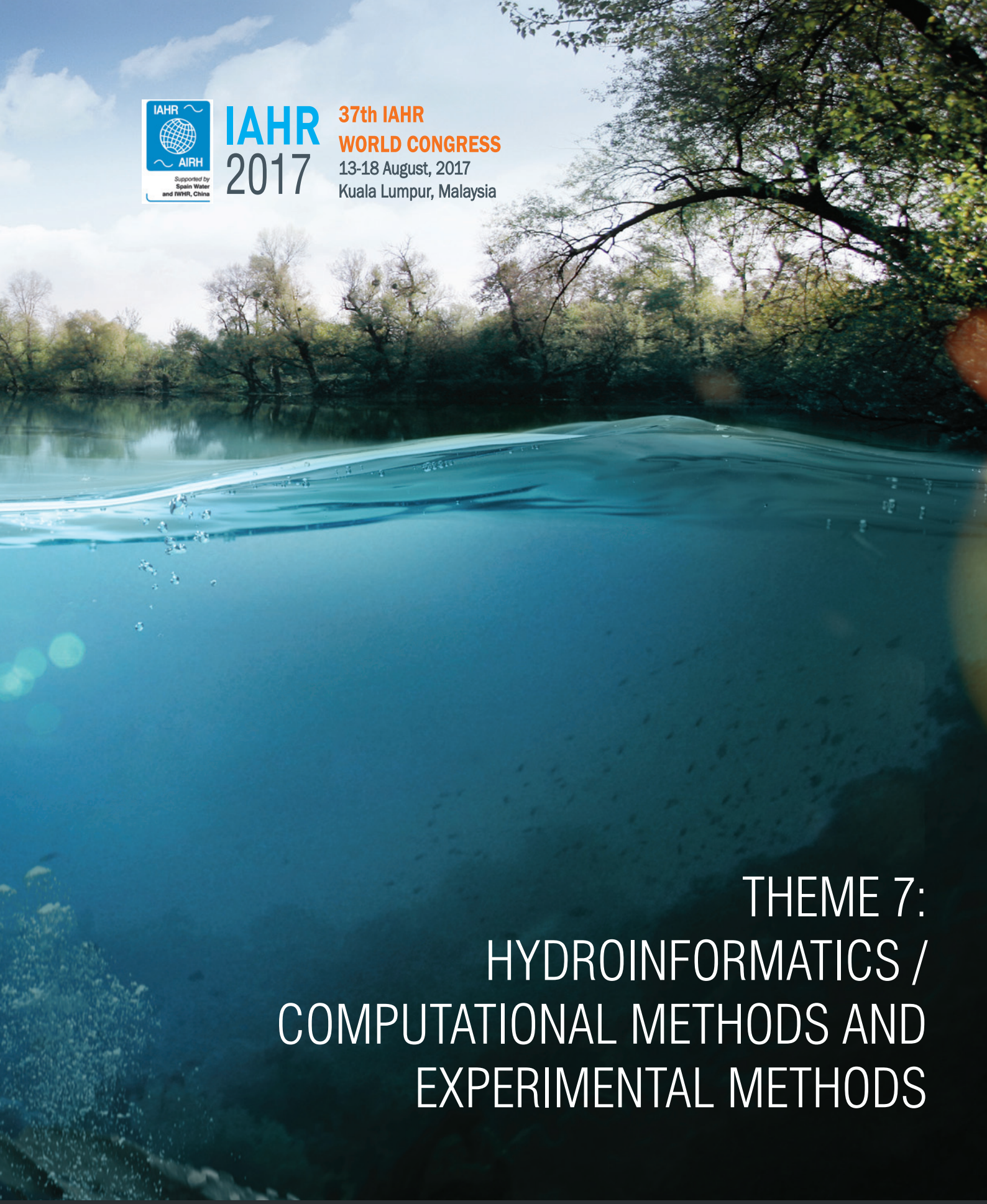
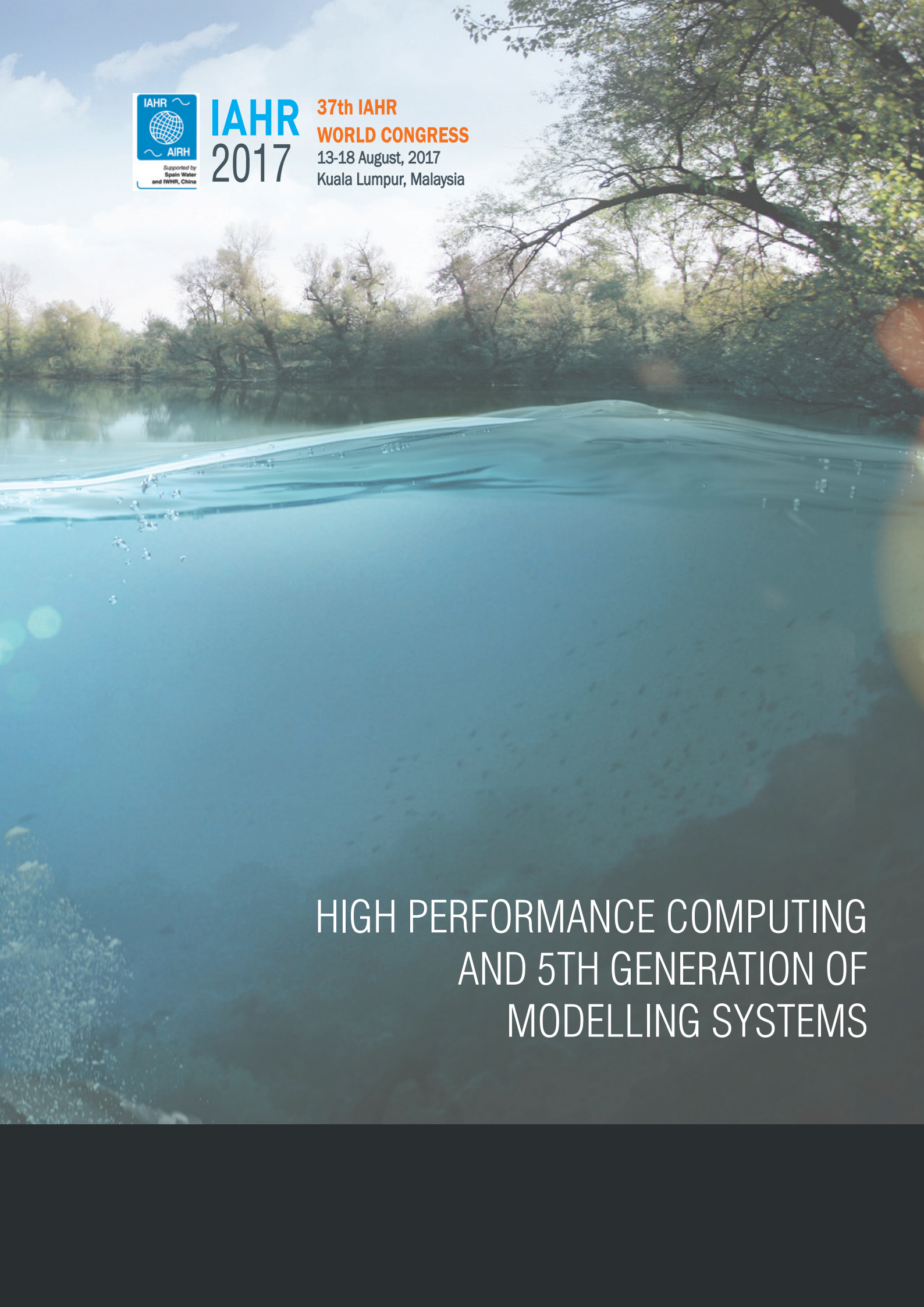


THEME 7: HYDROINFORMATICS / COMPUTATIONAL METHODS AND EXPERIMENTAL METHODS

Editor: Othman A. Karim


IAHR
AIRH
Supported by
Spain Water
and IWHR, China
IAHR
2017
37th IAHR
WORLD CONGRESS
13-18 August, 2017
Kuala Lumpur, Malaysia
HIGH PERFORMANCE COMPUTING
AND 5TH GENERATION OF
MODELLING SYSTEMS

UTILIZING GOOGLE EARTH FEATURES FOR HYDRAULIC MODEL DEVELOPMENT

KENNARD LAI[1] & PIERRE JULIEN[2]

[1, 2] Department of Civil & Environmental Engineering, Colorado State University, Fort Collins, CO 80523, USA.
kennard.lai@colostate.edu

ABSTRACT

This article explores the usage of Google Earth features in developing a hydraulic model. The authors share their experience of using Google Earth's satellite images and Street View images as new remote sensing tools to obtain a better understanding of the site through historical images, measure distances, as well as estimate river geometry and roughness. The features are found to be useful especially when modeling a river remotely when frequent site visits are prohibitive and as a supplement to available data. An example is presented where the information obtained from Google Earth is used in HEC-RAS (Hydrologic Engineering Center – River Analysis System) to develop a simple before-and-after hydraulic model to illustrate the effects of a stormwater tunnel outflow during a flood event on the Kerayong River in Malaysia before and after river improvement works has been completed. It is found that Google Earth satellite imagery and Street View features are new tools that can be used to aid in estimating channel geometry and roughness, as well as channel morphology.

Keywords: Hydraulic Modeling; Remote Sensing; Google Earth; Street View; HEC-RAS.

1 INTRODUCTION

Hydraulic modeling of streams and rivers are common yet vital component of most river projects and studies. Projects like stormwater management, bridge design, stream rehabilitation, or flood inundation mapping require some sort of hydraulic modeling. It is important for a modeler to have an understanding of the site for an effective model. Modeling a typical river requires data of sorts such as discharge, river geometry, and channel roughness. Traditionally, river geometry data such as reach length, cross-section geometry, stream bed/bank material are obtained from in-situ surveys. Field data collection is time consuming and costly. Many of these measurements are done frequently and are seen as a "privilege" of the developed world as such measurements are less available in developing countries. Challenges of modeling these infrequently measured rivers arise in the form of unavailable, outdated information, or inconsistent data records.

The advent of utilizing geographic information systems (GIS) in hydraulic modeling has brought upon improvements in data processing and results. One of the remote sensing advances is the introduction of Google Earth, a virtual globe, map, and geographical information program that was used to aid the development of the model discussed in this article. Released in 2005, Google Earth displays the superimposition of images from satellite imagery, aerial photography, and GIS onto a globe. Rusli et al. (2014) suggested that Google Earth's digital elevation model (DEM) is applicable as a data source in hydrological modeling as it is comparable with NASA's SRTM90 (Shuttle Radar Topography mission), ASTER10, and ASTER30 (Advanced Space-borne Thermal Emission and Reflection Radiometer). In 2008, Google introduced the Street View feature into the Google Earth program in which 360-degree panoramic street-level photos are taken by special equipped Google Street View Camera cars and stitched together, creating a seamless first-person virtual world. While most photography is done with a car, other methods like tricycles, push carts, and camera-outfitted backpack are used as well. As of February of 2015, Google made Google Earth Pro available for free. Google Earth's features and its zero-dollar price tag make it attractive to be utilized in studying a site for hydraulic model development.

Google Earth's features were used to develop a model to study the impact of the Stormwater and Road Tunnel (SMART) outflow on the hydraulics of its receiving river - the Kerayong River, Malaysia. The study will be used as an example throughout this article to illustrate the usage of Google Earth's features in developing the hydraulic model. Located in the Klang Watershed, the SMART Tunnel is a dual-purpose stormwater and road tunnel designed to divert stormwater away from the flood prone areas of Kuala Lumpur (Figure 1). During prolonged storm events, the tunnel discharges up to 300 m3/s at the outlet through Desa Pond and then into the lower reach of the Kerayong River. A one-dimensional steady flow model was performed in HEC-RAS to analyze water-surface elevation, velocity, and shear stress from the increased discharge of the SMART Tunnel. The Google Earth features that are used to develop the model are historical satellite images, the distant measure tools, and Street View.

2 DATA

In this study, the data acquired for the development of the hydraulic model are: (i) Cross section geometry of 54 river stations along the Kerayong River (survey year unknown), (ii) Stage and discharge data on the Kerayong River from 2008-2009 where the gage station is upstream of the outlet of Desa Pond, (iii) Site visit photos, (iv) River geometry measurements at a few locations along the reach. There are still a few unknown data that are required for the development of the model such as flood plain measurements, bed and bank material, channel roughness, and boundary conditions. As the study is conducted in a different country, site visits were cost and time prohibitive. The ease of use of Google Earth allowed for the remote study of the site and information about the site was extracted through the features in Google Earth.

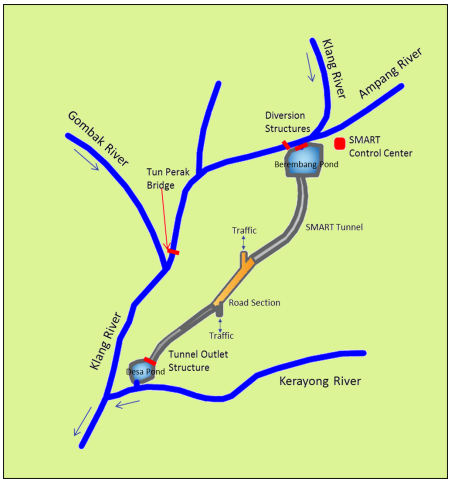


Figure 1. Components of SMART Tunnel

3 FEATURES AND METHODS

3.1 Google Satellite Images

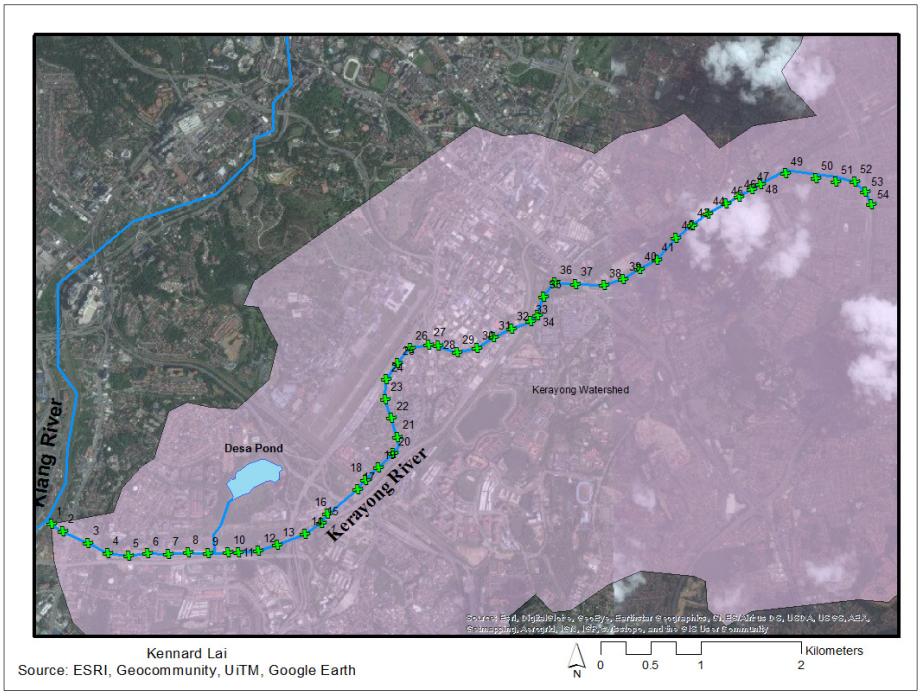


Figure 2. Kerayong River and Watershed

A general understanding of the site was obtained through the use of Google Earth Satellite Images. Figure 2 shows the site map generated with ArcMap with the location of the Kerayong River, the Kerayong Watershed, and locations of river stations (RS) where cross section geometry data were available. While the basemap in Figure 2 was from ESRI World Imagery, it did not provide a high enough resolution for the site to allow for a detection of the geometric features of the site (ESRI., 2016). Figure 3 shows the difference in

resolution between ESRI's World Imagery basemap and Google Earth at the site at a 1:5000 scale. While ESRI's World Imagery basemap resolution varies with locations, Google Earth's imagery ranges around 15m to 30m per pixel (Rusli et al., 2014). The images for the site of interest can be saved using the "Save Image" function in Google Earth and imported as a shapefile (.shp) or raster (.tiff) into ArcMap or other GIS software.

Figure 3. Comparison of image resolution from Google Earth and ESRI World Imagery Basemap

3.2 Historical Images

Google Earth's historical images feature allows for studying of the site through the times. In this example, the SMART tunnel was completed in 2007.The changes on the Kerayong River can be observed through multiple years since 2001, before the SMART Tunnel until after it was completed. It was observed that channel improvement works has been carried out sometime in 2008 and completed by 2010 (Figure 4). The "Time Slider" feature in Google Earth allows the model developers to analyze temporal changes of the river. This led to the study of the impact of SMART Tunnel outlet on the river before and after channel improvement works. The observation of historical images had also prompted a closer look on the available cross-section geometric data to determine its validity.

Figure 4. Reach before and after channel improvement works

3.3 Distance Estimation

Google Earth's "Ruler" feature allows for distance measurement that allows for the estimation of the geometric features of the river such as the length, and width. Figure 5 shows a screenshot of the feature in use in Google Earth. While it reports distance down to two decimal places, the varying resolution from 15m to 30m and different types of map projection used are recognized as sources of measurement errors. In this site, through comparisons with available data and measurements, the Google Earth ruler is determined to have an

error of +/- 0.1m. It is suggested that other projects should have a ground level measurement to verify the accuracy of Google Earth's distance measurements.

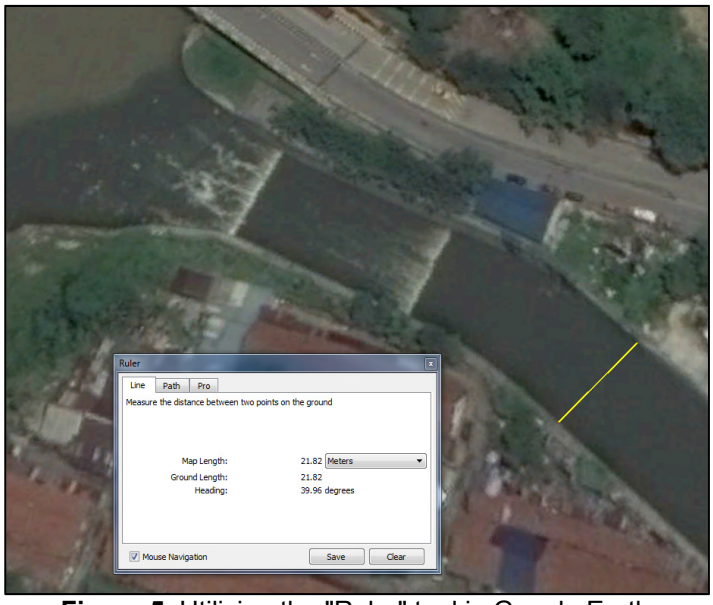


Figure 5. Utilizing the "Ruler" tool in Google Earth

3.4 Elevation estimation

Google Earth gives the elevation of a point which is displayed on the bottom corner of the window. It examined the suitability of extracting the DEM from Google Earth for watershed delineation and river bed elevation. In this case, the elevation indicator in Google Earth shows that the elevation of the river is around 26m. However, it is not known if the displayed elevation is the ground surface or water surface elevation. There are also errors associated with the elevation for channel bed elevation because the resolution in Google Earth varies from 15m to 30m grid. The width of the channel in this case does not exceed 30m. As the elevation is averaged in a grid, the DEM from Google Earth is deemed unsuitable to represent the channel bottom elevation because its resolution is not high enough to capture the bathymetry of small rivers.

On the other hand, the resolution of the DEM from Google Earth may be useful for the delineation of the watershed. While Google does not state if the DEM is a Digital Terrain Model (ground surface) or Digital Surface Model (surface of buildings/trees), the delineated watershed from Google Earth's DEM is comparable with the 3 arc-sec DEM from USGS. In this example, some elevation of the river's floodplain was estimated using the Google Earth DEM.

3.5 Utilizing Street View for channel geometry estimation

One of the newest additions to Google Earth is the Street View feature which aims to have information to be more universally accessible and useful (Anguelov et al., 2010). Street View is Google's feature where street-level 360 degree images are captured using vehicles equipped with cameras. While most of the images are captured on roads, pedestrian paths and trails are circumnavigated as well with special trikes, push carts, and snowmobiles. To develop the hydraulic model, familiarity with the site and the problems can help the modeler determine the required level of accuracy in the model. Street View allows a virtual first-person site visit that greatly increases the familiarity of the developers with the site.

One of the challenges encountered in this study is that the date at which the river geometry was surveyed was unknown; this gives the possibility that the data provided could be out-of-date. Another challenge is that the river geometry data does not include the geometry of the floodplain. For a model that would involve a flooding scenario, having the information of the floodplain such as elevation, geometry, and roughness would provide a better representation of the hydraulic behavior. In this study, modeling was done remotely in the USA whereas the site is in Malaysia; it was cost prohibitive for site visits to obtain field measurements. However, the gap could be reasonably filled with information obtained from Street View.

In the study, street view images were obtained at every vantage point where the river or its floodplain was visible. This included streets that are parallel to the river, bridges, and elevated highways (Figure 6). A total of 27 street view images were obtained for the Kerayong River. Paired with the locations of the river stations plotted on the map in ArcGIS, it is possible to identify the river station through Street View. Although exact lengths cannot be measured, by using careful judgement, lengths such as width and depth can be estimated and compared with available data. Google Street View images proved to be useful in verifying or modifying cross section geometry data. Figure 7 shows the updating of river geometry at a river station that includes modifying the channel side slopes and including a floodplain in HEC-RAS.

Figure 6. The Yellow Man icon can be dragged to the blue lines where Street View images are available

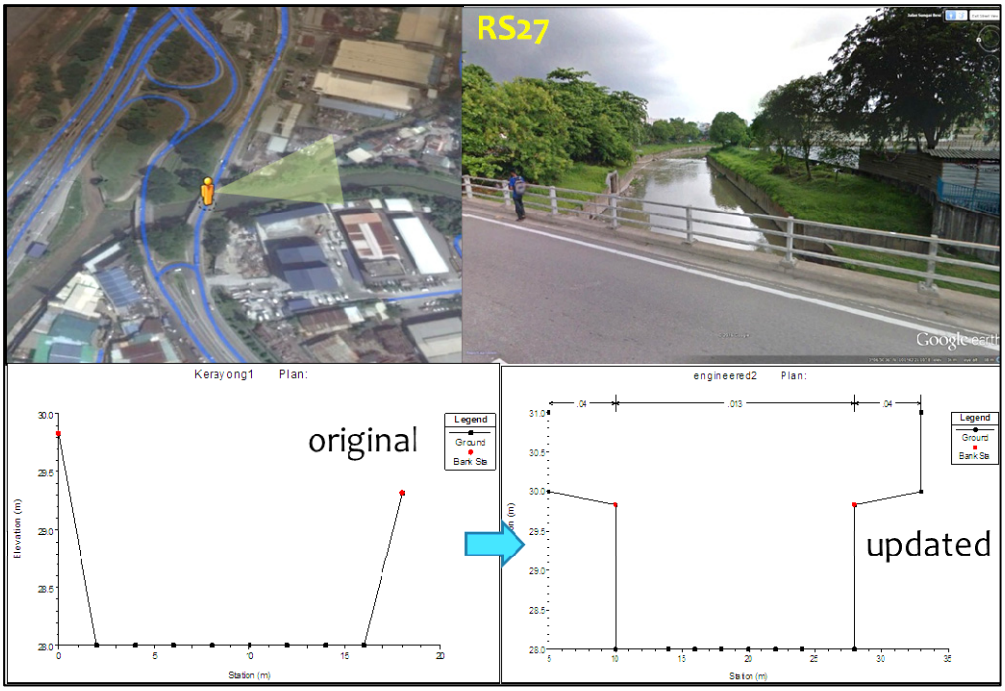


Figure 7. River geometry can be estimated and updated to include floodplain through Street View

3.6 Utilizing Street View to estimate channel roughness

Conveyance in HEC-RAS is described by Manning's equation. Bed and bank material as well as floodplain vegetation type can be observed in Street View. The corresponding Manning's n is that obtained from the HEC-RAS Reference Manual (USACE, 2010). On main channels Manning's n ranges from 0.017 for float finished concrete to 0.03 for clean, straight, natural channels. On floodplains, Manning's n ranges from 0.03 for short grass to 0.05 for scattered brush and heavy weeds. For example, the riprap material and bed material at the confluence of the Kerayong and Klang Rivers was estimated using Street View (Figure 8).

Figure 8. Street View can be used to identify bed and bank material

4 MODEL DEVELOPMENT & RESULTS

The information gathered from Google Earth was combined with available data such as discharge, channel geometry, and site visit photos to develop the one-dimensional steady state simulation of the river under multiple discharges and before and after channel improvement works (Figure 9). It could be observed through historical images that a previous “natural” channel was straightened after 2009. This may have been warranted by the increased discharge from the SMART Tunnel into the Kerayong River.

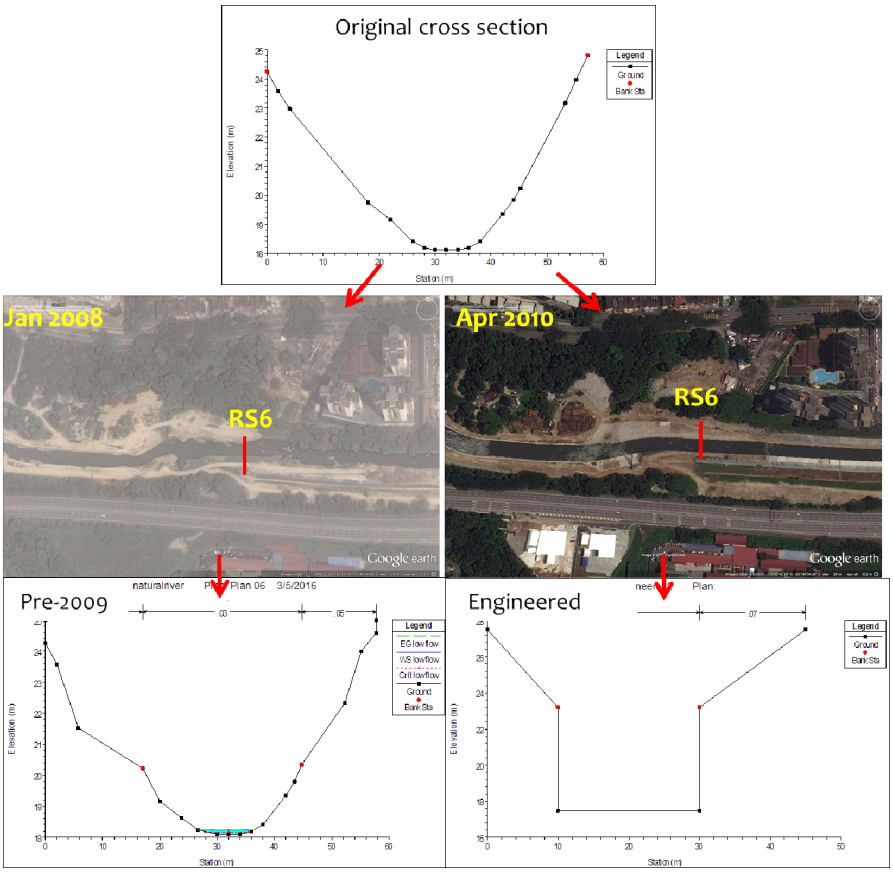


Figure 9. The same reach was modified to "Pre-2009" and "Engineered" conditions

Through flow duration analysis it was determined that the maximum flow that occurred in the Kerayong River is 484 m^3/s. When modeled with HEC-RAS, it was determined that the Pre-2009 river overtopped and caused flooding at 7 of 9 river stations of up to 3 m while the Engineered river only had about 23 cm of overtopping of 1 of 9 river stations (Figure 10).

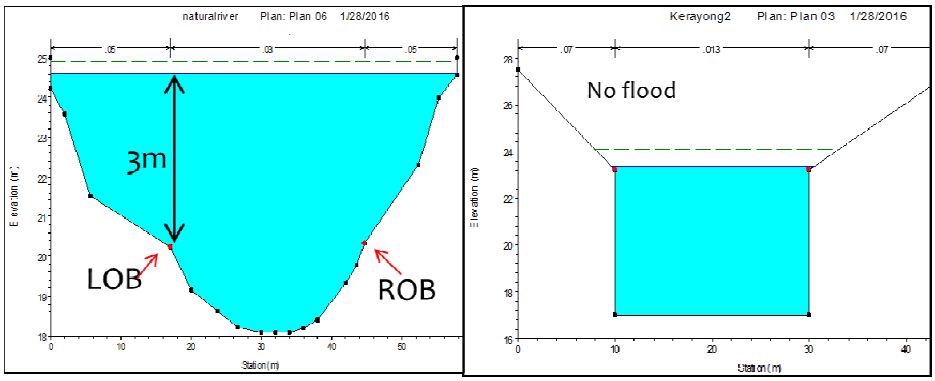


Figure 10. The engineered river was able to contain the increased discharge

The model was also used to predict bank stability and if future flooding will occur if discharge is increased by 10%. A "Future" flow data was used for the simulation and showed that the 10% increase in flow will cause 0.71 m of bank overtopping. The suitability of bed and bank material was determined through the estimation of particle size with Street View with the velocity and shear stress results from the HEC-RAS model and was found to be suitably designed.

4.1 Summary

A summary table of the suitability for use of each Google Earth feature is presented in Table 1.

Table 1. The suitability of each Google Earth feature

Feature	Suitability	Comments
Satellite Images	- Remote site study - Understanding terrain, land-use, and geomorphology	Images vary in resolution depending on location.
Historical Images	- Site study through time	Time when images are available is expected to increase in frequency.
Distance Ruler	- Reach length estimation - Channel width estimation	Distance should be validated with in-situ known measurements
DEM	- Useful for watershed delineation - Not suitable for channel depth estimation	DEM resolution seems to be a combination of SRTM90 and ASTER DEM data.
Street View	- Channel geometry estimation - Bed/bank material estimation	Should be used with careful judgement and as supplement to available data.

5 CONCLUSIONS

The utilization of Google Earth features in developing a hydraulic model has revealed some useful new features that would be helpful to the modeler. The benefits are that the modeler can obtain information remotely with a software that is free of charge. Google satellite images provide a good enough resolution for site studies and allows the modeler to familiarize him/herself with the project. Historical images provide information on the hydraulic features and behavior in the past. In this case, historical images reveal that channel improvement works has been carried out and changed the hydraulics of the river. The distance ruler tool in Google Earth can help the modeler measure distances to a certain degree of accuracy that helps with developing the model. While the resolution of the DEM of Google Earth is not high enough for delineation of the river geometry, it could be used for watershed delineation. Google Street View is perhaps the cherry on the cake in this situation where it gives the modeler a virtual, first-person perspective to estimate river geometry and bed/bank material. Overall it is recommended that for suitable projects, utilizing Google Earth may be beneficial in developing an effective model.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Dr. J. Abdullah at the MARA University of Technology (UiTM), A. Shafie at the Department of Irrigation of Drainage Malaysia, O.M. Chooi, and T. Lai for sharing knowledge and data for this study. The authors would also like to thank Dr. J. Ramirez, Dr. S. Kampf at Colorado State University for their guidance and input for the Master's project titled "Impact of the SMART Tunnel Outflow on the Hydraulics of the Kerayong River, Malaysia", the study from which this article is based on. The authors would also like to extend their gratitude to Google, Inc for permission of use of Google Earth and Google Earth Pro.

REFERENCES

Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L. & Weaver, J. (2010). Google Street View: Capturing The World At Street Level. *Computer*, 32-38.

ESRI. (2016). World Imagery. http://www.arcgis.com/home/item.html?id=10df2279f9684e4a9f6a7f08febac2a9 [Acessed 2016].

Rusli, N., Majid, M. R. & Din, A. H. M. (2014). Google Earth's derived digital elevation model: A comparative assessment with Aster and SRTM data. In *IOP Conference Series: Earth and Environmental Science*, Vol. 18, No. 1, 012065. IOP Publishing.

USACE(2010). *HEC-RAS Hydraulic Reference Manual.* US Army Corps of Engineers USACE HEC.

DEVELOPMENT OF SMOOTHED PARTICLE HYDRODYNAMICS METHOD TO SIMULATE WATER FLOW IN A CURVED CHANNEL

DWINANTI RIKA MARTHANTY[(1)]

[(1)] Civil Engineering Department, Universitas Indonesia, Depok, Indonesia,
dwinanti@eng.ui.ac.id

ABSTRACT

Smoothed particle hydrodynamics (SPH) is one most noticeable meshfree method and now become very popular, and particularly for free surface flows, it is a robust and powerful method for describing deforming media (Gomez-Gesteira, et al., 2010). SPH is a very promising method to answer 3D flow modeling in meander dynamics. Three basic characteristics of meandering process are flow structure, sediment transport, and morpho-dynamic. Helical flow as secondary current in flow structure plays the main role in characterizing of meandering dynamics. Meandering geometry is simplified as a curved channel boundary conditions. SPH procedures are developed from 3D fluid flow model and collision handling between water particles and a curved channel boundary conditions, as meandering geometry simplification. We used simple geometries based on Snell's law to represent basic particle responses to channel walls. We adapted SPH for nearly incompressible flow as an incompressible flow in a curved channel that is note bene. Viscosity plays the main role in initiating helical flow formation in the channel. Formation of helical flow is generated at downstream hemispheres part of the curved channel. This paper presents an application of SPH method to develop helical flow as a result of curvature, agreed with Camporeal et al. (2007), and even without sediment transport, agreed with Ferreira da Silva (2006) and Yalin (1993). Our contribution with this research is developing SPH method for modeling helical flow in a curved channel with the aim of simulating meandering dynamics.

Keywords: Smoothed particle hydrodynamics; free surface; water flow; curved channel; meandering dynamics.

1 INTRODUCTION

Since firstly introduced by (Gingold & Monaghan, 1977), and (Lucy, 1977), SPH gains popularity as a mesh-free method because it can overcome problems using fixed grid (Liu & Liu, 2003). The SPH method is modeled as an assembly of particles where the interaction zone is assumed to be around each of it. Thus, there is no need to describe all terms in governing equation on a fixed grid. The problem using grid arouses from the numerical diffusion as a consequence of the advection terms in the equations. On the free surface, large deformation yields severe numerical diffusion (Shao & Gotoh, 2005).

Einstein firstly explained the cause of the formation of meanders in 1926 where streams tend to flow in winding and turning course instead of following the downward slope as a result from Coriolis-force (Einstein, 1926). Even without bends, the helical movement still exists at cross-sections of its course. Meandering not only happens on alluvial streams but also on melting water channels on ice (Langbein & Leopold, 1966), wind tunnels (Beresh, Henfling, & Spillers, 2010), and submarine channels (Darby & Peakall, 2012).

Meandering channels research in general are separated, but still correlated, into two approaches: geomorphologic and fluid dynamics, where 3D flow modeling receives more attention for its ability to simulate helicoidal motion even though it is high in computational efforts and limited to simple geometry (Camporeal, Perona, Porporato, & Ridolfi, 2007). However, a computer's capability is growing hence meandering channel can be simulated with a powerful computational fluid dynamics (CFD) tools such as direct numeric simulation (DNS), large eddy simulation (LES), or $\kappa - \epsilon$ models (Wormleaton & Ewunetu, 2006).

CFD is traditionally using grid-based numerical methods, such as finite element methods (FEM) and finite volume methods (FVM), which have gained high acceptance ((Bates, Lane, & Ferguson, 2005), and (Wendt, 2009)). In spite of its success, grid-based method has limitations whenever dealing with free surface, deformable boundary, moving interface, and extremely large deformation and crack propagation because of the use of mesh. Complex geometry problems make generating mesh hard, expensive, and laborious (Liu & Liu, 2010). Meshfree methods have emerged as an alternative grid based methods to deliver better accuracy and stability numerical solutions for PDEs with all possible boundary conditions using particles (Liu & Liu, 2003).

Smoothed particle hydrodynamics (SPH) is one of the most noticeable meshfree methods and now have become very popular, and particularly for free surface flows, it is a robust and powerful method for describing deforming media (Gomez-Gesteira M. , Rogers, Violeau, Grassa, & Crespo, 2010). SPH main appeals are its ability to predict highly strained motions based on a set of particles, and its consistency with Lagrangian and Hamiltonian mechanics in terms of conservativity (Violeau, 2012). SPH applications for incompressible or

nearly incompressible flow in the last two decades are diverse involving dam breaks and plunging waves, gravity currents and multifluid phenomenon, bodies moving in fluids, non-Newtonian fluids, surface tension, and diffusion and precipitation (Monaghan J. J., 2012). Advanced hydraulics with SPH so far has been covered wave action upon waterworks, fish pass, floating oil spill containment boom, and dam spillway (Violeau, 2012).

SPH is also a hot topic in Computer Graphics (Kelager, 2006) including realistically animated fluids (Müller, Charypar, & Gross, 2003), fluid-fluid interaction (Muller et al., 2005), hydraulics erosion (Kristof, Beneš, Křivánek, & Šťava, 2009), and 2D shallow water simulation (Bender et al., 2011). SPH research in CFD, hydraulics, and computer graphics mostly do not focus on flow structures except in turbulence. Thus, SPH is a very promising method to answer 3D flow modeling in meander dynamics.

This research considers the most important characteristics of meandering process, which are flow structure, sediment transport, and morpho-dynamic. A curved channel boundary condition represents the simplification of meandering geometry. SPH procedures are developed from 3D fluid flow model and collision handling between water particles and curved channel boundary conditions, as meandering geometry simplification. This paper presents an application of SPH method to develop helical flow because of curvature and without sediment transport.

2 MEANDERING FLOW BASIC CHARACTERISTICS

In river modeling, one-dimensional model is often used due to its efficiency and applied in the study of long-term sedimentation problems in rivers. However, flows in curved channels give evidence of complex three-dimensional features that have big influences on sediment transport process. Thus, these phenomena must rely on three-dimensional model for all practical purposes. Here, the river scale is defined as reach scale where the domain is longer than the channel by a factor of at least 5 and up to 50 or more. Considering that reach scale is a common scale in management activities such as scour around bridge piers, in-channel sedimentation process, bank erosion and channel migration, thus it relates the flow and sediment transport around structures, and linkages strongly to morpho-dynamic and habitat process likewise the flow field itself (Bates et al., 2005). Development of meander dynamics model has to have a capability to simulate meander flow characteristic and sediment transport distribution patterns, or at least having the same capability as the finite element method (Wu, 2008). Here, the meander flow is characterized by having helical flow and coherent structures (bursts and sweeps), higher flow velocity at the outer banks and lower in the inner banks, sediment erosion at the outer banks and deposition in the inner banks, higher sediment concentration at the outer banks and lower in the inner banks (Ferreira da Silva, 2006).

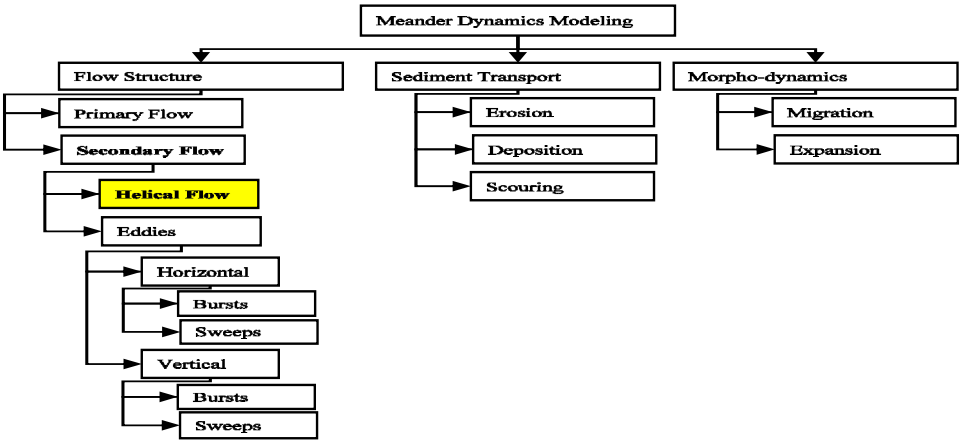


Figure 1. Basic Characteristics Meandering Process.

3 SMOOTHED PARTICLE HYDRODYNAMICS METHODS

3.1 Flow Equations

The Navier-Stokes equation for incompressible and isothermal fluid is presented by:

$$\rho \frac{d\mathbf{u}}{dt} = -\boldsymbol{\nabla} p + \mu \boldsymbol{\nabla}^2 \mathbf{u} + \mathbf{F}^{\mathbf{external}} \quad (1)$$

with ρ is mass density (kg/m3), u is vector velocity (m/s), t is time (second), p is pressure (N/m2), μ is kinematic viscosity (Ns/m2), and F is external force (N).

Equation (1) can be modified into:

$$\rho \frac{d\mathrm{u}}{dt} = \mathbf{F}^{\mathbf{internal}} + \mathbf{F}^{\mathbf{external}} = \mathbf{F} \quad (2)$$

Then for each particle I, the acceleration (m/s^2) becomes:

$$\mathbf{a}_i = \frac{d\mathbf{u}_\mathrm{i}}{dt} = \frac{\mathbf{F}_i}{\rho_i} \quad (3)$$

where $\mathbf{F}$ is a sum of force fields, internal forces $\mathbf{F}^{\mathbf{internal}}$ and external forces $\mathbf{F}^{\mathbf{external}}$.

Considered as internal forces, they are pressure force and viscosity force. For external forces, they are regarded as gravitation force, surface tension, and also buoyancy force.

SPH method uses velocity-pressure formulation to solve the momentum equation. Pressure is correlated to density in the thermodynamic equation of state. As introduced by Monaghan (1994) for free surface flow, the equation of state has form:

$$p = \mathrm{B}\left(\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right) \quad (4)$$

where B is coefficient where $B = \frac{200gH}{\rho\gamma}$, ρ_o is rest density (kg/m^3), and $\gamma = 7$.

If the pressure is known for each particle, at particle i, the pressure force is:

$$\boldsymbol{F}_i^{pressure} = -\nabla p(r_i) \quad (5)$$

Viscosity force at particle i with viscosity μ (Ns/m^2), is:

$$\boldsymbol{F}_i^{viscosity} = \mu\nabla^2\boldsymbol{u}(r_i) \quad (6)$$

Gravitation force is:

$$\boldsymbol{F}_i^{gravity} = \rho_i\boldsymbol{g} \quad (7)$$

The surface tension force is resulted from the force density spreading onto all potential particles.

$$\boldsymbol{F}_i^{surface} = -\sigma\nabla^2\mathrm{c_i}\frac{\mathbf{n_i}}{|\boldsymbol{n_i}|} \quad (8)$$

The buoyancy force is for gaseous fluids and caused by diffusion of temperatures. For isothermal fluid, an artificial buoyancy force can be used as:

$$\boldsymbol{F}_i^{buoyancy} = b(\rho_i - \rho_0)\boldsymbol{g} \quad (9)$$

If simulation is applied for isothermal water, then the buoyancy coefficient can be assumed as zero.

3.2 Particle Approximations

The mass-density at particle i is approximated as:

$$\rho_i = \sum_{j=1}^{N} m_j W(r_i - r_j, h) \quad (10)$$

The pressure force with symmetrical form can be written as:

$$\boldsymbol{F}_i^{pressure} = -\rho_i \sum_{j\neq 1}\left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right) m_j \nabla W(r_i - r_j, h) \quad (11)$$

The viscosity force with symmetrical velocity fields is defined as
:

$$\boldsymbol{F}_i^{viscosity} = \mu \sum_{j\neq 1}(\boldsymbol{u_j} - \boldsymbol{u_i})\frac{m_j}{\rho_j}\nabla^2 W(r_i - r_j, h) \quad (12)$$

The surface tension force at particle i with symmetrical from is approximated as:

$$\boldsymbol{F}_i^{surface} = -\sigma\left[\sum_{\mathrm{j}}\frac{\mathrm{m_j}}{\rho_\mathrm{j}}\nabla^2 W(r_i - r_j, h)\right]\frac{\left[\sum_{\mathrm{j}}\frac{\mathrm{m_j}}{\rho_\mathrm{j}}\nabla W(r_i - r_j, h)\right]}{|\mathrm{n_I}|} \quad (13)$$

where $|n_i| \geq l_i$, and $l_i > 0$ is some threshold concerning the particle concentration.

3.3 Smoothing Functions (Kernels)

The most common smoothing function (kernel) is Gaussian function as the best kernel assumption to interpret the first golden rule of SPH, according to (Monaghan J. J., 2005).

$$W(r,h) = \frac{1}{(2\pi h^2)^{\frac{3}{2}}} e^{-(\frac{|r|^2}{2h^2})} \ , \ h > 0 \tag{14}$$

where $|r|$ is scalar distance between particles. However, it is not always the best, where $W < 0$ for all r all particles within the domain contribute to the calculation. Consequently, the number of particles increases with the kernel computational cost in numerical error from particle approximation (Price, 2004).

Cubic spline kernels are often used for this reason, but in terms of computational accuracy, stability, and speed, (Müller, Charypar, & Gross, 2003) designed kernels that have second order interpolation errors. The kernels are all even and normalized. Zero kernels with disappearing derivatives at the boundary are conductive to stability. The kernel is called poly6, noticeable as a simple kernel, and preserve Gaussian bell curve (Kelager, 2006). More importantly according to (Müller, Charypar, & Gross, 2003), r is in form of square which can be calculated without computing its square roots.

$$W_{poly6}(r,h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - |r|^2)^3 \ , & 0 \leq |r| \leq h \\ 0 \ , & |r| > h \end{cases} \tag{15}$$

$$\nabla W_{poly6}(r,h) = -\frac{945}{32\pi h^9} r (h^2 - |r|^2)^2 \tag{16}$$

$$\nabla^2 W_{poly6}(r,h) = -\frac{945}{32\pi h^9} (h^2 - |r|^2)(3h^2 - 7|r|^2) \tag{17}$$

where $|r| = r_{ij}$ which is the particle distance between particle i and j.

