$$

where $\boldsymbol{r}_m$ is the normal distance vector from the cell center to side m. A limited gradient vector is used to avoid spurious oscillations in the linear reconstruction process. The corrector step is a conservative step over a full time step:

$$(AU)_{ij}^{n+1} = (AU)_{ij}^{n} - \Delta t\left(\sum_{m=1}^{M} F(U_m^L, U_m^R)^{n+\frac{1}{2}} \cdot L_m - (AS)_{ij}^{n+\frac{1}{2}}\right)$$

where $F(U^L,U^R)$ is the flux vector calculate by solving a local Riemann problem at each side of the cell; U_m (L and R) are the conservative variables at the left and right sides of cell interface. The HLL approximate Riemann solver (Harten et al., 1983) is used to solve the local Riemann problem with dry bed modification from Fraccarollo and Toro (1995). The flux at the interface is defined as:

$$F(U_m^L, U_m^R) = \begin{cases} F(U_m^L), \ if \ s_L \ge 0 \\ F^*(U_m^L, U_m^R), if \ s_L < 0 < s_R \\ F(U_m^R), if \ s_R \le 0 \end{cases}$$

RESULTS

Case 1: Triangular Sill (Soares-Frazao, 2007)

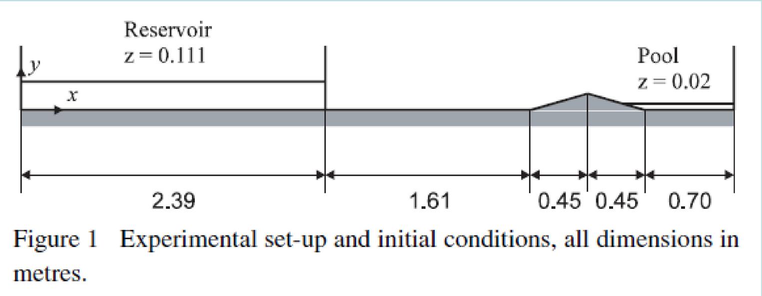


Figure 1 Experimental set-up and initial conditions, all dimensions in metres.

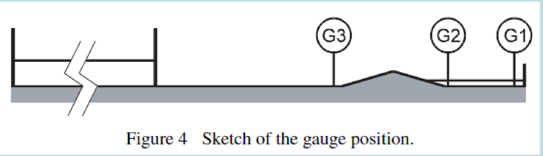
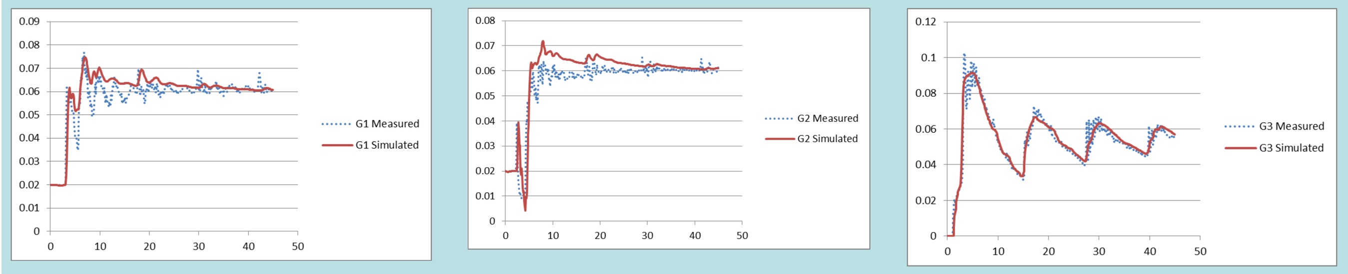
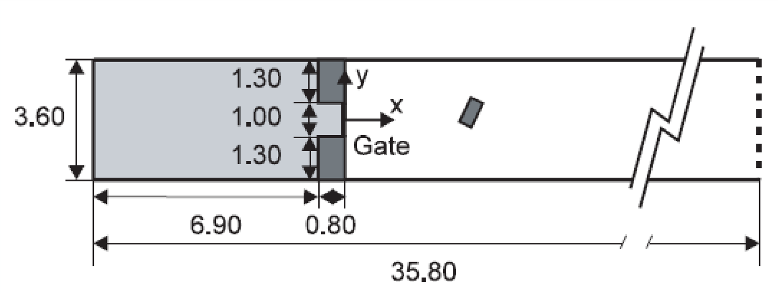
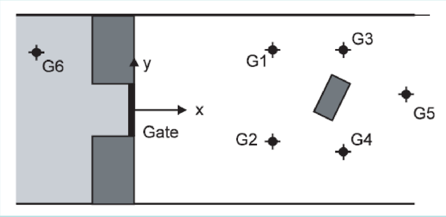