But kernel poly6 has disadvantage, under high pressure, particles using this kernel are likely to develop clusters. As a result, the repulsion force is gone whenever particles move very close to each other since the kernel gradient becomes zero at the center. Therefore, we are using spiky kernel as proposed by (Desbrun & Gascuel, 1996), (Müller, Charypar, & Gross, 2003) and (Kelager, 2006). Gradient and Laplacian of spiky kernel are vanished at the boundary. This is necessary to generate repulsion forces.

$$W_{spiky}(r,h) = \frac{15}{\pi h^6} \begin{cases} (h - |r|)^3 \ , & 0 \leq |r| \leq h \\ 0 \ , & |r| > h \end{cases} \tag{18}$$

$$\nabla W_{spiky}(r,h) = -\frac{45}{\pi h^6} \frac{r}{\}r|} (h - |r|)^2 \ , \tag{19}$$

$$lim_{r \to 0^-} \nabla W_{spiky}(r,h) = \frac{45}{\pi h^6} \ , \ lim_{r \to 0^+} \nabla W_{spiky}(r,h) = -\frac{45}{\pi h^6}$$

$$\nabla^2 W_{spiky}(r,h) = -\frac{90}{\pi h^6} \frac{1}{|r|} (h - |r|)(h - 2|r|) \ , \tag{20}$$

$$lim_{r \to 0} \nabla^2 W_{spiky}(r,h) = -\infty$$

Particular kernel for viscosity forces is designed by (Müller, Charypar, & Gross, 2003) to get positive result of the Laplacian. Negative result of viscosity forces due to negative Laplacian kernel increases particles relative velocity, thus it can cause instability numerical computation. The relative viscosity can be damped by the viscosity forces on condition that the Laplacian is positive everywhere in the calculation domain (Kelager, 2006).

$$W_{viscosity}(r,h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{|r|^3}{2h^3} + \frac{|r|^2}{h^2} + \frac{h}{2|r|} - 1 \ , & 0 \leq |r| \leq h \\ 0 \ , & |r| > h \end{cases}, \quad lim_{r \to 0} W_{viscosity}(r,h) = \infty \tag{21}$$

$$\nabla W_{viscosity}(r,h) = \frac{15}{2\pi h^3} r \left(-\frac{3|r|}{2h^3} + \frac{2}{h^2} - \frac{h}{2|r|^3} \right) \ , \tag{22}$$

$$lim_{r \to 0} W_{viscosity}(r,h) = +\infty \ , \quad lim_{r \to 0} W_{viscosity}(r,h) = -\infty$$

$$\nabla^2 W_{viscosity}(r,h) = \frac{45}{\pi h^6} (h - |r|) \tag{23}$$

4 NUMERICAL EXPERIMENTAL SETUPS

4.1 Algorithm

Input parameters were fluid properties, common physics and SPH properties. Initial conditions were velocity and position for each particle at zero time, and determined as input model. From the input, total force and density for each particle was calculated. Total force equals to the summation of internal forces (pressure

and viscosity), and external forces (gravity and surface tension). Then, it computes acceleration with (3) from total force divided by density for each particle.

Afterwards, it updates new velocity and new position from the acceleration for each particle at each time step. Later, this new velocity and new position will be used to calculate new total force and new density for each particle at next time step. The new total force and new density will be applied to compute new acceleration, and so on until it reaches maximum time step in the end.

If we define space, for each new velocity and new position, it checks collision between particles and its boundary conditions before it goes to the next time step. The space is set as a curved channel, and the collision determines water flow movement, in this case is the primary flow.

4.2 Collision Handling at Curved Channel

4.2.1 Boundary Conditions

Meander morphology is a very complex geometry. Thus, we need to simplify it into a curved channel in order to separate the effect of helical flow formation from complex geometry. Collision handling works whenever there is contact with the walls; four straight walls and four quarter-circle walls. At t = t, coordinate of a particle i is $(x(i), y(i), z(i))$, and at t = t + dt, its coordinate becomes $(xt(i), yt(i), zt(i))$. A contact point is $(xc(i), yc(i))$ where a particle collides with a wall at any $zc(i)$. We consider that curved channel has very high wall. At this time, we use Snell's law where angle and velocity of incidence equals to angle and velocity refraction. Center of curved channel has coordinate at x- and y-coordinate $(xr, yr) = (2, 3)$, and any z-coordinate zr. Radius of outer hemisphere from the center is $rr = 2\,meter$, and of inner hemisphere is $rr2 = 1\,meter$. The bed of curved channel is set up at $zbase = 0\,meter$ for all x- and y-coordinates. Coordinate in x-direction in meter for each plane wall are $xbase1 = 0$, $xbase2 = 1$, $xbase3 = 3$, and $xbase4 = 4$. Angle of refraction is in radians, and depends on quadrant of particle coordinate.

Area of particle movement is wherever (1) $y(i)$ is less than yr then particle collision uses algorithm for plane walls, or (2) $y(i)$ is greater than or equivalent to yr thus particle collision utilizes algorithm for curved walls. A particle collides a plane wall whenever (1) $xt(i) \leq xbase1$, (2) $xbase2 \leq xt(i) \leq xbase3$, or (3) $xbase4 \leq xt(i)$. Calculation of contact point is determined as in the following,

$xc(i) = xbase$, depends where a particle collides (24)

$$yc(i) = y(i) - \left(\frac{(x(i)-xc(i))*(y(i)-yt(i))}{x(i)-xt(i)}\right) \quad (25)$$

$$\gamma = atan\left(\frac{yc(i)-yt(i)}{xc(i)-xt(i)}\right) \quad (26)$$

Collision condition at curved channel whenever radius of a particle relatively to the center of a curved channel is (1) greater than outer hemisphere radius $(rxt > rr)$, or (2) less than inner hemisphere radius $(rxt < rr2)$.

$$rxt = \sqrt{(xt(i) - xr)^2 + (yt(i) - yr)^2} \quad (27)$$

Contact point is located at hemisphere. There are two initial guesses $xc(i)$ and $yc(i)$. Coordinate $yc(i)$ is estimated between $y(i)$ and $yt(i)$. Computation of $yc(i)$ uses bisection method (Chapra & Canale, 2010) and (Cheney & Kincaid, 2008), afterward $xc(i)$ is calculated under condition, if

rxt > rr then,

$$xc(i) = xr - \left(((rr + yc(i) - yr)^{0.5}) * ((rr - yc(i) + yr)^{0.5})\right), (xt(i) \leq xr) \quad (28)$$

$$xc(i) = xr + \left(((rr + yc(i) - yr)^{0.5}) * ((rr - yc(i) + yr)^{0.5})\right), (xt(i) > xr) \quad (29)$$

rxt < rr2 then,

$$xc(i) = xr - \left(((rr2 + yc(i) - yr)^{0.5}) * ((rr2 - yc(i) + yr)^{0.5})\right), (xt(i) \leq xr) \quad (30)$$

$$xc(i) = xr + \left(((rr2 + yc(i) - yr)^{0.5}) * ((rr2 - yc(i) + yr)^{0.5})\right), (xt(i) > xr) \quad (31)$$

After contact point coordinates is set up, radius of contact point from the center of hemisphere is calculated only to check whether the radius falls right at one of hemisphere walls.

$$rc = \sqrt{(xc(i) - xr)^2 + (yc(i) - yr)^2} \quad (32)$$

Refraction angle computation depends on where the location of contact point. The angle detection varies by hemisphere radius, slope between contact point and curved channel center, and contact point quadrant.

If $rxt > rr2$ where a particle lies outside outer hemisphere then,

$$mc = \left(\frac{(rr-xc(i)+xr)^{0.5}}{2.*((rr+xc(i)-xr)^{0.5})}\right) - \left(\frac{(rr+xc(i)-xr)^{0.5}}{2.*((rr-xc(i)+xr)^{0.5})}\right) \quad (33)$$

Else if $rxt < rr2$ where a particle remains inside inner hemisphere then,

$$mc = \left(\frac{(rr2-xc(i)+xr)^{0.5}}{2.*((rr2+xc(i)-xr)^{0.5})}\right) - \left(\frac{(rr2+xc(i)-xr)^{0.5}}{2.*((rr2-xc(i)+xr)^{0.5})}\right) \quad (34)$$

Next steps are calculation of refraction angle γ:

$$x1 = xr \quad (35)$$
$$y1 = yc(i) + mc * (x1 - xc(i)) \quad (35)$$
$$\alpha = atan\left(\frac{y1-yc(i)}{x1-xc(i)}\right) \quad (36)$$
$$\beta = atan\left(\frac{yt(i)-yc(i)}{xt(i)-xc(i)}\right) \quad (37)$$
$$\alpha = atan\left(\frac{y1-yc(i)}{x1-xc(i)}\right) \quad (38)$$
$$\theta = \beta - \alpha \quad (39)$$
$$\eta = atan\left(\frac{y(i)-yc(i)}{x(i)-xc(i)}\right) \quad (40)$$

The challenging part is where we have to identify the particle position quadrant, because previously we have to determine its incidence angle quadrant as relative to its contact point. After refraction angle of each particle is calculated, each particle position is updated where:

$$rcxt(i) = \left(\left(xt(i) - xc(i)\right)^2 + \left(yt(i) - yc(i)\right)^2\right)^{0.5} \quad (41)$$
$$xt(i) = xc(i) + rcxt(i) * \cos(\gamma) \quad (42)$$
$$yt(i) = yc(i) + rcxt(i) * sin(\gamma) \quad (43)$$
$$zt(i) = 2 * zbase - zt(i) \quad (44)$$
$$vxt(i) = ((vxt(i) ** 2 + vyt(i) ** 2) ** 0.5) * cos(\gamma) \quad (45)$$
$$vyt(i) = ((vxt(i) ** 2 + vyt(i) ** 2) ** 0.5) * sin(\gamma) \quad (46)$$
$$vzt(i) = -1 * vzt(i) \quad (47)$$

4.2.2 Initial Conditions, Parameters and Properties

The free surface water system is defined as shallow water and long wave (Ji, 2008). Time step is defined for each time integration calculation and based on number of Courant-Friedrichs-Lewy (CFL) to check the numerical stability in the computation (Kao & Chang, 2012) and (Chapra C. S., 1997).

$$\Delta t = CFL * \min\left(\frac{h}{c_i + |\vec{v}_i|}\right) \text{ for } 0 \leq CFL \leq 1 \quad (48)$$

where h is the SPH computation domain radius or smoothing length, c_i is gravity wave speed of propagation, and $|\vec{v}_i|$ is the magnitude of velocity for particles.

There are 4811 particle numbers. Initial time, set at $t = 0$, all particles forms as a cube with dimension of 1 meter width by 1 meter length by 1 meter height, and weighs 0.9622 kilogram. Initial velocity is set at variation afterwards in order to see particles flow behavior. Particle number independence test is done by using particle number resolution at initial from 1000 to 9622 particles. The solution is independent to particles number at 4811 particles. Parameters in numerical simulation are common, fluid, and SPH properties as shown in **Error! Reference source not found.**. Noted that simulations use 0.01 second for each time step. This value is chosen after several trials from 0.001 second up to 1 second, and checked with Courant number in equation (48). If the time step is smaller than 0.01, the simulations take a longer computation time especially for big particle numbers. If the time step is bigger than 0.01, the simulation computations become unstable and stuck in the middle of running program. Discretizing fluid system into large particle number results in consistent accuracy but also high runtime computation as stated by (Liu & Liu, 2003). In this research, we varied particle number from 10 up to 4000 particles. Nevertheless, this simulation uses 4811 particle numbers to gain the accuracy of the solution. Fluid properties and modeling parameter are simulated through several loops in order to see how the system responds under parameters variation of density and pressure.

5 RESULTS AND DISCUSSIONS

5.1 Results

SPH program run water simulations in a curved channel for inviscid, viscous flow, vorticity, mass, viscosity, gravity, and surface tension. Simulation runs from zero to fifteen seconds. Initial condition was inviscid flow where $vyo = 0.8\,m/s$. At first in straight channel, particles moved straight forward. When they entered in the curved part, they collided and turned directions as caused by collisions with curved walls. The collisions transferred momentum from outer to inner curved wall, then bounced back, and keep rebounding until they exit the curved part.

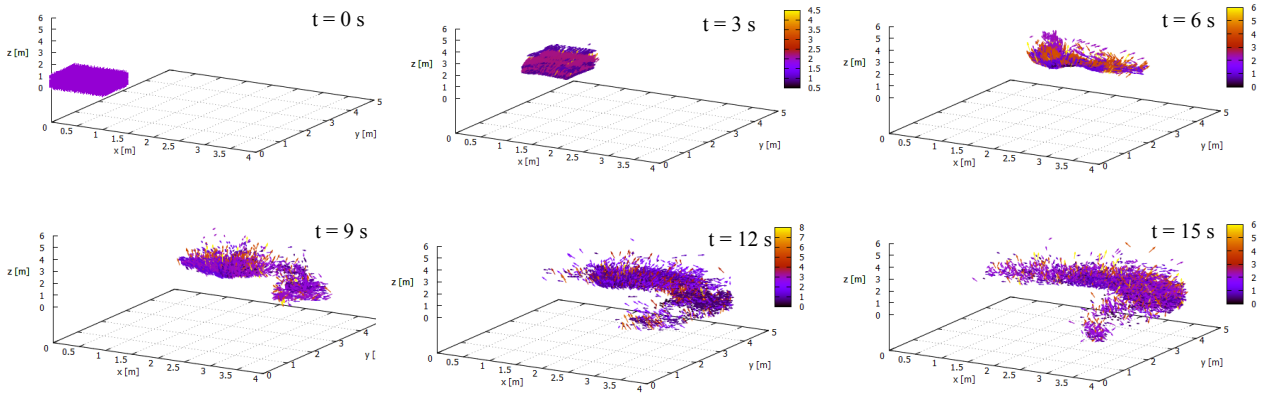


Figure 2. Particle flows with water properties from t = 0, 3, 6, 9, 12, and 15 second (from upper left in clockwise direction); colors represent velocity [m/s]

This pattern may appear as helical but if a particle flow path were to be looked closely, the refraction angle is to sharp, as in Figure 2. This can be happened due to collision handling method. At this time, the contact between particle and boundary is governed by Snell's law.

Figure **3** shows a particle flow path that run from 0 to 15 seconds from the simulation in preceding Figure 2. The pattern is quite satisfactory in a plan view, but in 3D view we see a hooping bug movement or a particle jumping in vertical direction in upper left

Figure **3**. This is happened because the magnitude of gravity force is rather dominant than pressure force, viscosity force, and surface tension force. The ratio of gravity force to other force ranges between 100 to 150 times bigger.

In order to solve the jumping particles movement, the main parameters in vertical forces are gravity and particle mass, and these will be varied. Firstly, we reduce the gravity acceleration into 1% of its magnitude. Secondly, we set zero gravity, and lastly, we decrease the mass into 10% mass as displayed in

Figure **3**. In upper right figure, since we use 1% coefficient of gravity, the particle movement in time is no longer jumping. However, if we set zero gravity, the particle movement is expanding and ascending towards the initial location as in lower left figure. The last one, we reduce the mass into 10%, the particle moves randomly without any pattern as in lower right figure. We can conclude that flow simulation using water properties develops velocity predominantly governed by gravity. Despite the fact above, we have settled that all parameter magnitudes are within sensitivity range and none is dominant. Consequently, a question emerges why the magnitude is imbalanced between gravity forces with other forces. Therefore, we need to consider the effect of gravity magnitude, and each fluid properties that were employed in momentum equation. We run eight cases of water flow simulations to test the which parameter(s) is(are) sensitive in generating helical flow.

SPH simulations with initial viscous flow have shown that viscous effect plays significant role in the formation of helical flow. In order to magnify the strength of helical formation, the simulation adds both vorticity and viscous effects at initial velocities. Particles move faster, smoothly swirl through the hemisphere, and flow towards the downstream. As expected, the helical flow starts to form in the downstream of the hemispheres.

From discussion above, we just focus in the hemisphere part of curved channel. Then we look farther at the downstream of hemisphere. Helical flow is formed after water flows exit the hemisphere of Figure 4. Some particles move from channel bottom towards outer wall then they swirl up backward against outer wall. This motion initiates helical flow. The helical shape can be improved by varying magnitudes of initial vorticity and viscous flow.

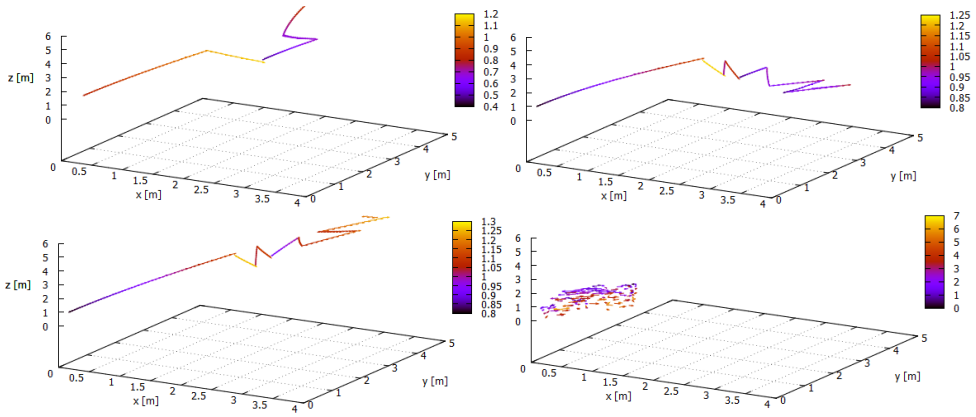


Figure 3. Particles collision handling (upper left) with gravity, (upper right) with 1% * gravity, (lower left) without gravity, and (lower right) with 10% mass; colors represent velocity [m/s]

5.2 Discussions

Wang and Liu (2015) described their findings from experimental investigation of flow structures in a bend flume. We use these experimental results to compare with SPH results and RMA results for verification. The channel geometry in SPH is adjusted with respect to their experimental set up. Width channel is 0.30 meter, initial depth is set at 0.16 meter, center coordinate of the hemispheres is (0.80, 0.80, z) meters, inner hemisphere radius is 0.50 meter, and outer hemisphere radius is 0.80 meter.

At first, when coefficient gravity was used, particles were flowing and expanding like gas. At the cross section intended in Figure 4, depth of particles flow became 11 meter instead of 0.2 meter. It was because the gravity force was too weak due to the coefficient we set at the equation. Then, coefficient is taken out from the gravity force equation. Afterwards, particles behave like water as in Figure 4 where the average depth is less than 0.2 meter, its velocity range close range between 0.50 m/s and 2.50 m/s.

Particles start to move in helical formation even though it is not strong. From the bottom, some particles move towards outer wall then they turn back against outer wall. From particles calculation, at t = 2.35 s, average velocity is 1.29 m/s with minimum 0.04 m/s, and maximum 2.82 m/s. Average relative pressure is 0.034 Pa, with minimum 0.003 Pa and maximum 0.188 Pa. Average density is 4383.8 kg/m3, with minimum 1392.6 kg/m3 and maximum 19798.9 kg/m3. At this point, that the particles calculation can be considered to behave close to the experiment result as in Figure 4.

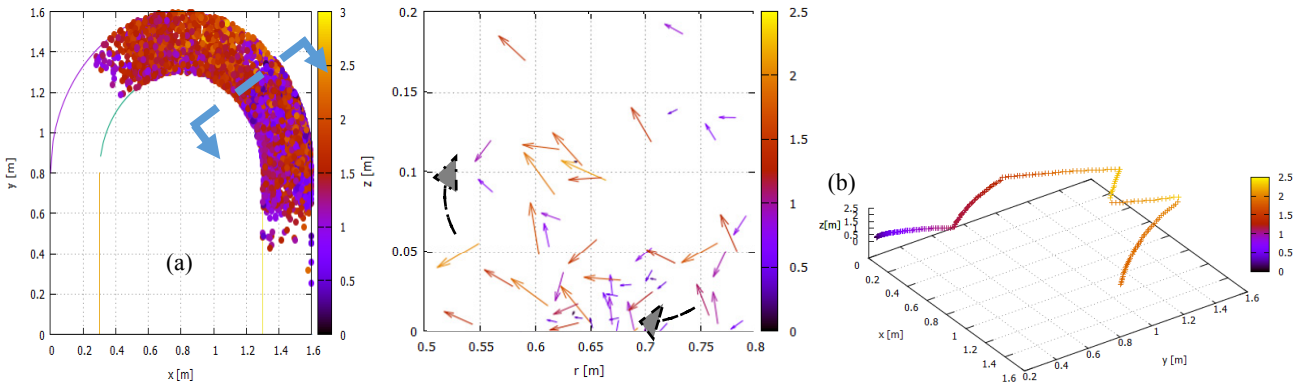


Figure 4. Helical formation with SPH simulation, initial vorticity viscous flow, and results at t = 2.35 seconds, (b) Checking angle refraction for a particle interaction; colors represent velocity [m/s]

In Figure 5, high pressure and high density particles coordinates are located with the same low velocity particle coordinates. This is reasonable since crowded particles push particles to its surrounding but it is limited to space. Particles movement was decelerated. Thus, particles have high pressure.

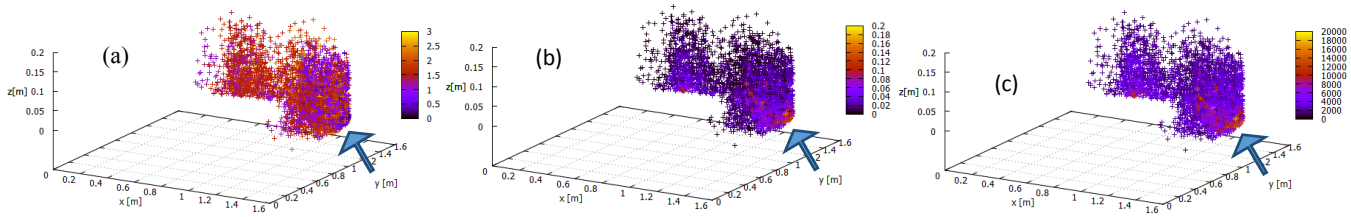


Figure 5. Particles value of: (a) velocity [m/s], (b) pressure [Pa], and (c) density [kg/m3]

Simulation with RMA (Resources Model Association is run with the same setting with Wang and Liu's (2015) experiments. Here, we used RMA-10 version 87e in 20 November 2012) (King, 2012). Helical flow pattern from SPH model in Figure 4 has the same pattern with helical flow patterns from RMA model in Figure 6 (a), and follows the pattern with the experiment results in Figure 6 (b). Helical formation with SPH, RMA, and even in the experiments is not fully in circular shape but rather has tendency to form like a spinal cord. At the bottom channel, velocity vectors move toward outer bank then sweeps back to inner bank near water surface, and forms helical motion. This helical flow pattern is consistent with the patterns from very recently experiment investigation by Wang and Liu (2015) in Figure 6 (b), and theoretical sketch of helical flows in a curved channel by Wormleaton and Ewunetu (2006).

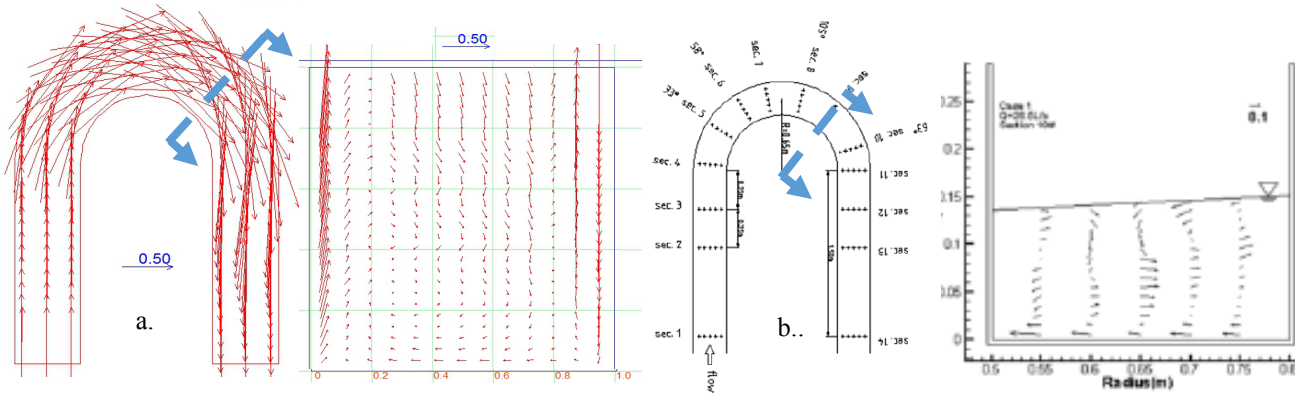


Figure 6. (a) Helical formation with RMA simulation at t = 1.5 seconds; colors represent velocity [m/s], (b) Helical formation from experiment investigation of flow structures (Wang & Liu, 2015)

6 CONCLUSIONS

Initial velocities drift the fluid particles and work as an advective term, while the densities and pressures expand the fluid's volume and act as a dispersive term. Up to this point, the 3D flow SPH program gives a stable result as expected from fluid properties. In running time integration based on Verlet method, the program runs smoothly for each given time. The basic of SPH is presented in the order from its formulations, integral representations and kernel functions, and particle approximations. The SPH formulations are developed for 3D fluid flow and sedimentation transport equations. The logical frame of the program development is drawn in an algorithm, and the program code is written in FORTRAN language. The numerical experimental results show that the program has capability to simulate the basic behavior of fluid properties as the basis characteristics of meandering river. The calculation is stable during the time looping for the given initial particles conditions. The very basic characteristic in meandering dynamics is helical flow, which is initiated by adding up viscous flow and vorticity at initial conditions. Its formation is generated at downstream hemispheres part of the curved channel. Viscous flow plays the main role in the development of helical flow. The helical flow pattern is consistent with the patterns from very recently experiment investigation by Wang and Liu (2015), and theoretical sketch of secondary flows in a curved channel by Wu (2008) and Wormleaton & Ewunetu (2006). SPH method is able to predict realistically helical flow as a result of curvature. agreed with Camporeal et al. (2007), and even without sediment transport, agreed with da Silva (2006) and Yalin (1993).

SPH becomes popular in Hydraulics community. Our contribution with this research is developing SPH method for modeling helical flow in a curved channel with the aim of simulating meandering dynamics. This is aligned with the advancement of SPH in Hydraulics. Four grand challenges in SPH applications in Hydraulics, according to SPHERIC community (Violeau & Rogers, 2015), are convergence, numerical stability, boundary conditions, and adaptivity. This research participates in the two of the SPH challenges; (1) boundary conditions where we used simple geometries based on Snell's law to represent basic particle responses to channel walls, and (2) adaptivity where we adapted SPH for nearly incompressible method for basic hydraulics phenomenon in a curved channel that is note bene an incompressible flow.

REFERENCES

Bates, P. D., Lane, S. N. & Ferguson, R. I. (Eds.). (2005). *Computational Fluid Dynamics: Applications in Environmental Hydraulics*. John Wiley & Sons.

Beresh, S. J., Henfling, J. F. & Spillers, R. W. (2010). Meander of a Fin Trailing Vortex and The Origin of Its Turbulence. *Experiments in fluids*, *49*(3), 599-611.

Bender, J., Erleben, K. & Galin, E. (2011). Sph based shallow water simulation. *Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS*.

Camporeal, C., Perona, P., Porporato, A. & Ridolfi, L. (2007). Hierarchy of Models for Meandering Rivers and Related Morphodynamic Processes. *Reviews of Geophysics*, 45(1), 1-28.

Chapra, C. S. (1997). *Surface Water-Quality Modeling (International ed.)*. McGraw-Hill.

Chapra, S. C. & Canale, R. P. (2010). *Numerical Methods for Engineers (Sixth ed.)*. New York: McGraw-Hill.

Cheney, W. & Kincaid, D. (2008). *Numerical Mathematics and Computing (Sixth ed.)*. Thomson Brooks/Cole.

Darby, S. E. & Peakall, J. (2012). Modelling the Equilibrium Bed Topography of Submarine Meanders That Exhibit. *Geomorphology*, 163-164, 99-109.

Desbrun, M. & Gascuel, M. -P. (1996). Smoothed Particles: A New Paradigm for Animating Highly Deformable Bodies. In *Computer Animation and Simulation '96*. Springer Vienna, 61-76.

Einstein, A. (1926). The Cause of the Formation of Meanders in the Courses of Rivers and of the So-Called Baer's Law. *Die Naturwissenschaften*, 14(11), 223-224.

Ferreira da Silva, A. M. (2006). On Why and How Do Rivers Meander. *Journal of Hydraulic Research*, 579-590.

Gingold, R. A. & Monaghan, J. J. (1977). Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars. *Monthly Notices of the Royal Astronomical Society*, 181(3), 375-389.

Gomez-Gesteira, M., Rogers, B. D., Dalrymple, R. A. & Crespo, A. C. (2010). State-of-The-Art of Classical SPH for Free-surface Flows. *Journal of Hydraulic Research*, 48(S1), 6–27.

Gomez-Gesteira, M., Rogers, B. D., Violeau, D., Grassa, J. M. & Crespo, A. J. (2010). Foreword: SPH for Free-Surface Flows. J*ournal of Hydraulic Research,* 48(S1), 3-5.

Ji, Z. G. (2008). *Hydrodynamics and Water Quality: Modeling Rivers, Lakes, and Estuaries*. John Wiley & Sons

Kao, H. -M. & Chang, T. J. (2012). Numerical Modeling of Dambreak-Induced Flood and Inundation Using Smoothed Particle Hydrodynamics. *Journal of Hydrology*, 448-449, 232-244.

Kelager, M. (2006). *Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics,* University of Copenhagen: Department of Computer Science.

King, I. (2012). *A Finite Element Model for Stratified Flow RMA-10 Users Guide (Version 8.7e ed.)*. Sydney: Resource Modelling Associates.

Kristof, P., Beneš, B., Křivánek, J. & Šťava, O. (2009). Hydraulic Erosion Using Smoothed Particle Hydrodynamics. *Eurographics*, 28(2), 219 - 228.

Langbein, W. B. & Leopold, L. B. (1966). *River Meanders - Theory of Minimum Variance (Geological Survey Professional Paper 422 H ed.)*. Washington: US Government Printing Office.

Liu, G. R. & Liu, M. B. (2003). *Smoothed Particle Hydrodynamics: a meshfree particle method*. World Scientific.

Liu, M. B. & Liu, G. R. (2010). Smoothed Particle Hydrodynamics (SPH): An Overview and Recent Developments. *Archives of Computational Methods in Engineering*, 17(1), 25-76.

Lucy, L. B. (1977). A Numerical Approach to the Testing Of The Fission Hypothesis. *The Astronomical Journal*, 82(12), 1013-1024.

Monaghan, J. J. (1994). Simulating Free Surface Flows with SPH. *Journal of Computational Physics*, 110, 399-406.

Monaghan, J. J. (2005). Smoothed Particle Hydrodynamics. *Reports on progress in physics*, 68(8), 1703.

Monaghan, J. J. (2012). Smoothed Particle Hydrodynamics and Its Diverse Applications. *Annual Review of Fluid Mechanics*, 44, 323-346.

Müller, M., Charypar, D. & Gross, M. (2003). Particle-Based Fluid Simulation for Interactive Applications. Eurographics/SIGGRAPH Symposium on Computer Animation.

Müller, M., Charypar, D. & Gross, M. (2003, July). Particle-based Fluid Simulation for Interactive Applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 154-159.

Müller, M., Solenthaler, B., Keiser, R. & Gross, M. (2005). Particle-Based Fluid-Fluid Interaction. Eurographics/ACM SIGGRAPH Symposium on Computer Animation, 1-7.

Müller, M., Solenthaler, B., Keiser, R. & Gross, M. (2005, July). Particle-based fluid-fluid interaction. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 237-244.

Papanicolaou, A. N., Elhakeem, M., Krallis, G., Prakash, S. & Edinger, J. (2008). Sediment Transport Modeling Review—Current and Future Developments. *Journal of Hydraulic Engineering*, 134(1), 1-14.

Price, D. J. (2004). Magnetic fields in Astrophysics, *PhD Thesis*. University of Cambridge, UK.

Raudkivi, A. J. (1998). *Loose boundary hydraulics*. CRC Press.
Shao, S. & Gotoh, H. (2005). Turbulence Particle Models for Tracking Free Surfaces. *Journal of Hydraulic Research*, 43(3), 276-289.
Violeau, D. (2012). *Fluid Mechanics and the SPH Method: Theory and Applications*. Oxford University Press.
Violeau, D. & Rogers, B. D. (2015). The SPHERIC Community. *Hydrolink*, 73.
Wang, X. & Liu, X. (2015). Experimental Investigation of Flow Structures and Bed Deformation with Small Width-to-Depth Ratio in a Bend. *Korean Society of Civil Engineers*, 1, 1-12.
Wendt, J. F. (Ed.). (2009). *Computational Fluid Dynamics: An Introduction (Third ed.)*. Berlin, Heidelberg: Springer-Verlag.
Wormleaton, P. R. & Ewunetu, M. (2006). Three-dimensional k–∈ numerical modelling of overbank flow in a mobile bed meandering channel with floodplains of different depth, roughness and planform. *Journal of Hydraulic Research*, 44(1), 18-32.
Wu, W. (2008). *Computational River Dynamics*. London, UK: Taylor and Francis Group.
Yalin, M. S. (1992). *River Mechanics*. Ontario: Pergamon Press Lt.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

APPLYING OBJECT-ORIENTED DESIGN CONCEPTS FOR THE COUPLED NUMERICAL SIMULATION OF SHALLOW WATER FLOW AND RELATED PROCESSES

ILHAN ÖZGEN[(1)], JIAHENG ZHAO[(2)], FINN AMANN[(3)] & REINHARD HINKELMANN[(4)]

[(1,2,3,4)] Chair of Water Resources Management and Modeling of Hydrosystems, Technische Universität Berlin, Berlin, Germany, ilhan.oezgen@wahyd.tu-berlin.de

ABSTRACT

In a broad range of applications of hydro- and environmental system modeling, the flow field is computed by the shallow water equations. The flow field is then used to compute the physical processes that depend on it, e.g., transport of a passive tracer, sediment transport and morphodynamics or infiltration. Thus, a scientific code for the computation of these type of hydro- and environmental problems is required to be flexible and extendable such that different processes can be added or removed to the existing code with reasonable effort. Object-Oriented Programming (OOP) concepts that enable abstraction by class encapsulation, polymorphism and class inheritance, are very suitable to design a code that fulfills the aforementioned requirements. In this contribution, we present the in-house scientific modeling framework Hydroinformatics Modeling System (hms), developed at the Chair of Water Resources Management and Modeling of Hydrosystems, Technische Universität Berlin, as an example of OOP in application. We discuss the design patterns used in the framework and show the advantages of the OOP approach in test cases that involve coupled processes. Firstly, contaminant transport in an idealized section of Panke river, Berlin, is simulated. Here, the contaminant does not influence the flow and the coupling is only in one way. In a second test case, we couple sediment transport and morphodynamics with the shallow water flow. This is a more complex case, because the transported sediment concentration acts as a momentum sink term in the shallow water flow. Thus, the processes interact with each other in both directions. We conclude that while OOP slightly increases the computational cost and the complexity of the code, applying the OOP concepts results in the code that allows easy implementation and coupling of different physical processes in an existing framework.

Keywords: Shallow water model; coupled modeling; object-oriented programming; java for scientific computing; design patterns.

1 INTRODUCTION

The depth-averaged two-dimensional shallow water equations are used in a broad range of simulations concerning hydro- and environmental systems. Some examples are the modeling of river hydraulics coupled with the spreading of contaminants or habitats (Lange et al., 2015; Matta et al., 2016), rainfall-runoff with infiltration (Mügler et al., 2011; Simons et al., 2014), coupled with subsurface flow (Liang et al., 2007; Viero et al., 2014) and dam-break flow simulations with sediment and bedload transport and morphodynamics (Cao et al., 2004; Zhao et al., 2016).

As pointed out in Busse et al. (2012), traditional scientific code is usually targeting computational efficiency and is limited to compute a limited number of processes. Adding new physical processes into these type of codes is usually difficult and might require rewriting a large part of the code. However, the interdisciplinary and complex nature of modeling hydro- and environmental systems requires modeling approaches that enable adding physical processes from different scientific disciplines, e.g., implementing sediment transport and morphodynamics into an existing shallow water model code, with a reasonable amount of effort. Thus, it is desirable that a large part of the code is implemented in a generalized way and encapsulated such that it can be applied to more than one process. In software engineering, this is often referred to as *code modularity* and *reusability* (Tchon, 1995).

Object-Oriented Programming (OOP) concepts enable to obtain modularity and reusability by *class encapsulation*, *polymorphism* and *class inheritance* (Katsurayama et al., 2004). In addition, techniques such as *generic programming* provide further tools for abstraction and generalization (Stroustrup, 2007). The idea of applying OOP to scientific code for the solution of partial differential equations has been suggested by several researchers (Tchon, 1995; Cambier and Gazaix, 2002; Katsurayama et al., 2004; Rodríguez-Gómez et al., 2004; Heng and Mackie, 2009; Busse et al., 2012; Simons et al., 2014) as it gives the code flexibility, because different objects, i.e. processes, can be implemented into the framework without the need of extensive modification. As Katsurayama et al. (2004) note, the choice of class extraction, i.e. the decision of what type of objects to create, is very significant. Here, a poor choice often leads to complicated code which runs slowly and is difficult to read and maintain. Thus, special care has to be given here.

In contrast, non-object-oriented programming, also referred to as *procedural programming*, enables abstraction only at a functional level (Cambier and Gazaix, 2002). It is generally agreed on that procedural

programming yields more computationally efficient code that is more difficult to extend and maintain (Tchon, 1995; Cambier and Gazaix, 2002; Katsurayama et al., 2004). In Katsurayama et al. (2004), for the specific case of a shock-tube problem for the two-dimensional Euler equations, the OOP code is 7 times slower than the procedural code.

As an example of OOP, we present and discuss the in-house scientific modeling framework Hydroinformatics Modeling System (hms), which is being developed at the Chair of Water Resources Management and Modeling of Hydrosystems, Technische Universität Berlin, since 2005. The hms is written entirely in the Java programming language in an object-oriented way. The aim of hms is to enable the coupling of different physical processes for research purposes, thus its main priority is to enable a comfortable way to implement new processes and technologies rather than computational efficiency. However, it is noted that if the computational efficiency is low, the code can also not be considered as suitable for research. The main issue then becomes to the finding of a good trade-off between flexibility and efficiency.

The aim of this paper is to present an OOP concept-based model design for the coupled simulation of shallow water flow and related processes. The paper is structured as follows: the abstractions and design patterns used in hms are presented. Then, we show the advantages of OOP concepts in test cases that involve coupled processes: contaminant transport in an idealized river system and dam-break flow coupled with sediment transport and morphodynamics. Finally, conclusions are given.

2 NUMERICAL SCHEME

The depth-averaged two-dimensional shallow water, tracer transport and morphodynamics equations presented in Cao et al. (2004) are solved using a cell-centered second order MUSCL scheme (Hou et al., 2013), with an explicit two-stage Runge-Kutta scheme in time. Numerical fluxes are calculated using a Harten, Lax and van Leer approximate Riemann solver with the contact wave restored (HLLC) (Toro et al., 1994).

3 OBJECT-ORIENTED PROGRAMMING

3.1 Geometry and mesh construction

In hms, all geometry objects are derived from the *geometry*-class. The most fundamental geometry objects are the *point*-object, the *line*-object, the *polyline*-object and the *polygon*-object, cf. Figure 1 (left).

In Busse et al. (2012), several existing scientific software codes are analyzed and the following similarities are pointed out: all codes run simulations on a mesh in order to update state variables, whereby a mesh can be abstracted as a collection of elements and variables can be abstracted as vectors. Consequently, the *cell*-object and the *edge*-object are derived from the *element*-class. As shown in Figure 1 (right), a cell can either be a triangular or a quadrilateral cell. Since the mesh is considered as a collection of element-objects, it can consist of either triangular or quadrilateral cells. Neighborship relations are stored at the edge-elements. This way, the topology of the mesh is determined independently from the cell type. Since each edge is also an element, state variables can also be stored at the edges. This has been utilized, e.g., in Özgen et al. (2016), where a specialized form of the shallow water equations has been implemented in hms that requires to store porosity terms at each edge. In addition to state variables, each element also stores its centroid, which is a point-object.

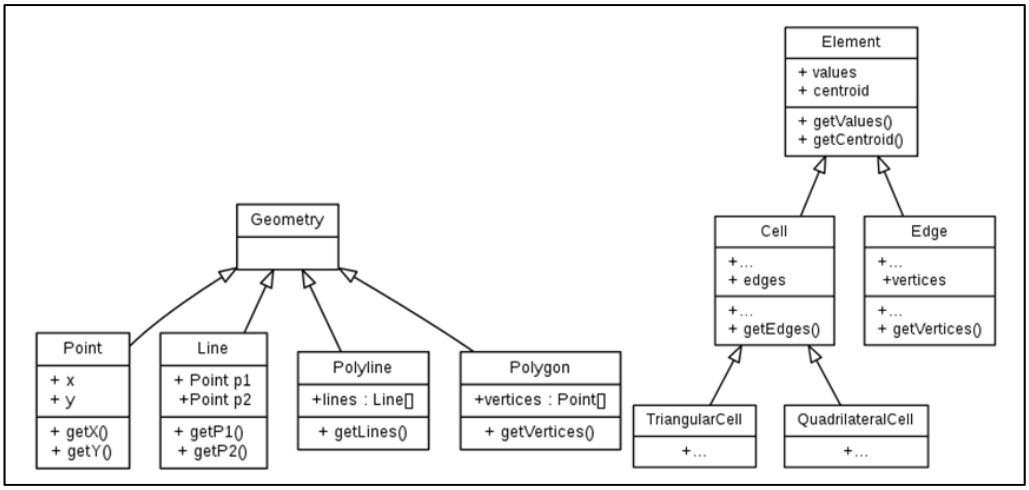


Figure 1. Class inheritance in *geometry*-objects (left) and *element*-objects (right).

3.2 Layer-based design concept for data structure

Simons et al. (2014) pointed out that physical processes usually can be written in the form of conservation or balance laws that can be solved with the same numerical methods, e.g. a Godunov-type finite

volume scheme (Godunov, 1959). This fact is then exploited to encapsulate different physical processes as separate objects called *layer*s. The layer-object describes the state variables of a physical process, cf. Figure 2 (top). For example, if the layer describes shallow water flow, it contains the state variables water depth, water elevation, bed elevation, flow velocity and unit discharge. The tracer transport layer contains the state variables water depth, flow velocity, unit discharge, concentration and tracer discharge. In a coupled transport simulation, the values for water depth, flow velocity and unit discharge would be computed in the shallow water flow layer and then mapped to the tracer transport layer. Note that in the layer-objects, no discrete values for the state variables are assigned. The layer-object solely declares which variables exist in the physical process and stores indices for the variables (integer values). The discrete values of and the relationship between the state variables are defined in the *value*s-objects, which are defined for each layer separately. During the computation, the values of the state variables can then be accessed by the *getValues(int)*-method of the values-object, where the input is the variable index defined in the layer-object, cf. Figure 2 (bottom).

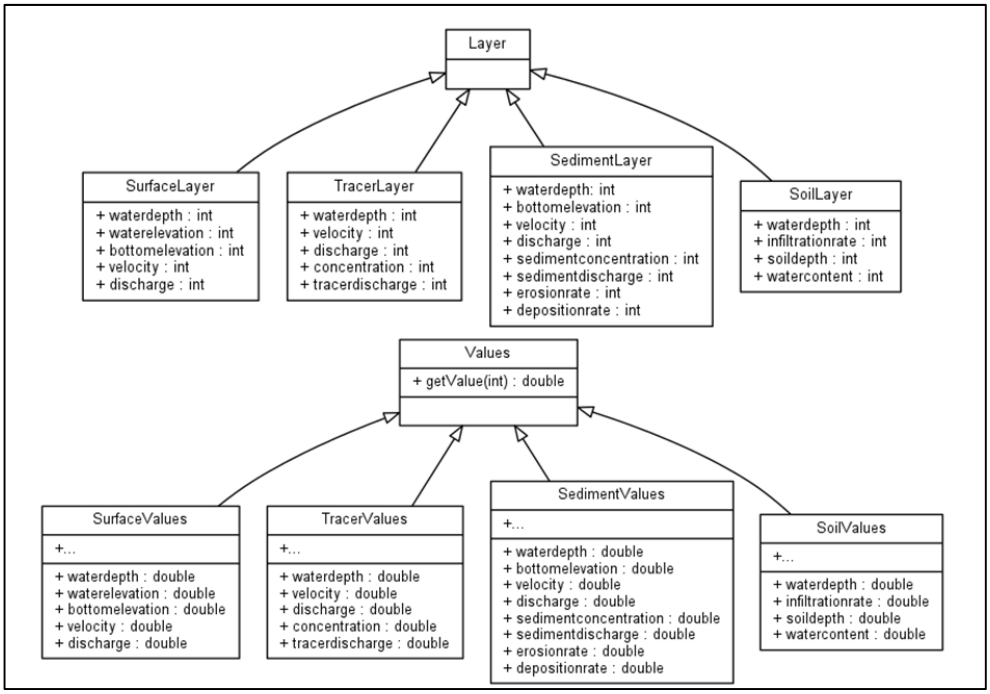


Figure 2. Class inheritance in *layer*-objects (top) and *values*-objects (bottom).

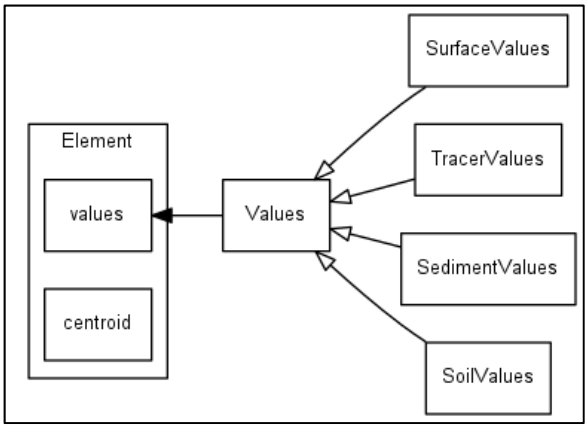


Figure 3. *Strategy* pattern applied to *values*-objects.

In hms, a *strategy* pattern was then applied to specify the type of state variables inside the element as shown in Figure 3. We note that each values-object has different state variables, as shown in Figure 2 (bottom) and since the element-object does not need to know what specific class of values it contains, exchanging the state variables, i.e. exchanging physical processes, becomes easy (Katsurayama et al., 2004). We acknowledge that the proposed data encapsulation may introduce additional cost to access data. This is a known issue of OOP (Cambier and Gazaix, 2004), however, the benefit of the additional flexibility and readability of the code which reduces maintenance cost may outweigh the performance penalty.

3.3 Generalized numerical scheme and solver for updating state variables

A generalized numerical scheme to solve conservation laws can be abstracted as follows:

1. initialize state variables;
2. preparation step prior to global loop;
3. compute values of conservative variables;
4. compute flux terms;
5. compute source terms;
6. update independent state variables;
7. update dependent state variables (required if there are two layers that depend on each other).

Using this, a general *scheme*-interface can be implemented, and different numerical schemes can be derived by class inheritance as shown in Figure 4 (left).

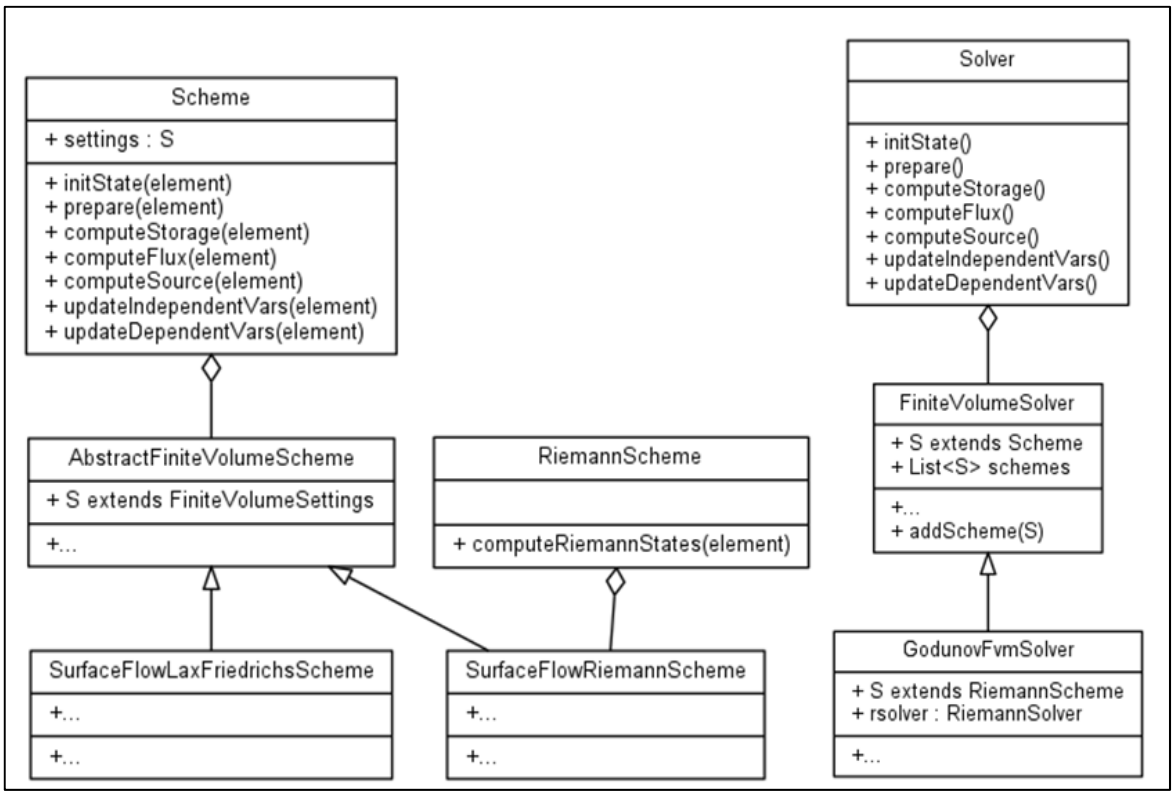


Figure 4. Class inheritance in *scheme*-objects (left) and in *solver*-objects (right).

We note that in Figure 4 (left), the Lax-Friedrichs scheme does not need to implement the Riemann scheme interface, because it does not require the computation of Riemann states.

In hms, the scheme object operates locally on the cells. The global loop over the elements in the whole mesh was carried out by the *solver*-object that calls the functionalities of the scheme-object while iterating over the entire mesh. The class inheritance of the solver-object is shown in Figure 4 (right). Using generic programming, a list of generic schemes was created (in Java-syntax *List<S extends Scheme> schemes*). Using a *strategy* pattern, schemes were added and removed easily from the solver-object. This way, one can add a shallow water flow scheme to the list, then add a tracer flow scheme to the list, and the solver would solve the governing equations for each scheme. As a last remark, we note that several approximate Riemann solvers exist, e.g., HLLC Riemann solver (Toro et al., 1994) and Roe Riemann solver (Roe, 1981), that are encapsulated as separate objects. Again, a strategy pattern was applied to the *GodunovFvmSolver* (Figure 4 (right)), to enable easy exchange of approximate Riemann solvers.

3.4 Generalized numerical engine for time stepping

While the solver is responsible for the global loop over the mesh elements, the time loop was carried out by the *engine*-object (Figure 5 (left)). The engine-object contains the *gear*-object (Figure 5 (right)), which specifies the task to be carried out by means of so-called *job*-objects, denoted in a generic programming way as *J* in Figure 5 (right). A job is essentially a so-called *enumerated type*, and can be one of the following: *prepare*, *solve*, *solve_2nd_order*, *update*, *initiate_cells*, *initiate_edges*. The gear-object implements the *Runnable*-interface, which is a standard Java language interface that enables multithreading, i.e. parallel computation. In essence, the engine sets the type of job to be carried out by the gear-object, which then

delegates the job to the solver-object (Figure 4 (right)). The job *solve_2nd_order* indicates second order accurate time stepping via a two-stage Runge-Kutta method, while the job *solve* indicates a forward Euler time step.

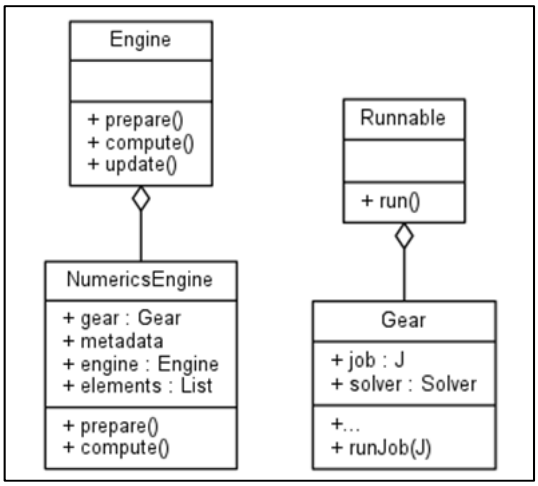


Figure 5. Definition of *engine*-object (left) and *gear*-object (right).

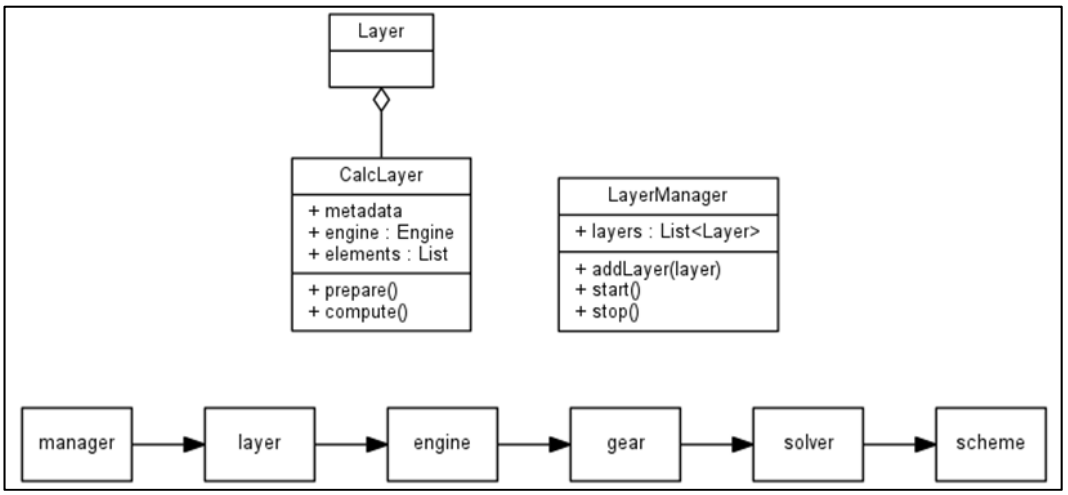


Figure 6. Definition of the *calculation-layer*-object (top left) and the *layer-manager*-object (top right), call-hierarchy during one simulation run (bottom).

3.5 Calculation layers and layer management

The *calculation-layer*-object is a special type of *layer*-object that defines the logic of the simulation in itself. The calculation-layer-object takes the *engine*-object and the metadata of a physical process layer, e.g. surface flow layer, as input (cf. Figure 6 (top left)). Each physical process then becomes a *calculation-layer*-object. During the simulation, the *prepare()*- and *compute()*-methods are delegated to the engine-object by the calculation-layer.

Finally, the *layer-manager*-object, shown in Figure 6 (top right), provides the functionality to add and remove layer-objects, which were stored in a list. The *layer-manager*-object further triggers the start and the stop of the simulation run. During the simulation, the layer-manager-object loops through all calculation-layer-objects and triggers each of them to compute their new state. As discussed above, the calculation-layer-objects then delegate the task to their corresponding engine-object and from there the task is delegated to the gear-, the solver- and finally the scheme-object which solves the conservation law at cell-level. The call-hierarchy is illustrated in Figure 6 (bottom).

3.6 Summary

We briefly presented the layer-based OOP design of hms as an example how different physical processes can be coupled such that the adding and removing of processes at runtime is easy and implementing new processes can be done with reasonable effort. The layer-manager starts the simulation, observes it and then stops it when the time horizon of the simulation is exceeded. The engine-object is responsible for the time loop and the solver-object is responsible for the spatial loop. The scheme-object defines, how and which conservation law is solved. The most fundamental object herein is the layer-object,

which combines the state variables with the numerical methods. The most commonly used design pattern is the strategy-pattern. We note that in order to simplify the object creation, hms uses a *factory*-pattern, i.e. a much simpler interface (the *factory*) is presented to the user and the complex objects are created in the background. The simulation is further monitored by so-called monitor-objects, that write out model results. This is known as the *observer*-pattern (Gamma et al., 1995). An in-depth discussion on the software design of hms is found in Busse (2015).

4 COMPUTATIONAL EXAMPLES

The numerical schemes implemented in hms have been extensively verified in Simons et al. (2014). In the following, two computational examples of shallow water flow coupled with another physical process are presented. The first example considers the transport of a contaminant that does not influence the flow and hence, the coupling is only in one direction. The second example, which deals with sediment transport and morphodynamics, is more complex because the transported sediment concentration appears in the momentum balance as a source term. Furthermore, the bed elevation changes in dependency of the transported sediment. The change in bed elevation then influences the whole flow dynamics. Hence, the processes are coupled in both directions.

4.1 Contamination transport in an idealized section of Panke river, Berlin, Germany

Panke river has its source in Brandenburg, Germany, and flows over the Barnim plateau, through the northern urbanized part of Berlin into the Spandauer channel in Berlin, Germany. It is classified as a heavily modified water body, as its bed has been straightened, artificial embankments have been constructed and the river is incorporated into the wastewater system of the region (Lange et al., 2015). In 2015, a previously undetected contaminant (traces of the medication gabapentin) was discovered in the water work at Lake Tegel coming from the wastewater treatment plant Schönerlinde, which currently cannot filter gabapentin out of the water sufficiently. As a consequence, the treatment plant is forced to discharge its treated water in a tributary of the Panke river instead of Lake Tegel, as the lake is a source for drinking water (Senatsverwaltung für Stadtentwicklung und Umwelt, 2015). The location where the tributary joins the Panke river is shown in Figure 7 (left).

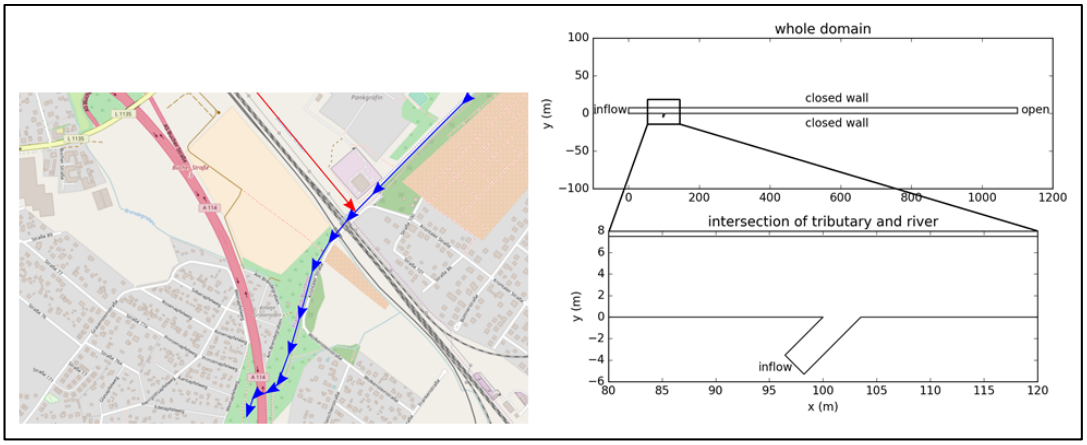


Figure 7. Location of Panke river (blue arrows) and the tributary (red arrow) in Berlin, Germany, ©OpenStreetMap-contributors (left), idealized section of the river with definition of boundary conditions (right).

In this example, we study the influence of the increased discharge in the tributary on the flow conditions in Panke river and the spreading of the contaminant that is released from the wastewater treatment plant. Since these results were obtained from preliminary studies, the geometry of Panke river as well as its tributary were idealized (Amann, 2016). The contaminant was treated as a passive scalar. The real world mean flow discharge in the Panke river was about 0.777 m^3/s and in the tributary about 0.06 m^3/s. As the wastewater treatment plant discharges into the tributary, the discharge will increase. Thus, we use an estimated discharge of 1.0 m^3/s in the tributary as an inflow boundary condition at the tributary. The inflow boundary of the Panke river used the real world mean flow discharge. Boundary conditions were sketched in Figure 7 (right). The contaminant was injected with a concentration of 1.0 over the boundary of the tributary.

The advantage of the OOP approach can be seen here, because from the model user point of view, we simply create and add a surface-layer- and a tracer-layer-object to the layer-manager-object. The coupling of both processes was then carried out as discussed above.

The simulation was run until steady state was reached. The steady state flow velocities were plotted in Figure 8. The tributary leads to an average flow velocity of 0.25 m/s at the upstream of the intersection and an average velocity of 0.53 m/s at the downstream. As seen in Figure 8, the magnitude of flow velocity also

varies spatially. The maximum velocities were observed at the outlet of the tributary, and in general, the velocities in the middle of the river are higher than the velocities at the boundaries.

For a case where both inflow boundaries were set to the real world mean discharges (status quo), the velocity at the upstream of the intersection was 0.38 m/s, and we observed that the increased discharge in the tributary yields a decrease in the flow velocity at the upstream. The downstream velocities of the status quo case were about 0.4 m/s, and here we observed a significant increase due to the increased discharge in the tributary.

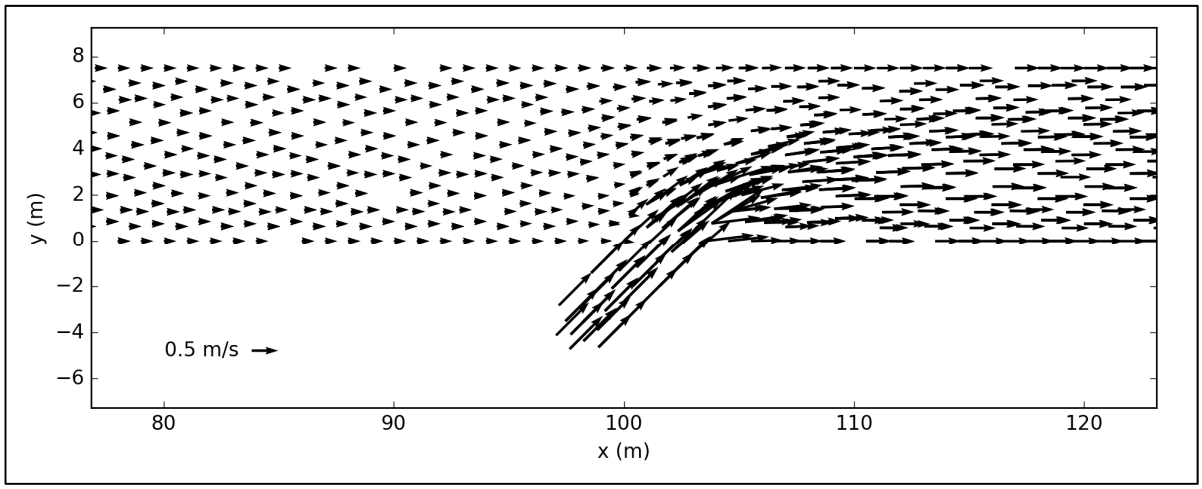


Figure 8. Model results for the steady state flow field at the intersection of tributary and Panke river with increased discharge.

It is expected that the flow field has significant influence on the contaminant spreading. Figure 9 shows snapshots at different time steps of the simulation. It is observed that the contaminant is transported downstream without contaminating the water at the upstream of the intersection. Furthermore, the boundary at the opposite side of the intersection shows less contamination than the boundary at the side of the intersection. This is of course determined by the flow field. Turbulent diffusion leads to a spreading perpendicular to the flow direction and thus it is observed that the contaminant spreads more towards the opposite boundary as the distance to the intersection increases.

4.2 Dam-break flow over mobile bed

This academic test case was initially presented and discussed in Cao et al. (2004). Later, it was replicated in Simpson and Castelltort (2006) and Zhao et al. (2016). The governing equations solved in this test case are presented in Simpson and Castelltort (2006). In addition to shallow water flow, the transport of sediment concentration was solved and erosion and deposition rates were calculated that determined the evolution of the bed morphology. In addition, the erosion and deposition rates act as a momentum sink or source term and thus influence the shallow water flow. The domain was 50000 m long with the dam located at 25000 m. At the left-hand side, an initial water level of 40 m, at the right-hand side an initial water level of 2 m is defined. The initial bed elevation was constant at zero datum. The sediment diameter was chosen as 8 mm. All model parameters were identical with those in Cao et al. (2004).

Again, we emphasize that the OOP approach significantly simplifies the simulation of coupled processes. From the developer point of view, the physics of morphodynamics are encapsulated in the corresponding layer and no rewriting of existing code is required. From the model user point of view, we simply add the morphodynamics layer-object to the layer-manager.

Model results were compared with the results by Simpson and Castelltort (2006) in Figure 10 after 60 s. It is seen that a large erosion of the bed occurs at the location of the dam, and that the erosion is less at the downstream of the dam. A hydraulic jump, which manifests itself in depth-averaged models as a sudden discontinuity in water elevation, is observed near the location of the dam. All of these observations are in agreement with Simpson and Castelltort (2006). The model shows good agreement with the reference solution as both water elevation and bottom elevation were replicated with an L_1-error of 0.38 m and 0.09 m, respectively. We note that the reference solution was extracted from the figures in Simpson and Castelltort (2006) and therefore can only be viewed as an approximation to the real reference solution. However, given the good agreement with hms, we are confident that the error margin of this approximation is sufficiently small for comparison purposes.

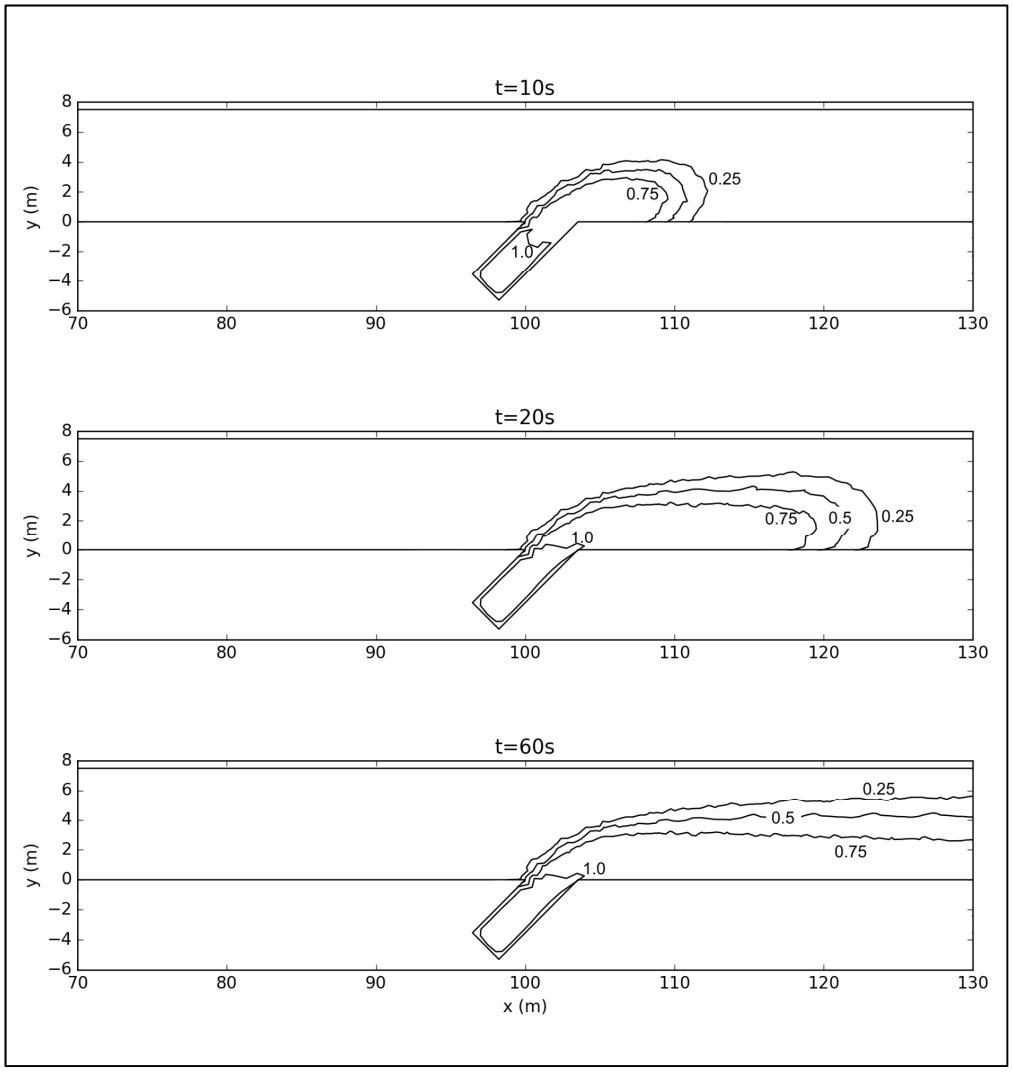


Figure 9. Model results for tracer concentration at different time steps.

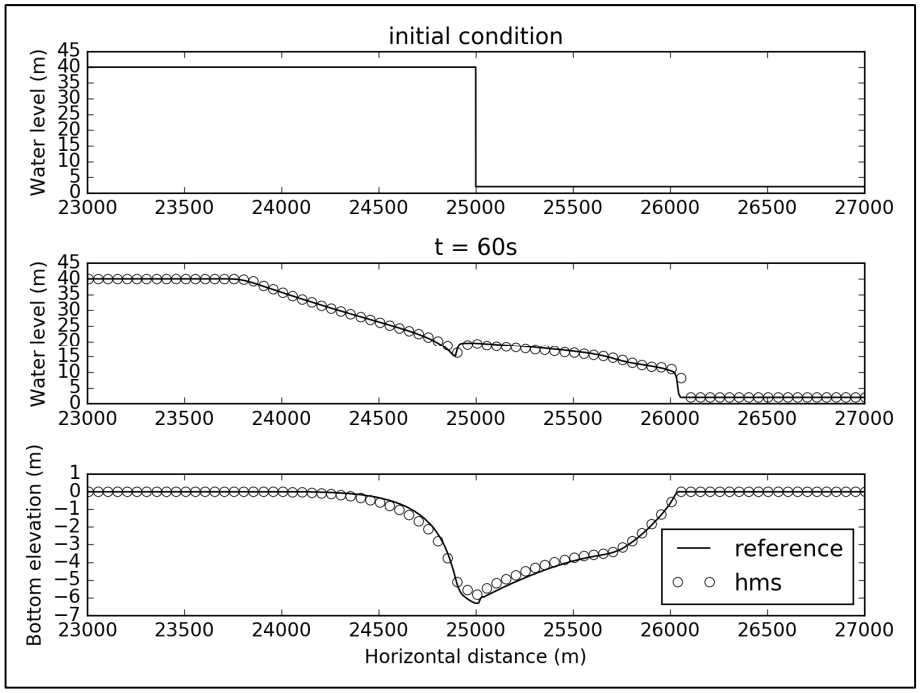


Figure 10. Model results for dam-break flow over mobile bed compared with reference solution by Simpson and Castelltort (2006).

5 CONCLUSIONS

Object-oriented design patterns that enable a comfortable way to simulate coupled physical processes were suggested and applied to shallow water flow and related processes within the framework Hydroinformatics Modeling System. It is concluded that while OOP slightly increases the computational overhead and the complexity of the code, applying the OOP concepts results in the code that allows easy implementation and coupling of different physical processes. Compared to procedural programming, OOP enhances the reusability and flexibility of the code significantly.

ACKNOWLEDGEMENTS

This work was supported by the DFG Research Training Group 'Urban Water Interfaces' (DFG-GRK2032). The scholarship granted to J. Zhao by the Chinese Scholarship Council is acknowledged. The research by T. Busse and F. Simons, that led to the implementation of hms in its current form, is acknowledged.

REFERENCES

Amann, F. (2016). Zweidimensionale Strömungs- und Transportsimulation an einem idealisierten Gewässerabschnitt der Panke in Berlin (German), *BSc Thesis*. Technische Universität Berlin.

Busse, T. (2015). HMS - A component-based hydroinformatics modelling system for the engineering of customised and integrated software applications, *Phd Thesis*. Technische Universität Berlin.

Busse, T., Simons, F., Mieth, S., Hinkelmann, R. & Molkenthin, F. (2012). HMS: A Generalized Software Design to Enhance the Modelling of Geospatial Referenced Flow and Transport Phenomena. *Proceedings of the 10th International Conference on Hydroinformatics, Hamburg, Germany.*

Cambier, L. & Gazaix, M. (2002). elsA: An Efficient Object-Oriented Solution to CFD Complexity. *Proceedings of the 40th Aerospace Sciences Meeting and Exhibit*, Nevada, 108.

Cao, Z., Pender, G., Wallis, S. & Carling, P. (2004). Computational Dam-Break Hydraulics over Erodible Sediment Bed. *Journal of Hydraulic Engineering*, 130(7), 689-703.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston, USA.

Godunov, S.K. (1959). A Difference Method for Numerical Calculation of Discontinuous Solutions of the Equations of Hydrodynamics. *Matematicheskii Sbornik*, 83(9), 271-306.

Senatsverwaltung für Stadtentwicklung und Umwelt (2015). *Reply to the written enquiry from 28 July 2015 by MP Silke Gebel (Bündnis90/Die Grünen) ``Spurenstoffbelastung durch Medikamente im Berliner Trinkwasser und den Gewässern'' (German)*. Berlin, Germany.

Heng, B.C.P. & Mackie, R.I. (2009). Using Design Patterns in Object-Oriented Finite Element Programming. *Computers and Structures*, 87(15), 952-961.

Hou, J., Liang, Q., Simons, F. & Hinkelmann, R. (2013). A 2D Well-Balanced Shallow Flow Model for Unstructured Grids with Novel Slope Source Treatment. *Advances in Water Resources*, 52, 107-131.

Katsurayama, H., Komurasaki, K. & Arakawa, Y. (2004). Application of Object-Oriented Design Patterns to a CFD Code with an Adaptive Mesh Refinement Technique. *Proceedings of the 42nd Aerospace Sciences Meeting and Exhibit, Nevada, USA.*

Lange, C., Schneider, M., Mutz, M., Haustein, M., Halle, M., Seidel, M., Sieker, H., Wolter, C. & Hinkelmann, R. (2015). Model-Based Design for Restoration of a Small Urban River. *Journal of Hydro-environment Research*, 9(2), 226-236.

Liang, D., Falconer, R.A. & Lin, B. (2007). Coupling Surface and Subsurface Flows in a Depth Averaged Flood Wave Model. *Journal of Hydrology*, 337(1), 147-158.

Matta, E., Selge, F., Gunkel, G., Rossiter, K., Jourieh, A. & Hinkelmann, R. (2016). Simulations of Nutrient Emissions from a Net Cage Aquaculture System in a Brazilian Bay. *Water Science & Technology*, 73(10), 2430-2435.

Mügler, C., Planchon, O., Patin, J., Weill, S., Silvera, N., Richard, P. & Mouche, E. (2011). Comparison of Roughness Models to Simulate Overland Flow and Tracer Transport Experiments under Simulated Rainfall at Plot Scale. *Journal of Hydrology*, 402(1), 25-40.

Özgen, I., Zhao, J., Liang, D. & Hinkelmann, R. (2016). Urban Flood Modeling Using Shallow Water Equations With Depth-Dependent Anisotropic Porosity. *Journal of Hydrology*, 541, 1165-1184.

Rodríguez-Gómez, G., Muñoz-Arteaga, J. & Fernández-del-Busto, R. (2004). Scientific software design through scientific computing patterns. *In: Proceedings of the 4th IASTED International Conference, Hawaii, USA.*

Roe, P.L. (1981). Approximate Riemann Solvers, Parameter Vectors and Difference Schemes. *Journal of Computational Physics*, 43(2), 357-372.

Simons, F., Busse, T., Hou, J., Özgen, I. & Hinkelmann, R. (2014). A Model for Overland Flow and Associated Processes within the Hydroinformatics Modelling System, *Journal of Hydroinformatics*, 16(2), 375-391.

Simpson, G. & Castelltort, S. (2006). Coupled Model of Surface Water Flow, Sediment Transport and Morphological Evolution. *Computers & Geosciences*, 32(10), 1600-1614.

Stroustrup, B. (2007). Evolving a Language in and for the Real World: C++ 1991-2006. *Proceedings of the 3rd ACM SIGPLAN History of Programming Languages Conference, California, USA*.

Tchon, K.F. (1995). Object-Oriented Programming for a Temporal Adaptive Navier-Stokes Solver. *Proceedings of the 33rd Aerospace Sciences Meeting and Exhibit,* 574.

Toro, E.F., Spruce, M. & Speares, W. (1994). Restoration of the Contact Surface in the HLL-Riemann Solver. *Shock Waves*, 4(1), 25-34.

Viero, D.P., Peruzzo, P., Carniello, L. & Defina, A. (2014). Integrated Mathematical Modeling of Hydrological and Hydrodynamic Response to Rainfall Events in Rural Lowland Catchments. *Water Resources Research*, 50(7), 5941-5957.

Zhao, J., Özgen, I., Liang, D., Simons, F. & Hinkelmann, R. (2016). Comparison of capacity and non-capacity sediment transport models for dam break flow over movable bed. *Proceedings of the 13th International Symposium on River Sedimentation*, 113.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

NUMERICAL SIMULATION OF CAVITATING FLOW ON STEPPED BEDS

REZA ESMAILI[1], SEYED ALI AKBAR SALEHI NEYSHABOURI[2] & AMIR REZA ZARRATI[3]

[1,2] Civil and Environmental Engineering Department
Tarbiat Modares University, Tehran, Iran
reza.esmailli@gmail.com
[3] Civil and Environmental Engineering Department
Amirkabir University of Technology, Tehran, Iran
zarrati@aut.ac.ir

ABSTRACT

The objective of the present study is to numerically simulate the cavitating flow behavior on stepped bed similar to a stepped spillway using a finite volume, two-phase solver available in the framework of the OpenFOAM (Open Field Operation and Manipulation) software package. These simulations were performed using the realizable k-ε turbulence model and volume of fluid (VOF) technique. A transport equation model for the local volume fraction of vapor was solved, and a finite rate mass transfer model was used for simulation of the vaporization and condensation processes. The mass transfer model of Kunz was used, and its validation was performed through comparisons between numerical simulations and the available experimental data conducted in a cavitation tunnel. The cloud of cavitation predicted by the numerical model agreed with the experimental results. The results also revealed that larger step heights are more prone to cavitation than smaller steps. The numerical model similar to experimental results showed that cavitation potential increased by increasing the steps angle corresponding to horizontal direction. According to the numerical results, the step tips are the main location of cavitation formation. It was also shown that proceeding downstream of the cavitation tunnel, as the average pressure in flow section decreases, the cavitation potential and thickness of the formed cavitation cloud increases.

Keywords: Numerical model; stepped bed; cavitation potential; openfoam; mass transfer model.

1 INTRODUCTION

As a fundamental property of liquid, cavitation appears when the local pressure in liquid falls below the saturated vapor pressure (Chen et al., 2015). A liquid is said to cavitate when vapor bubbles form and grow as a consequence of pressure reduction. When the phase transition results from hydrodynamic pressure changes, a two-phase flow composed of a liquid and its vapor is called a "cavitating flow" (Eisenberg, 1967).

Cavitation damage may occur if projections or depressions such as chamfers into the flow and other irregularities during construction of flood release structures exist on the chute surface. A serious problem is cavitation damage. Hydraulic practice has demonstrated that severe cavitation damage occurred even for irregularities of a few millimeters (Dong et al., 2010).

The experimental researches on cavitation have been extensively carried out so far for varieties of flows, most of which are around hydrofoils or propellers. Rouse and McNown (1948) carried out a series of experiments on the cavitating flows about axisymmetric configurations with cylindrical after-body (Chen et al., 2015).

A non-dimensional parameter called the cavitation number, σ is a tool of characterizing the level of cavitation and is defined by (Novak et al., 2010) as:

$$\sigma = \frac{P - P_v}{\frac{1}{2}\rho U^2} \qquad [1]$$

where U, P, P_v and ρ are the mean reference velocity, the local reference pressure, the vapor pressure and the water density, respectively.

Cavitation inception occurs when the cavitation number, σ becomes lower than a critical value σ_i (Cavitation Inception Number). The general method to study the incipient cavitation number is mainly by means of experiment in cavitation tunnels or in depression tanks (Zhang et al., 2011).

Due to their significant energy dissipation, stepped spillways have gained interest and popularity among researchers and dam engineers (Cheng et al., 2006). There is however a long discussion on the possible cavitation potential over the macro roughnesses of these spillways in higher specific discharges (Frizell et al., 2013).

In order to reduce cavitation damages, it is necessary to accurately predict the extent and behavior of the cavitating flow on the body surface. There has been considerable uncertainty in how to address cavitation on stepped beds which has caused conservative design practices. Flow on a stepped channel forms a highly intense shear layer along the line connecting successive step tips (also called the pseudo-bottom) (Frizell et al., 2013). Pfister et al. (2006) proposed treating each step as a singular bottom irregularity with risk of cavitation. To avoid cavitation, it is recommended to limit the specific discharge (Amador et al., 2009). The use of specific discharge as a design recommendation seems a bit misguided without knowing the actual conditions of when and if cavitation will form (Frizell et al., 2013). A negative pressure approach was adopted by Gomes (2006), on the basis of which an incipient cavitation number was estimated. Critical unit discharges and mean velocities were also proposed for typical stepped spillway geometries (Gomes, 2006; Gomes et al., 2007).

Frizell et al. (2013) conducted laboratory experiments in a specialized facility to study the formation of cavitation and they have concluded that cavitation can form on stepped bed geometries that are representative of typical stepped spillways currently in service.

In the last few decades, several CFD (Computational Fluid Dynamics) methods were developed to numerically investigate cavitating flow phenomena. Accurate prediction of the flow condition, where sharp interfaces might occur between the different phases, represents a numerical problem of considerable difficulty (Owis et al., 2003). In the CFD framework for cavitation simulation, different kinds of two-phase flow approaches have been developed. In the past two decades, the model in which the cavitating flow field is treated as filled with a homogeneous mixture composed of the liquid and vapor phases, and the flow is governed by only one set of equation, has been widely used (Chen et al., 2015). Different methods were proposed to evaluate the variable density field (Esmaili et al., 2015). Two common methods imply either a barotropic law (Coutier-Delgosha et al., 2003; Barre et al., 2009) or a transport equation model for void ratio with the source terms simulating the mass transfer due to cavitation (Merkle et al., 1998; Kunz et al., 2000; Singhal et al., 2002; Senocak et al., 2002).

The objective of the present work is to numerically simulate the cavitating flow behavior on stepped bed similar to a stepped spillway with different slopes and steps' heights and study the possibility of simulating cavitation potential on such geometries. The mass transfer model of Kunz et al. (2000) was used and the realizable k-ε turbulence model along with volume of fluid (VOF) technique was utilized. These simulations were performed using a finite volume, two phase solver available in the framework of the OpenFOAM (Open Field Operation and Manipulation) software package (OpenFOAM, 2013).

2 GOVERNING EQUATIONS

Due to the phase changes from liquid to vapor, a multiphase flow model has to be employed to describe the flow. Usually the two-phase mixture governing equation is employed to describe the multiphase flow for cavitation (Shang, 2013).

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial t} = 0 \qquad [2]$$

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial \left(\rho u_i u_j\right)}{\partial x_i} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_i}\left(\tau_{tij}\right) + \rho g_i + f_\sigma \qquad [3]$$

where u_i and p are the velocity and pressure, respectively. The mixture density ρ and viscosity μ are defined by:

$$\rho = \alpha_v \rho_v + \alpha_l \rho_l \qquad [4]$$

$$\mu = \alpha_v \mu_v + \alpha_l \mu_l \qquad [5]$$

in which $\alpha_v + \alpha_l = 1$, ρ_v and μ_v are vapor density and viscosity, ρ_l and μ_l are liquid density and viscosity, α_v and α_l are the local vapor and liquid volume fraction, respectively. The local liquid volume fraction α_l is governed by:

$$\frac{\partial \alpha_l}{\partial t} + \frac{\partial (\alpha_l u_i)}{\partial x_i} = -\frac{\dot{m}}{\rho_l} \qquad [6]$$

where ṁ is the mass transfer rate between the phases (Shang, 2013).

In the above set of equations, some important terms such as Reynolds stress, τ_{tij}, the surface tension, f_σ and the mass transfer rate, ṁ need to be modeled.

The Reynolds stress can be modeled through Boussinesq hypothesis by the following equation:

$$\tau_{tij} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \left(\rho k + \mu_t \frac{\partial u_k}{\partial x_k} \right) \delta_{ij} \quad [7]$$

where δ_{ij} is Kronecker delta and μ_t is the turbulent viscosity.

If the turbulent viscosity is calculated, the Reynolds stress will be determined and the system of equations will be closed. In the present work, realizable k-ε model was chosen as the turbulence model for calculating turbulent viscosity.

The interface between the liquid and vapor phases is captured by volume of fluid (VOF) method (Hirt et al., 1981). The VOF equation can be developed from Eq. [7] and described as the following (Shang, 2015):

$$\frac{\partial \alpha_v}{\partial t} + \frac{\partial (\alpha_v u_i)}{\partial x_i} + \frac{\partial [\alpha_v (1 - \alpha_v) u_r]}{\partial x_i} = \frac{\dot{m}}{\rho_v} \quad [8]$$

where u_r is the relative velocity at the interface between vapor and liquid, which can be described as:

$$u_r = \min(C_\alpha |u_i|, \ \max|u_i|) \frac{\partial \alpha_v / \partial x_i}{|\partial \alpha_v / \partial x_i|} \quad [9]$$

where c_α is the compression coefficient for the interface. The VOF equation can be solved by the implicit multidimensional universal limiter with explicit solution (MULES) algorithm (Rusche, 2002; Gomaa et al., 2011).

Since it is necessary to calculate the interface between water and vapor using VOF, the surface tension should be calculated in Eq. [3] using the following formula:

$$f_\sigma = \sigma \frac{\partial}{\partial x_i} \left(\frac{\partial \alpha_v / \partial x_i}{|\partial \alpha_v / \partial x_i|} \right) \frac{\partial \alpha_v / \partial x_i}{|\partial \alpha_v / \partial x_i|} \quad [10]$$

where σ is surface tension (Shang, 2013).

The mass transfer model of Kunz et al. (2000) was employed in the present work. This model is based on the work of Merkle et al. (1998) and is currently one of the mass transfer models implemented in the OpenFOAM CFD Open Source Library. In this model, the mass transfer is based on two different strategies for creation and destruction of liquid. The transformation of liquid to vapor was evaluated as being proportional to the amount by which the pressure is below the vapor pressure. The transformation of vapor to liquid, otherwise, is based on a third order polynomial function of volume fraction, α. The specific mass transfer rate is defined as $\dot{m}=\dot{m}^+ + \dot{m}^-$ (Kunz et al., 2000):

$$\dot{m}^+ = \frac{C_c \alpha_l^2 (1 - \alpha_l) \rho_v}{t_\infty} \quad [11]$$

$$\dot{m}^- = \frac{C_v \alpha_l \rho_v \min[p - p_{sat}, 0]}{\left(\frac{1}{2} \rho_l U_\infty^2 \right) t_\infty} \quad [12]$$

In the above equations, U_∞ is the reference velocity, $t_\infty = L/U_\infty$ is the mean flow time scale, where L is the characteristic length scale. C_c and C_v are the two empirical coefficients.

3 VERIFICATION OF NUMERICAL MODEL

In the first step, the experiments of Rouse and McNown (1948) were chosen to verify the cavitation model. The experiments were conducted in a cavitation tunnel with a cylindrical test object of length l=0.24 m and diameter D=0.024 m, such that l/D=10. The flow attacks at the hemispherical head shape with zero angle under a constant Reynolds number of 2.67×10^5 (Figure 1). The numerical simulations were carried out with the same conditions as the experiments, and with the cavitation numbers of 0.3 and 0.5. Only one half of the flow domain was simulated using symmetry conditions to reduce the computation time. The flow enters the domain with a velocity of 10 (m/s), and exits at the outlet boundary under a fully-developed condition, where a fixed pressure condition was set. A 60,960-cellgrid was used for the 2D axisymmetric simulations.

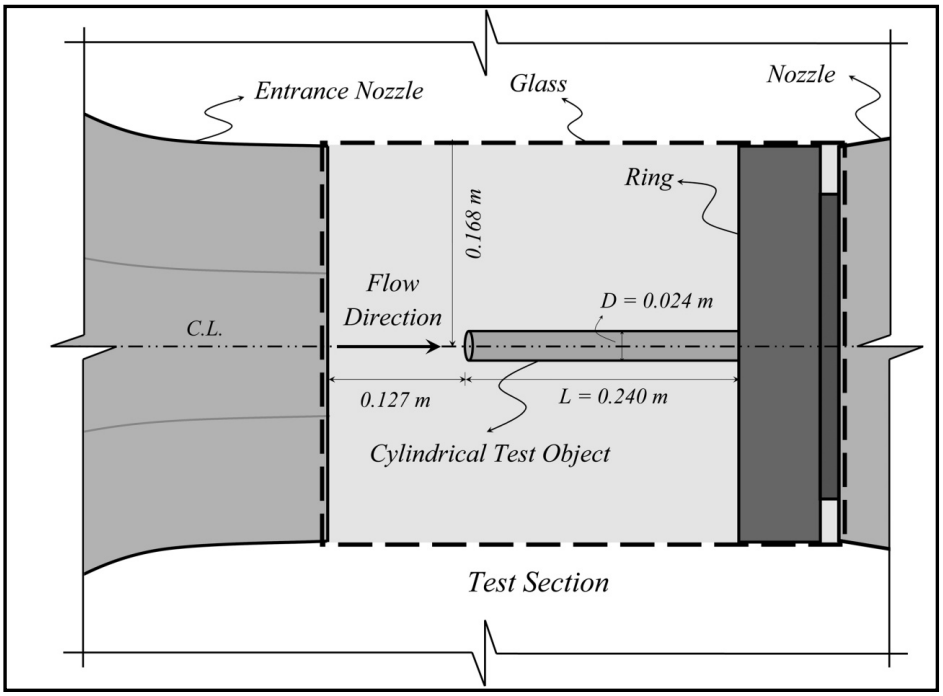


Figure 1. Outline of the test section.

The resulting pressure coefficient (C_p) distribution along the cylinder surface defined by Eq. [13] was compared to the measured data by Rouse and McNown (1948) in Figure 2. In this figure, "s" is the distance between a surface point and the edge of the cylinder head along the body surface and D is the cylinder diameter. It is found that the numerical results basically agree with the experiment under various cavitation conditions.

$$C_p = -\frac{P_{loc} - P}{\frac{1}{2}\rho_l U^2} \qquad [13]$$

where P_{loc} is the local static pressure.