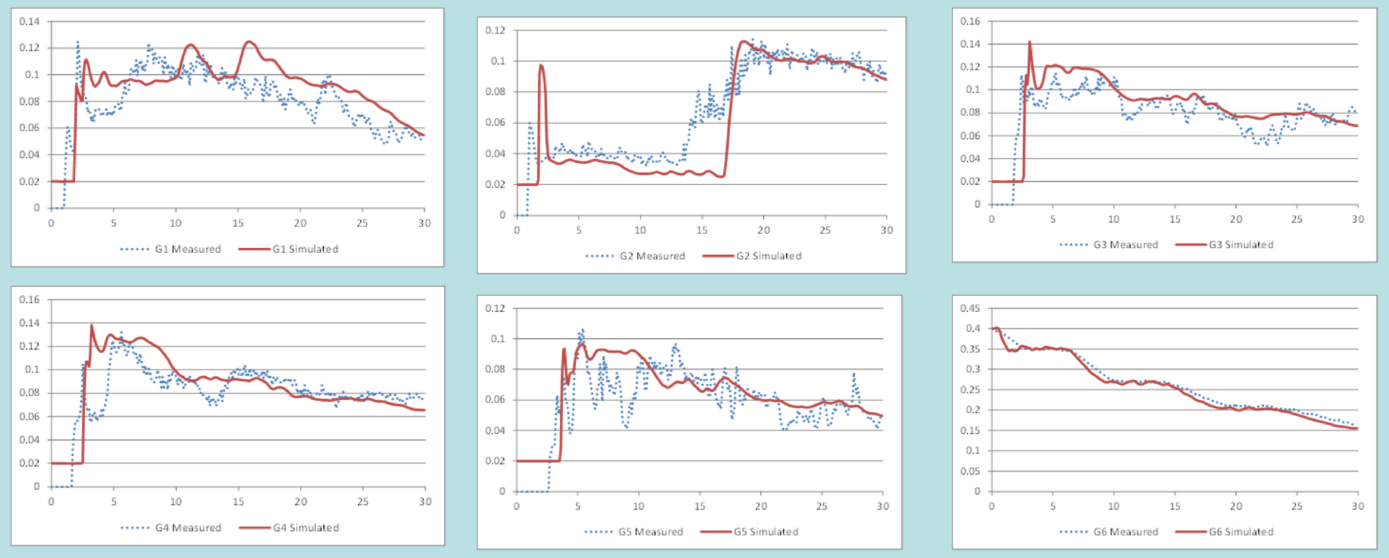
Figure 4 Sketch of the gauge position.



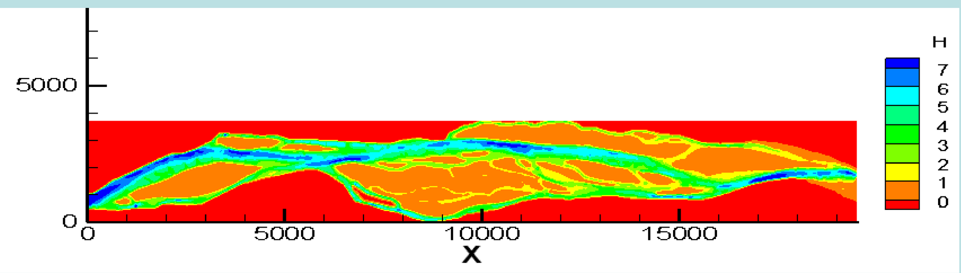
Case 2: Isolated obstacle: Soares-Frazao and Zech, 2007







Case 3: Unsteady flood flow through Huron Island Reach, Mississippi River.