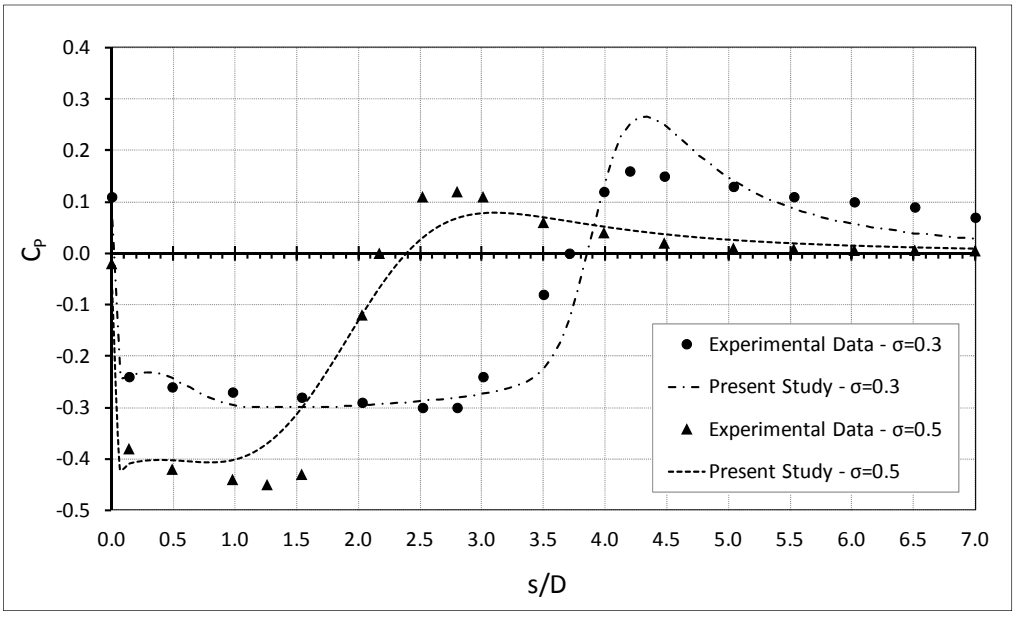


Figure 2. Comparison of the pressure coefficient distributions along the cylinder surface for cavitation numbers of 0.3 & 0.5 with Rouse and McNown's experimental data (Rouse and McNown, 1948).

In addition, shapes of cavity agree well with the experimental data, as shown in Figure 3.

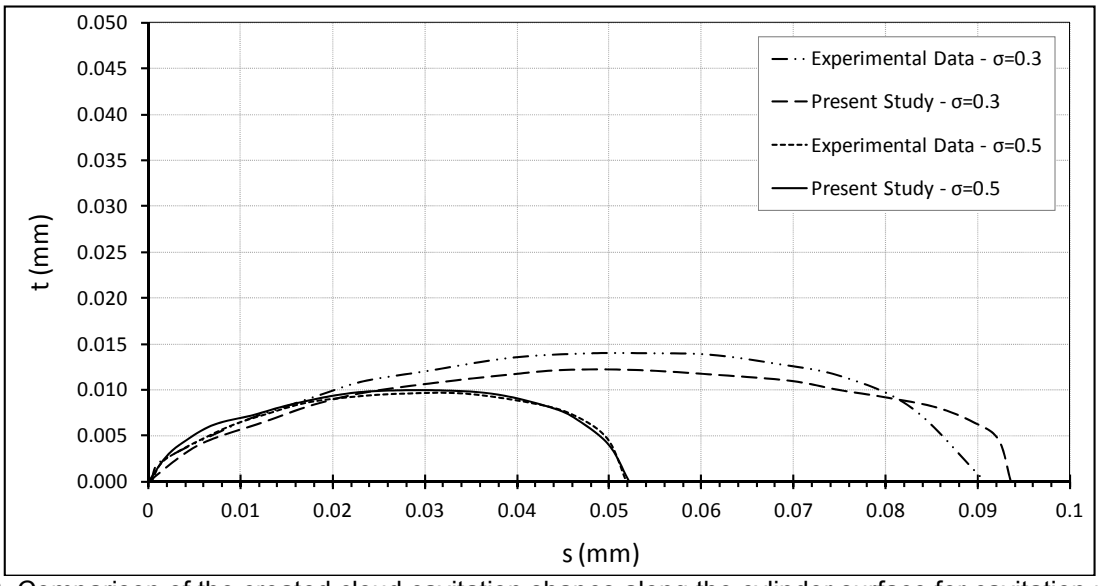


Figure 3. Comparison of the created cloud cavitation shapes along the cylinder surface for cavitation numbers of 0.3 and 0.5 with Rouse and McNown's experimental data (Rouse and McNown, 1948).

4 CAVITATING FLOW ON STEPPED BEDS

After validation, the numerical model was used to simulate the cavitating flow on stepped beds. Cavitation potential of 4 types of stepped beds was studied experimentally by Frizell et al. (2013). Experiments were conducted in a specialized low-ambient pressure chamber (LAPC) at the laboratory of the Bureau of Reclamation in Denver, Colorado. A rectangular closed-conduit with machined anodized aluminum steps on the invert and smooth acrylic sidewalls and lid was placed within the chamber, with water entering the conduit from a pressure tank with flow straightening vanes and exiting submerged into a small free-surface reservoir, allowing the chamber pressure to act on the fluid. The conduit dimensions are 148 mm high (pseudo-bottom to lid), 203.2 mm wide and 2.21 m long (Frizell et al., 2013). The conduit orientation is horizontal, with the angle of the step treads above the horizontal direction (θ) representing the spillway slope. Two different step heights (h) of 50.8 mm and 25.4 mm (by adding small rectangular blocks at each larger step) with two different slopes (θ) of 21.8° and 68.2° were tested (Figure 4). The flow enters the domain with a uniform velocity, V_0, and exits at the outlet boundary under a fixed pressure condition.

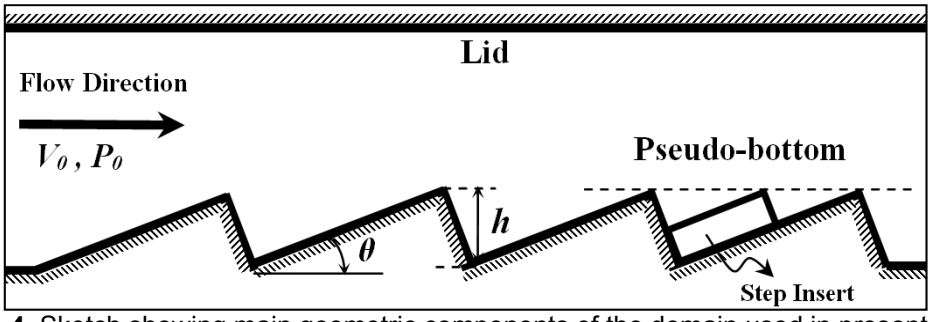


Figure 4. Sketch showing main geometric components of the domain used in present study.

Frizell et al. (2013) reported that the cavities form just above the step tips along the pseudo-bottom. They have found that at a given slope, larger step heights are slightly more prone to cavitation than smaller steps. Also, they have concluded that the steeper slope of steps has a greater cavitation potential (Frizell et al., 2013).

Figure 5 shows the fields of vapor volume fraction, for all four simulated stepped beds in the present study. As can be seen, the step tips, along the pseudo-bottom are the main locations of cavitation formation which is also reported by Frizell et al. (2013). It was also shown that proceeding downstream along the steps, as the average pressure in flow section decreases the cavitation potential of the formed cavitation cloud increases where this results also conforms to Frizell et al. (2009) observations.

The formed cavitation cloud from present study along with the single-frame images presented by Frizell et al. (2013), at the last step are shown in Figure 6. It can be seen that the patterns of the cavitation clouds predicted by the numerical model agrees with the experimental results.

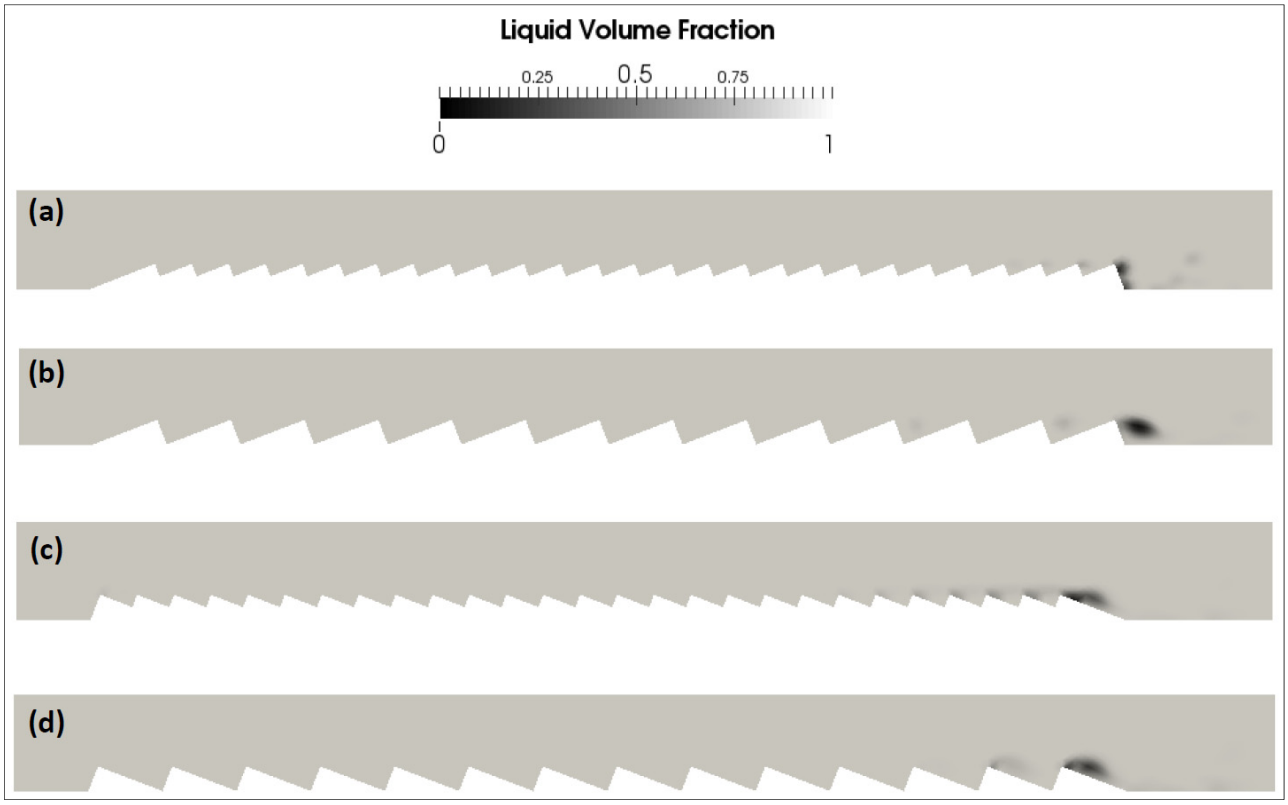


Figure 5. Numerical results for the created cloud cavitation on the stepped beds;
(a) θ=21.8° & h=25.4 mm, (b) θ=21.8° & h=50.8 mm, (c) θ=68.2° & h=25.4 mm and (d) θ=68.2° & h=50.8 mm.

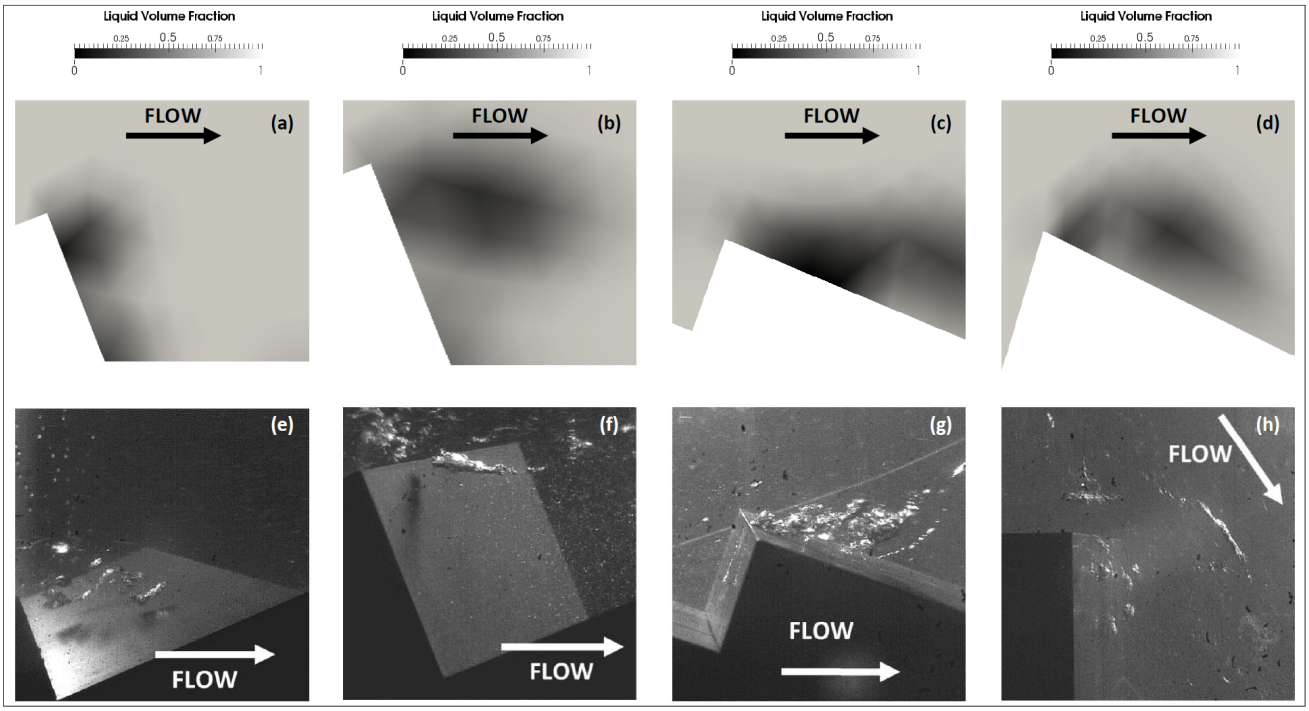


Figure 6. Comparison of the created cloud cavitation for present study (at the top) with Frizell et al. (2013) experimental results (at the bottom) for each of four test conditions at the last step:
(a) & (e): θ=21.8° & h=25.4 mm, (b) & (f): θ=21.8° & h=50.8 mm,
(c) & (g): θ=68.2° & h=25.4 mm and (d) & (h): θ=68.2° & h=50.8 mm.

To know the effects of step height and spillway slope on cavitation potential, additional numerical simulations were carried out for estimation of the cavitation inception condition in four types of tested stepped beds (σ_i). To find the inception condition, by keeping the outlet pressure constant, the inlet velocity increased from a low value until vapor began to form.

Table 1 shows the computed incipient cavitation numbers ($\sigma_{i\ Num.}$). As can be seen from the table, results of the numerical model are of the same trend as the experimental data obtained by Frizell et al. (2013). It can be seen that similar to experimental results, increasing the step height (h) resulted in increasing σ_i and therefore increasing the cavitation potential. Also, by increasing the steps' slope (θ), σ_i increased.

Table 1. Cavitation inception numbers on the stepped beds.

Type	θ (deg)	h (mm)	$\sigma_{i\ Num.}$
a	21.8	25.4	0.24
b	21.8	50.8	0.30
c	68.2	25.4	0.48
d	68.2	50.8	0.59

5 CONCLUSIONS

The cavitating flow on 4 types of stepped beds similar to stepped spillways, with two different step heights and slopes, have been studied numerically. A transport equation model for the local volume fraction of vapor was solved and the finite rate mass transfer model of Kunz et al. (2000) was used for simulation of the vaporization and condensation processes. These simulations were performed using a finite volume, two phase solver available in the framework of the OpenFOAM (Open Field Operation and Manipulation) software package. The numerical model was validated with experimental data of pressure distribution and shape of cavitation cloud along a cylinder tested in cavitation tunnel. The comparisons show that the numerical results are in good agreements with the corresponding data.

The cloud of cavitation over 4 types of stepped beds predicted by the numerical model agreed with the experimental results. Effects of step height and slope on the cavitation potential of the stepped bed were also analyzed by computing cavitation inception number. It was concluded that the numerical model's results show the same trend as was reported in experiments. According to the numerical results, stepped beds with larger step heights are more prone to cavitation than smaller steps. It was also concluded that cavitation potential increased by increasing the slope of stepped bed. According to the numerical results, the step tips are the main location of cavitation formation. Based on the present work, the numerical model is a strong tool for studying cavitation potential in high speed flows.

REFERENCES

Amador, A., Sanchez-Juny, M. & Dolz, J. (2009). Developing Flow Region and Pressure Fluctuations on Steeply Sloping Stepped Spillways. *Journal of Hydraulic Engineering*, 135(12), 1092–1100.

Barre, S., Rolland, J., Boitel, G., Goncalves, E. & Patella, R.F. (2009). Experiments and Modelling of Cavitating Flows in Venturi: Attached Sheet Cavitation. *European Journal of Mechanics, B/Fluids*, 28(3), 444–464.

Chen, Y., Lu, C., Chen, X. & Cao, J. (2015). Numerical Investigation on the Cavitation Collapse Regime around the Submerged Vehicles Navigating with Deceleration. *European Journal of Mechanics, B/Fluids*, 49, 153–170.

Cheng, X.J., Chen, Y.C. & Luo, L. (2006), Numerical Simulation of Air-Water Two-Phase Flow over Stepped Spillways. *Science in China Series E-Technological Sciences*, 49(6), 674-684

Coutier-Delgosha, O., Reboud, J.L. & Delannoy, Y. (2003). Numerical Simulation of the Unsteady Behaviour of Cavitating Flows. *International Journal for Numerical Methods in Fluids*, 42(5), 527–548.

Dong Zh., Wu Y. & Dong Zh. (2010). Cavitation Characteristics of Offset-Into-Flow and Effect of Aeration. *Journal of Hydraulic Research*, 48 (1), 74-80.

Eisenberg Ph. (1967). *Cavitation*. Hydronautics Incorporated.

Esmaili, R., Salehi Neyshabouri, S.A.A. & Zarrati, A.R. (2015). Numerical Simulation of Cavitating Flow Past Steps-Into-Flow with Different Heights. *Proceedings of 36th IAHR World Congress*.

Frizell, K.W. & Renna, F.M. (2009). Cavitation Potential of the Folsom Auxiliary Stepped Spillway. *HL-2009-07*. Bureau of Reclamation. Denver. Colorado.

Frizell, K.W., Renna, F.M. & Matos, J. (2013). Cavitation Potential of Flow on Stepped Spillways. *Journal of Hydraulic Engineering. ASCE*, 139(6), 630-636.

Gomaa, H., Kumar, S., Huber, C., Weigand, B. & Peters, B. (2011). Numerical Comparison of 3D Jet Breakup Using a Compression Scheme and an Interface Reconstruction based VOF-Code. *ILASS-Europe 2011, Proceedings of 24th European Conference on Liquid Atomization and Spray Systems*.

Gomes, J.F. (2006). Campo de pressões: Condições de incipiência à cavitação em vertedouros em degraus com declividade 1V:0,75H, *Ph.D. Thesis*. Universidade Federal do Rio Grande do Sul. Porto Alegre. Brazil (in Portuguese).

Gomes, J., Marques, M. & Matos, J. (2007). Predicting cavitation inception on steeply sloping stepped spillways. *Proceedings of 32nd IAHR Congress,* Venice, Italy.

Hirt, C.W. & Nichols, B.D. (1981). Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries. *Journal of Computational Physics*, 39(1), 201–225.

Kunz, R.F., Boger, D.A., Stinebring, D.R., Chyczewski, T.S., Lindau, J.W., Gibeling, H.J., Venkateswaran, S. & Govindan, T.R. (2000). A Preconditioned Navier–Stokes Method for Two-Phase Flows with Application to Cavitation Prediction. *Journal of Computers and Fluids*, 29, 849–875.

Merkle, C.L., Feng, J. & Buelow, P.E.O. (1998). Computational Modeling of the Dynamics of Sheet Cavitation. *Proceedings of Third International Symposium on Cavitation*, Grenoble, France.

Novak, P., Guinot, V., Jeffrey, A. & Reeve, D. E. (2010). *Hydraulic modeling: An introduction*. Spon Press. London.

OpenFOAM. 2013. *The Open Source CFD Toolbox User Guide.* Version 2.2.2.

Owis, F.M. & Nayfeh, A.H. (2003). Computations of the Compressible Multiphase Flow over the Cavitating High-Speed Torpedo. *Journal of Fluids Engineering*, 125(3), 459-468.

Pfister, M., Hager, W.H. & Minor, H.E. (2006). Bottom Aeration of Stepped Spillways. *Journal of Hydraulic Engineering,* 132(8), 850–853.

Rouse, H. & McNown, J.S. (1948). *Cavitation and Pressure Distribution, Head Forms at Zero Angle of Yaw*. Studies in Engineering, Bulletin 32, State University of Iowa. Iowa. USA.

Rusche, H. (2002). *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. PhD Thesis of Imperial College of Science, Technology and Medicine. UK.

Senocak, I. (2002). *Computational methodology for the simulation of turbulent cavitating flows*. Ph.D. Thesis. University of Florida. USA.

Shang, Zh. (2013). Numerical Investigations of Supercavitation around Blunt Bodies of Submarine Shape. *Journal of Applied Mathematical Modelling*, 37 (20), 8836–8845.

Singhal, A.K., Athavale, M.M., Li, H. & Jiang, Y. (2002). Mathematical Basis And Validation Of The Full Cavitation Model. *Journal of Fluids Engineering*, 124 (3), 617–624.

Zhang, J.M., Yang, Q., Wang, Y.R., Xu, W.-l. & Chen, J.G., (2011). Experimental Investigation of Cavitation in a Sudden Expansion Pipe. *Journal of* Hydrodynamics, Series B, 23(3), 348-352.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

CFD MODELLING OF BREAKING AND UNDULAR TIDAL BORES WITH PHYSICAL VALIDATION

XINQIAN LENG[(1)], PIERRE LUBIN[(2)] & HUBERT CHANSON[(3)]

[(1)] The University of Queensland, School of Civil Engineering, Brisbane QLD 4072, Australia,
xinqian.leng@uqconnect.edu.au
[(2)] Université de Bordeaux, I2M, Laboratoire TREFLE, Pessac, France,
lubin@enscbp.fr
[(3)] The University of Queensland, School of Civil Engineering, Brisbane QLD 4072, Australia,
h.chanson@uq.edu.au

ABSTRACT

A tidal bore may form during spring tide conditions when the tidal range exceeds 4 to 6 m in a natural estuary with a funnel shaped river mouth and shallow initial water level. The propagation of tidal bores is a highly unsteady turbulent process associated with intensive sediment scouring and mixing. To date, few physical and numerical studies documented the unsteady turbulent process of tidal bore propagation. Recent numerical CFD models lacked careful experimental validations. The present study aims to provide new results on CFD numerical modelling of tidal bore propagation with a wide range of Froude numbers (1.2 to 2.1) and systematic experimental validations. The model solved the incompressible Navier-Stokes equations in its two-phase flow forms using Large Eddy Simulation (LES). Both 2D and 3D simulations were conducted; the inlet turbulence of the 3D models was simulated by a Synthetic Eddy Method (SEM). The physical experiments were based upon an ensemble-average technique, with measurements of water depth and velocity repeated 25 times for each flow condition. The 2D CFD simulations showed good agreement in terms of free-surface elevations with experimental results, for the range of tested Froude numbers. Mesh grid refinement only improved the accuracy for some but not all flow conditions. The 2D velocity data showed qualitative and quantitative agreement, but only at a selective range of vertical elevation beneath the free-surface, where the inlet velocity were correctly reproduced. The 3D simulation highlighted a numerical boundary layer, the thickness of which agreed with the experimental results. The time-averaged velocity and velocity RMS of the numerical model data showed a closer agreement with the physical model outside the boundary layer. The development of the numerical boundary layer was clearly observed in the CFD model results.

Keywords: Tidal bores; CFD modelling; large eddy simulation LES; physical model validation; turbulence.

1 INTRODUCTION

The propagation of tidal bores is a highly unsteady turbulent process, associated with intensive sediment scouring and mixing. The occurrence of a tidal bore is marked by a steep rise in free-surface propagating upstream. A tidal bore could form during spring tide conditions when the tidal range exceeds 4 to 6 m in a natural system with a funnel shaped river mouth and shallow initial water level (Chanson, 2011) (Figure 1). The strength of a tidal bore is characterised by its Froude number, defined as:

$$Fr_1 = \frac{U + V_1}{\sqrt{g \times \frac{A_1}{B_1}}} \quad [1]$$

where U is the bore celerity positive upstream, V_1 is the streamwise velocity of the initially steady flow posirive downstream, g is the gravitational acceleration: g = 9.8 m/s^2, A_1 and B_1 are the cross-sectional area and width respectively for the initially steady flow. When the Froude number is less than unity, a bore cannot form. When the Froude number is between 1 and 1.2-1.3, the bore is undular (Treske, 1994; Koch and Chanson, 2008; Leng and Chanson, 2016a). When the Froude number exceeds 1.3-1.5, the bore is breaking, and its strength increases with increasing Froude number (Hornung et al., 1995; Koch and Chanson, 2009; Chanson, 2010; Leng and Chanson, 2016a).

To date, limited physical studies studied the unsteady turbulent propagations of tidal bores (Hornung et al., 1995; Koch and Chanson, 2009; Chanson, 2010; 2011; Docherty and Chanson, 2012; Khezri and Chanson, 2012b; Leng and Chanson, 2015a; 2015b; 2016a; 2017). Recent numerical studies were performed using Computational Fluid Dynamics (CFD) models (Furuyama and Chanson, 2010; Lubin et al., 2010a, 2010b; Reichstetter, 2011; Simon et al., 2011; Chanson et al., 2012; Khezri, 2014; Simon, 2014), albeit over a limited range of Froude numbers and most studies lacked careful experimental validations. The present work

presents new results on a numerical CFD model of the unsteady turbulent propagations of tidal bores in open channel flows, with a wide range of Froude numbers (from 1.2 to 2.1). Both two-dimensional and three-dimensional CFD models were simulated. The modelling results were validated against physical experiments conducted systematically for the exact flow conditions used in the CFD models. The experiments were performed in a 19 m long and 0.7 m wide rectangular prismatic channel. Ensemble-averaged flow measurements were performed, where experiments were repeated 25 times for each controlled flow conditions and the results were ensemble-averaged. The free-surface variations were measured by a series of acoustic displacement meters (ADMs) along the channel centerline, and the velocity characteristics were measured using a Nortek^{TM} Vectrino+ acoustic Doppler velocimeter (ADV). All instruments were sampled at 200 Hz and synchronized within ±0.01 s.

(a) Breaking tidal bore of Qiantang River (Laoyanchang, China), photograph taken on 23 September 2016.

(b) Undular tidal bore of Dordogne River (St. Pardon, France), photograph taken on 2 September 2015.

Figure 1. Photographs of tidal bores in rivers.

2 NUMERICAL METHOD AND CFD MODEL CONFIGURATION

2.1 Numerical method and implementation

The numerical modelling was conducted using the CFD code Thétis developed by the I2M laboratory, University of Bordeaux, France. The model solves the Navier-Stokes equations in its incompressible two-phase flow form between non-miscible fluids. In this case, the two fluids were air and water, respectively. A phase function C, called the color function, is used to locate the different fluids with C = 0 in the air, C = 1 in water, with C = 0.5, the value assumed to characterise the interface location. The governing equations are presented below, which is the Large Eddy Simulation (LES) of an incompressible fluid flow classically derived by applying a convolution filter to the unsteady Navier-Stokes equations:

$$\nabla \cdot \vec{u} = 0 \qquad [2]$$

$$\rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u}\right) = -\nabla p + \rho \vec{g} + \nabla \cdot (\mu + \mu_t)[\nabla \vec{u} + \nabla^T \vec{u}] \qquad [3]$$

and,

$$\frac{\partial C}{\partial t} + \vec{u} \cdot \nabla C = 0 \quad [4]$$

where $\vec{u}$ is the velocity, C is the phase function, t is the time, p is the pressure, $\vec{g}$ is the gravity vector, ρ is the density, μ is the dynamic viscosity and μ_t is the turbulent viscosity. The turbulent viscosity, μ_t is calculated with the Mixed Scale model (Sagaut, 2006), which was found to be accurate for coastal applications (Helluy et al., 2005; Lubin et al., 2006). The magnitude of the physical characteristics of the fluids was calculated based upon the phase function as:

$$\rho = C \times \rho_1 + (1 - C) \times \rho_0 \quad [5]$$

$$\mu = C \times \mu_1 + (1 - C) \times \mu_0 \quad [6]$$

The densities ρ_0, ρ_1 and dynamic viscosities μ_0, μ_1 are the respective properties of fluids 0 and 1: in this case, air and water, respectively. The velocity and pressure coupling is solved with a pressure correction method (Goda, 1979).

The space derivatives of the inertial term are discretised by a hybrid upwind-centered scheme and the viscous terms is approximated by a second-order centered scheme (Lubin et al., 2006). The interface tracking was done using a Volume of Fluid (VOF) method with a piecewise linear interface calculation (PLIC). This method has the advantage of building a sharp interface between air and water. The time discretization is implicit and the equations are discretised on a staggered grid thanks to a finite volume method. The MPI library HYPRE was used to solve the linear system of the prediction and correction steps (Falgout et al., 2006). The time steps were dynamically calculated to insure a CFL condition inferior to 0.2. The numerical model has been proved accurate through a variety of coastal applications and numerous test cases (Lubin, 2004; Lubin et al., 2006). Earlier CFD studies of tidal bores by Simon et al. (2011) and Khezri (2014) were also based upon this model.

2.2 Numerical model configuration

Both 2D and 3D CFD simulations were conducted in the present study. For 2D CFD simulations, breaking bores of Fr_1 = 1.5 and 2.1, and undular bores of Fr_1 = 1.2 were modelled numerically. The numerical domain was 12 m in the longitudinal (stream-wise) direction and 1 m in the vertical direction (Figure 2). A no-slip condition was imposed at the lower boundary (z = 0 m) and a Neumann condition was used at the upper boundary (z = 1 m). At the end of domain (x = 12 m), a wall boundary was imposed to act like a closed gate to reproduce the experimental generation process. The opening under the gate h_{out} could be set to introduce a Neumann condition between the bed (z = 0 m) and the bottom of the gate (z = h_{out}). The initial conditions of the 2D models consisted of a water trapezoid, with higher depth at the inlet (d_{in}) and lower depth at the outlet (d_{out}) to approximate the gradually-varied flow in the physical open channel. All initial and boundary parameters were taken from the experimental measurements.

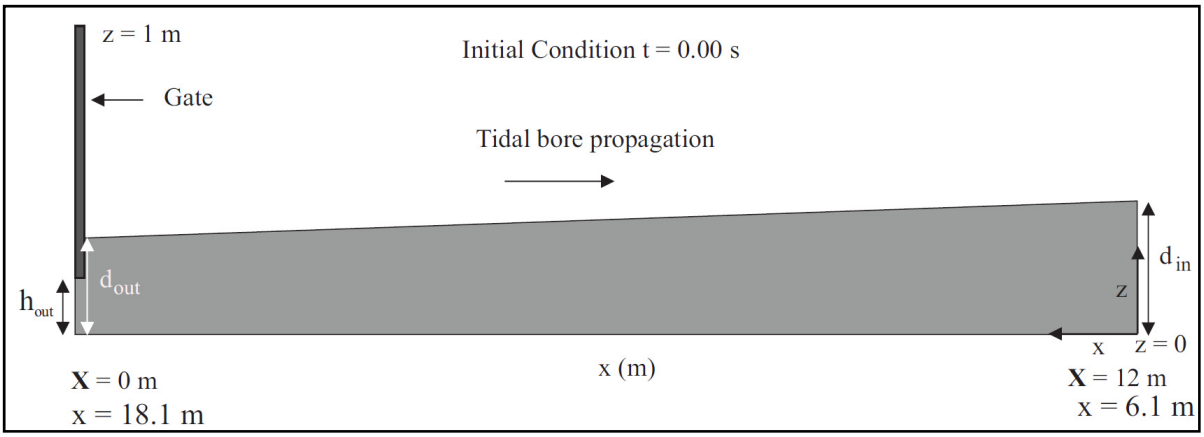


Figure 2. Definition sketch of numerical domain configurations; **X** is the distance from the left boundary (i.e. gate) of the numerical domain; x is the distance from the upstream end of the physical channel.

For 3D CFD simulations, the work is still in progress. Herein, only data of the initially steady flow before the bore arrival will be presented. The flow conditions simulated by the 3D CFD model corresponded to the initially steady flow before the breaking bore of Fr_1 = 2.1. The 3D model was based upon the 2D model configuration, extruded in the third direction being the transverse y dimension. The coordinate y was positive

towards the left side wall and the 3D numerical domain was 0.7 m wide. In this case, no-slip conditions were applied to both lateral walls and bottom of the domain. The Synthetic Eddy Method (SEM) was used in the 3D model to inject turbulence at the inlet of the domain (Jarrin et al., 2006; 2009). The input parameters for this method, including mean velocity and velocity fluctuations, were all extracted from the experimental data of Leng and Chanson (2016a). The number of eddies was set at 2000 and the size of eddies was 0.010 m, which was an order of magnitude higher than the experimental data (Leng and Chanson, 2017). Jarrin et al. (2006; 2009) found that the method gave better results with over-estimated eddy size. Table 1 documents detailed configurations of the 2D and 3D numerical models. In the table, S_o stands for the channel slope in the longitudinal direction. The opening under the gate after rapid closure is denoted h_{out}.

Table 2 summarises the experimental flow conditions corresponding to the three Froude number modelled by CFD (Fr_1 = 1.2, 1.5, 2.1). The reference depth, d_1 and celerity, U was taken at the velocity sampling location, which was located 9.6 m upstream of the gate for both physical and numerical channels.

Table 1. Initial configuration of the 2D and 3D numerical simulations.

Reference	Domain (m)	Mesh grid density	Fr_1	Q (m^3/s)	S_o	d_{in} (m)	d_{out} (m)	h_{out} (m)	Bore type
2D_Fr1.2	12×1	1600×100	1.2	0.101	0	0.208	0.19	0.071	Undular
2D_FR1.5	12×1	2400×200	1.5	0.101	0	0.18	0.16	0	Breaking
2D_FR2.1	12×1	1600×140	2.1	0.101	0.0075	0.1	0.1	0	Breaking
3D_FR2.1	12×1×0.7	1600×250×200	2.1	0.101	0.0075	0.093	0.093	0	Breaking

Table 2. Flow conditions of the experimental data used to validate the numerical model.

Reference	Fr_1	Q (m^3/s)	S_o	d_1 (m)	U (m)	Bore type	Instrumentation
Leng and	1.2	0.101	0	0.210	0.71	Undular	ADMs and ADV
Chanson	1.5	0.101	0	0.180	1.13	Breaking	ADMs and ADV
(2016a)	2.1	0.101	0.0075	0.100	1.00	Breaking	ADMs and ADV

3 2D SIMULATION OF TIDAL BORE PROPAGATION

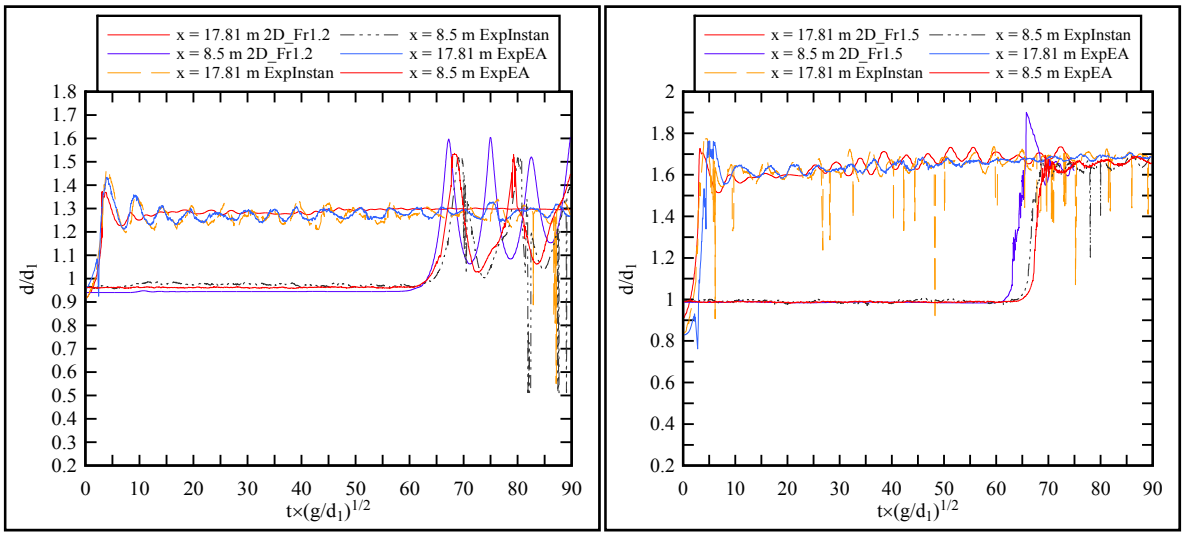
3.1 Free-surface comparisons

During the physical experiments, both instantaneous and ensemble-averaged measurements were performed to characterise the free-surface and velocity properties. Up to 10 acoustic displacement meters (ADMs) were installed at different longitudinal positions x (x = longitudinal distance from upstream end) on the channel centerline, all sampling at 200 Hz, to record the free-surface variations with time at different locations along the channel. During the numerical simulations, the free-surface variations with time were recorded at the same locations as those of the ADMs to validate the numerical results. Comparisons between numerically simulated free-surface evolution and experimental observations were conducted, and typical results are presented in Figure 3 for both undular and breaking bores. Note that the gate was located at x = 18.1 m.

Overall, the free-surface variations simulated by the 2D CFD model agreed well with the experimental data (instantaneous or ensemble-averaged) at all longitudinal locations, quantitatively and qualitatively, for all Froude numbers. Some deviations were observed in terms of the bore height and bore celerity. At generation (close to the gate), the numerical model tended to estimate relatively accurately the free-surface rise mechanism, with almost identical depth gradient with time. However, the bore height was underestimated for both breaking and undular bores at generation. As the bore propagated upstream towards mid-channel, the numerical model overestimated the bore celerity and bore height, resulting in differences in terms of bore arrival time at x = 8.5 m (Figure 3). For undular bores, the numerical model was associated with secondary wave periods which differed from the experimental data. The wave forms of the numerical model appeared to be more regular than the experimental data, due to the two-dimensional constraint and the absence of side wall effects.

3.2 Velocity comparisons

Ensemble-averaged velocity measurements were performed using a Nortek^{TM} Vectrino+ acoustic Doppler velocimeter (ADV) located on the channel centerline at mid-channel (x = 8.5 m). The ADV was equipped with a three-dimensional side looking head, able to record velocity in the longitudinal, transverse and vertical directions. The ADV was sampled at 200 Hz, synchronised with the ADMs, and measured velocity at a number of vertical elevations (z/d_1 = 0.1, 0.4, 0.8). During the numerical CFD modelling, velocity data at the same dimensionless vertical elevations was recorded at x = 8.5 m for validation purposes. Figure 4 presents typical comparisons between numerical and experimental results for bores with Fr_1 = 1.2 and 1.5. The time frames of the numerical and experimental data was synchronized using the numerical time line.

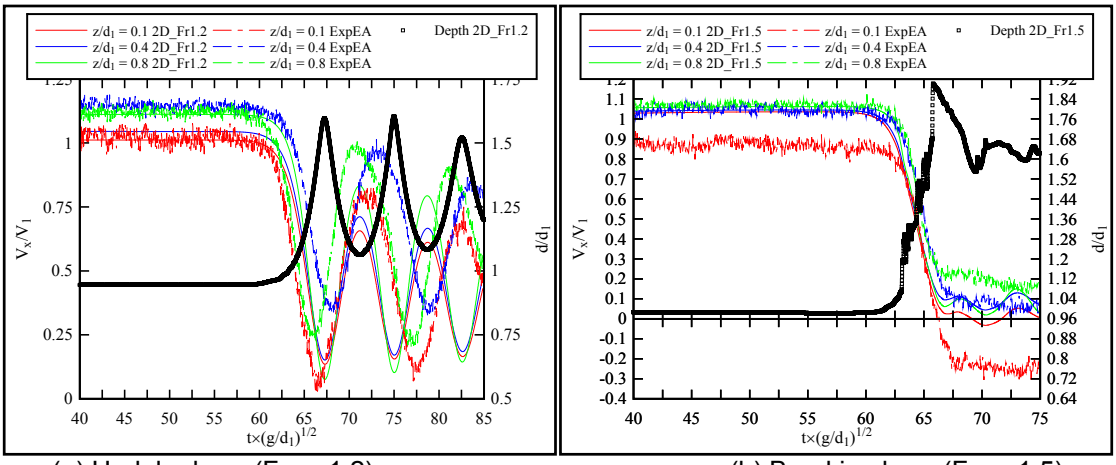


(a) Undular bore (Fr_1 = 1.2) (b) Breaking bore (Fr_1 = 1.5)

Figure 3. Dimensionless free-surface time evolution for undular and breaking bores of Fr_1 = 1.2 and 1.5, respectively; Comparisons between numerical simulation (2D_Fr), instantaneous experimental data (ExpInstan) and ensemble-averaged experimental data (ExpEA).

Overall, the time-variations of the numerically-simulated velocity data agreed well with the experimental data at all vertical elevations and for all velocity components. The longitudinal velocity was associated with a sharp deceleration following the arrival of breaking bores. The vertical velocity showed a sharp acceleration, then deceleration, following the breaking bore arrival. One feature which was absent from the numerical data set was the presence of a boundary layer in the initially steady flow as shown in the experimental results. Although a no-slip condition was imposed at the bottom boundary of the model, resulting in slightly lower steady flow velocity for lower vertical elevations (Figure 4), no obvious boundary layer such as highlighted in the experimental data was observed. This would be further addressed in the 3D simulation documented in Section 4.

Despite the absence of boundary layer, the present numerical data highlighted some longitudinal recirculation at the lowest vertical elevation during the propagation of breaking bores with complete gate closure (Figure 4b). Field and experimental studies have documented the presence of longitudinal recirculation next to the bed (often $0 < z/d_1 < 0.3$-0.5) beneath the bore front of both the breaking and undular bores (Wolanski et al., 2004; Chanson and Toi, 2015; Leng and Chanson, 2016a). This is characterised by a negative longitudinal velocity at the end of the deceleration following the bore passage, highlighted by the experimental data of breaking bores at z/d_1 = 0.1 in Figure 4b. At the same dimensionless vertical elevation next to the bed, the numerical data showed qualitatively some longitudinal recirculation velocity, however, with a time delay and lesser strength (red solid curve in Figure 4b).



(a) Undular bore (Fr_1 = 1.2) (b) Breaking bore (Fr_1 = 1.5)

Figure 4. Dimensionless time-variations of longitudinal velocity for undular and breaking bores of Fr_1 = 1.2 and 1.5, respectively; Comparisons between numerical simulation (2D_Fr) and ensemble-averaged experimental data (ExpEA).

The undular bore was characterised by a gentle free-surface rise, followed by a series of well-formed secondary waves. Meanwhile, the longitudinal velocity decreased in a smooth manner with the rise of free-surface, and oscillated quasi-periodically with the free-surface by a phase difference of π. The vertical velocity increased with the free-surface rise, then oscillated quasi-periodically with the free-surface by a phase difference of π/2. The numerical results reproduced the periodic oscillations in the free-surface, longitudinal and vertical velocity variations, however, with periods different from the experimental data. More specifically, the numerical data was associated with a shorter period. The minimum longitudinal velocity reached by the numerical model was higher than the experimental data, whereas the maximum longitudinal velocity of the secondary waves was lower than the experimental data (Figure 4). The difference was believed to be due to the lack of the third dimension and of sidewall friction, and hence resulting in a more energetic behavior of the bore.

3.3 Discussion

The 2D numerical modelling results showed large vortical structures formed both underneath and behind the front of breaking bores (Figure 5). Previous 2D CFD modelling by Khezri (2014) observed large vortical structures formed close to the bed, with a vertical dimension L_z up to 0.1 m ($L_z/d_1 \approx$ 0-0.5) underneath breaking bores of Froude number 1.5. The present study highlighted two types of vortical structures, one near the fluctuating free-surface behind the roller, and one close to bed. The vortical structures near the upper free-surface were observed almost immediately downstream of the breaking roller, with length scales of $L_x \approx 0.1$ m ($L_x/d_1 \approx 0.56$), and were advected downstream as the bore front propagated upstream. This type of vortical structures were formed by the plunging mechanism of the steepened bore front into the flow and was typically associated with pockets of air.

The vortical structures next to the bed were observed to be flat and elongated, with much larger length scales in the longitudinal direction compared to the vertical direction. The height of these vortical structures was approximately $L_z \approx 0.02$ m ($L_z/d_1 \approx 0.11$). Experimental studies of turbulent scales by Leng and Chanson (2017) highlighted presence of anisotropic vortical structures underneath the bore front. The vertical length scale L_z was up to 0.02 m ($L_z/d_1 \approx$ 0-0.11) for breaking bores with relatively large Froude numbers ($Fr_1 \geq 1.5$). The numerical results agreed with the experimental findings in terms of the vertical length scale of the vortical structures near the bed. The occurrence of vortical structures in the numerical model was found to be approximately 1 m downstream of the bore front, which was greatly delayed compared to experimental observations and past numerical results. The reason could be due to the 2D constraint, where bubble breakup mechanism was not allowed. Some numerical air bubbles were still observed more than 2 m ($\Delta x/d_1 \approx$ 0-11) following the leading edge of the bore, while the entrained bubbles would have risen to the free-surface in a real physical flow (Wang et al., 2016).

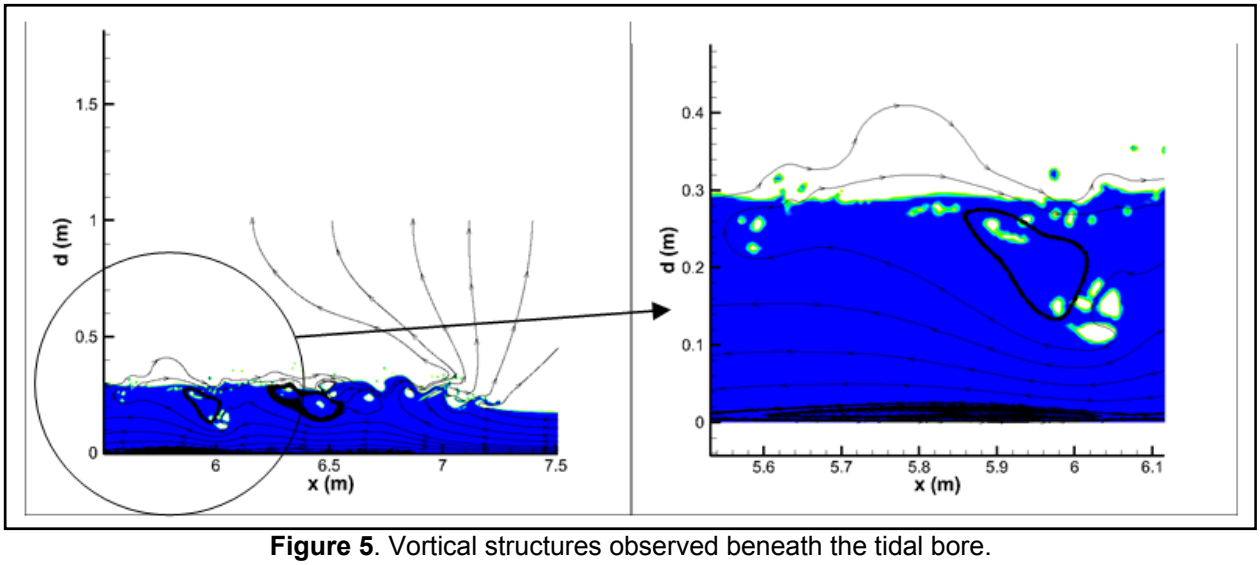
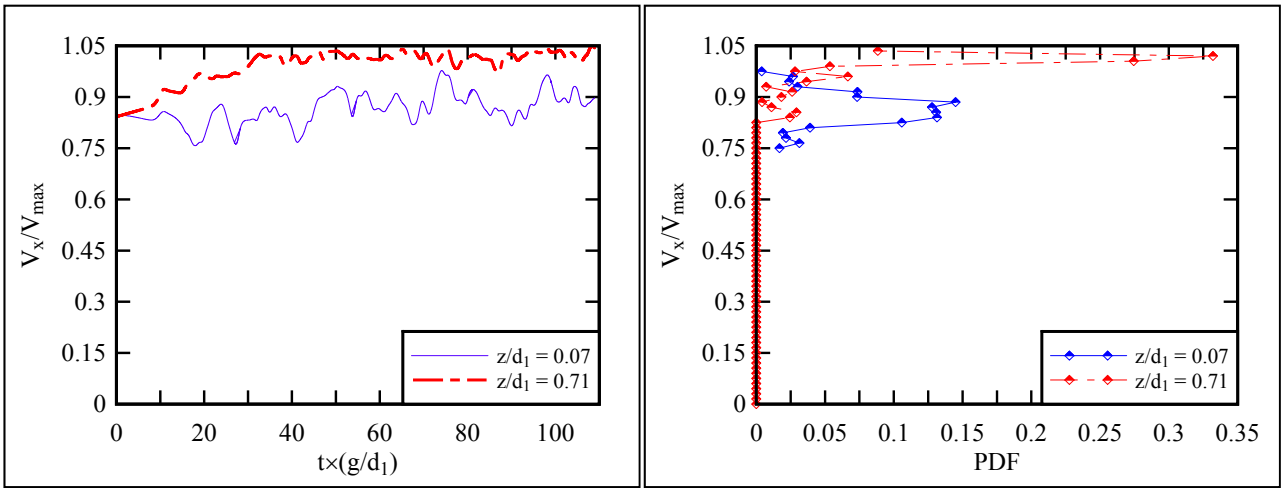


Figure 5. Vortical structures observed beneath the tidal bore.

4 3D STEADY FLOW SIMULATION

Figure 6 presents typical time series and PDFs of the longitudinal velocity components simulated at x = 8.5 m by the 3D CFD model (3D_Fr2.1) in steady flow at two different vertical elevations z. The 3D CFD model was run for a physical time span of 10.609 s (dimensionless time $t\times(g/d_1) \sim 110$), and the turbulence arrived at the velocity sampling location approximately after 1 s (dimensionless time $t\times(g/d_1) \sim 10$). Compared to the 2D results in the steady flow phase (Figure 3 and 4), the present 3D results clearly highlighted some turbulent fluctuations in all velocity components at all vertical elevations. Moreover, from the time series and

PDFs, variations in longitudinal velocity magnitudes with vertical elevations can be observed. Namely, the mean velocity at higher vertical elevation (z/d_1 = 0.71) was larger than that at lower elevation (z/d_1 = 0.07). This indicated the successful reproduction of a turbulent boundary layer.



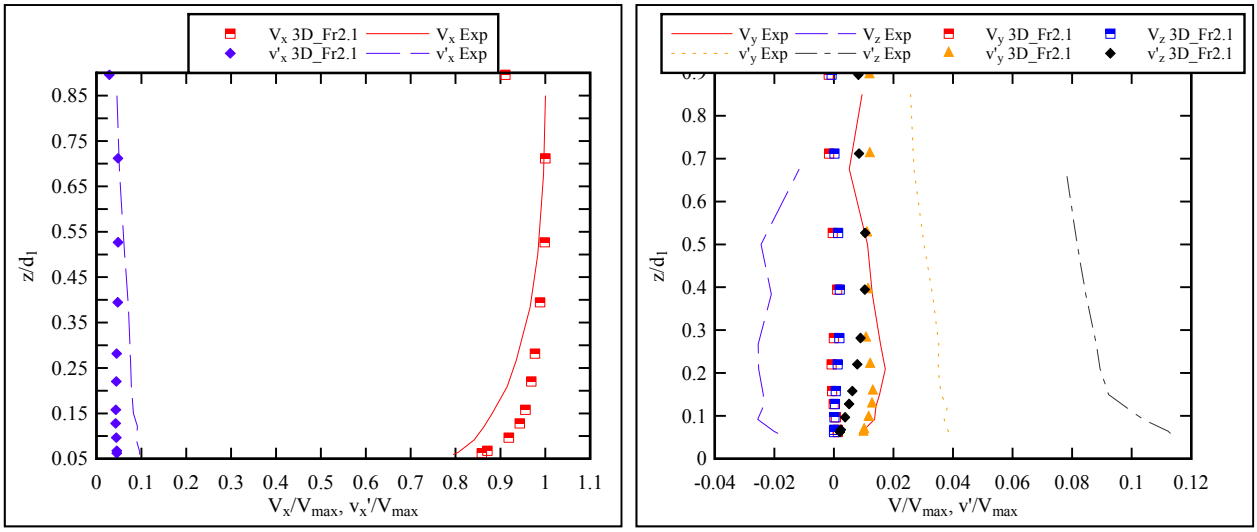
(a) Time-variations of instantaneous longitudinal velocity (b) PDF of longitudinal velocity

Figure 6. (a) Time-variations of the instantaneous longitudinal velocity simulated at x = 8.5 m, z/d_1 = 0.07 and 0.71 by the numerical model (3D_Fr2.1); (b) the corresponding probability density functions at two vertical elevations.

Figure 7 presents the vertical profile of the time-averaged velocity V and velocity fluctuations v' simulated at x = 8.5 m. The numerical results were compared to the experimental data. The numerical results were time-averaged over the entire 10.609 s time span. The experimental data was time-averaged over 30 s. As highlighted by Figure 7a, the numerical profile of the longitudinal velocity clearly showed a bottom boundary layer. The vertical profile of the longitudinal velocity simulated by the numerical model agreed well in shape and values with the experimental data, especially at the outer edge and outside of the boundary layer, except for the highest vertical elevation. Inside the boundary layer, the numerical data agreed better with the experimental data close to bed, then deviated from the experimental curve, before they coincided again near the outer edge. The longitudinal velocity fluctuations, highlighted by its standard deviation, agreed quantitatively and qualitatively with the experimental data, with better fit near the outer edge and outside of the boundary layer.

The vertical profile of the transverse and vertical velocity components simulated by the numerical model showed very small, near-zero magnitudes (Figure 7b). The values were orders of magnitudes smaller than the experimental data throughout the vertical range. The velocity fluctuations in the transverse and vertical directions were associated with values higher than the corresponding velocity component magnitudes, highlighted by the numerical data. Furthermore, the numerical data showed increase in velocity fluctuations with increasing vertical elevations, which agreed in terms of trend with the experimental data. Nevertheless, the experimental fluctuations were orders of magnitudes higher than the numerical ones, especially in the vertical direction. This could be attributed to the arrangement of the ADV head, which was known to over-estimate the vertical velocity fluctuations because of beam reflections and echo effects on the bed.

From the time series, turbulence was observed to arrive at the velocity sampling point after approximately 1 s and was fully developed after 5 s. The evolution of the numerical boundary layer was thus studied by time-averaging the longitudinal velocity at different vertical elevations over different time spans, from the entire time span of 10.609 s down to every 2 s. The results are presented in Figure 8, with comparison to experimental data. Overall, all numerical profiles were associated with the presence of boundary layers. The point at the highest vertical elevations (z/d_1 = 0.90) was associated with outlying values at all time. This was previous observed by Simon (2014) using the same numerical tool. For all time spans, the numerical data showed better agreement near the bottom and close to the outer edge of the boundary layer. Throughout the vertical range, the accuracy of the data seemed to be unaffected by the time span over which the numerical data was averaged, as long as the turbulence has reached the velocity sampling location. The findings seemed to suggest that, once the flow became turbulent and the boundary layer started to develop in the numerical model, the steady flow data would not evolve with longer time of simulation. The finding indicated good quality of numerically simulated turbulence, and highlighted the importance of using the actual experimental data for the numerical inlet boundary.



(a) Longitudinal velocity (b) Transverse and vertical velocity

Figure 7. Vertical time-averaged velocity profile of the steady flow longitudinal (a), transverse and vertical (b) velocity components at x = 8.5 m; Comparison between numerical (3D_Fr2.1) and experimental results (Exp).

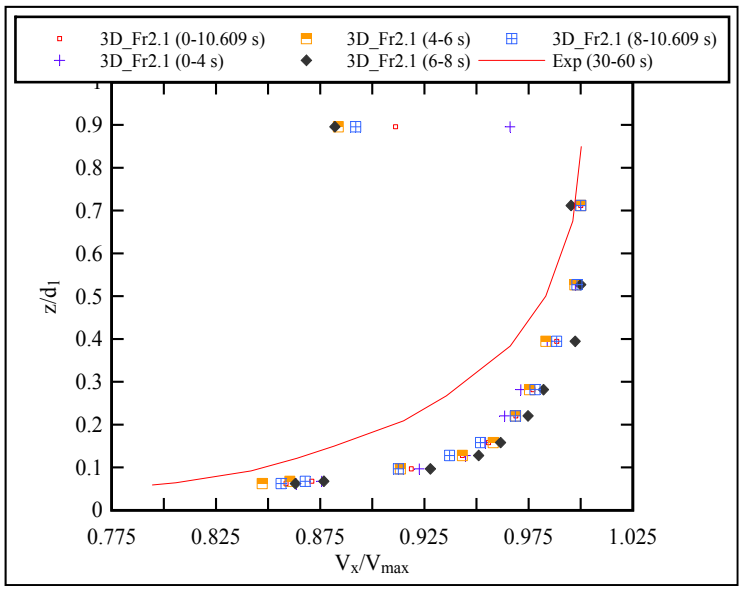


Figure 8. Vertical profile of time-averaged longitudinal velocity at x = 8.5 m, simulated by the numerical model and averaged over different time span (3D_Fr2.1).

5 CONCLUSIONS

The propagation of tidal bores in open channels was investigated numerically using 2D and 3D CFD model with Large Eddy Simulations. The numerical modelling investigated tidal bores with Froude numbers ranging from 1.2 to 2.1. The results of the numerical studies were compared and validated against experimental data collected under the same flow conditions. Overall, the 2D numerical results gave good approximation of the free-surface elevation associated with the bore propagation for the range of Froude numbers. The modelled velocity data agreed well with the experimental data outside the boundary layer, and highlighted some longitudinal recirculation underneath breaking bores, although with a delayed occurrence. The numerical model tended to over-estimate the bore celerity and under-estimate the bore height for both breaking and undular bores. Streamline tracing of the numerical results highlighted elongated thin vortical structures behind the bore front and close to bed, with vertical length scale comparable to experimental findings. The 3D steady flow results highlighted a developing boundary layer in the initially steady flow before the bore generation. The thickness of the boundary layer was comparable to experimental results, and the numerical and experimental velocity profiles agreed closely next to the bottom and near the outer edge of the boundary layer. The development of the numerical boundary layer over time was examined. Results suggested that after the turbulence has reached the velocity sampling location, the numerical boundary layer changed very little with longer simulation time.

ACKNOWLEDGEMENTS

The authors acknowledge Dr. Stéphane Glockner (Université de Bordeaux, I2M, Laboratoire TREFLE, Pessac, France) for his expert assistance. The authors also wish to thank the Aquitaine Regional Council for financial support towards a 432-processor cluster investment, located in the I2M laboratory. This work was granted access to the HPC resources of CINES, under allocation 2016-x2012026104 made by GENCI (Grand Equipement National de Calcul Intensif). Computer time for this study was also provided by the computing facilities at MCIA (Mésocentre de Calcul Intensif Aquitain) of the Université de Bordeaux and of the Université de Pau et des Pays de l'Adour. The financial support of the Graduate Student International Travel Award provided by UQ Graduate School is acknowledged.

REFERENCES

Chanson, H. (2011). *Tidal bores, Aegir, Eagre, Mascaret, Pororoca: Theory and Observations.* World Scientific, Singapore, 220.

Chanson, H. & Toi, Y.H. (2015). Physical Modelling of Breaking Tidal Bores: Comparison with Prototype Data. *Journal of Hydraulic Research*, 53(2), 264-273.

Chanson, H., Lubin, P. & Glockner, S. (2012). Unsteady Turbulence in a Shock: Physical and Numerical Modelling in Tidal Bores and Hydraulic Jumps. *Turbulence: Theory, Types, and Simulation*, Nova Science Publishers, Hauppauge, N.Y: R.J. Marcuso, 3, 113–148.

Falgout, R., Jones, J. & Yang, U. (2006). *The Design and Implementation of Hypre, a Library of Parallel High Performance Preconditioners.* Numerical solution of partial differential equations on parallel computers, Lecture notes in computational science and engineering. Springer Berlin Heidelberg, 267-294.

Furuyama, S.I. & Chanson, H. (2010). A Numerical Solution of a Tidal Bore Flow. *Coastal Engineering Journal*, 52(3), 215-234.

Goda, K. (1979). A Multistep Technique with Implicit Difference Schemes for Calculating Two-Or Three-Dimensional Cavity Flows. *Journal of Computational Physics,* 30(1), 76-95.

Helluy, P., Golay, F., Caltagirone, J.P., Lubin, P., Vincent, S., Drevard, D. & Lesage, A.C. (2005). Numerical Simulations of Wave Breaking. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(3), 591-607.

Hornung, H.G., Willert, C. & Turner, S. (1995). The Flow Field Downstream of a Hydraulic Jump. *Journal of Fluid Mechanics*, 287, 299-316.

Jarrin, N., Benhamadouche, S., Laurence, D. & Prosser, R. (2006). A Synthetic-Eddy Method for Generating Inflow Conditions for Large-Eddy Simulations. *International Journal of Heat and Fluid Flow*, 27(4), 585–593.

Jarrin, N., Prosser, R., Uribe, J.C., Benhamadouche, S. & Laurence, D. (2009). Reconstruction of Turbulent Fluctuations for Hybrid RANS/LES Simulations Using a Synthetic-Eddy Method. *The Seventh International Symposium on Engineering Turbulence Modelling and Measurements, ETMM7,* 30(3), 435–442.

Khezri, N. (2014). Modelling Turbulent Mixing and Sediment Process beneath Tidal Bores: Physical and Numerical Investigations, *Ph.D. thesis*. School of Civil Engineering, The University of Queensland, Brisbane, Australia, 267.

Koch, C. & Chanson, H. (2008). Turbulent Mixing Beneath an Undular Bore Front. *Journal of Coastal Research*, 24(4), 999-1007.

Koch, C. & Chanson, H. (2009). Turbulence Measurements in Positive Surges and Bores. *Journal of Hydraulic Research*, 47(1), 29-40.

Leng, X. & Chanson, H. (2015a). Turbulent Advances of a Breaking Bore: Preliminary Physical Experiments. *Experimental Thermal and Fluid Science*, 62, 70-77.

Leng, X. & Chanson, H. (2015b). Breaking Bore: Physical Observations of Roller Characteristics. *Mechanics Research Communications*, 65, 24-29.

Leng, X. & Chanson, H. (2015c). *Unsteady Turbulence During the Upstream Propagation of Undular and Breaking Tidal Bores: An Experimental Investigation*, Hydraulic Model Report No. CH98/15, School of Civil Engineering, The University of Queensland, Brisbane, Australia, 235.

Leng, X. & Chanson, H. (2016a). Coupling between Free-Surface Fluctuations, Velocity Fluctuations and Turbulent Reynolds Stresses during the Upstream Propagation of Positive Surges, Bores and Compression Waves. *Environmental Fluid Mechanics*, 16(4), 695-719.

Leng, X. & Chanson, H. (2016b). Unsteady turbulent velocity profiling in open channel flows and tidal bores using a Vectrino profiler, *Hydraulic Model Report No. CH101/15*, School of Civil Engineering, The University of Queensland, Brisbane, Australia, 118.

Leng, X. & Chanson, H. (2017). Integral Turbulent Scales in Unsteady Rapidly Varied Open Channel Flows. *Experimental Thermal and Fluid Science*, 81, 382-395.

Lubin. P. (2004). Large Eddy Simulation of Plunging Breaking Waves, *Ph.D. Thesis*. Université Bordeaux I, France (in English).

Lubin, P., Chanson, H. & Glockner, S. (2010a). Large Eddy Simulation of Turbulence Generated by a Weak Breaking Tidal Bore. *Environmental Fluid Mechanics,* 10(5), 587-602.

Lubin, P., Glockner, S. & Chanson, H. (2010b). Numerical Simulation of a Weak Breaking Tidal Bore. *Mechanics Research Communications,* 37(1), 119-121.

Lubin, P., Vincent, S., Abadie, S. & Caltagirone, J.P. (2006). Three-Dimensional Large Eddy Simulation of Air Entrainment under Plunging Breaking Waves. *Coastal Engineering,* 53(8), 631-655.

Reichstetter, M. (2011). Hydraulic Modelling of Unsteady Open Channel Flow: Physical and Analytical Validation of Numerical Models of Positive and Negative Surges, *MPhil Thesis*. Brisbane, Australia: School of Civil Engineering, The University of Queensland.

Sagaut, P. (2006). *Large Eddy Simulation for Incompressible Flows: An Introduction*. 3rd. Berlin, Heidelberg, New York: Springer-Verlag Berlin and Heidelberg.

Simon, B. (2014). Effects of Tidal Bores on Turbulent Mixing: A Numerical and Physical Study in Positive Surges, *PhD Thesis*. School of Civil Engineering, The University of Queensland, Brisbane, Australia, 259.

Simon, B., Lubin, P., Glockner, S. & Chanson, H. (2011). Three-Dimensional Numerical Simulation of the Hydrodynamics Generated by a Weak Breaking Tidal Bore. *34th IAHR World Congress, 33rd Hydrology and Water Resources Symposium and 10th Conference on Hydraulics in Water Engineering*.

Treske, A. (1994). Undular Bores (Favre-Waves) in Open Channels-Experimental Studies. *Journal of Hydraulic Research*, 32(3), 355-370.

Wang, H., Leng, X. & Chanson, H. (2016). Hydraulic Jumps and Breaking Bores: Modelling and Analysis. *Proceedings of the Institution of Civil Engineers-Engineering and Computational Mechanics*, 1-18.

Wolanski, E., Williams, D., Spagnol, S. & Chanson, H. (2004). Undular Tidal Bore Dynamics in the Daly Estuary, Northern Australia. *Estuarine, Coastal and Shelf Science*, 60(4), 629-636.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

A GPU MODEL FOR SIMULATING HYDRODYNAMIC HAZARDS: APPLICATION TO TSUNAMI BENCHMARKS

FAHAD MOHAMMED[(1)]

[(1)] Center for Property Risk Solutions,
FM Global, Research Division, Norwood MA, USA
fahad.mohammed@fmglobal.com

ABSTRACT

This paper describes a computationally efficient, fast and accurate model to simulate tsunami generation, propagation and inundation. Tsunamis are simulated using the non-linear shallow water wave (NSWE) model in both the Spherical and the Cartesian coordinate system to simulate trans-oceanic wave propagation and inundation. The NSW equations are solved using a finite volume based shock capturing algorithm. The positivity preserving algorithm accurately preserves the steady "C" state of the NSWE equations. The solver is implemented on GPGPUs (General-Purpose computing on Graphics Processing Unit) using NVIDIA's Compute Unified Device Architecture (CUDA) programming platform. The model has been successfully verified and validated with analytical, experimental and field cases. In this paper, the model validation is summarized against the benchmark cases provided by the National Tsunami Hazard Mitigation Program (NTHMP) of National Oceanic and Atmospheric Administration (NOAA) USA. The validation cases include analytical solutions, laboratory experiments and field data.

Keywords: GPGPU; tsunami; high-performance computing; non-linear shallow water wave model; shock capturing methods.

1 INTRODUCTION

Tsunamis are long water waves that are primarily caused by earthquakes. They are amongst the most dangerous natural disasters and pose considerable risk on most of the world's coastlines. Tsunami events may result in large scale ocean basin wide destruction and fatalities, and have the potential to damage local and global economies. The scale of the destruction and damage due to tsunamis requires efficient risk mapping for tsunami hazard mitigation, humanitarian efforts and reduce impact to global financial markets. Numerical modeling of tsunami propagation and inundation is an effective and an essential tool that can provide risk assessment, evacuation planning and real-time forecasting. Due to the length scales of the tsunami waves, numerical solutions of non-linear shallow water wave (NSWE) equations are commonly used to simulate tsunami generation, propagation and inundation on land.

Numerical modeling of tsunami generation, propagation and inundation have been historically performed by solving the NSWE in both the spherical and the Cartesian coordinate systems for trans-oceanic as well as local simulations. Various approaches have been employed in solving the NSWEs including the finite difference method, finite volume method, finite element method and smoothed particle hydrodynamics (SPH). Earlier tsunami models have been based on finite difference leap frog schemes such as MOST by Titov and Synolakis (1995), TUNAMI by Goto et al. (1997) and COMCOT by Wang and Liu (2006).

More recently finite volume based Godunov schemes have been implemented successfully to simulate tsunami propagation and inundation (Kim et al., 2007; Popinet, 2011; Leveque at al., 2011). The advantages of these schemes include better conservation property, shock-capturing ability including wet-dry transition and ease of implementation on both structured and unstructured grids. Because of these reasons, a second-order shock capturing finite volume model is used to solve the NSWE in this work for tsunami simulations. However, simulating tsunami propagation and inundation on large grids render these methods computationally expensive.

Due to the advancement of computational hardware, various methods have been employed to speed up these simulations. Leveque et al. (2011) used adaptive block meshes to increase the speed of their finite volume solver. Popinet (2011) implemented their finite volume solver on a dynamically adaptive quadtree grids. Liang et al. (2015) demonstrated another simplified adaptive grid system to speed up their model simulations. While these techniques have demonstrated model acceleration by several orders (Liang et al., 2015), they have also been subject to problems dealing with full conservation of mass and flux gradients during the grid adaptation.

A more recent high performing computing capability has emerged involving the use of graphical processing units (GPUs). While the GPUs have been commonly used in the gaming and digital image processing regimes for quite some time, they are gaining popularity and show promise in their application on

high performance computing (Brodtkorb, 2010). The advantage stems from the availability of hundreds or streaming processing units on a single GPU compared with the CPU, which provide powerful parallel computing capability. Recently numerous GPU-accelerated models have been developed and used in many areas of scientific computing such as CFD, magneto-hydrodynamics, gas dynamics and planetary dynamics. (Wang et al., 2010; Kuo et al., 2011; Rossinelli et al., 2011; Schive et al., 2012)

In the present work, a GPU-based NSWE solver is presented for tsunami simulations. The model solves the NSWE using an explicit second order TVD based shock capturing algorithm which is positivity preserving in nature. The model preserves the exact "C" state of the NSWE system and accurately captures the wet-dry interface. The model has been fully verified and validated using the series of benchmark cases designed by NTHMP, NOAA (Synolakis et al., 2007). Here these validations are presented.

2 NUMERICAL MODEL

The 2D hyperbolic conservation form of the NSWE in Cartesian coordinates may be written as

$$\frac{\partial \boldsymbol{u}}{\partial t}+\frac{\partial \boldsymbol{f}}{\partial x}+\frac{\partial \boldsymbol{g}}{\partial y}=\boldsymbol{S} \qquad [1]$$

where (x,y) are the Cartesian coordinates, t denotes time, $\boldsymbol{u}$ is the vector of conservative variables, $(\boldsymbol{f},\boldsymbol{g})$ are the vectors containing $x-$ and $y-$ direction flux functions and $\boldsymbol{S}$ is a vector containing the source terms and diffusive fluxes. The conservative form of the NSW equations in a spherical coordinate system may be written as:

$$\frac{\partial \boldsymbol{u}}{\partial t}+S_o\frac{\partial \boldsymbol{f}}{\partial x}+\frac{\partial \boldsymbol{g}}{\partial y}+\boldsymbol{f_c}=\boldsymbol{S} \qquad [2]$$

where the spherical coordinates (ψ,ϕ) representing the longitudes and the latitudes are scaled following Shi et al. (2012) as $x=Rcos\phi_o(\psi-\psi_o)$ and $y=R(\psi-\psi_o)$, R is earth's radius, $S_o=cos\phi_o/\cos\phi$ and $\boldsymbol{f_C}$ is the vector containing additional Coriolis force terms. Without loss of generality, the conservative variables and the flux functions are defined below:

$$\boldsymbol{u}=[h,hu,hv]^T \qquad [3a]$$

$$\boldsymbol{f}=\left[hu,hu^2+\frac{1}{2}gh^2,huv\right]^T \qquad [3b]$$

$$\boldsymbol{g}=\left[hv,huv,hv^2+\frac{1}{2}gh^2\right]^T \qquad [3c]$$

$$\boldsymbol{f_c}=[0,-fhv,fhu]^T \qquad [3d]$$

where h is the total water depth, (u,v) are the 2-D horizontal depth-averaged velocities of the water column, $b(x,y)$ is the topography/bathymetry above a datum and $g=9.81\,m^2/s$ is the gravitational acceleration. The source vector varies depending on the choice of the coordinate system used as:

$$\boldsymbol{S}=\left[\begin{array}{c|c}0 & \frac{1}{R}hv\tan\phi \\ -gh\frac{\partial b}{\partial x}-C_f u\sqrt{u^2+v^2} & -S_o gh\frac{\partial b}{\partial x}-C_f u\sqrt{u^2+v^2} \\ -gh\frac{\partial b}{\partial y}-C_f v\sqrt{u^2+v^2} & -gh\frac{\partial b}{\partial y}-C_f v\sqrt{u^2+v^2}\end{array}\right] \qquad [4]$$

where the first column of the source vector $\boldsymbol{S}$ is used in the Cartesian coordinate system for nearfield inundation and the second column is used in the Spherical coordinate system for global basin wide tsunami propagation. $C_f=gn^2/h^{1/3}$ is the bed roughness coefficient with n as the Manning's friction coefficient.

The governing equations given in Eq. [1] and [2] are solved using a finite volume, Godunov-type shock capturing based algorithm. The algorithm is a 2nd order TVD ENO-MUSCL based scheme resulting in a second order accuracy in both space and time (Kurganov et al., 2000; Van Leer, 1979, Leveque, 2002; Toro, 1999). The partial differential equations in Eqs. [1, 2] are transformed into a series of ordinary differential equations by integrating the equations over a finite volume. The resulting system of the ODE's may be written in cell (ij) as:

$$\frac{d\boldsymbol{u_{ij}}}{dt}+\frac{\boldsymbol{f_E}-\boldsymbol{f_W}}{\Delta x}+\frac{\boldsymbol{g_N}-\boldsymbol{g_S}}{\Delta y}=\boldsymbol{S_b}+\boldsymbol{S_F} \qquad [5]$$

where the source terms are split into a bed slope term $\boldsymbol{S_b}$ and a friction term $\boldsymbol{S_F}$. The subscripts $(\boldsymbol{E, W, N, S})$ represent the cell edges. The left and right values of the conservative variables on either side of a cell interface are obtained by a second-order linear reconstruction of the variable considering a MUSCL slope limiter (van Leer, 1979). The MUSCL reconstruction is employed to avoid spurious oscillations of the variables in case of discontinuities and steep gradients. The minmod limiter based on the cell-center values of the variables is used, which offers good numerical stability in an efficient computational step. The interface cell flux is constructed from the left and right values of the variables using the Lax-Friedrich flux shown for the east interface for illustration as:

$$\boldsymbol{f_E} = \frac{1}{2}\left(\boldsymbol{f_E^l} + \boldsymbol{f_E^r}\right) + \frac{\alpha}{2}\left(\boldsymbol{u_E^l} - \boldsymbol{u_E^r}\right) \quad [6]$$

Where $\boldsymbol{u_E^l}$ and $\boldsymbol{u_E^r}$ are values of the linear reconstruction of the variable $\boldsymbol{u}$ around the cell interface, $\boldsymbol{f_E^l}$ and $\boldsymbol{f_E^r}$ are the left and right flux values and $\alpha = \max_{\mathrm{ij}} \lambda_{ij}$ is the maximum characteristic wave speed in the computational domain, with $\lambda_{ij} = \max(|u| + \sqrt{gh}, |v| + \sqrt{gh})$ is the characteristic wave speed in cell ij. A semi-implicit formulation for the frictional term is employed which simply results in the following time-stepping function:

$$\boldsymbol{u_{ij}^{n+1}} = \frac{\left(\boldsymbol{u_{ij}^n} - \frac{\Delta t}{\Delta x}\left(\boldsymbol{f_E} - \boldsymbol{f_W}\right) - \frac{\Delta t}{\Delta y}\left(\boldsymbol{g_N} - \boldsymbol{g_S}\right) + \Delta t \boldsymbol{S_b}\right)}{\left(1 - f_{Sf}\Delta t\right)} \quad [7]$$

where $\boldsymbol{f_{Sf}} = [0, gn^2\sqrt{u^2+v^2}/h^{2/3}, gn^2\sqrt{u^2+v^2}/h^{2/3}]$ is the Manning's friction term. Thus, the simplified form of Eqs. [1,2] can be reduced as:

$$\boldsymbol{u^{n+1}} = \boldsymbol{u^n} + \Delta \mathrm{t}\mathcal{L}(\boldsymbol{u^n}) \quad [8]$$

A positivity preserving reconstruction of the water depth and the velocities given by Liang (2010) is employed to accurately track the wet-dry moving interface, ensure non-negativity of the water depth and balancing the exact "C" state of the NSW equations. As a part of this reconstruction, a central differencing scheme is employed to discretize the bed slope terms which ensures the exact reconstruction of the still water surface at rest. The time integration is performed by a 2nd order TVD Runge-Kutta scheme (Gottlieb and Shu, 1998) given as:

$$\begin{aligned} \boldsymbol{u^{(1)}} &= \boldsymbol{u^n} + \Delta \mathrm{t}\mathcal{L}(\boldsymbol{u^n}) \\ \boldsymbol{u^{(2)}} &= \boldsymbol{u^{(1)}} + \Delta \mathrm{t}\mathcal{L}\left(\boldsymbol{u^{(1)}}\right) \\ \boldsymbol{u^{n+1}} &= \frac{1}{2}\left(\boldsymbol{u^n} + \boldsymbol{u^{(2)}}\right) \end{aligned} \quad [9]$$

The scheme given in Eqs. [1] to [9] is a semi-implicit finite volume based shock capturing scheme which ensures non-negativity of the depth and accurate capturing of the wet-dry interface. The numerical methodology is stable as long as the time step satisfies the Courant-Friedrich-Levy (CFL) condition:

$$\Delta t \leq \frac{1}{2}\min\left[\frac{\Delta x}{\alpha}, \frac{\Delta y}{\alpha}\right] \quad [10]$$

The boundary conditions are imposed based on the nature of the boundary condition during the explicit time stepping algorithm. For inflow and outflow boundary conditions, a local Riemann problem is solved at the appropriate boundary cell by imposing the local flow conditions. For open boundaries, a zero normal flow boundary condition is imposed which prevents the rise of any spurious oscillations at the boundaries. Wall boundary conditions are imposed by negative flow velocities and zero water depth at the wall boundaries. Initial conditions are imposed based on the nature of the problem being solved. For tsunamis, an earthquake fault model specified by the geometry of the rupture area and the slip value is specified. The bottom bed rupture is computed from the fault parameters using the Okada solution (Okada, 1985) and is assumed to rupture instantaneously at $t = 0$ providing the initial condition for tsunami wave generation.

3 GPGPU IMPLEMENTATION

The finite volume based NSW solver given in Eq. [9] is implemented for high performance computing on GPUs using the CUDA programming framework. A GPU is made up of multiple cores which are all designed to perform stream processing of vector operations simultaneously. The finite volume method is an ideal solver to run on the GPUs since almost all the cells in the computational domain undergo the same mathematical

operations, making the solver computationally fast and efficient. The NSW solver is implemented in a massively parallel environment on the GPU's using CUDA/C while the pre- and post-processing components of the software are programmed in C. The parallel component of the algorithm is identified and efficiently implemented as GPU kernels (NVIDIA, 2016).

The program can be split into three distinct components. In the first pre-processing step, appropriate memory is allocated using the Unified memory which is accessible to both the CPU and the GPU and the initial bathymetry/topography data is loaded into the memory. The computational grid is split into blocks of 16×16 cells. The computational grid is transformed into a CUDA grid which consists of multiple 2D computational blocks. The overlapping blocks are designed to improve the speed of data access using the available shared memory on the GPUs. In the second step, the mathematical calculations of the NSW solver are performed wholly on the GPUs by the parallel kernels of the program. In the final step, after the required calculations are completed, the data is exported for post-processing and analyses of the results. Two main kernels are defined in the program. The first is a reduction kernel which is used to compute the maximum characteristic wave speed in the system.

Then the maximum permissible time step is computed based on the characteristic speed. The second kernel is the actual computational kernel which uses several defined device functions to compute the Minmod limiters, MUSCL reconstruction, flux calculation and time stepping to obtain the solution at the new time level. Within each time step, the main kernel is invoked twice to account for the two-step Runge-Kutta scheme. Both the kernels are executed entirely on the GPUs.

During the execution, the kernels launch a grid of thread blocks. The individual threads correspond to each computational cell in the domain of the system. Each block is made up of 16×16 threads and has an overlap of one row on the edges. The blocks are launched asynchronously and the overlap ensures that the blocks can be launched in any order without depending on the data from the adjacent blocks. The threads within a block are launched synchronously and the data has to be coalesced to ensure that there is no "out of warp" data access, which can slow down the computational time. In this work, Unified global memory, with 16×16 overlapping thread blocks and shared memory is implemented in the program.

4 VERIFICATION & VALIDATION

4.1 NSW Model Verification

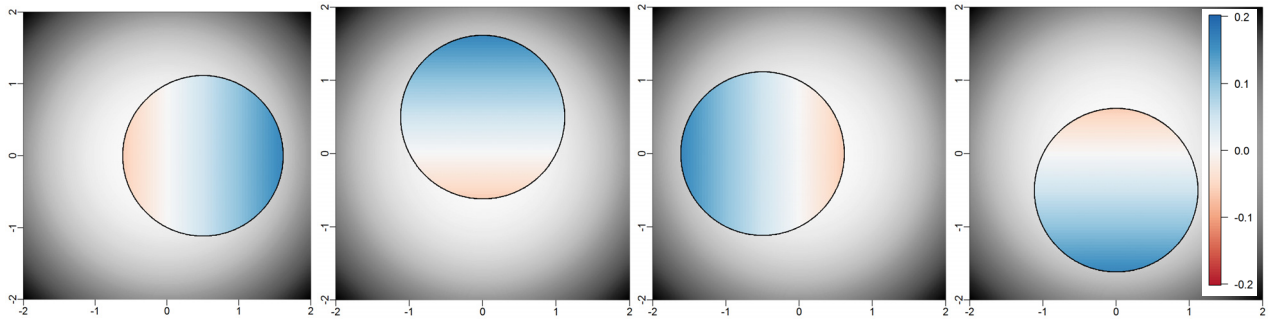
The GPU-NSWE model accuracy was verified through the solutions of Thacker's 2D planar problem in a parabolic basin (Thacker, 1981). The bottom of the parabolic basin is given as

$$b(x,y) = b_o\left(1 - \frac{r^2}{a_o^2}\right) \qquad [11]$$

For the planar case, the solution of the free surface and the velocity components are given as

$$\begin{aligned} \eta(x,y,t) &= \frac{\eta_o}{h_o a_o^2}(2xcos(\omega t) + 2ysin(\omega t)) \\ u(x,y,t) &= -\eta\omega \sin(\omega t) \\ v(x,y,t) &= \eta\omega \cos(\omega t) \end{aligned} \qquad [12]$$

with $\omega = \sqrt{2gh_o/a_o}$, $\eta_o = 0.5, a_o = 1, h_o = 0.1$ and period of oscillation, $T = 2\pi/\omega$.



(a) $t = 4T$ **(b) $t = 16T/4$** **(c) $t = 9T/4$** **(d) $t = 19T/4$**

Figure 1. Numerical solution of Thacker's planar solution shown at $t =$ (a) $4T$, (b) $16T/4$, (c) $9T/4$ and (d) $19T/4$. The parabolic basin is shown in grayscale in the background. The shoreline is shown by the black line.

The free surface solution for Thacker's 2D planar solution is shown in Figure 1. The simulation has been performed for a total time of $10T$. The results are shown starting at $4T$ for every $T/4$ s. The solutions are exact

after $4T$ and compares well with the exact analytical solution. The RMS error of the water depth compared with the analytical solution is measured to conform the model accuracy. The simulations are performed for various spatial discretization $\Delta x = 10^{-1} \sim 10^{-6}$ and temporal discretization, $\Delta t = 10^{-3} \sim 10^{-6}$. The RMS error approximately follows a second order law $\epsilon \sim O(\Delta x^2, \Delta t^2)$, verifying the second order accuracy of the model in space and time as shown in Figure 2.

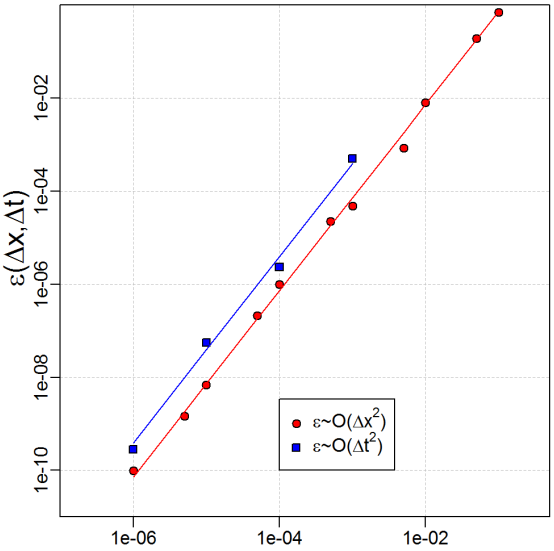


Figure 2. RMS errors for Thacker's planar solutions compared with the analytical solution is shown for different orders of spatial and temporal discretization at $5T$.

4.2 NSW Model Validation

Model validation is essential, especially if the model is used in the production of accurate hazard mitigation, public safety and risk tool products. A suite of benchmark cases for validating the tsunami models were established based on the OAR-PMEL-135 report (Synolakis et al., 2007). The benchmark cases were designed for model verification and validation to include analytical solutions, laboratory experiments and field measurements. The cases chosen test certain features of the model, but not all of the tsunami models. The list of the benchmark cases that were used for the model validation is shown in Table 1.

Table 1. NTHMP benchmark test cases based on the recommendations of Synolakis et al. (2007).

Benchmark Test	Category	Description
BP1	Analytical Solution	Single wave on a simple beach
BP4	Laboratory	Solitary wave on a simple beach
BP6	Experiment	Solitary wave on a conical island
BP9	Field Measurements	Okushiri Island Tsunami

The NTHMP model validation is quantified by three error quantities. Normalized root mean square deviations error in an area of interest of span of time, relative error for maximum wave height or runup and relative error for multiple runup values in a specific region were recorded.

The allowable errors for the three main categories of the benchmarking tests as recommended by OAR-PMEL-135, suggested by Synolakis et al. (2007) are summarized below in Table 2. It is suggested to have error criteria set at <10% for laboratory experiments for both breaking and non-breaking waves. The normalized root mean square deviation (RMS) is applied to all the observed data points withing a space segment or a time period. It is defined as

$$RMS = \frac{1}{\eta_{e_{max}} - \eta_{e_{min}}} \sqrt{\frac{\sum_{i=1}^{n} \left(\eta_{e_i} - \eta_{m_i}\right)^2}{n}} \qquad [13]$$

where the subscript e denotes the observed or expected value and m denotes the modeled value. The model's predictive accuracy for maximum wave amplitude or runup is quantified by the MAX error, irrespective of the location in time and space. It is quantified as the relative difference between the expected and the modeled maximum value. Finally a relative error ERR is defined for multiple values collected in a specific location or region. The local data set is reduced to define three values, minimum, maximum and mean, which

represents the physics at a particular location (for e.g. runup or drawdown). The error is then defined similarly as the MAX error with these localized values. The allowable errors according to Synolakis et al. (2007) are shown in Table 2.

Table 2. Allowable errors in the NTHMP benchmarking tests suggested by Synolakis et al. (2007).

Category	Allowable errors (max/err/rms)
Analytical Solution	<5%
Laboratory Measurements	<10%
Field Measurements	<20%

4.2.1 BP1 Analytical solution: Solitary wave on a simple beach

The test case is based on the analytical solution of a solitary wave propagation and runup on a simple beach (Titov and Synolakis, 1995). The bathymetry consists of a channel of constant depth d connected to a plane sloping beach of angle $\beta = 2.88°$. $x = 0$ is the initial shoreline location and the toe of the beach is situated at $x = X_o = dcot\beta$. A solitary wave of height H is initially centered at $x = X_s$ at $t = 0$. The benchmark test is evaluated for the case of $H = 0.0185d$. The initial conditions for the test is set as

$$\eta(x,0) = 0.0185\,\mathrm{sech}^2\left(\sqrt{3 * \frac{0.0185}{4}}(x - X_s)\right)$$
$$u(x,0) = -\sqrt{\frac{g}{d}}\eta(x,0) \qquad [14]$$
$$v(x,0) = 0$$

While the analytical solution is one-dimensional, the 2D model with one cell width is set up to solve this test case. The model simulations of the free surface profiles are compared with analytical solutions at specified time steps. Further the free surface elevations at two locations are also compared with the analytical solutions. The model errors for the BP1 test case are summarized in Table 3, which are all such that $\epsilon \leq 5\%$.

Table 3. NTHMP model errors for BP1 shown for surface profiles at $t = (35{\sim}65)$ along with the mean error in the entire duration of the computation and water surface time series at $x = (0.25{,}9.95)$ shown along with the maximum amplitude and run up error.

	RMS	MAX
$t = 35$	1%	1%
$t = 40$	1%	1%
$t = 45$	1%	0%
$t = 50$	1%	0%
$t = 55$	0%	1%
$t = 60$	0%	2%
$t = 65$	0%	1%
MEAN	1%	1%
$x = 9.95$	1%	1%
$x = 0.25$	1%	1%
Mean	1%	1%

4.2.2 BP4 Laboratory experiment: Solitary wave on a simple beach

This test case coincides with the BP1 test case and requires model comparison with measured experimental data. The experiment was performed in a 32 meter long wave tank at California Institute of Technology (Hammack, 1972; Goring, 1978; Synolakis 1986). More than 40 experiments with solitary waves were conducted with $H/d = 0.021 {\sim} 0.626$. The wave with $H/d = 0.0185$ resulted in a non-breaking wave test case, simulating a realistic tsunami.

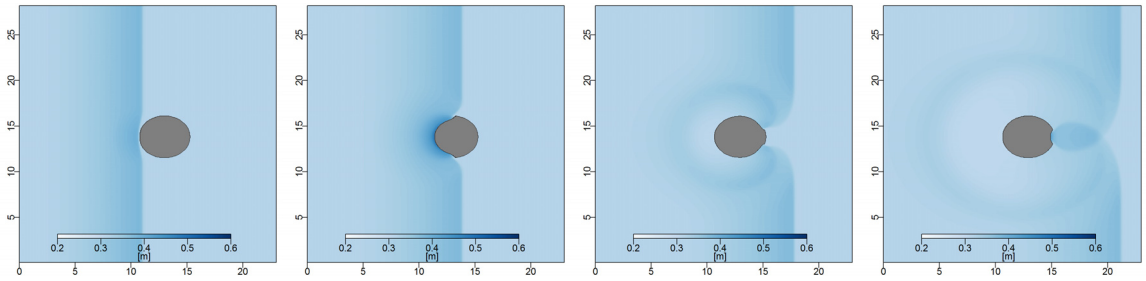
Hence this case was used as a test case for the benchmark validation. The initial and boundary conditions for BP4 are the same as BP1 with a constant water depth of 0.3 m. The measured free surface and runup locations are compared as part of the BP4 test case. The model errors for BP4 test cases are summarized in Table 4 which are all such that $\epsilon \leq 10\%$.

Table 4. NTHMP model errors for BP4 shown for surface profiles at $t = (30{\sim}70)$ along with the mean error in the entire duration of the computation shown for the cases of $H = 0.0185$ and $H = 0.3$.

	RMS	MAX
	$\boldsymbol{H = 0.0185}$	
$t = 30$	3%	4%
$t = 40$	3%	5%
$t = 50$	4%	3%
$t = 60$	6%	8%
$t = 70$	6%	4%
Mean	6%	5%
	$\boldsymbol{H = 0.3}$	
$t = 15$	4%	5%
$t = 20$	5%	6%
$t = 25$	4%	6%
$t = 30$	3%	4%
Mean	6%	5%

4.2.3 BP6 Laboratory experiment: Solitary wave on a conical island.

This test case was chosen to perform the validation of a three-dimensional case. This test case is based upon a laboratory experiment designed to study the inundation of Babi Island due to the December 12, 1992 tsunami (Yeh et al., 1994). While the tsunami attacked the north face of the conical island, extremely high inundation was observed on the south side of this island. A model of this island was built and tested in the US Army Engineer Waterways Experimental Station (Briggs et al., 1995). The test was conducted in a 29.3×30m wave basin with a wavemaker on one end of the wave basin. The island was designed as a truncated right circular cone with diameters of 7.2 m and the toe and 2.2m at the crest. The vertical height of the island was approximately 0.625 m with a beach face slope of $14°$. The island was tested in a water depth of 0.32 m in the wave basin. The solitary wave propagation around the conical island is shown in Figure 3. The wave approaches the conical island shown in Figure 3(a). The solitary wave splits around the island (Figure 3(b)), wraps around the island (Figure 3(c)) and runups on the back face of the island (Figure 3(d)).