GOVERNING EQUATION

The governing equations of shallow-water flow, the shallow water equation (SWE), are reduced from Navier-Stokes equations with the hydrostatic assumption.

$$\frac{\partial U}{\partial t} + \nabla \cdot F = D + S$$

in which

$$U = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} \quad F = \begin{pmatrix} hV \\ huV + \frac{1}{2}gh^2\vec{i} \\ hv^2V + \frac{1}{2}gh^2\vec{j} \end{pmatrix}$$

where $h=\eta - z_b$ is flow depth, g is gravity acceleration; u and v are the velocity components in x and y direction; $\boldsymbol{V}$ is the velocity vector defined as $V = u\vec{i} + v\vec{j}$

D is the diffusion term, including both the kinematic viscosity and turbulent viscosity.

$$D = \begin{pmatrix} 0 \\ \frac{\partial h\tau_{xx}}{\partial x} \\ \frac{\partial h\tau_{xy}}{\partial x} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\partial h\tau_{xy}}{\partial y} \\ \frac{\partial h\tau_{yy}}{\partial y} \end{pmatrix}$$

S is the source term, which consists of gravity force and bed surface friction,

in which $S = S_b + S_f$

$$S_b = \begin{pmatrix} 0 \\ ghS_x \\ ghS_y \end{pmatrix}, S_f = \begin{pmatrix} 0 \\ -\frac{\tau_x}{\rho} \\ -\frac{\tau_y}{\rho} \end{pmatrix}$$

The Integral form of shallow water equation,

$$\frac{\partial}{\partial t}\int_A U\,dA + \int F \cdot n\,dS = \int_\forall S\,d\forall + \int_\forall D d\forall$$

where $\boldsymbol{n}$ is the outward normal vector; A is the area enclosed by the surface S. The governing equation is hyperbolic differential equation, which exhibits discontinuity. Numerical scheme needs to be compatible with discontinuous solution.

In which $\tau_{xx} = 2(\nu+\nu_t)\frac{\partial u}{\partial x}, \tau_{xy} = (\nu+\nu_t)(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}), \tau_{yy} = 2(\nu+\nu_t)\frac{\partial v}{\partial y}$

WET AND DRY METHOD

For the dry bed problem, according to Fraccarollo and Toro (1995), the waves are calculated by the following equations, for the right dry bed:

$$s_L = V^L \cdot n_m - \sqrt{\phi^L}\ ,\ s_R = V^L \cdot n_m + 2\sqrt{\phi^L}$$

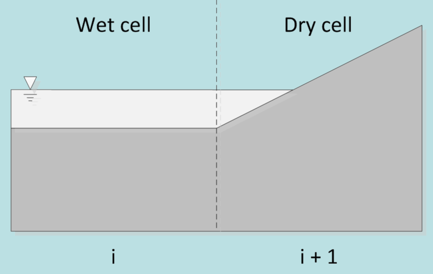
For the left side dry bed:

$$s_L = V^R \cdot n_m - \sqrt{\phi^R}\ ,\ s_R = V^R \cdot n_m + 2\sqrt{\phi^R}$$

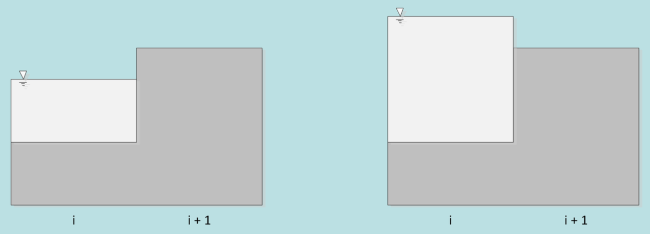
where $\varphi = g(\eta - z_b)$.

In the case when the bed elevation of the dry cell is higher than the surface elevation of its neighboring wet cell, the HLL Riemann solver would lose its stability. So it's necessary to track the wet-dry front and cut the dry cell out of the computational domain.

- The position of the wet-dry front is interpolated linearly using bed elevation:



- Other dry/wet cells need to be cut using under reflection/non-reflection condition.



CONCLUSIONS

- Cartesian cut-cell method does not require a mesh generator.
- Cartesian grids are uniform that can save computational cost comparing to irregular skewed meshes.
- Using Cartesian cut-cell method can identify wet-dry front automatically at runtime.
- The MUSCL-Hancock finite volume method is efficient in Cartesian Cut-cell method.
- Because of simple grid, parallel computing can be easily implemented.