(a) $\boldsymbol{t = 5.68\ s}$ **(b)** $\boldsymbol{t = 8.46\ s}$ **(c)** $\boldsymbol{t = 12.35\ s}$ **(d)** $\boldsymbol{t = 15.75\ s}$

Figure 3. Solitary wave runup around a conical island shown at $t =$ (a) $5.68s$, (b) $8.46s$, (c) $12.35s$ and (d) $15.75s$ for $H/d = 0.181$.

As part of this test, comparison of the model simulations are made with wave gauge measurements in the wave basin and along the slope of the island. The model errors for BP6 test are summarized in Table 5. It was found that the mean errors were such that $\epsilon \leq 10\%$.

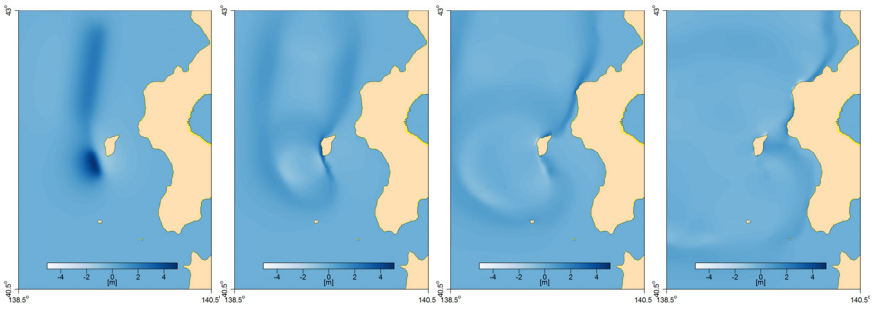
Table 5. NTHMP model errors for BP6 shown at 4 gauge locations for three cases of $H = 0.045$, $H = 0.096$ and $H = 0.181$. The RMS, MAX and runup errors are summarized compared with laboratory measurements.

	CASE A: $\boldsymbol{H = 0.045}$		CASE B: $\boldsymbol{H = 0.096}$		CASE C: $\boldsymbol{H = 0.181}$	
	RMS	MAX	RMS	MAX	RMS	MAX
Gauge #6	5%	1%	6%	6%	8%	11%
Gauge #9	6%	5%	7%	1%	7%	2%
Gauge #16	9%	8%	5%	7%	9%	2%
Gauge #22	7%	12%	5%	11%	5%	13%
Mean	6%	8%				
Runup	11%	6%	8%	2%	8%	3%
Mean (Runup)	11%	6%				

4.2.4 BP9 Field measurements: Okushiri Island tsunami

This test case refers to the July 12, 1993 tsunami event in Japan. The tsunami was generated by $M_w = 7.8$ Hokkaido-Nansei-Oki earthquake that severly inundated coastal areas in the north of Japan. Majority of the damage was concentrated around the Okushiri Island located west of Hokkaido. Up to 31.7 m

of tsunami runup was reported close to Monai village. As part of the validation case, high-resolution bathymetry was provided to perform the numerical test. Initial wave conditions are provided as boundary conditions for the simulations. The tsunami runup around Okushiri Island was measured by the Hokkaido Tsunami Survey Group (1993) which reported up to 32m of tsunami wave runup. The snapshots of the tsunami simulation around Okushiri Island is shown in Figure 4. The initial surface elevation assuming instantaneous surface deformation is provided by the NTHMP test case. The simulation in Figure 4 is shown for the parent grid with resolution of 450m. Nested grids are used to accurately simulate tsunami inundation on Okushiri Island with grids of resolution 450m down to 50m. The accuracy of runup measurements in the simulations are dependant on the grid resolution of the DEM that was used.



(a) $t = 0\ s$ **(b)** $t = 392\ s$ **(c)** $t = 713\ s$ **(d)** $t = 1271\ s$

Figure 4. Tsunami wave surface elevations during the 1993 Japan tsunami event shown at t = (a) $0\ s$, (b) $392\ s$, (c) $713\ s$ and (d) $1271\ s$.

The numerical results are validated with the tide gauge measurements at Iwanai and Esahi. Further comparisons are made with the measured runup values around the Coast of Okushiri Island. The free surface elevation errors with the gauge measurements are within the desired range of expected errors as shown in Table 6. However, the errors in runup measurements are quite large in some locations, beyond the recommended 20%. These runup errors may be explained through the uncertainty in the input bathymetry, topography and initial condition data. Further, data on local friction on land and grid resolution can also affect the simulation results. Hence, it is suggested to quantify the uncertainties in real field simulations for the model in the future.

Table 6. NTHMP model errors for BP9 shown at measured tide gauge locations and run up locations.

Locations	MAX/ERR
IWANAI	11%
ESASHI	15%
RUNUP LOC #1	9%
RUNUP LOC #2	4%
RUNUP LOC #3	15%
RUNUP LOC #4	4%
RUNUP LOC #5	1%
RUNUP LOC #6	2%
RUNUP LOC #7	23%
RUNUP LOC #8	18%
RUNUP LOC #9	1%
RUNUP LOC #10	0%
RUNUP LOC #11	14%
RUNUP LOC #12	2%
RUNUP LOC #13	12%
RUNUP LOC #14	21%
RUNUP LOC #15	8%
RUNUP LOC #16	3%
RUNUP LOC #17	4%
RUNUP LOC #18	11%
RUNUP LOC #19	6%
Mean	9%

5 CONCLUSION

A fast shock capturing finite volume based NSW solver is implemented on GPUs using NVidia's CUDA library. The solver is positivity preserving and preserves the exact "C" state of the NSW solution at rest. The NSW solver is second order in space and time and computationally fast due to the implementation on the GPUs. The solver is designed to simulate tsunami generation due to earthquakes, propagation and inundation on land. The model has been verified against analytical solutions to confirm the salient features of the NSWE and the implementation of the solver on the GPUs. The model has been validated with the benchmark cases specified by NTHMP, NOAA. The benchmarking test cases include analytical solutions, laboratory experiments and field measurements. The RMS, MAX and ERR has been measured in all the test cases to compare with the expected results. The computed errors are below the suggested rates of $\epsilon \leq 5\%$ for analytical solutions, $\leq 10\%$ for laboratory experiments and $\leq 20\%$ for field observations and can be successfully implemented for the purpose of tsunami risk assessment and hazard mitigation.

REFERENCES

Briggs, M.J., Synolakis, C.E., Harkins, G.S. & Green, D. (1995). Laboratory Experiments of Tsunami runup on a Circular Island. *Pure & Applied Geophysics*, 144, 569–593.

Brodtkorb, A.R. (2010). Scientific Computing on Heterogeneous Architectures, *PhD Thesis.* University of Oslo.

Goring D.G. 1978. *Tsunamis-The Propagation of Long Waves onto a Shelf,* WM. Keck Laboratory of Hydraulics and Water Resources, California Institute of Technology, Report No. KH-R-38.

Goto, C., Ogawa, Y., Shuto, N. & Imamura, F. (1997). *Numerical Method of Tsunami Simulation with the Leap-frog Scheme (IUGG/IOC Time Project),* IOC Manual, UNESCO, No. 35.

Gottlieb, S. & Shu, C.-W. (1998). Total Variation Diminishing Runge-Kutta Schemes. *Mathematics of Computation*, 67(221), 73-85.

Hammack, J.L. (1972). *Tsunamis- A model for their Generation and Propagation*, WM. Keck Laboratory of Hydraulics and Water Resources, California Institute of Technology, Report No. KH-R-28.

Hokkaido Tsunami Survey Group (1993). Tsunami Devastates Japanese Coastal Regions. *EOS, Transactions AGU*, 74(37), 417-432.

Kim, D.H., Cho, Y.S. & Yi, Y.K. (2007). Propagation and run-up of near Shore Tsunamis with HLLC approximate Riemann Solver. *Ocean Engineering*, 34(8-9), 1164-1173.

Kuo, F.A., Smith, M.R., Hsieh, C.W., Chou, C.Y. & Wu, J.S. (2011). GPU Acceleration for General Conservation Equations and its Application to Several Engineering Problems. *Computers and Fluids*, 45(1), 147- 154.

Kurganov, A. & Tadmor, E. (2000). New High Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equations. *Journal of Computational Physics*, 160, 241–282.

Leveque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press.

Leveque, R.J., George, D.L. & Berger, M.J. (2011). Tsunami Modelling with Adaptively Refined Finite Volume Methods. *Acta Numerica*, 20, 211-289.

Liang, Q. (2010). Flood Simulation using a Well Balanced Shallow Flow Model. *Journal of Hydraulic Engineering*, 136(9), 669-675.

Liang, Q., Hou, J. & Amouzgar, R. (2015). Simulation of Tsunami Propagation Using Adaptive Cartesian Grids. *Coastal Engineering Journal*, 57(4), 1-30.

NVidia. (2016). *CUDA Toolkit Documentation,* © NVidia.

Okada, Y. (1985). Surface Deformation due to Shear and Tensile faults in a Half Space. *Bulletin of Seismological Society of America*, 75(4), 1135-1154.

Popinet, S. (2011). Quadtree Adaptive Tsunami Modelling. *Ocean Dynamics*, 61(9), 1261-1285.

Rossinelli, D., Hejazialhosseini, B., Spampinato, D.G. & Koumoutsakos, P. (2011). Multicore/Multi GPU Accelerated Simulations of Multiphase Compressible Flows using Wavelet adapted Grids. *SIAM Journal on Scientific Computing*, 33(2), 512-540.

Schive, H.Y., Zhang, U.H. & Chiueh, T. (2012). Directionally Unsplit Hydrodynamic Schemes with Hybrid MPI/OpenMP/GPU Parallelization in AMR. *International Journal of High Performance Computing Applications*, 26(4), 367-377.

Shi, F., Kirby, J.T., Harris, J.C., Geiman, J.D. & Grilli, S.T. (2012). A high Order Adaptive Time Stepping TVD Solver for Boussinesq Modeling of Breaking Waves and Coastal Inundation. *Ocean Modelling*, 43-44, 36-51.

Synolakis, C.E. (1986). The Runup of Long Waves, *Ph.D. Thesis.* California Institute of Technology, Pasadena, California, 91125, 228.

Synolakis, C.E., Bernard, E.N., Titov, V.V., Kânoğlu, U. & González F.I. (2007). *Standards, Criteria and Procedures for NOAA Evaluation of Tsunami Numerical Models*, NOAA Tech. Memo. OAR PMEL-135, NOAA/Pacific Marine Environmental Laboratory, Seattle, WA, 55.

Thacker, W.C. (1981). Some Exact Solutions to the Nonlinear Shallow Water Equations. *Journal of Fluid Mechanics*, 107,499-508.

Titov, V.V. & Synolakis, C.E. (1995). Modeling of Breaking and Nonbreaking Long Wave Evolution and runup using VTCS-2. *Journal of Waterway, Port, Coastal and Ocean Engineering – ASCE*, 121(6), 308-316.

Toro, E.F. (1999). *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer-Verlag.

van Leer, B. (1979). Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method. *Journal of Computational Physics*, 32, 101–136.

Wang, P., Abel, T. & Kaehler, R. (2010). Adaptive Mesh Fluid Simulations on GPU. *New Astronomy*, 15(7), 581-589.

Wang, X. & Liu, P.L-.F. (2006). An Analysis of 2004 Sumatra Earthquake Fault Plane Mechanisms and Indian Ocean Tsunami. *Journal of Hydraulic Research*, 44(2), 147-154.

Yeh, H., Liu, P.L.-F., Briggs, M. & Synolakis, C.E. (1994). Tsunami Catastrophe in Babi Island. *Nature*, 372, 6503–6508.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

COMPUTATIONAL HYDRODYNAMICS SIMULATIONS WITH UPC PARALLEL ARCHITECTURE

TUNG T.VU[(1),] ALVIN W.Z. CHEW[(2),] ADRIAN WING-KEUNG LAW[(3)]

[(1,3)]Environmental Process Modelling Centre, Nanyang Environment and Water Research Institute, Nanyang Technological University, Singapore
[(2,3)]School of Civil and Environmental Engineering, Nanyang Technological University, Singapore
cwklaw@ntu.edu.sg

ABSTRACT

In this study, an approach for computational hydrodynamics (CHD) simulations with high performance computing using the parallel architecture, PGAS-UPC, is developed. UPC hybrids the advantages of MPI's scalability and OpenMP's direct memory access, and could minimize the possible stagnation in computational speedup with an increasing number of CPUs deployed for the simulations. For evaluation, the computational performance of UPC-CFD is compared with that of OpenMP and MPI. Two incompressible CHD cases, namely (a) Blasius boundary layer flow, and (b) Poiseuille's flow, are tested. The accuracy of the proposed UPC-CHD model for both CHD cases is first verified by comparing with the respective analytical predictions. The computational efficiency of UPC-CHD is then demonstrated by comparing with the respective versions of MPI and OpenMP at their basic designs, in a SGI UV-2000 server with 100 CPUs. The positive results from the computational performance analysis have demonstrated that UPC-CFD is a viable alternative for future large-scale CHD simulations.

Keywords: Computational hydrodynamic simulations; PGAS-UPC; MPI; OpenMP; high performance computing.

1 INTRODUCTION

Computational hydrodynamics (CHD) simulations have garnered importance for evaluating and optimizing engineering design and applications. Examples include flow simulations within the tight spacers of membrane modules in attempts to minimize fouling tendency and flow short-circuiting (Y. L. Li et al., 2011; P. Sousa et al., 2014; S. S. Bucs et al., 2014), and flow simulations in porous-media related applications with complex geometrical representation of the granular materials (W. Sobieski and Q. Zhang, 2017; D. Jajcevic et al., 2013). These flow simulations would usually involve large-scale numerical domains which are expensive computationally. For example, a n million two-dimensional (2D) mesh involving the flow continuity and fluid momentum equations would result in an approximate $3n$ million cells information to be managed during each iteration. Thus, high performance parallel computing with multiple processors (CPUs) is now commonly adopted for these simulations to speed up the simulation time.

Message Passing Interface (MPI) and Open Message Interface (OpenMP) are currently the two most common computer architectures for large scale CHD simulations. In MPI, the communication among the CPUs involves either blocking or non-blocking methodologies (S. Jamshed, 2015) The former holds the program execution until the message buffer slots within the computer memory is ready, which might incur significant idle time for a large number of CPUs. The latter proceeds on with the program execution and does not wait for the communication buffer to be completed. Thus, idle time is eliminated, however, data loss might be experienced. The in-built architecture of MPI is postulated to be inefficient for CHD architectures having the large number of CPUs and high levels of memory hierarchy (S. Jamshed, 2015; N. Gourdain et al., 2009). On the other hand, OpenMP employs a shared memory architecture and does not require the message passing in MPI, which renders its ease in applicability. However, its scalability is limited for applications which involves significant parallelization (S. Jamshed, 2015).

This study aims to develop an approach for CHD simulations using the parallel computer architecture of the Partitioned Global Address Space (PGAS). PGAS provides the following benefits: (a) data having direct affiliation with the respective computing thread (CPU) are stored in the private/local memory component which can possibly be accessed at the fastest time possible, (b) all threads/CPUs have direct access to the singular global shared memory, and (c) an ability to achieve a good balance among programmability, performance scalability and portability. The extension of PGAS to the ISO C language, known as Unified Parallel C (UPC), was carried out in this study to demonstrate its computational effectiveness in a shared memory SGI system.

This paper is structured as follows. In section 2, we describe the specific numerical scheme to resolve two 2D incompressible CHD flow cases: (CHD case A) Blasius boundary layer (BL) flow and (CHD case B) Poiseuille's flow. Section 3 describes the developed PGAS-UPC architecture, which is then followed by comparing its computational performance in the shared memory SGI system in section 4. Lastly, section 5 summarizes the important findings from this study.

2 NUMERICAL DISCRETIZATION

The representation of the viscous, unsteady and incompressible flow in full conservative form (E. F. Toro, 2009; M. Kermani and E. Plett, 2001) with the absence of external body forces, can be expressed in the compact form of Equation 1 as follow:

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=\frac{\partial G_{Vx}}{\partial x}+\frac{\partial G_{Vy}}{\partial y} \quad [1]$$

where Q is the conservative temporal term, F and G are the convective flux vectors in the x and y directions respectively, and G_{Vx} and G_{Vy} are the viscous flux vectors in the x and y directions respectively. The exact representation of Q, F, G, G_{Vx} and G_{Vy} (E. F. Toro, 2009; M. Kermani and E. Plett, 2001) are shown in Equation 2.

$$Q=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ E_t\end{bmatrix}, F=\begin{bmatrix}\rho\\ p+\rho u^2\\ \rho uv\\ (E_t+p)u\end{bmatrix}, F=\begin{bmatrix}\rho\\ \rho uv\\ p+\rho v^2\\ (E_t+p)v\end{bmatrix}, G_{Vx}=\mu\begin{bmatrix}0\\ u_x\\ v_x\\ uu_x+vu_x\end{bmatrix}, G_{Vy}=\mu\begin{bmatrix}0\\ u_y\\ v_y\\ uv_x+vv_y\end{bmatrix} \quad [2]$$

where ρ is the density of water (kg/m^3), u is the horizontal velocity (m/s), v is the vertical velocity (m/s), p is the pressure $(kg/m^2.s)$, E_t is the energy term $(kg.m^2/s^2)$, μ is the dynamic viscosity of water $(kg/m.s)$, u_x is the x-derivative of the u velocity, u_y is the y-derivative of the u velocity, v_x is the x-derivative of u and v_y is the y-derivative of v.

Equation 2 can be discretized over the control volume as shown in Figure 1 (E. F. Toro, 2009; M. Kermani and E. Plett, 2001). All others nodes within the numerical domain undergo the same discretization procedure. The resultant discretized form is shown in Equation 3.

$$\frac{\partial \overline{Q}}{\partial t}+\frac{F_E-F_W}{\Delta x}+\frac{G_N-G_S}{\Delta y}=\frac{G_{Vx,E}-G_{Vx,W}}{\Delta x}+\frac{G_{Vy,N}-G_{Vy,S}}{\Delta y} \quad [3]$$

where $\partial\overline{Q}$ is the discretized form of Q which is defined as either $(Q^{n+\frac{1}{2}}-Q^n)$ for the predictor step, or $(Q^{n+1}-Q^n)$ for the corrector step.

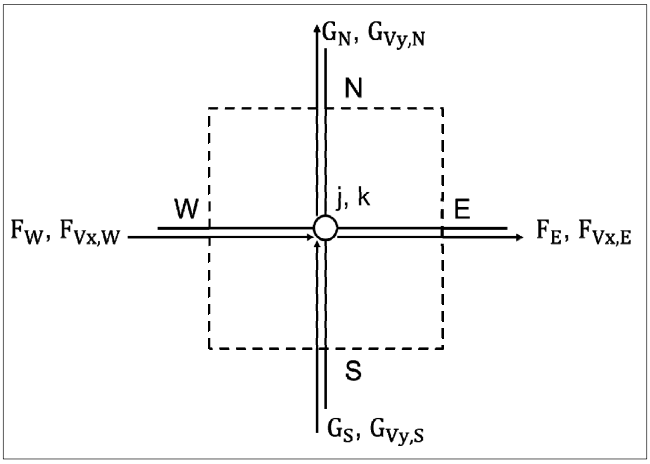


Figure 2.1. Representative control volume of a single node for discretizing the viscous, unsteady and incompressible Navier-Stokes (NS) in Equation 1.

The temporal term in Equation 3 undergoes the 2-step explicit scheme from the Lax-Wendroff family of predictors-correctors. The predictor step computes the flow condition at the half-time step as shown in Equation 4, whereas the corrector step corrects the predicted values by employing a central differencing method with respect to time at the full-time step in Equation 5. For the respective computations at the half- and full time step, the convective and viscous fluxes are first computed before computing for $Q^{n+1/2}$ and Q^{n+1}.

$$Q^{n+1/2}=\left[\left(\frac{G_{Vx,E}-G_{Vx,W}}{\Delta x}\right)^n+\left(\frac{G_{Vy,N}-G_{Vy,S}}{\Delta y}\right)^n-\left(\frac{F_E-F_W}{\Delta x}\right)^n-\left(\frac{G_N-G_S}{\Delta y}\right)^n\right]\frac{\Delta t}{2}+Q^n \quad [4]$$

$$Q^{n+1} = \begin{bmatrix} \left(\frac{G_{Vx,E} - G_{Vx,W}}{\Delta x}\right)^{n+\frac{1}{2}} + \left(\frac{G_{Vy,N} - G_{Vy,S}}{\Delta y}\right)^{n+\frac{1}{2}} - \\ \left(\frac{F_E - F_W}{\Delta x}\right)^{n+\frac{1}{2}} - \left(\frac{G_N - G_S}{\Delta y}\right)^{n+1/2} \end{bmatrix} \Delta t + Q^n \qquad [5]$$

The convective fluxes (F and G) in Equation 3 are computed by the Roe scheme which is represented by Equations 6 and 7 respectively (E. F. Toro, 2009; M. Kermani and E. Plett, 2001). The computed F_E at node (j, k) equates to F_W of node (j+1, k), and the computed G_N at node (j, k) equates to G_S of node (j, k-1). .

$$F_E = 0.5\left[F_E^L + F_E^R\right] - 0.5\sum_{k=1}^{4}\left|\hat{\lambda}_E^k\right| \delta w_E^{(k)} \hat{T}_E^k \qquad [6]$$

$$G_N = 0.5\left[G_N^L + G_N^R\right] - 0.5\sum_{k=1}^{4}\left|\hat{\lambda}_N^k\right| \delta w_N^{(k)} \hat{T}_N^k \qquad [7]$$

where F_E^L and F_E^R are the inner and outer values of F computed at the east face, G_N^L and G_N^R are the inner and outer values of G computed at the north face, $\hat{\lambda}_E^k$ and $\hat{\lambda}_N^k$ are the respective eigenvalues of the Jacobian matrix determined at the Roe's averaged condition, $\hat{T}_E^k$ and $\hat{T}_N^k$ are the respective eigenvectors corresponding to the determined eigenvalues, and both $\delta w_E^{(k)}$and $\delta w_N^{(k)}$ are the respective wave amplitudes.

To compute the eigenvalues, corresponding eigenvectors and wave amplitudes of both F and G, similar computational procedures are undertaken. Here, only the computations for the required parameters of F are shown for brevity. For further details, the reader is referred to references (E. F. Toro, 2009; M. Kermani and E. Plett, 2001). For each of the nodes within the numerical domain, the left (L) and right (R) flow conditions are first computed by Equations 8 and 9 respectively based on the 3rd order upwind-biased algorithm (E. F. Toro, 2009; M. Kermani and E. Plett, 2001).

$$q_E^L = q_{j,k} + 0.25\left[\left(\frac{2}{3}\right)\left(q_{j,k} - q_{j-1,k}\right) + \left(\frac{4}{3}\right)\left(q_{j+1,k} - q_{j,k}\right)\right] \qquad [8]$$

$$q_E^R = q_{j+1,k} - 0.25\left[\left(\frac{2}{3}\right)\left(q_{j+2,k} - q_{j+1,k}\right) + \left(\frac{4}{3}\right)\left(q_{j+1,k} - q_{j,k}\right)\right] \qquad [9]$$

where q represents the primitive variables (u, v).

Subsequently, the Roe's averaged conditions are determined in Equations 10 to 13 (E. F. Toro, 2009; M. Kermani and E. Plett, 2001). Since the fluid is incompressible, no averaged condition is required for the density (ρ) parameter.

$$\widehat{u_E} = \frac{\sqrt{\rho_W} u_E^R + \sqrt{\rho_W} u_E^L}{2\sqrt{\rho_W}} \qquad [10]$$

$$\widehat{v_E} = \frac{\sqrt{\rho_W} v_E^R + \sqrt{\rho_W} v_E^L}{2\sqrt{\rho_W}} \qquad [11]$$

$$\widehat{H_E} = \frac{\sqrt{\rho_W} H_E^R + \sqrt{\rho_W} H_E^L}{2\sqrt{\rho_W}} \qquad [12]$$

$$\hat{c}_E = \sqrt{(\gamma - 1)(\widehat{H_E} - 0.5(\widehat{u_E}^2 + \widehat{v_E}^2)} \qquad [13]$$

where $\widehat{u_E}$, $\widehat{v_E}$, $\widehat{H_E}$ and $\hat{c}_E$ are the Roe averaged conditions and γ =1.33 for the ratio of specific heats applied to water

With the Roe's averaged conditions, the values of $\hat{\lambda}_E^k$, $\hat{T}_E^k$ and $\delta w_E^{(k)}$ are then computed in the following (E. F. Toro, 2009; M. Kermani and E. Plett, 2001).

$$\begin{bmatrix} \hat{\lambda}_E^1 \\ \hat{\lambda}_E^2 \\ \hat{\lambda}_E^3 \\ \hat{\lambda}_E^4 \end{bmatrix} = \begin{bmatrix} \widehat{u_{\perp E}} - \hat{c_E} \\ \widehat{u_{\perp E}} \\ \widehat{u_{\perp E}} \\ \widehat{u_{\perp E}} + \hat{c_E} \end{bmatrix} \quad [14]$$

$$\begin{bmatrix} \delta w_E^{(1)} \\ \delta w_E^{(2)} \\ \delta w_E^{(3)} \\ \delta w_E^{(4)} \end{bmatrix} = \begin{bmatrix} \frac{\delta_{PE} - \rho_W \delta_{u\perp E} \hat{c_E}}{2\hat{c_E}^2} \\ \rho_W \delta u_{||E} \\ -\frac{\delta_{PE} - \hat{c_E}^2 \delta_{\rho E}}{\hat{c_E}^2} \\ \frac{\delta_{PE} + \rho_W \delta_{u\perp E} \hat{c_E}}{2\hat{c_E}^2} \end{bmatrix} \quad [15]$$

$$T_E{}^1 \begin{bmatrix} 1 \\ \widehat{u_E} - \hat{c_E}\cos\theta_E \\ \widehat{v_E} - \hat{c_E}\sin\theta_E \\ \widehat{H_E} - \widehat{u_{\perp E}}\hat{c_E} \end{bmatrix}, T_E{}^2 = \begin{bmatrix} 0 \\ -\sin\theta_E \\ \cos\theta_E \\ u_{||E} \end{bmatrix}, T_E{}^3 = \begin{bmatrix} 1 \\ \widehat{u_E} \\ \widehat{v_E} \\ \frac{\widehat{u_E}^2 + \widehat{v_E}^2}{2} \end{bmatrix}, T_E{}^4 \begin{bmatrix} 1 \\ \widehat{u_E} + \hat{c_E}\cos\theta_E \\ \widehat{v_E} + \hat{c_E}\sin\theta_E \\ \widehat{H_E} + \widehat{u_{\perp E}}\hat{c_E} \end{bmatrix} \quad [16]$$

where $\delta_{PE} = p_{j+1,k} - p_{j,k}$, $\delta u_{||E} = u_{||j+1,k} - u_{||j,k}$, $\delta_{\rho E} = \rho_{j+1,k} - \rho_{j,k}$, $\delta_{u\perp E} = u_{\perp j,k+1} - u_{\perp j,k}$ and $u_{||E}{}^2 = \overrightarrow{V_E}.\overrightarrow{V_E} - u_{\perp E}^2$.

Lastly, the viscous terms (G_{Vx} and G_{Vy}) in Equation 3 can be discretized using the 2nd order central differencing scheme (E. F. Toro, 2009; M. Kermani and E. Plett, 2001) as shown.

$$\frac{G_{Vx,E} - G_{Vx,W}}{\Delta x} = \frac{G_{(i+1,j)} - 2G_{(i,j)} + G_{(i-1,j)}}{(\Delta x)^2} \quad [17]$$

$$\frac{G_{Vy,N} - G_{Vy,S}}{\Delta y} = \frac{G_{(i,j+1)} - 2G_{(i,j)} + G_{(i,j-1)}}{(\Delta y)^2} \quad [18]$$

The implemented numerical scheme as described above was examined for the two incompressible CHD flow cases: CHD case A - Blasius boundary layer flow and CHD case B - Poiseuille's flow. The respective boundary and initial conditions adopted are represented in Figures 2.2 and 2.3. The water density and temperature of all numerical domains were kept at $1000 kg/m^3$ and 293.15K. To validate the accuracy of the implemented numerical scheme, a comparison between the attained numerical results and analytical values was carried out for both cases which will be discussed in section 4 of this paper.

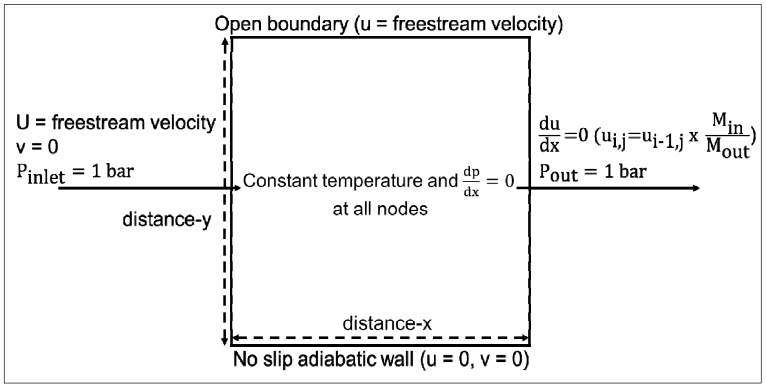


Figure 2.2. Adopted numerical domain for CHD case A (Blasius boundary layer flow).

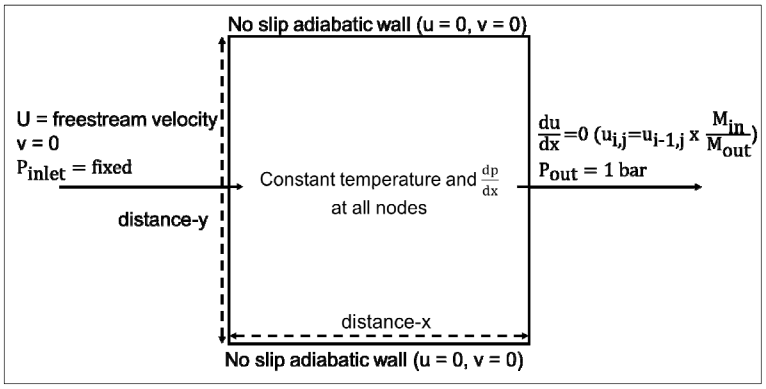


Figure 2.3. Adopted numerical domain for CHD case B (Poiseuille's flow).

3 UPC IMPLEMENTATION OF CHD

The UPC-CHD model was developed by the described numerical scheme in section 2. Firstly, the time-consuming functions and types of data dependences within the model were identified. The data dependences and model workflow dictated the forms of algorithms employed for data divisions and storage. The work sharing function of PGAS was then implemented to parallelize the developed UPC-CHD model.

The functionality of the proposed UPC-CHD model is summarized as follows. The flux predictor at the (n+1/2) time level is first computed, which is then followed by the flux correction at the (n+1) time level by repeating the prediction process with the fluxes from the half-time step as the input data. Both the flux predictor and flux corrector are within the nested loop for which the complexity of the algorithm is identified as O(N^2) where N is the number of nodes undirectionally. The developed parallel algorithm in UPC-CHD aims to minimize the total run time of the predictor and corrector fluxes within each cell, as both occupy the largest portion of the total computational time. Within the model, the original nested loop is first divided into multi sub-loops to prevent data confliction issue. Several functions, unique to the PGAS-UPC architecture, were employed within the UPC-CHD model as described in Table 3.1.

Table 3.1. Description of functions unique to PGAS-UPC architecture.

UPC function/identifier	Functionality
upc_barrier	To synchronize all threads (CPUs) before going to the next function after every nested loop
upc_forall	To distribute the computations within the nested loops in UPC-CHD model
THREADS	To determine total number of threads at run time
MYTHREAD	To identify each thread

All threads with MYTHREAD from 0 to THREADS-1 run through the identical code (i.e. the nested loop) except for the fluxes computation at the first- and last-row of each sub-domain respectively. Each thread is designed to calculate the fluxes on the different sub-domains. To demonstrate the effectiveness of the PGAS-UPC architecture, the accuracy and performance of the developed UPC-CHD model for the two incompressible CHD flow test cases are examined.

4 MODEL VERIFICATION AND PERFORMANCE EVALUATION

Validation of the numerical scheme was first carried out by comparing the obtained numerical results with the respective analytical predictions for both CHD test cases. The physical dimensions and initial hydrodynamic conditions of the deployed numerical domains for CHD case A and B are described in Table 4.1. For CHD case A, the numerical results were compared with the analytical predictions of (White, 1991) whereas Equation 19 was employed to validate the numerical results for CHD case B. As observed from Figures 4.1 and 4.2, there is a good agreement between the two for both cases.

$$\frac{u_x}{U} = \frac{(y^2 - h^2)}{2 * U * \mu}\left(\frac{\partial p}{\partial x}\right) \qquad [19]$$

where u_x is the horizontal velocity value obtained (m/s), U is the freestream velocity (m/s), y is the respective y-distance (m), h is the total vertical height of the domain (m), μ is the dynamic viscosity of the fluid in the domain $(kg/m.s)$, x is the total horizontal distance of the domain (m) and p is the pressure $(kg/m.s^2)$.

Table 4.1. Dimensions and initial hydrodynamic conditions of numerical domains for CHD cases A and B.

Parameter	Case A	Case B
Distance-x & Distance-y (m)	0.3 & 0.02	0.5 & 0.00016
No. of nodes	65 x 65	300 x 45
Freestream velocity, U $(\mathrm{m/s})$	10	26
$\frac{\mathrm{dp}}{\mathrm{dx}}\ (\frac{\mathrm{Pa}}{\mathrm{m}})$	0	1.22 x 10^7
delta-t (s)	10^{-6}	10^{-6}
Total runtime (s)	0.1	0.1
Temperature (K)	293.15	293.15
Kinematic viscosity (m^2/s)	10^{-6}	10^{-6}

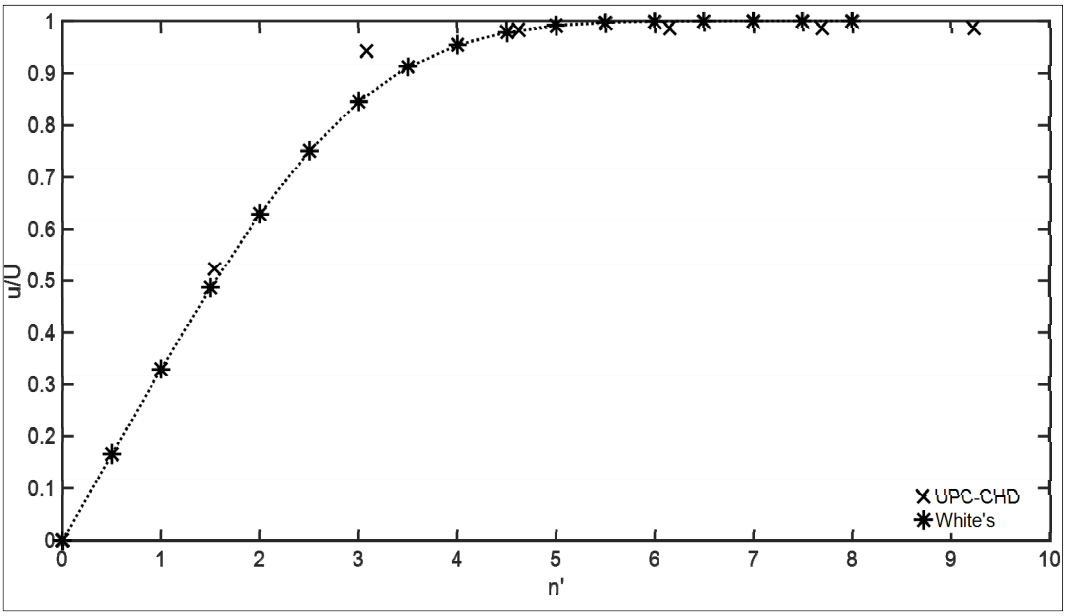


Figure 4.1. Comparison between numerical values from UPC-CHD model with White's analytical predictions for CHD case A (Blasius boundary layer flow).

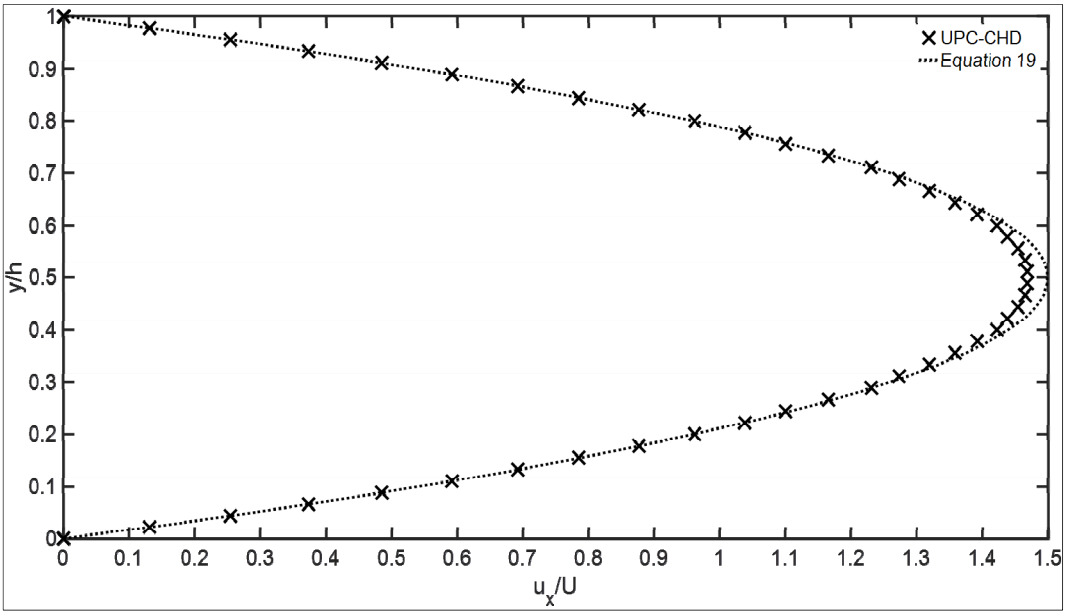


Figure 4.2. Comparison between numerical values from UPC-CHD model with Equation 19 for CHD case B (Poiseuille's flow).

Comparison of the computational performance of UPC against MPI and OpenMP implementation at their basic designs in the SGI UV-2000 server was carried out for both CHD flow cases by computing the respective speedup. The UPC-CHD model was operated for a total duration of 100s with a regular Cartesian

grid of 4 million cells for case A and 100 million cells for case B. The respective speedup attained, as computed in Equation 20, for the varying number of CPUs deployed in UPC, OpenMP and MPI were summarized in Tables 4.2 and 4.3 for CHD case A and CHD case B respectively. The following observations were made from the attained speedup results.

- UPC and OpenMP exploited the advantages of data locality by reading and writing directly to the internal memory section without incurring any delays with the number of CPUs below 16;
- MPI was still effective in its speedup with the number of CPUs below 16;
- UPC was still effective in its speedup with the number of CPUs beyond 100 with no signs of deterioration;
- UPC and MPI performed more effectively than OpenMP with the number of CPUs beyond 16;
- OpenMP showed signs of deterioration with the number of CPUs beyond 16 which could be attributed to its over-accessing of the shared-memory section in the SGI UV-2000 server;
- Speedup with MPI began to be less significant with the number of CPUs beyond 64 which could be attributed to the outweighing of the actual computational time by the total message processing time on each CPU;
- The preceding could further be ascribed to the overloading of synchronized messages to be processed in the system at each time step which exceeded the total data processing time.

$$\text{Speedup} = \frac{1}{\left[\left(\frac{\text{P}}{\text{N}}\right) + \text{S}\right]} \qquad [20]$$

where P is the parallel fraction, S is the serial fraction and N is the number of CPUs available.

Table 4.2. Comparison of speedup among UPC, MPI and OpenMP for varying number of CPUs for CHD case A (Blasius boundary layer flow).

Cores	UPC speedup	MPI speedup	OpenMP speedup
1	1.0	1.0	1.0
2	2.0	2.0	1.9
4	4.0	4.0	3.2
8	7.9	7.9	6.3
16	15.7	15.8	6.9
32	28.6	26.7	2.4
64	52.9	45.1	2.8
100	80.9	46.9	2.3

Table 4.3. Comparison of speedup among UPC, MPI and OpenMP for varying number of CPUs for CHD case B (Poiseuille's flow).

Cores	UPC speedup	MPI speedup	OpenMP speedup
1	1.0	1.0	1.0
2	2.0	2.0	1.9
4	4.0	4.0	3.5
8	7.8	7.9	6.3
16	15.8	15.8	7.2
32	30.1	26.5	2.8
64	54.6	46.7	3.8
100	82.9	47.2	2.6

5 CONCLUSIONS

In this study, we developed a model for computational hydrodynamic simulations using the PGAS-UPC computing architecture for parallelization. Two flow cases, namely the (a) Blasius boundary layer flow and (b) Poiseuille's flow, were simulated with the developed model. The accuracy of the proposed UPC-CHD model was first verified by comparing the simulation results with the respective analytical solutions for each case. The model computational speedup was then evaluated by comparing the total run time with that of MPI and OpenMP at their basic designs using a SGI UV-2000 server. The results showed that UPC performed better than MPI and OpenMP with a near linear speedup until the maximum 100 CPUs available, which underscored UPC's capability to expedite the computational process by exploiting the data locality during parallelism. We are hopeful that the proposed CHD approach with PGAS-UPC computer architecture can serve as an effective alternative for large scale simulations in the future.

ACKNOWLEDGEMENTS

This research study is funded by the internal core funding from the Nanyang Environment and Water Research Institute (NEWRI), Nanyang Technological University (NTU), Singapore. The first author is grateful

to NTU's Interdisciplinary Graduate School (IGS) for the 4-year PhD scholarship awarded for his study. The second author is grateful to NTU for the 4-year Nanyang President Graduate Scholarship (NPGS) for his PhD study.

REFERENCES

Bucs, S.S., Radu, A.I., Lavric, V., Vrouwenvelder, J.S. & Picioreanu, C. (2014). Effect of Different Commercial Feed Spacers on Biofouling of Reverse Osmosis Membrane Systems: A Numerical Study. *Desalination,* 343, 26-37.

Gourdain, N., Gicquel, L., Montagnac, M., Vermorel, O., Gazaix, M. & Staffelbach, G. (2009). High Performance Parallel Computing of Flows in Complex Geometries: I. Methods. *Computational Science & Discovery,* 2, 1-27.

Jajcevic, D., Siegmann, E., Radeke, C. & Khinast, J.G. (2013). Large Scale CFD–DEM Simulations of Fluidized Granular Systems. *Chemical Engineering Science,* 98, 298-310.

Jamshed, S. (2015). *Chapter 3 - The Way the HPC Works in CFD, Using HPC for Computational Fluid Dynamics*. Academic Press, 41-79.

Kermani, M. & Plett, E. (2001). Roe Scheme in Generalized Coordinates. I – Formulations. *39th Aerospace Sciences Meeting and Exhibit*, 86.

Kermani, M. & Plett, E. (2001). Roe Scheme in Generalized Coordinates. II - Application to Inviscid and Viscous Flows. *39th Aerospace Sciences Meeting and Exhibit*, 87.

Li, Y.-L. Lin, P.-J. & Tung, K.-L. (2011). CFD Analysis of Fluid Flow through a Spacer Filled Disk Type Membrane Module. *Desalination,* 283, 140-147.

Sobieski, W. & Zhang, Q. (2017). Multi Scale Modeling of Flow Resistance in Granular Porous Media. *Mathematics and Computers in Simulation,* 132,159-171.

Sousa, P., Soares, A., Monteiro, E. & Rouboa, A. (2014). A CFD Study of the Hydrodynamics in a Desalination Membrane filled with Spacers. *Desalination,* 349, 22-30.

Toro, E.F. (2009). *The Riemann Solver of Roe, Riemann Solvers and Numerical Methods for Fluid Dynamics.* A Practical Introduction, Springer Berlin Heidelberg, Chapter 11, pp. 345-376.

White, F.M. (1991). *Viscous Fluid Flow.* McGraw-Hill, Chapter 7, 457-528.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

SPH MODELING OF OSCILLATING CHARACTERISTICS OF HYDRAULIC JUMPS

DIANA DE PADOVA[1], MICHELE MOSSA[2] & STEFANO SIBILLA[2]

[1,2]Department of Civil, Environmental, Land, Building Engineering and Chemistry (DICATECh), Polytechnic University of Bari, Bari, Italy
diana.depadova@poliba.it; michele.mossa@poliba.it
[3] Department of Civil Engineering and Architecture, Pavia, Italy
stefano.sibilla@unipv.it

ABSTRACT

The oscillating characteristics and cyclic mechanisms in hydraulic jump are investigated and reproduced using a weakly-compressible XSPH scheme which includes both an algebraic mixing-length model and a two-equation turbulence model to represent turbulent stresses. The numerical model is applied to analyze oscillations of different jump types based on the laboratory experiments by Mossa (1999). Experimental investigations were carried out in the hydraulic laboratory of the Mediterranean Agronomic Institute of Valenzano (Bari) in a channel of 7.72 m long with a rectangular cross section of a width of 0.30 m and a height of 0.40 m. The comparison between SPH and experimental results shows an influence of different turbulence models, such as mixing length or Standard k-ε, on the amplitude spectrum and peak amplitude of the time-dependent surface elevation upstream and downstream of the jump. Numerical results show satisfactory agreement with measurements and most of the peculiar features of the flow are qualitatively and quantitatively reproduced. By analyzing a single cycle of the oscillating phenomena of a simulated jump (periodic formation of different jump types) it is possible to indicate the correlation between the vortex structures of the roller. Vortices are characterized by a clockwise or anti-clockwise rotation depending on which type of jump is present, particularly for configurations with alternations of wave and B jumps.

Keywords: Hydraulic jumps; smoothed particle hydrodynamics models; oscillating characteristics.

1 INTRODUCTION

Some researchers pointed out the existence of oscillating phenomena, and particularly of cyclic variation of jump types over long-duration experiments, under some flow conditions (e.g. Nebbia, 1942; Hager and Bretz 1986; Ohtsu and Yasuda, 1991; Abdel Ghafar et al., 1995; Mossa and Tolve, 1998; Mossa, 1999; Wang and Chanson, 2015; Wang et al., 2014).

These oscillating characteristics can be: a) changes of the different types of hydraulic jumps (variation from one type to another); b) horizontal movements of the jump toe (Long et al., 1991); c) variations of the velocity components and pressure in the region close to the jump roller; d) process of formation, development and coalescence of the large-scale flow structures.

All standard experiments by Mossa (1999), carried out in a channel with a non-erodible, moulded bed profile, took into account previous studies by Abdel Ghafar et al. (1995) and Nebbia (1942) to investigate the oscillating characteristics of hydraulic jumps.

Numerical modeling of the hydraulic jump has always proved to be difficult for traditional Eulerian techniques, owing to its oscillating characteristics which lead to the propagation of short breaking waves which prevented from an accurate capturing of the free-surface (Mossa, 1999). In this view, mesh less Lagrangian techniques appear to be more suitable to represent the complex and highly-unsteady free-surface pattern which characterizes a hydraulic jump. Among the mesh less techniques, one of the most versatile techniques is Smoothed Particle Hydrodynamics (SPH) which is developed since 1977 to simulate astrophysical problems (Liu and Liu, 2006; Lucy, 1977). It is presently one of the most popular mesh-free methods for numerical simulations of complex, free-surface flows (e.g. Dalrymple and Rogers, 2006; Capone et al., 2010; Manenti et al., 2012; De Padova et al., 2013; De Padova et al., 2014; Jonsson et al., 2016; De Padova et al., 2016).

The method is fully Lagrangian and obtains, through a discrete kernel approximation. The solution of the equations of motion for each of the fluid particles in which the flowing volume is discretized.
The free surface requires, therefore, no special approach, such as in the case of the Volume-of-Fluid method or of Lagrangian surface-tracking techniques. Furthermore, the method can easily treat rotational flows with vorticity and turbulence. The basics of the methodology are described in textbooks and review papers (e.g.: Violeau, 2012; Violeau and Rogers, 2016).

The SPH turbulence models used for engineering applications have been based on RANS (Reynolds-averaged Navier–Stokes) approaches with first-order closure (eddy viscosity models), using mixing length (Violeau et al., 2002) or k-models (Violeau, 2004). Both turbulence model were successfully applied to SPH

analyses of rotational flows, such as wave overtopping (Shao, 2006), or spilling breakers (De Padova et al., 2016).

The purpose of this paper is to use a weakly-compressible XSPH scheme, together with either an algebraic mixing-length turbulence model or a two-equation turbulence model, to study oscillating characteristics and cyclic mechanisms in different jump types based on the laboratory experiments by Mossa (1999).

2 EXPERIMENTAL SET UP

Experimental investigations were carried out in the hydraulic laboratory of the Mediterranean Agronomic Institute (hereafter referred to as IAM) in Valenzano (Bari). It measured at 7.72 m long and 0.3 m wide rectangular channel with sidewall height of 0.40 m (Fig.1). The walls and bottoms of both channels were made of Plexiglas.

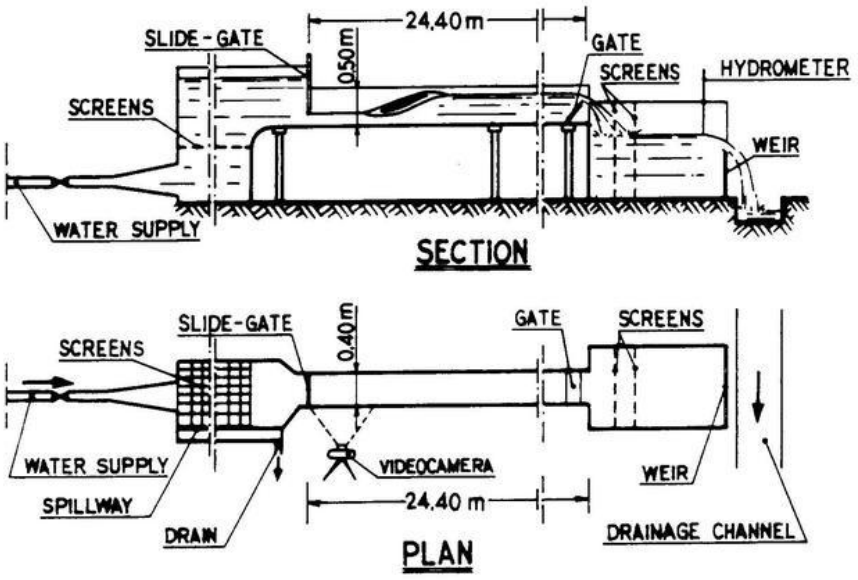


Figure 1. Sketch of the channel at the hydraulic laboratory of the Mediterranean Agronomic Institute (IAM) of Valenzano (Bari).

Discharges were measured with a triangular sharp-crested weir. Measurements of the upstream and downstream water depths were carried out with electric hydrometers, type point gauge supplied with electronic integrators which allowed the estimate of the time-averaged flow depth. The hydrometers were supplied with verniers allowing measurement accuracy of ±0.1 mm.

Water discharge and hydrodynamic conditions were regulated by two gates placed at the upstream and downstream ends of the channels. The desired tailwater level was maintained by using the downstream gate. Figure 2 shows the shape of the wooden bed that had been used for the configuration carried out in the channel. The profile was equal to one of those measured by Abdel Ghafar et al. (1995) in the central longitudinal section of the channel, when equilibrium scour profiles were reached.

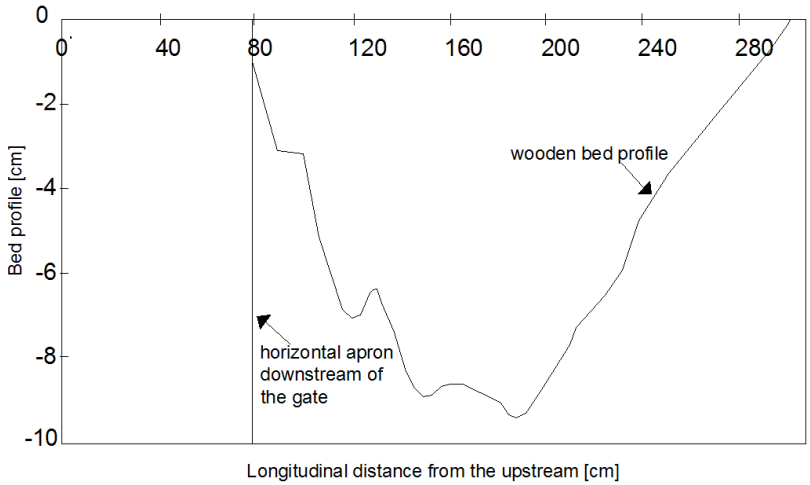


Figure 2. Shape of the wooden bed used for configurations carried out in the channel.

Table 1 lists the main experimental parameters of the investigated hydraulic jumps, where Q is the discharge; y_1 is the water depth in section 1, located at the downstream end of the horizontal apron, i.e. where the cavity begins; y_t is the water depth downstream of jump where a second horizontal apron is located; $F_1=V_1/(gy_1)^{0.5}$ is the Froude number in section 1 and $Re=V_1y_1/v$ is the Reynolds number, with V_1 indicating the mean velocity of supercritical flow in section 1, v the water kinematic viscosity and g the acceleration of gravity. During all experiments, water temperature was measured by a thermometer with an accuracy of 10^{-1} °C.

Table 1. Experimental parameters of the analyzed undular hydraulic jump

Q [l/s]	y_1 [cm]	y_t [cm]	y_1/y_t	F_1	Re	Jump type
9.60	2.50	7.70	3.08	2.58	31894	B and wave

3 SPH NUMERICAL METHOD

SPH simulations were obtained by a standard weakly compressible SPH (WCSPH) scheme (Monaghan, 1992), where a reduced compressibility modulus of 10^6 Pa was assumed for water. A pressure smoothing procedure was applied at every time step to show the difference between local and hydrostatic pressure in order to reduce the numerical noise in pressure evaluation.

The SPH code was developed at the Fluid Mechanics Laboratory of the Department of Civil Engineering and Architecture of the University of Pavia, and had been used extensively to simulate several kinds of hydraulic phenomena, such as scouring downstream of sluice gates (Sibilla, 2008), hydraulic jumps (De Padova et al., 2013) or spilling breakers (De Padova et al., 2016). SPH particle approximations of flow quantities and their derivatives were obtained by the use of the Wendland-C4 kernel function W (Wendland, 1995), renormalized to improve the consistency and accuracy of the method (Sibilla, 2015).

An algebraic mixing-length model and a two-equation model were used to represent turbulent stresses.

The mixing-length model is based on the introduction of a mixing-length $l_m = f_i \min(\kappa z, l_{max})$, where κ =0.41 is the Von Kármán constant, z is the distance from the wall, l_{max} is a cutoff maximum value and

$$f_i = \min\left[1, \left|\sum_j \frac{m_j}{\rho_j}\nabla W_{ij}\right|^{-3}\right] \qquad [1]$$

is a damping function which is less than unity; only near the free surface. The effect of f_i is lower l_m near the free surface, where a non-physical growth of the eddy viscosity may lead to numerical instabilities (De Padova et al, 2009; 2010).

The two-equation model is a SPH version of the Standard k-ε turbulence model (Launder and Spalding, 1974):

$$\begin{aligned}
\frac{Dk_i}{Dt} &= P_{k_i} + \frac{1}{\sigma_k}\sum_j m_j \frac{\nu_{T_i}+\nu_{T_j}}{\rho_i+\rho_j}\frac{k_i-k_j}{r_{ij}^2+0.01h^2}\vec{r}_{ij}\cdot\vec{\nabla}W_{ij} - \varepsilon_i \\
\frac{D\varepsilon_i}{Dt} &= \frac{1}{\sigma_\varepsilon}\sum_j m_j \frac{\nu_{T_i}+\nu_{T_j}}{\rho_i+\rho_j}\frac{\varepsilon_i-\varepsilon_j}{r_{ij}^2+0.01h^2}\vec{r}_{ij}\cdot\vec{\nabla}W_{ij} + \\
&+ C_{\varepsilon_1}\frac{\varepsilon_i}{k_i}P_{k_i} + C_{\varepsilon_2}\frac{\varepsilon_i}{k_i}\sum_j \frac{m_j}{\rho_j}\varepsilon_j W_{ij}
\end{aligned} \qquad [2]$$

where P_k is the production of turbulent kinetic energy depending on the local rate of deformation, ν_T is the eddy viscosity and σ_k = 1, σ_ε = 1.3, $C_{\varepsilon 1}$ = 1.44 and C_{e2} = 1.92 are model constants whose values are those proposed for the standard k-ε formulation.

According to the standard XSPH approach, at each time steps a new particle velocity $\vec{v}^T$ was obtained by explicit integration of the momentum equation (2) while a smoothed value $\vec{v}^S$ of velocity is obtained by:

$$\vec{v}^S(\vec{x}_i) = (1-\varphi)\vec{v}^T(\vec{x}_i) + \varphi\frac{\sum_j \frac{m_j}{\rho_j}\vec{v}^T(\vec{x}_j)W_{ij}}{\sum_j \frac{m_j}{\rho_j}W_{ij}} \qquad [3]$$

and used for particle movement. The smoothing parameter φ controls the intensity of velocity filtering. According to a sensitivity analysis, the SPH simulation of the experimental test on hydraulic jump was performed by adopting a velocity smoothing coefficient in the XSPH scheme $\varphi = 0.01$.

4 NUMERICAL TEST AND RESULTS

The 2D simulation of the hydraulic jump reported in Table 1 was performed in a physical domain consisting a rectangle of 3.3m long and 0.4 m high, shorter than the real channel in the test facility. The shorter domain was chosen in order to reduce the computational cost without influencing the quality of the numerical solution. The current incompressible SPH computations were executed up to t=50s from the beginning and it costs about 33 hours of CPU time using a personal computer (CPU 2.70 GHz and RAM 4.0 GB PC).

The accuracy of particle methods was influenced by the initial particle spacing Σ, as well as by the smoothing length, η. Both parameters must be taken into account when attempting to improve the resolution of the numerical simulation. It has been shown that the efficiency of the SPH kernel function depends also on the choice of the η/Σ ratio (Dehnen and Aly, 2012) and that a value $\eta/\Sigma \geq 1.2$ should be preferred (De Padova et al., 2008).

The SPH simulation has been performed with initial particle spacing Σ equal to 0.008 m and constant value of $\eta/\Sigma = 1.5$. A sensitivity analysis was performed on turbulence modelling influence. According to the sensitivity analysis shown by De Padova et al. (2013) the test T1 was performed by adopting a mixing length turbulence model with $l_{max} = 0.5\ h_2$. It was chosen to adopt two-equation model (2) for test T2. Some of the results are summarized here. Table 2 lists the principal characteristics of SPH simulations.

Table 2. Principal characteristics of SPH simulations

Test	Turbulence model	η/Σ	Np
T1	mixing-length model	1.5	6000
T2	k-ε turbulence model	1.5	6000

The results of the SPH simulations were carried out to analyze oscillations of different jump types. Their analysis was useful for the study of the phenomena, which appeared cyclic as well for the formation with evolution of the jump vortices. Both the mixing length model and the *k-ε* model yield similar results and were able to predict the oscillating characteristics and cyclic mechanisms in hydraulic jumps.

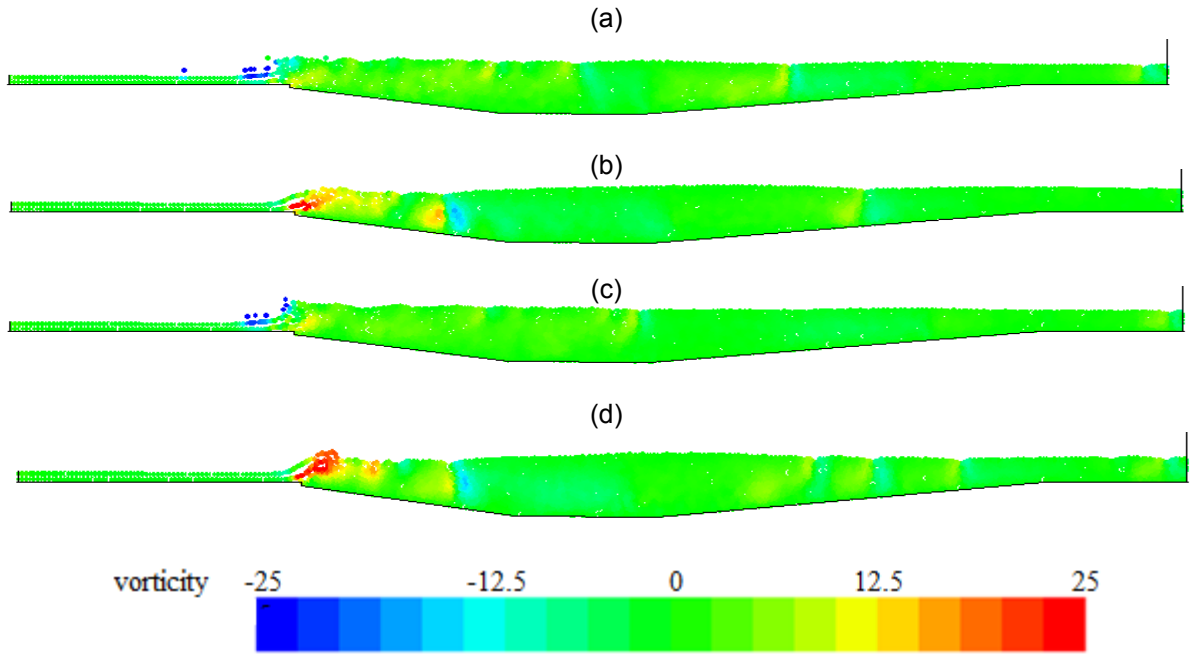


Figure 3. Instantaneous SPH vorticity field in the SPH simulation of Test T1 a) t=16s; (b) t=20s; (c) t=24s; (d) t=28s.

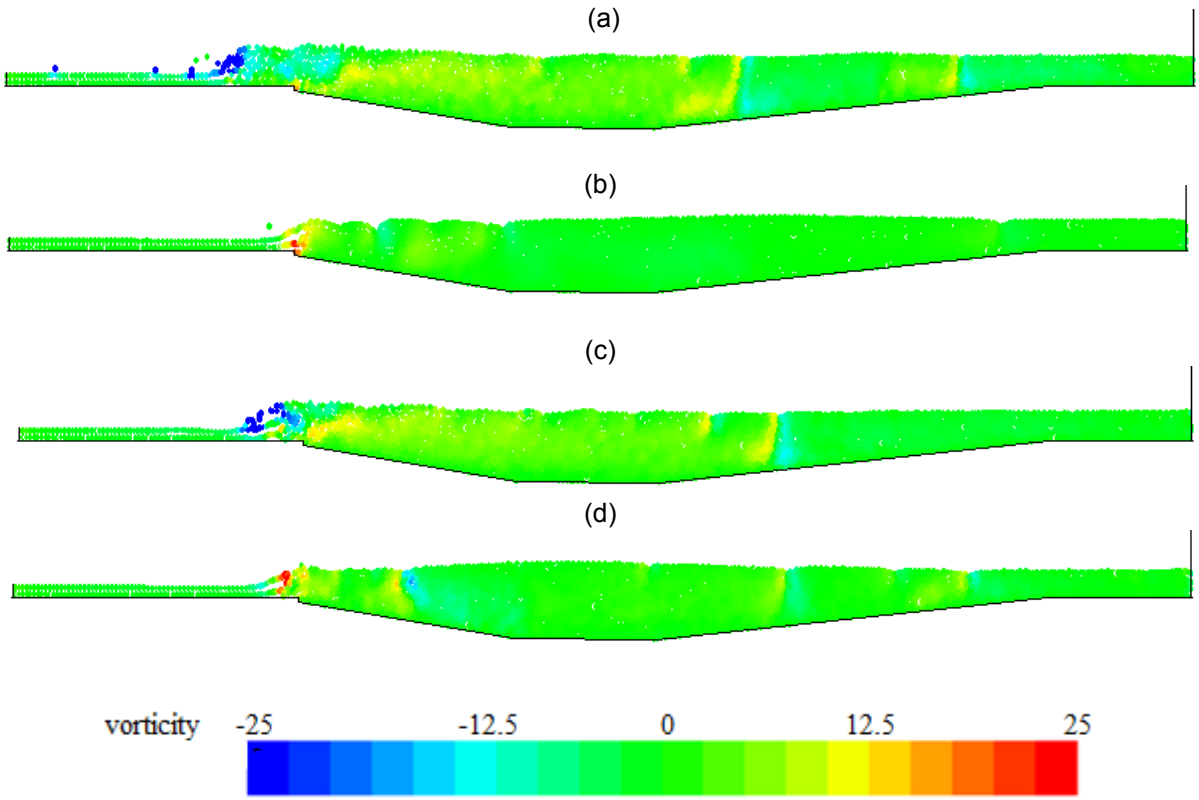


Figure 4. Instantaneous SPH vorticity field in the SPH simulation of Test T2 a) t=15s; (b) t=21s; (c) t=27s; (d) t=32s.

The instantaneous vorticity field (Figs. 3a ÷ 4d) of the configuration with oscillations of the B and wave jump clearly indicates the transition phase between the jump types with both the turbulence models (T1 and T2). Vortices were characterized by a clockwise or anti-clockwise rotation depending on which type of jump was present. In particular, vortices were characterized by a clockwise rotation in the presence of the wave jump (Figs. 3b - 3d and Figs. 4b - 4d) and an anti-clockwise for the B jump (Figs. 3a - 3c and Figs. 4a - 4c) respectively.

Figures 5a - 6b show the amplitude spectra of the time series of the surface elevations, upstream and downstream of the jump for tests T1 and T2. From the analysis of the previously mentioned figures it is possible to observe the existence of a peak in each spectrum, as shown by Mossa (1999). Consequently, as suggested by Mossa (1999), oscillating characteristic of wave and B jumps is quasi-periodic. Furthermore, as the spectra in Figures. 5a and 6a show a peak at a frequency similar to the one of the spectra in Figuress. 5b and 6b. It is possible to conclude that fluctuations of the surface profile downstream of the jump depended on alternations of jump types. The periodic formation of different jump types produces small waves downstream of the hydraulic jump.

Although both turbulence models yielded similar results, the detailed comparison of the amplitude spectra of the time series of the surface elevations upstream and downstream of the jump for tests T1 and T2, shows that the mixing-length results are closer to the experimental data than the k-ε ones.

Actually, figures 5a - 5b show a peak in each spectrum at a frequency slightly higher than 0.1 Hz, as shown by Mossa (1999), while figures 6a ÷ 6b show a peak at a frequency lower than 0.1 Hz.

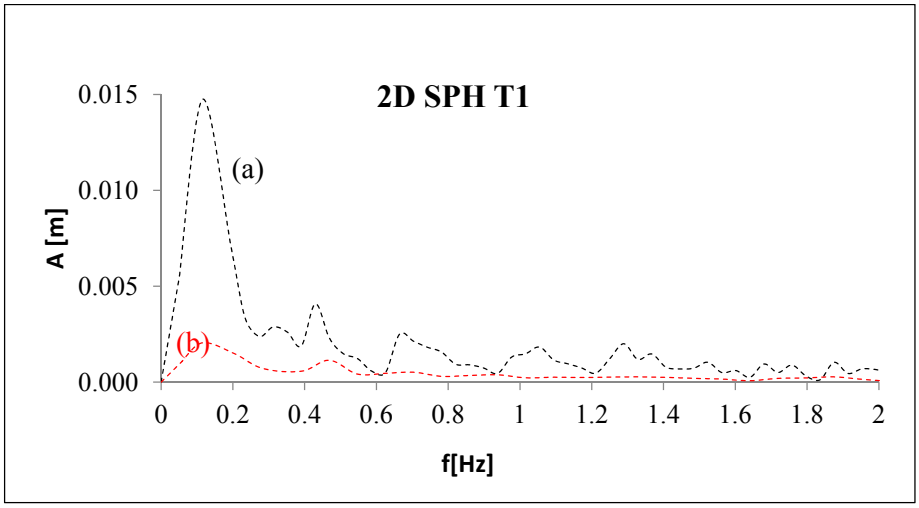


Figure 5. Test T1: Amplitude spectrum of the time series of the surface elevations (a) upstream and (b) downstream of the jump

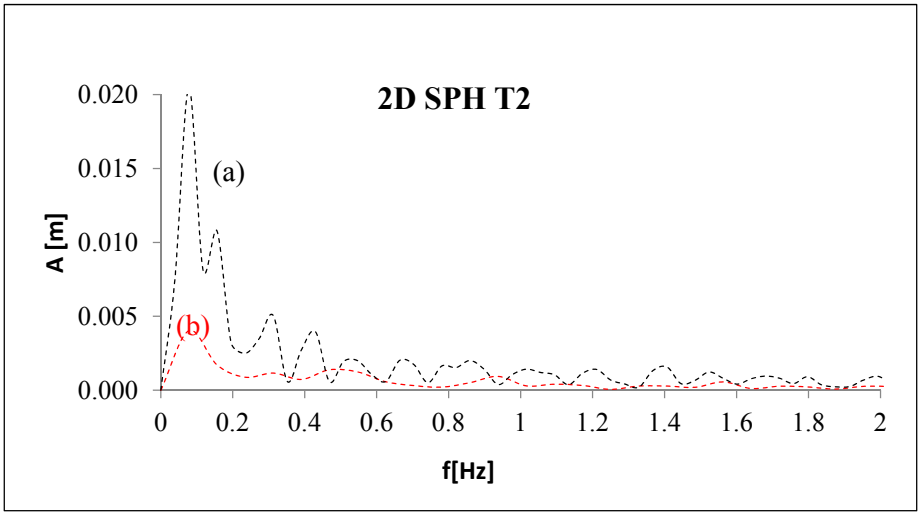
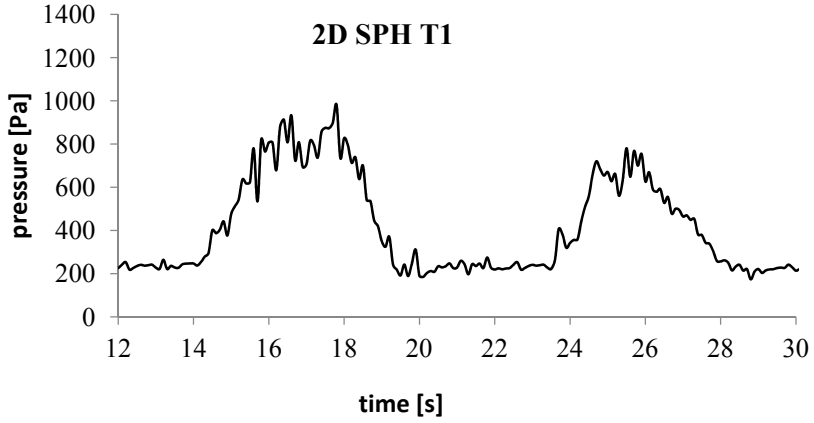


Figure 6. Test T2: Amplitude spectrum of the time series of the surface elevations (a) upstream and (b) downstream of the jump

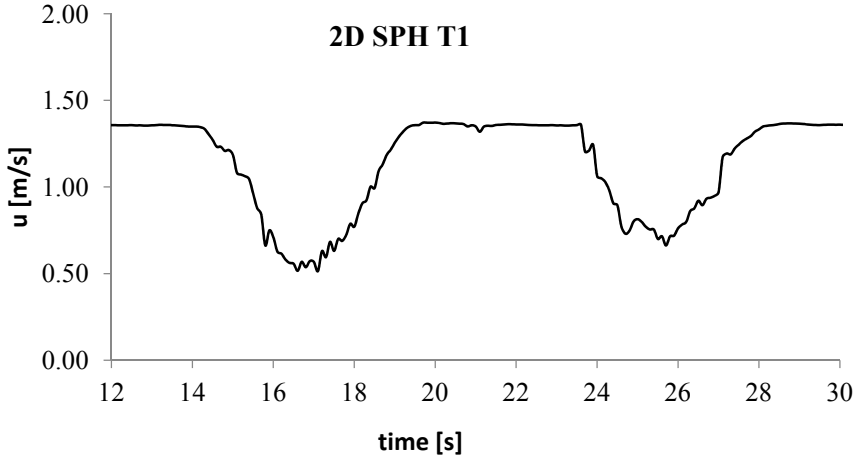
The quasi-periodic oscillations between the two jump types were accompanied by drastic instantaneous changes in the pressure and in the velocity components under the jump (e.g. Mossa, 1999). Figures 7a - 8c show a part of the time history and the amplitude spectrum of the pressure. The horizontal (u) and vertical (v) velocity components measured at the bottom under the hydraulic jump T1 of Table 2, near the time –averaged position of the hydraulic jump toe are shown. From the analysis of the pressure amplitude spectrum (Fig. 8a) it is clear that even the pressure fluctuations were quasi-periodic and so strongly influenced by the oscillations between the B and wave jump types, as it is shown by the peak amplitude at the same frequency of the elevation spectra.

In the regions close to the roller, the amplitude spectra of the velocity components were characterized by a dominant peak at the same frequency of the elevation (Fig. 5a) and the pressure spectra (Fig. 8a).

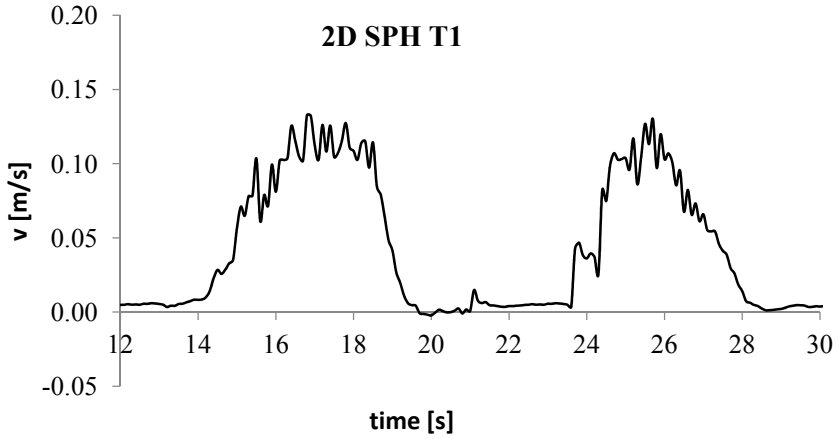
Therefore, the analysis of the oscillating phenomena indicates a correlation among the surface profile elevations, velocity components and pressure fluctuations. In particular, the time histories (Fig. 7) shows that low pressures (low water depths) can be correlated with horizontal flow upstream of the wave-jump, while higher pressures (and depths) can be correlated with the upward flow caused by the roller in the B-jump phase.



(a)

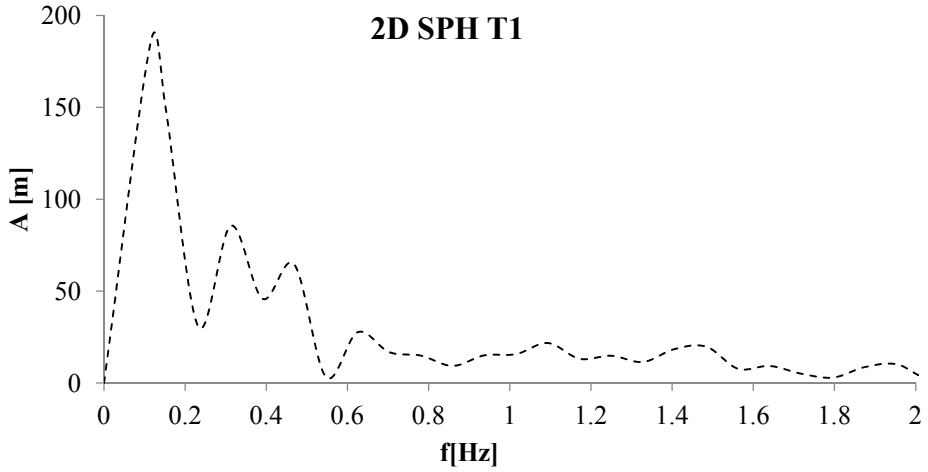


(b)

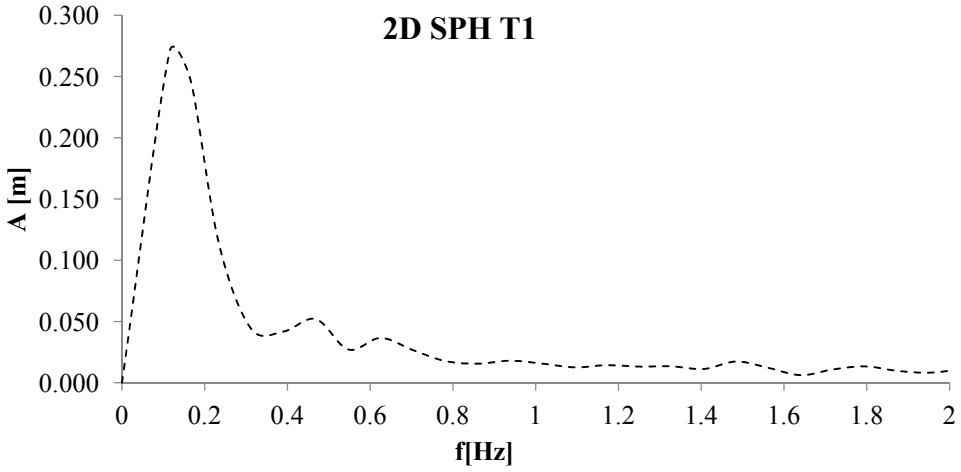


(c)

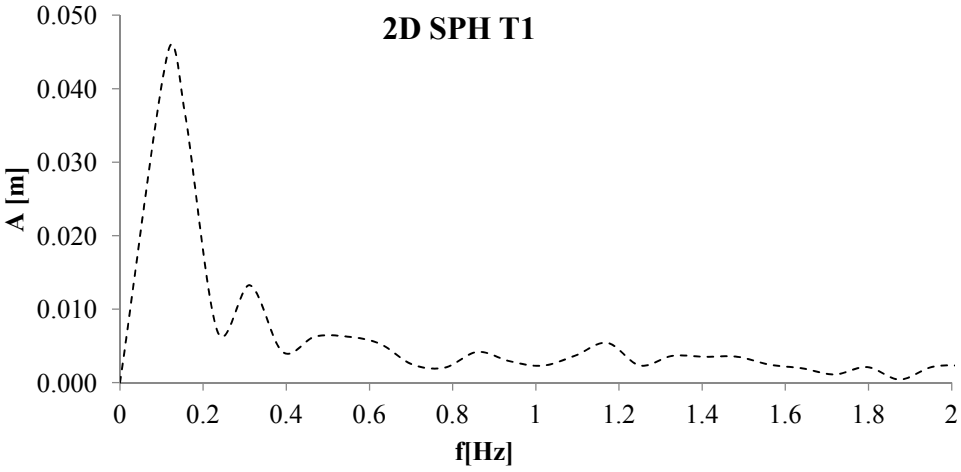
Figure 7. Test T1: Time series of (a) the pressure; (b) the horizontal (u) velocity component and (c) the vertical (v) velocity component measured at the bottom under the hydraulic jump



(a)



(b)



(c)

Figure 8. Test T1: Amplitude spectrum of (a) the pressure fluctuations; (b) the horizontal (u) velocity component and (c) the vertical (v) velocity component measured at the bottom under the hydraulic jump

5 CONCLUSIONS

The 2D SPH model is applied to the modelling of cyclic mechanisms in hydraulic jumps realized in the hydraulic laboratory of the Mediterranean Agronomic Institute of Valenzano (Bari). Oscillating characteristics in the hydraulic jump are investigated and reproduced using a weakly-compressible XSPH scheme, together with either an algebraic mixing-length or a two-equation turbulence model to represent turbulent stresses. A sensitivity analysis was therefore performed on the influence of different turbulence model.

Both the mixing length model and the k-ε model proved to be able to predict the oscillating characteristics and cyclic mechanisms in hydraulic jumps. The numerical results showed indeed a satisfactory agreement with measurements, and most of the peculiar features of the flow were reproduced. By analyzing a single cycle of the oscillating phenomena of a simulated jump (periodic formation of different jump types) for both tests (T1 and T2), it is possible to indicate that there is correlation with the vortex structures of the roller. Vortices are characterized by a clockwise or anti-clockwise rotation depending on which type of jump is present, particularly for configurations with alternations of wave and B jumps.

Although both turbulence models yielded similar results, the detailed comparison of the amplitude spectra of the time series of the surface elevations upstream and downstream of the jump for tests T1 and T2, shows that the mixing-length results proved to be closer to the experimental data than the k-ε ones.

As observed experimentally by Mossa (1999), these numerical results show the existence of a peak at a similar frequency in the amplitude spectra of the time series of the surface elevations upstream and downstream of the jump. In the amplitude spectra of the pressure and in the amplitude spectra of the velocity components fluctuations are measured under the hydraulic jump.

It is possible to conclude that the fluctuations of the surface profile downstream of the jump depended on alternations of jump types, and that the periodic formation of different jump types produces small waves downstream of the hydraulic jump. Therefore, the analysis of the oscillating phenomena indicates a correlation among the surface profile elevations, velocity components and pressure fluctuations, which is consistent with a regular, periodic alternation between a wave-jump and a B-jump, leading to the propagation of downstream waves.

REFERENCES

Abdel Ghafar, A., Mossa, M. & Petrillo, A. (1995). Scour from flow downstream of a sluice gate after a horizontal apron., *Sixth International Symposium on River Sedimentation - Management of Sediment - Philosophy, Aims, and Techniques*, New Delhi, Editors C.V.J. Varma and A.R.G. Rao, Oxford & IBH Publishing Co. Pvt. Ltd., 1069-1088.

Capone, T., Panizzo, A. & Monaghan, J.J. (2010). SPH Modelling of Water Waves Generated by Submarine Landslides. *Journal of Hydraulic Research*, 48, 80–84.

Dalrymple, R.A. & Rogers, B.D. (2006). Numerical Modelling of Waves with The SPH Method. *Coastal Eng.,* 53, 131–147.

De Padova, D., Dalrymple, R.A., Mossa, M. & Petrillo, A.F. (2008). An Analysis of SPH Smoothing Function Modelling a Regular Breaking Wave. *Proc. Nat. Conf. XXXI Convegno Nazionale di Idraulica e Costruzioni Idrauliche*, Perugia, Italy, 182-182.

De Padova, D., Mossa, M. & Sibilla, S. (2009). Laboratory Experiments and SPH Modelling of Hydraulic Jumps. *Proc. Int. Conf. 4th Spheric Workshop*, Nantes, 255–257.

De Padova, D., Mossa, M., Sibilla, S. & Torti, E. (2010). Hydraulic Jump Simulation by SPH. *Proc. Int. Conf. 5th Spheric Workshop*, Manchester, 50–55.

De Padova, D., Mossa, M., Sibilla, S. & Torti, E. (2013). 3D SPH Modelling of Hydraulic Jump in a Very Large Channel. *J Hydraulic Res*, 51, 158-173.

De Padova, D., Dalrymple, R.A & Mossa, M. (2014). Analysis of the Artificial Viscosity in The Smoothed Particle Hydrodynamics Modelling of Regular Waves. *J Hydraulic Res,* 52, 836-848.

De Padova, D., Mossa, M. & Sibilla, S. (2016). SPH Numerical Investigation of The Velocity Field and Vorticity Generation within a Hydrofoil-Induced Spilling Breaker. *Environmental Fluid Mechanics,* 16, 267-287.

Dehnen, W. & Aly, H. (2012). Improving Convergence in Smoothed Particle Hydrodynamics Simulations without Pairing Instability. *Monthly Not. Royal Astr. Soc.*, 425, 1068-1082.

Hager, W.H. & Bretz, N.V. (1986). Hydraulic Jumps at Positive and Negative Steps. *Journal of Hydraulic Research*, IAHR, 24 (4), 237-252.

Jonsson, P., Andreassona P., Gunnar, J., Hellströma, I. Pär Jonsénb & Staffan Lundströma, T. (2016). Smoothed Particle Hydrodynamic Simulation of Hydraulic Jump Using Periodic Open Boundaries, *Applied Mathematical Modelling,* 40 (19–20), 8391–8405.

Launder, B.E. & Spalding D.B. (1974). The Numerical Computation of Turbulent Flows. *Comp Meth Appl Mech Eng,* 3, 269–289.

Manenti, S., Sibilla, S., Gallati, M., Agate, G. & Guandalini, R. (2012). SPH Simulation of Sediment Flushing Induced by A Rapid Water Flow. *Journal of Hydraulic Engineering*, 138(3), 272–284.

Mossa, M. & Tolve, U. (1998). Flow Visualization in Bubbly Two-Phase Hydraulic Jump. *Journal of Fluids Engineering*, ASME, 120(1), 160-165.

Mossa, M. (1999). On the Oscillating Characteristics of Hydraulic Jumps. *Journal of Hydraulic Research*, Taylor & Francis 37(4), 541-558.

Nebbia, G. (1942). Su Taluni Fenomeni Alternativi in Correnti Libere. *L'Energia Elettrica*, fasc. I – vol. XIX, 1-10.

Ohtsu, I. & Yasuda, Y. (1991). Transition from Supercritical to Subcritical Flow at an Abrupt Drop. *Journal of Hydraulic Research*, IAHR, 29 (3), 309-328.

Shao, S. (2006). Incompressible SPH Simulation of Wave Breaking and Overtopping with Turbulence Modeling. *International Journal for Numerical Methods in Fluids*, 50, 597–621.

Sibilla, S. (2008) A SPH-based Method to Simulate Local Scouring. *Proc. 19th IASTED International Conference on Modelling and Simulation*, MS2008, Quebec, 9-14.

Sibilla, S. (2015) An Algorithm to Improve Consistency in Smoothed Particle Hydrodynamics. *Computer and Fluids*, 118, 148-158.

Violeau, D. (2004). One and Two-Equations Turbulent Closures for Smoothed Particle Hydrodynamics. *Proc. 6th Int. Conf. Hydroinformatics*, 87–94.

Violeau, D. (2012). Fluid Mechanics and The SPH Method: Theory and Applications. *Oxford University Press*, Oxford.

Violeau, D., Piccon, S. & Chabard, J.P. (2002). Two Attempts of Turbulence Modelling in Smoothed Particle Hydrodynamics. *Advances in Fluid Modelling and Turbulence Measurements*, World Scientific, 339–346.

Violeau, D. & Rogers B.D, (2016). Smoothed Particle Hydrodynamics (SPH) for Free-Surface Flows: Past, Present and Future. *Journal of Hydraulic Research*, 54(1), 1-26, DOI: 10.1080/00221686.2015.1119209

Wang, H., Murzyn, F. & Chanson, H. (2014). Total Pressure Fluctuations and Two-Phase Flow Turbulence in Hydraulic Jumps. *Exp. Fluids,* 55.11(2014), 1847, 1e16 (DOI: 10.1007/s00348-014-1847-9).

Wang, H. & Chanson, H. (2015). Experimental Study of Turbulent Fluctuations in Hydraulic Jumps. *J. Hydraul. Eng.*, 141(7) http://dx.doi.org/10.1061/(ASCE) HY.1943-7900.0001010. Paper 04015010, 10 pages.

Wendland, H. (1995). Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree. *Advances in Computational Mathematics*, 4, 389–96.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

A NODAL DISCONTINUOUS GALERKIN MODEL WITH QUADRILATERAL MESH FOR 2D SHALLOW WATER EQUATIONS

LONGXIANG LI [(1)] & QINGHE ZHANG [(2)]

[(1,2)]State Key Laboratory of Hydraulic Engineering Simulation and Safety, Tianjin University, Tianjin, China
li12242@tju.edu.cn; qhzhang@tju.edu.cn

ABSTRACT

The discontinuous Galerkin methods are very attractive to shallow water models nowadays, as they can give more accurate results and achieve high parallel efficiency. However, recent researches reported that the discontinuous Galerkin methods on the unstructured triangle meshes suffer from being computationally expensive when compared with those traditional numerical methods. Therefore, a nodal discontinuous Galerkin model with quadrilateral meshes is proposed for shallow water equations in this study. The arbitrary quadrilateral elements along with the quadrature-free approach are introduced to improve the computational efficiency of nodal discontinuous Galerkin methods. Some numerical methods are also implemented in the model, such as the wetting and drying treatment, the maintaining of the well-balanced properties, and the positivity preserving of water depth. The developed model is verified with the oscillatory flow in the parabolic bowl problem. It is shown that the model with quadrilateral meshes is approximately 1.21 to 1.62 times more efficient, and the numerical error is 1.11 to 1.72 times smaller when comparing with the triangle meshes.

Keywords: Shallow water equations; nodal discontinuous Galerkin method; quadrilateral elements; computational efficiency.

1 INTRODUCTION

Nowadays, the discontinuous Galerkin (DG) method has become a popular approach in modeling the shallow water equations (SWE). Many researchers have implemented this new method for various hydrodynamic problems and reported its good performance of less numerical error and higher parallel efficiency. Kubatoko et al. (2009), analyzed the performance of DG method comparing with finite element (FE) method in ADCIRC models. The result indicated that the error of DG was less than that of FE by an order of magnitude, and the DG model also had displayed better convergence rates and higher parallel efficiency. Kesserwani and Wang (2014) compared the 2nd-order Runge-Kutta DG (RKDG) method with the MUSCL-type finite volume (FV) method in modeling of the flood inundation problem, and showed that the RKDG2 method can provide better results in modeling of shock propagations and reflections on the same mesh resolution. However, these researches also pointed out that the DG method suffered from being too computational expensive when comparing with the traditional FV and FE methods. This is mainly due to the fact that DG method contains a larger number of degrees of freedom, and the nonlinear terms in SWE lead to an excessive computational cost in numerical integration.

With an aim of enhancing the computational efficiency, the nodal DG methods have been developed in recent decade. Coupling with the quadrature-free approach, the nodal DG methods are able to get rid of the full order integration in the basic implementation, and save the major computational cost. Besides, more recent studies also reported that the use of quadrilateral mesh could reduce the computational cost of nodal DG method (Wirasaet et al., 2010, 2014). The number of elements and edges in a quadrilateral mesh would be much less than that in a triangular mesh of the same element size. Consequently, the computational efficiency will be significantly improved with less computation in evaluating of volume and edge integral. In this regard, it is worthwhile to develop the nodal DG model with quadrilateral mesh for the sake of improving computational efficiency.

The motivation of this study is to develop a nodal DG model for two-dimensional SWE with arbitrary quadrilateral elements. A set of nodal basis is constructed from the Lagrangian interpolating polynomials, which is associated with the Legendre–Gauss–Lobatto points. To apply the model for various problems, several numerical treatments are implemented. The well-balanced property is maintained by approximating the bottom topography with a set of piecewise continuous functions. For the flooding and drying problems, a wetting and drying treatment of the thin-layer technique is employed. This method can preserve the mass conservation and introduce a much smaller computational cost, as elements remain unchanged and maintain their connectivity throughout the computation. As high-order methods may generate spurious oscillations near discontinuities and lead to numerical instability, a slope limiting approach is also applied. For the flooding and drying problems, a wetting and drying treatment of the thin-layer technique is employed. Moreover, a positivity-preserving operator from (Xing et al., 2010) is adopted to ensure the positivity of water depth.

The paper is arranged as follows. In Section 2, we introduce the discretization of SWE with the nodal DG method, and some numerical treatments of the SWE are also represented. In Section 3, the results of the oscillatory flow in a parabolic bowl problem are investigated to address the accuracy and computational efficiency of the quadrilateral model. Finally, we give some conclusions in Section 4.

2 NUMERICAL DISCRETIZATION OF THE SHALLOW WATER EQUATIONS

2.1 Shallow water equations

The conservative form of the two-dimensional SWE is written in terms of the water depth h and the flow discharge $(q_x, q_y) = (hu, hv)$, given by

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}) \tag{1}$$

where $\mathbf{U}$, $\mathbf{F}(\mathbf{U}) = [\mathbf{E}(\mathbf{U}), \mathbf{G}(\mathbf{U})]^T$, and $\mathbf{S}(\mathbf{U})$ are the vectors of conservative variables, flux terms, and source terms, respectively, and are expressed by

$$\mathbf{U} = \begin{bmatrix} h \\ q_x \\ q_y \end{bmatrix}, \quad E = \begin{bmatrix} q_x \\ q_x^2 / h + gh^2 / 2 \\ q_x q_y / h \end{bmatrix}, \quad G = \begin{bmatrix} q_y \\ q_x q_y / h \\ q_y^2 / h + gh^2 / 2 \end{bmatrix}. \tag{2}$$

The source term $\mathbf{S}(\mathbf{U})$ accounts for the friction and bottom topography effects and is defined as

$$\mathbf{S} = \begin{bmatrix} 0 \\ gh\left(S_{ox} - S_{fx}\right) \\ gh\left(S_{oy} - S_{fy}\right) \end{bmatrix} = \begin{bmatrix} 0 \\ gh\left(-\partial z/\partial x - un^2\sqrt{u^2+v^2}\,/h^{4/3}\right) \\ gh\left(-\partial z/\partial y - vn^2\sqrt{u^2+v^2}\,/h^{4/3}\right) \end{bmatrix}, \tag{3}$$

where the friction terms S_{fx} and S_{fy} are estimated by the Manning law and n is the Manning number.

2.2 Discretization of SWE with nodal DG method

The first step of implementing the nodal DG method is to subdivide the computation domain into N_e non-overlapping elements Ω_e. Then we multiply Eq. [1] by a test function $\boldsymbol{\phi}=\{\phi_i\} \in V_p(\Omega_e)$ and integrate by parts to obtain the weak formulation in each element Ω_e:

$$\int_{\Omega_e} \boldsymbol{\varphi} \frac{\partial \mathbf{U_h}}{\partial t} \mathrm{d}x - \int_{\Omega_e} \nabla \boldsymbol{\varphi} \cdot \mathbf{F}(\mathbf{U_h}) \mathrm{d}x + \oint_{\partial \Omega_e} \boldsymbol{\varphi} \left(\mathbf{F}^*(\mathbf{U_h}) \cdot \mathbf{n}\right) \mathrm{d}s = \int_{\Omega_e} \boldsymbol{\varphi} \mathbf{S}(\mathbf{U_h}) \mathrm{d}x \tag{4}$$

where the basis and test functions belong to the same finite dimensional space $V_p(\Omega_e)$, which consists of Lagrangian polynomials in element Ω_e with degree at most p, and allows discontinuities at element boundaries. $\mathbf{U}_h(x,t) = \sum U_i(t)\boldsymbol{\phi}_i(x)$ is the local approximation of conservative variables $\mathbf{U}$ with the nodal basis functions, where the main aspect of the DG framework is to seek these time-varying coefficients. To allow information transfer between elements, the dual-valued flux terms in the boundary integral are replaced by a numerical flux function $\mathbf{F}^*$. In this study, the HLL flux is adopted, which is widely used in SWE modeling and can provide less dissipative result than Lax-Friedrich flux for small degree p (Cockburn and Shu, 2001; Eskisson and Sherwin, 2004).

To enhance the computational efficiency with nodal DG method, a quadrature-free approach is adopted. A set of interpolation points is first determined on the reference quadrilateral $\Omega_{st} = [-1,1]^2$, while any arbitrary quadrilateral element can be projected to the standard element with an appropriate mapping $x_e(\xi)$: $\Omega_{st}(\xi) \rightarrow \Omega_e(x)$ (Wirasaet et al., 2010). In this study, we use the Legendre–Gauss–Lobatto (LGL) interpolation points that include the boundary points on $\partial\Omega_{st}$. Then some discrete matrices are required

$$M = \int_{\Omega_{st}} \varphi_i \varphi_j \mathrm{d}x, \qquad \mathbf{N} = \int_{\Omega_{st}} \nabla \varphi_i \varphi_j \mathrm{d}x, \quad \mathbf{D} = M^{-1}\mathbf{N},$$
$$M_e = \int_{\partial\Omega_{st}} \varphi_i \varphi_j \mathrm{d}x, \quad J = \left|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right|_{\Omega_e}, \qquad J_e = \left|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right|_{\partial\Omega_e}, \tag{5}$$

where M and M_e is the mass matrix of volume and boundary integrals, **N** is the stiffness matrix, and **D** is the derivative matrix. Numerically, each element of these matrices can be calculated with the Gauss-quadrature rule of integration. J and J_e are diagonal matrices, where each element of the diagonal is the Jacobian at the nodal interpolation points. Now the discretization of Eq. [2] can be written as

$$\frac{\partial \mathbf{U}_i}{\partial t} = -\mathbf{D}\cdot\mathbf{F}_i - \left(J^{-1}J_e\right)\cdot\left(M^{-1}M_e\right)\left(\mathbf{F}_i^* \cdot \mathbf{n}_i\right) + S(\mathbf{U}_i) \tag{6}$$

It is important to point out that in the RHS of Eq. [6], the terms (-**D**·**F**) and ($M^{-1}M_e$(**F***·**n**)) are calculated by the matrix–vector multiplication, while the Jacobian factor ($J^{-1}J_e$) and the source term S(**U**) are multiplied and added to each element of the result. Comparing the computational cost with triangular elements, Wirasaet et al. (2010) analyzed the computation time of nodal DG method on quadrilaterals and triangles for convection problem numerically, and gave the conclusion that the use of quadrilateral elements is less expensive when the degree p is not greater than 3. Consequently, we will restrict the degree p to 1 in this study.

A semi-discrete weak formulation of Eq. [6] is then obtained and can be written as an ordinary differential equation (ODE) as follows:

$$\frac{\partial \mathbf{U}_i}{\partial t} = L(\mathbf{U}_i) \tag{7}$$

The resulting ODE is solved by the explicit four-order five-stage Runge-Kutta method, which can be defined in the general form

$$\begin{aligned} &\mathbf{U}^{(0)} = \mathbf{U}^n, \\ &\mathbf{U}^{(i)} = \sum_{l=0}^{i-1}\{\alpha_{il}\mathbf{U}^{(l)} + \Delta t\beta_{il}L(\mathbf{U}^{(l)})\}, \quad i = 1,2,\cdots,s \\ &\mathbf{U}^{n+1} = \mathbf{U}^{(s)} \end{aligned} \tag{8}$$

where α_{il} and β_{il} are the coefficients of the Runge-Kutta method and $\mathbf{U}^n$ and $\mathbf{U}^{n+1}$ are the results at different times t and $t + \Delta t$. Some numerical treatments are applied in each RK stage to eliminate numerical oscillation and enforce the stability of the scheme, which will be described in the next section.

2.3 Numerical treatment of SWE method

To preserve the equilibrium states, the well-balancing (*C*-property) is one of the most important properties for SWE models, which can be provided by a large number of numerical schemes. In this model, the well-balanced DG model is guaranteed by adopting a continuous piecewise polynomial function z_h to approximate the bottom topography $z(x)$, which is the same as Bollermann et al. (2013). For discontinuous topography, one may also use the hydrostatic reconstruction method (Audusse et al., 2004) by adding a correction term to the Eq. [6].

Another important property of SWE model is the ability of handling the wetting and drying process. An efficient wetting and drying treatment from Bunya et al. (2009) is adopted. This treatment involves a threshold of water depth h_0 for detecting the dry region. For the elements Ω_e with $\bar{h}_e < h_0$ ($\bar{h}_e$ is the averaged depth), the gravity terms is canceled by setting g = 0, and the numerical fluxes **F*** between dry elements are also neglected to preserve the conservation of water mass and momentum.

Some numerical treatments are also needed to preserve the numerical stability for DG schemes. One is related with the calculation of the flux Jacobian $|\mathbf{u}| + \sqrt{gh}$. When the water depth is sufficient small, a spurious large flow rate many be derived by (q_x/h, q_y/h) with a small truncation error in (q_x, q_y). This will lead to a very small time step from the Courant-Friedrichs-Lewy (CFL) criterion. In this study, we simply ignore the flow rate when the averaged water depth is less than h_0. For high order schemes, a stabilization technique is required to remove the numerical oscillations. A modified Barth-Jespersen slope limiter (Barth and Jespersen, 1989) for arbitrary quadrilateral elements is used in this model, which is only used at the local extrema and is

efficient to ensure the stability. A positivity-preserving limiter is also used to preserve the positivity of water depth. The operator is first proposed by Xing, et al. (2010) and defined as

$$\mathbf{U}_i^* = \theta\left(\mathbf{U}_i - \bar{\mathbf{U}}_e\right) + \bar{\mathbf{U}}_e, \quad \theta = \min\left\{1, \frac{\bar{h}_e}{\bar{h}_e - h_i^{\min}}\right\} \tag{9}$$

where the notation $\mathbf{U}_i$ denotes the conservative variables at each node, and $\bar{\mathbf{U}}_e = \left(\bar{h}_e, \bar{q}_{x,e}, \bar{q}_{y,e}\right)^T$ is the averaged variables of each element. It is clear that the modified depth h_i^* is positive at all nodes, and the conservation of mass and momentum is also ensured in each cell. For dry cells with a negative mean water depth $\bar{h}_e < 0$, we simply set $\bar{h}_e = 0$.

3 NUMERICAL TEST

3.1 Oscillatory flow in a parabolic bowl

A periodic oscillatory flow problem is used to analyze the accuracy and performance of the new quadrilateral mesh model. The computation domain is a [-4000, 4000] m × [-4000, 4000] m rectangle with a parabolic bottom topography $z(x) = ar^2$, where $r^2 = x^2 + y^2$ and a is a positive constant of 1.6×10^{-7} m^{-1}. The bottom is smooth with Manning number $n = 0$. The analytical solutions of the water depth and momentum are given by

$$h(r,t) = \frac{1}{X + Y\cos\omega t} + \frac{\alpha\left(Y^2 - X^2\right)r^2}{\left(X + Y\cos\omega t\right)^2} \tag{10}$$

$$\left(u, v\right) = -\frac{Y\omega\sin\omega t}{X + Y\cos\omega t}\left(\frac{x}{2}, \frac{y}{2}\right)$$

where the parameter $\omega^2 = 8ga$, X and Y are set to 1 m^{-1} and -0.41884 m^{-1} respectively. The period of the solution is $T = 2\pi/\omega = 1773.13$ s. The initial condition of $\mathbf{U}_h$ is defined by Eq. [10] at $t = 0$ s and the model runs for two periods and ends at $t = 2T$. The CFL number is set to 0.3 and the time step is determined by the following formula

$$\Delta t = CFL \cdot \min_{\Omega_e} \frac{\delta_e}{\left(\lambda_{\max}\right)_i} \tag{11}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the Jacobian of the flux term and δ_e is the elemental length, which can be calculated by $\sqrt{A}$ for quadrilaterals and $\sqrt{2A}$ for triangles, with A being the area of element. To reduce the accuracy reduction from the slope limiter, the Barth-Jespersen limiter only acts on the conservative variables $\mathbf{U}_h = (h, q_x, q_y)$ in wet regions, while other numerical operators, such as the wet/dry treatment and positivity-preserving operator, are applied to the whole domain. The threshold of h_0 is set to 10^{-4} m and the gravitation constant g is 9.81 m^2/s.

In Figure 1, the numerical results on a coarse uniform quadrilateral mesh with 80×80 elements are presented. At the first half period, the water runs-up onto the parabolic bowl and submerges a large area of the domain as shown in Figure 1 (a), (b) and (c). Then the water retreats and finally stops at its initial position in the second half period as illustrated in Figure 1 (d), (e) and (f). In Figure 2, the numerical results on section $x = 0$ m are compared with exact solutions at different times, where both the water depth and flux are nicely approximated by the numerical results. Small wiggles are observed in the solution h during the retreating process, which are mainly caused by the wetting and drying treatments from the wet/dry interface and can be rapidly reduced with finer meshes.

To assess the convergence rate of DG methods, we calculate the norm error on several uniform quadrilateral meshes of $N_e \times N_e$ elements (N_e = 80, 100, 120, 140, 160 and 180 respectively), where N_e is the number of elements on each edge. The norm error of L_2 is computed on each mesh at $t = T$, and the convergence rate is given by

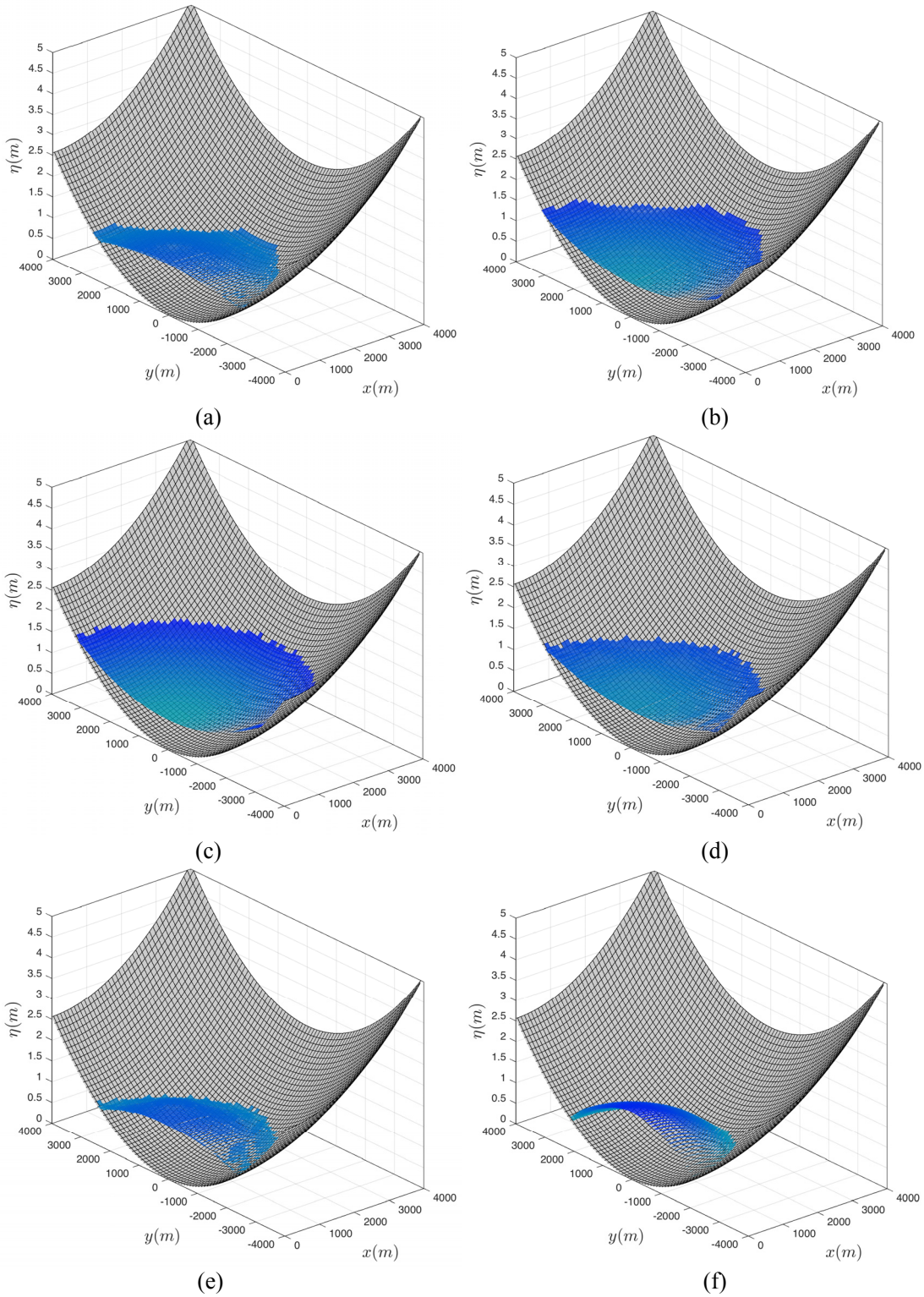


Figure 1. The results of water depth at different times: (a) $t = T/6$, (b) $t = T/3$, (c) $t = T/2$, (d) $t = 2T/3$, (e) $t = 5T/6$, and (f) $t = T$.

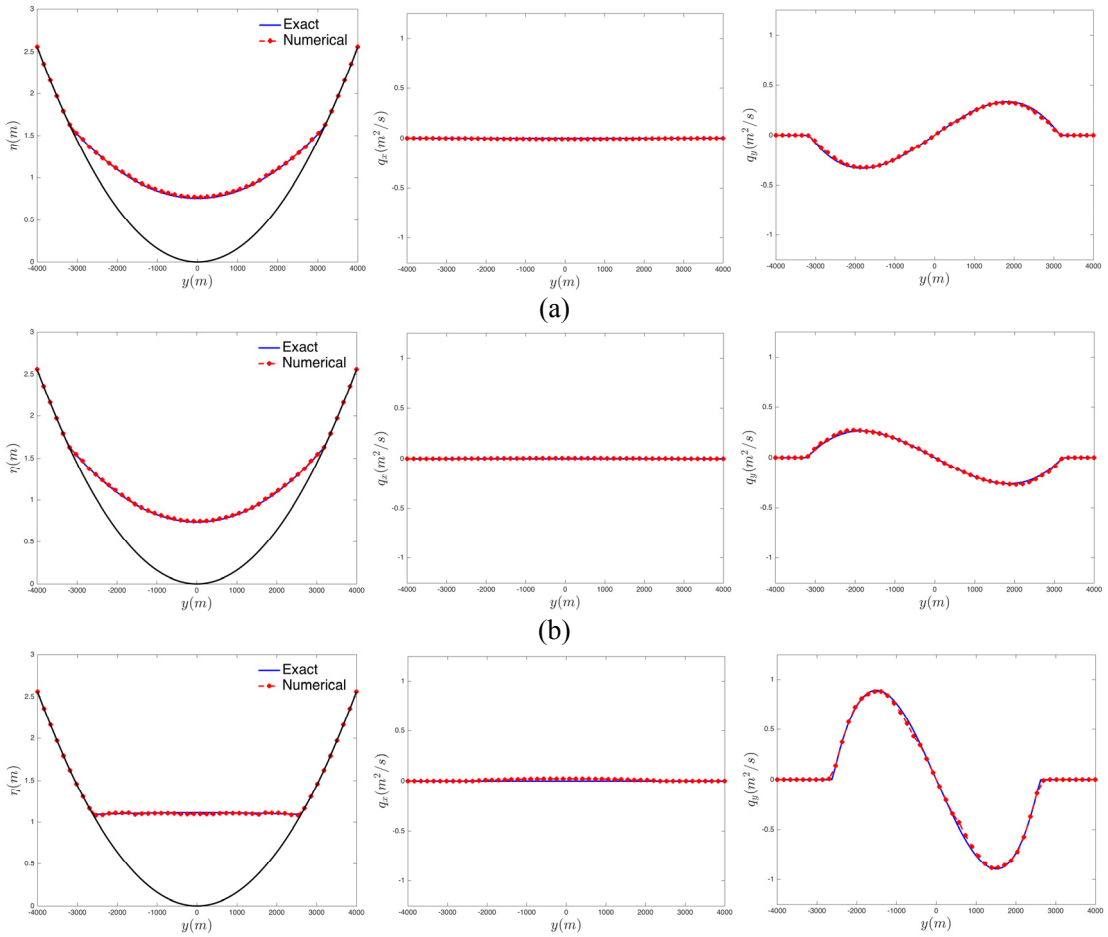


Figure 2. Comparisons of the numerical and exact solutions of h (left), q_x (middle) and q_y (right) on section x = 0 m at different times: (a). $t = 2T/5$, (b) $t = 3T/5$, and (c) $t = 4T/5$.

$$r = \frac{\log(e_c / e_f)}{\log(\delta_c / \delta_f)} \qquad [11]$$

where the e_c, e_f and δ_c, δ_f are the norm error and element length of the coarse and fine meshes, respectively. The errors and convergence rates of variables h, q_x and q_y are shown in Table 1, where the fitted convergence rates are obtained by the least square methods. The convergence rates for water depth h and flux q_x and q_y in the L_2 norm are close to 1.25, 1.00 and 1.00, respectively. In this test, the discontinuities appear in the solutions at the wet/dry interface, and it is well known that the slope limiter will degenerate the accuracy of the schemes to first order, so it is unexpected that all of these rates are equal to or greater than the optimal degree. Nonetheless, the convergence rates are within an acceptable range and the quadrilateral mesh-based model is able to offer accurate solutions with very fine mesh resolution. Similar results can be found in Bunya et al. (2009).

Table 1. Errors and convergence rates of the oscillatory flow problem at $t = T$.

N_e	δ_e	h		q_x		q_y	
		$L_2(h)$	r	$L_2(q_x)$	r	$L_2(q_y)$	r
80	98.7	4.54E-03	\	9.93E-03	\	9.93E-03	\
100	80.0	3.38E-03	1.32	7.62E-03	1.19	7.62E-03	1.19
120	66.7	2.59E-03	1.45	6.23E-03	1.10	6.23E-03	1.10
140	57.1	2.18E-03	1.11	5.50E-03	0.82	5.50E-03	0.82
160	50.0	1.86E-03	1.19	4.90E-03	0.85	4.90E-03	0.85
180	44.4	1.66E-03	0.99	4.34E-03	1.04	4.34E-03	1.04

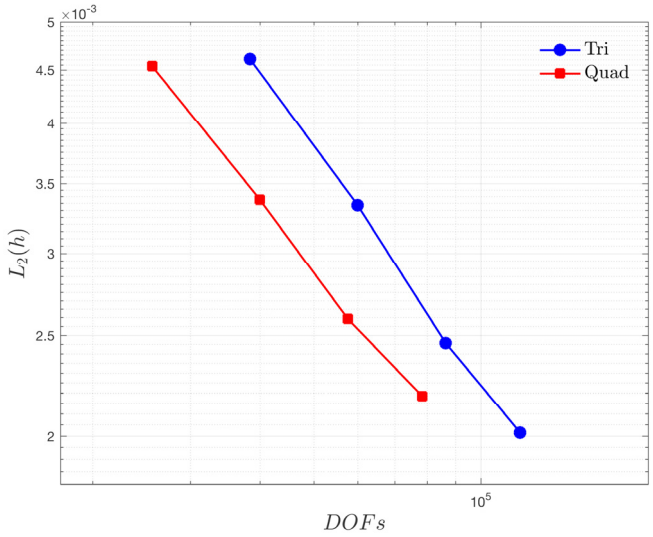


Figure 3. Comparisons of the numerical error $L_2(h)$ at $t = 2T$ on triangle and quadrilateral meshes for different DOFs.

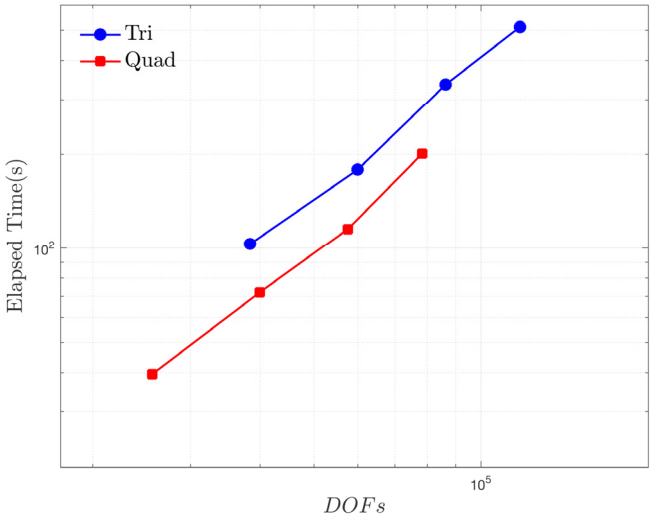


Figure 4. Comparisons of the computation time used on triangle and quadrilateral meshes for different DOFs.

To demonstrate the high computational efficiency of the quadrilateral-based model, we compared the error and the computational time with the triangle mesh-based model in the same framework. For the uniform triangle mesh, the elements are created by dividing the square elements along the diagonal (N_e = 80, 100, 120 and 140), while the time step Δt are the same with corresponding quadrilateral mesh. The results of numerical error $L_2(h)$ and the computation time used for different N_e are shown in table 2. Figure 3 illustrates the logarithmic plot of the error $L_2(h)$ and the degrees of freedoms (DOFs) for different meshes at the end of the simulation. It is obvious that the logarithms of the errors are proportional to the logarithms of DOFs for both of triangle and quadrilateral meshes. For the same number of N_e, the $L_2(h)$ errors on triangle meshes are corresponding with that on quadrilateral meshes, however, the triangle meshes contains more DOFs which are 1.5 times more than that on quadrilateral meshes. Therefore, with the same number of DOFs, the errors of quadrilateral meshes are only 1.11 to 1.72 times smaller than the error of triangle meshes. Similarly, in Figure 4, the logarithmic of the computing time used and the DOFs are also proportional as expected. This probably because that the DOFs has a close relationship with the computational cost of matrix-vector multiplication. As the quadrilateral meshes contain a smaller DOFs, the computation with quadrilateral meshes are approximately 1.21 to 1.62 times more efficient than using triangle meshes when the DOFs is held fixed. Overall, combing the information in Figure 3 and 4, we can give the conclusion that with the same number of DOFs, the quadrilateral-based DG model can provide more accurate results with less computation time. It is important to point out that this conclusion applies for low order degree p = 1, while for high order p it still needs further investigation.

Table 2. Error and computation time for triangles and quadrilaterals on different mesh resolutions at $t = T$.

N_e	*Tri*				*Quad*			
	DOFs	***$L_2(h)$***	***r***	***time(s)***	***DOFs***	***$L_2(h)$***	***r***	***time(s)***
80	38400	4.61E-03	\	102.93	25600	4.53E-03	\	39.548
100	60000	3.34E-03	1.45	178.42	40000	3.38E-03	1.32	72.072
120	86400	2.46E-03	1.68	335.11	57600	2.59E-03	1.45	114.95
140	117600	2.01E-03	1.29	511.33	78400	2.18E-03	1.11	200.70

4 CONCLUSIONS

In this study, we proposed the nodal DG methods on quadrilateral meshes for the two-dimensional SWE. The motivation is to enhance the computational efficiency of the nodal DG method for SWE model. In addition, some numerical treatments are also introduced, and the model is able to preserve stable in handling the complex wetting and drying problem. To verify the new model, the oscillatory flow in the parabolic bowl problem is simulated. The numerical solutions indicate that the model can give robust and accurate results, and the expected order of convergence rate is achieved. Small wiggles on water depth h are produced at the wet/dry interface in the coarse meshes, but these errors are rapidly reduced with finer meshes. Moreover, to demonstrate the performance of accuracy and efficiency, the quadrilateral model is compared with the triangular model in the same framework. The solutions show that the quadrilateral meshes require less computational time (1.21 to 1.62 time shorter) while providing a relatively small numerical error (1.11 to 1.72 times smaller) when the DOFs is held fixed. From this point of view, it is more efficient to use the DG SWE model on quadrilateral meshes. It is necessary to point out that this study restricts to low order with $p = 1$, there still needs further investigation for high order degrees.

ACKNOWLEDGEMENTS

The Research Fund supported the research for the Doctoral Program of Higher Education of China (Grant No. 20120032130001) and the Science Fund for Creative Research Groups of the National Natural Science Foundation of China (Grant No. 51621092).

REFERENCES

Audusse E., Bouchut F., Bristeau M.-O., Klein R. & Perthame B.T. (2004). A Fast and Stable Well-Balanced Scheme with Hydrostatic Reconstruction for Shallow Water Flows. *SIAM Journal on Scientific Computing*, 25(6), 2050–2065.

Barth, T. & Jespersen, D. (1989). The Design and Application of Upwind Schemes on Unstructured Meshes. *27th Aerospace sciences meeting*, 366.

Bollermann A., Chen G., Kurganov A. & Noelle S. (2013). A Well-Balanced Reconstruction of Wet/Dry Fronts for the Shallow Water Equations. *Journal of scientific computing*, 56(2), 267–290.

Bunya S., Kubatko E.J., Westerink J.J. & Dawson C. (2009). A Wetting and Drying Treatment for the Runge–Kutta Discontinuous Galerkin Solution to The Shallow Water Equations. *Computer Methods in Applied Mechanics and Engineering*, 198(17-20), 1548–1562.

Cockburn B. & Shu C.-W. (2001). Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. *Journal of Scientific Computing*, 16(3), 173–261.

Eskilsson C. & Sherwin S.J. (2004). A Triangular Spectral/Hp Discontinuous Galerkin Method for Modelling 2D Shallow Water Equations. *International Journal for Numerical Methods in Fluids*, 45(6), 605-623.

Gassner G.J., Lörcher F., Munz C.-D. & Hesthaven J.S. (2009). Polymorphic Nodal Elements and Their Application in Discontinuous Galerkin Methods. *Journal of Computational Physics*, 228(5), 1573-1590.

Hesthaven J.S. & Warburton T. (2002). Nodal High-Order Methods on Unstructured Grids. *Journal of Computational Physics*, 181(1), 186-221.

Kesserwani G. & Wang Y. (2014). Discontinuous Galerkin Flood Model Formulation: Luxury or Necessity? *Water Resources Research*, 50(8), 6522-6541.

Kubatko, E.J., Bunya, S., Dawson, C. & Westerink, J.J. (2009). A Performance Comparison of Continuous and Discontinuous Finite Element Shallow Water Models. *Journal of Scientific Computing*, 40(1), 315-339.

Wirasaet D., Tanaka S., Kubatko E.J., Westerink J.J. & Dawson C. (2010). A Performance Comparison of Nodal Discontinuous Galerkin Methods on Triangles and Quadrilaterals. *International Journal for Numerical Methods in Fluids*, 64(10‐12), 1336-1362.

Wirasaet D., Kubatko E.J., Michoski C.E., Tanaka S., Westerink J.J. & Dawson C. (2014). Discontinuous Galerkin Methods with Nodal and Hybrid Modal/Nodal Triangular, Quadrilateral, and Polygonal Elements for Nonlinear Shallow Water Flow. *Computer Methods in Applied Mechanics and Engineering*, 270(C), 113–149.

Xing Y., Zhang X. & Shu C.-W. (2010). Positivity-Preserving High Order Well-Balanced Discontinuous Galerkin Methods for The Shallow Water Equations. *Advances in Water Resources*, 33(12), 1476–1493.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

A 3D NUMERICAL SCHEME FOR NON-HYDROSTATIC FLOWS

FRANCESCO GALLERANO[(1)], GIOVANNI CANNATA[(2)],FRANCESCO LASAPONARA[(3)] & CHIARA PETRELLI[(4)]

[(1,2,3,4)] Department of Civil, Construction and Environmental Engineering (DICEA) Sapienza University of Rome, Italy
francesco.gallerano@uniroma1.it; giovanni.cannata@uniroma1.it; francesco.lasaponara@uniroma1.it; chiara.petrelli@uniroma1.it

ABSTRACT

In this paper, three-dimensional simulation of hydrodynamic phenomena (wave refraction, diffraction, shoaling and breaking) are performed by a finite-volume non-hydrostatic and shock-capturing three-dimensional model. The Navier-Stokes equations are solved on a time dependent coordinate system. The solution procedure of the equations of motion is based on a high-order WENO reconstruction procedure and a third order accurate Runge-Kutta (SSPRK) fractional-step method and adopts a pressure corrector formulation in order to obtain a divergence-free velocity field at each stage. The proposed model is validated against several benchmark test cases.

Keywords: 3D model; non-hydrostatic flows; shock-capturing scheme; time dependent coordinate system; pressure-correction method.

1 INTRODUCTION

The main problem to deal with in the study of nearshore hydrodynamics is the wave transformation simulation from deep water to coastal regions. An efficient and well-tested tool for the simulation of the above-mentioned phenomena is given by Boussinesq-type wave models with improved non-linearity and dispersion characteristics (Gallerano et al., 2014; 2016).

A different approach is based on the direct solution of the Navier-Stokes equations by using a tracking technique in order to follow the free surface elevation (maker-and-cell (MAC) and volume-of-fluid (VOF)). In many 3D models, Cartesian coordinates are used and thus the free surface crosses the computational cell arbitrarily. This leads to the difficulty of applying the pressure boundary condition on the free surface and may affect the accuracy of the velocity computation. An alternative approach can be obtained by assuming that the free surface is always placed at the upper computational boundary. Thus, the pressure boundary condition can be fixed over it. In order to solve the Navier-Stokes equations, the pressure is decomposed in a hydrostatic and a non-hydrostatic component (Stelling and Zijlema, 2003; Yuan and Wu, 2004a; 2004b; Young and Wu, 2010). Stelling and Zijlema (2003) proposed a model in which the pressure is located at the cell faces rather than the cell centres (known as the Keller-box method) in order to replace staggered grids in the vertical direction. In this model, the pressure boundary condition at the free surface is exactly assigned to zero without any approximation. Many authors simulate the fluid dynamic fields on unstructured grids (Oka and Ishihara, 2009; Cioffi and Gallerano, 2006).

The numerical schemes that are adopted for the solution of the equations of motion ensure convergence to the correct weak solutions if these equations are expressed in an integral form and in terms of conserved variables (Toro, 2009). In order to track the actual location of the wave breaking without using any criterion, the above-mentioned numerical schemes should be of shock-capturing type.

In this paper, the equations of motion are expressed in a new integral form and in terms of conserved variables. In order to express the equations in a time-dependent coordinate system, the time-varying coordinates of the physical domain are mapped to a uniform transformed space. A third order Runge-Kutta (SSPRK) fractional-step method is involved in the solution procedure. A divergence free velocity field is obtained at each stage by a pressure corrector formulation. A shock-capturing numerical procedure, a high order WENO reconstruction technique (Gallerano and Cannata, 2011) and an approximate HLL Riemann solver (Harten et al., 1983) are on the basis of the discretization of the equations of motion in the predictor phase. The solution of the Poisson equation in the corrector phase is based on an alternate Zebra four-colour Gauss-Siedel relaxation method with a V-cycle multigrid strategy (Trottenberg et al., 2001).

2 EQUATIONS OF MOTION

We define the water depth as $H(x^1,x^2,t) = h(x^1,x^2,t) + \eta(x^1,x^2,t)$ where h is the still water depth and η is the free surface elevation. In order to accurately represent the bottom and free surface geometry and to correctly assign on them the pressure and the kinematic conditions, we consider a transformation from the Cartesian coordinates (x^1,x^2,x^3,t) to the transformed coordinates (ξ^1,ξ^2,ξ^3,τ)

$$\xi^1 = x^1 \quad , \quad \xi^2 = x^2 \quad , \quad \xi^3 = \frac{x^3+h}{H} \quad , \quad \tau = t, \qquad [1]$$

The time dependent coordinate system moves with velocity given by $v_m = \left\{0, 0, \frac{\partial x^3}{\partial \tau}\right\}$. This coordinate transformation basically maps the varying vertical coordinates in the physical domain to a uniform transformed space where ξ^3 spans from 0 to 1.

Let $\vec{g}_{(l)} = \partial \vec{x}/\partial \xi^l$ and $\vec{g}^{(l)} = \partial \xi^l/\partial \vec{x}$ be respectively the covariant and the contravariant base vectors. Let us also define $g_{(lm)}$, $g^{(lm)}$ and $\sqrt{g}$ respectively as the metric tensor, its inverse and the Jacobian of the transformation. For the considered coordinate transformation (Eq. [1]), the Jacobian is equal to the water depth, $\sqrt{g} = H$. Let $\Delta V(t)$, $\Delta A(t)$ and ΔV^*, ΔA^* be the volume and surface elements expressed in the physical space and transformed space, respectively, as $\Delta V(t) = \Delta x^1 \Delta x^2 \Delta x^3 = \sqrt{g}\Delta\xi^1\Delta\xi^2\Delta\xi^3$, $\Delta A(t) = \Delta x^\alpha \Delta x^\beta = \sqrt{g}\Delta\xi^\alpha\Delta\xi^\beta$ and $\Delta V^* = \Delta\xi^1\Delta\xi^2\Delta\xi^3$, $\Delta A^* = \Delta\xi^\alpha\Delta\xi^\beta$ (in which $\alpha, \beta = 1,2,3$ are cyclic). Let us define u_l ($l = 1, 3$) as the fluid velocity in the Cartesian reference coordinate system. H and u_l are the primitive variables while H and Hu_l are the conserved variables. We define the cell averaged value of the primitive and conserved variables in the transformed space, respectively, as

$$\bar{H} = \frac{1}{\Delta V^*}\int_{\Delta V^*} H d\xi^1 d\xi^2 d\xi^3 \ , \ \bar{u}_l = \frac{1}{\Delta V^*}\int_{\Delta V^*} u_l d\xi^1 d\xi^2 d\xi^3 \ , \ \overline{Hu_l} = \frac{1}{\Delta V^*}\int_{\Delta V^*} Hu_l \, d\xi^1 d\xi^2 d\xi^3 \qquad [2]$$

We introduce a restrictive condition on the control volume $\Delta V(t)$: in the following, $\Delta V(t)$ must be considered as a volume element defined by surface elements bounded by curves lying on the coordinate lines. By this condition, the integral form of the momentum equation in the transformed coordinates, for an incompressible fluid, can be written as

$$\begin{aligned}\frac{\partial \overline{Hu_l}}{\partial \tau} = &-\frac{1}{\Delta V^*}\sum_{\alpha=1}^{3}\left\{\int_{\Delta A^{*\alpha+}}\left[Hu_l(u_m - v_m)g_m^{(\alpha)} + GH^2 g_l^{(\alpha)}\right]d\xi^\beta d\xi^\gamma \right.\\ &\left. - \int_{\Delta A^{*\alpha-}}\left[Hu_l(u_m - v_m)g_m^{(\alpha)} + GH^2 g_l^{(\alpha)}\right]d\xi^\beta d\xi^\gamma\right\}\\ &+\frac{1}{\Delta V^*}\sum_{\alpha=1}^{3}\left\{\int_{\Delta A^{*\alpha+}} GhHg_l^{(\alpha)} d\xi^\beta d\xi^\gamma - \int_{\Delta A^{*\alpha-}} GhHg_l^{(\alpha)} d\xi^\beta d\xi^\gamma\right\}\\ &-\frac{1}{\Delta V^*}\frac{1}{\rho}\int_{\Delta V^*}\frac{\partial q}{\partial \xi^k} g_l^{(k)} H \, d\xi^1 d\xi^2 d\xi^3 +\\ &\frac{1}{\Delta V^*}\sum_{\alpha=1}^{3}\frac{1}{\rho}\left\{\int_{\Delta A^{*\alpha+}} T_{lm} g_m^{(\alpha)} H d\xi^\beta d\xi^\gamma - \int_{\Delta A^{*\alpha-}} T_{lm} g_m^{(\alpha)} H d\xi^\beta d\xi^\gamma\right\}\end{aligned} \qquad [3]$$

Here the indexes α, β and γ are cyclic. $g_m^{(\alpha)}$ indicates the m-th component (in the Cartesian reference system) of the contravariant base vector $\vec{g}^{(\alpha)}$. $\Delta A^{*\alpha+}$ and $\Delta A^{*\alpha-}$ indicate the contour surfaces of the volume element on which ξ^α is constant and which are respectively located at the larger and at the smaller value of ξ^α. G is the gravity constant, ρ is the fluid density, q is the dynamic pressure and T_{lm} is the stress tensor. From Eq. [3] it can be noticed that the conserved variables are advanced in time within a transformed space which is not time varying. The time variation of the geometric components is expressed by the time variation of the metric terms.

Let $\Delta A^*_{xy} = \Delta\xi^1\Delta\xi^2$ be the horizontal surface element in the transformed space. By taking into account the surface and bottom kinematics boundary conditions and by considering that the water depth H does not depend on the ξ^3 coordinate (ξ^3 spanning from 0 to 1) and that ΔV^* is not time-varying, we obtain the following equation

$$\frac{d}{d\tau}\int_{\Delta V^*} H \, d\xi^1 d\xi^2 d\xi^3 = \int_{\Delta A^*_{xy}}\frac{\partial H}{\partial \tau} \, d\xi^1 d\xi^2 \qquad [4]$$

in which the right-hand side can also be written as

$$\int_{\Delta A^*_{xy}}\frac{\partial H}{\partial \tau} \, d\xi^1 d\xi^2 = \Delta A^*_{xy}\frac{\partial \bar{H}}{\partial \tau} \qquad [5]$$

By using Eqs. [4] and [5] and by applying the bottom and surface kinematic boundary conditions, the integral form of the continuity equation in the transformed coordinates can be written as

$$\frac{\partial \bar{H}}{\partial \tau} + \frac{1}{\Delta A^*_{xy}}\int_0^1 \sum_{\alpha=1}^{2}\left[\int_{\xi^{\alpha+}} Hu_\alpha d\xi^\beta - \int_{\xi^{\alpha-}} Hu_\alpha d\xi^\beta\right] d\xi^3 = 0 \qquad [6]$$

in which $\xi^{\alpha+}$ and $\xi^{\alpha-}$ indicate the contour lines of the surface element ΔA^* on which ξ^α is constant and which are located at the larger and at the smaller value of ξ^α respectively. Equation [6] represents the governing equation for the free surface movements. Equations [3] and [6] represent the three-dimensional equations of motion as a function of the $\overline{Hu_l}$ and $\bar{H}$ conserved variables in the time dependent coordinate system

$(\xi^1, \xi^2, \xi^3, \tau)$. The numerical integration of these equations makes it possible to simulate the fully dispersive wave phenomena.

3 NUMERICAL SCHEME

Eqs. [3] and [6] are solved by using a combined finite-difference/finite-volume scheme. A grid staggering strategy is used (Stelling and Zijlema, 2003) where the fluid velocities are located at the centres of the computational cells and the fluid pressure is defined in correspondence of the horizontal faces of the cells. The time discretization of Eqs. [3] and [6] is done by means of a third-order accurate Strong Stability-Preserving Runge-Kutta method (Gottlieb et al., 2009). At each time level, a divergence-free velocity field is obtained by using a pressure-correction method. The proposed numerical procedure can be summarized as follows.

(1) At the face centre of each computational cell, two point values of the conserved variables are reconstructed by means of a high-order upwind WENO scheme (Gallerano et al., 2012).
(2) These two values are used as initial data of a local approximate HLL Riemann solver that gives the updated volume flux on the cell face.
(3) The volume fluxes, obtained at the cell faces by step (2), are used to calculate a non-hydrostatic (predictor) velocity field that is not, in general, divergence-free.
(4) The divergence of the predictor velocity field is used as the right-hand side of a Poisson pressure equation that is numerically solved by using a four-colour Zebra line Gauss-Seidel alternate method and a multigrid strategy.
(5) The pressure field, as calculated at step (4), is used to correct the predictor field in order to obtain a divergence-free velocity field.
(6) The non-hydrostatic resulting velocity field is used to advance in time the total water depth by Eq. [6].

4 RESULTS

4.1 Spatial evolution of waves propagating over a longshore bar.

The experimental test proposed by Beji and Batties (1993) is numerically reproduced in order to verify the ability of the proposed model to simulate the spatial evolution of waves propagating over a longshore bar.

The experimental test is carried out in a flume of length $37.7\ m$ where a submerged trapezoidal bar is present, consisting of a slope of $1:20$ and a $2m$ horizontal crest, $0.3\ m$ above the bottom of the flume, followed by a $1:10$ slope. The still water level in the deep region is $0.4\ m$ and reduced to $0.1m$ above the horizontal bar part. Periodic incident non-breaking waves with frequency 0.5 Hz and wave height $2\ cm$ were experimentally generated.

The numerical simulation reproduces the experimental flume (Figure 1) and the wave conditions. Waves are generated on the left boundary and a plane sloped beach of $1:25$ is placed on the opposite boundary in order to dissipate the wave motion. The time discretization is $0.001s$ and the spatial discretization is $\Delta x = 0.01\ m$. Three vertical layers are used for the simulation of the dispersive and non-linear phenomena related to this test.

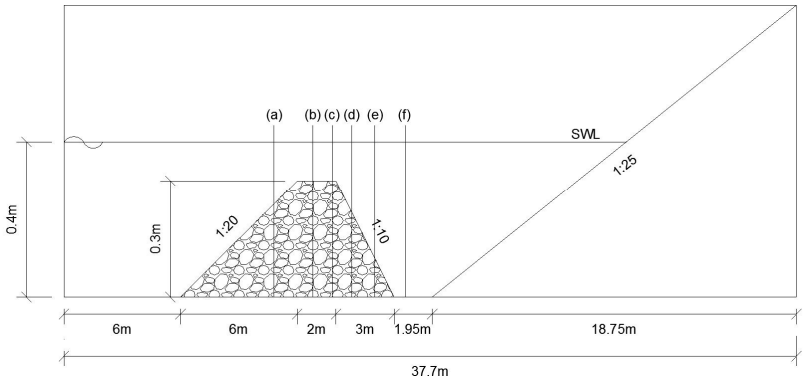


Figure 1. Spatial evolution of waves propagating over a longshore bar. Bottom topography and location of the experimental measurement stations: $x(a) = 10.8\ m$; $x(b) = 12.8\ m$; $x(c) = 13.8m$; $x(d) = 14.8\ m$; $x(e) = 16\ m$; $x(f) = 17.6\ m$.

Figures 2 (a) – (f) show the comparison between the numerical and experimental results of the temporal and spatial evolution of the waves at the measurement stations, indicated in Figure 1. From Figure 2 (a) it can be seen that on the offshore slope of the bar the waves hold sinusoidal. From Figures 2 (b) and (c) it can be noted the steepening of the incident waves above the horizontal bar part, due to the wave-structure interaction. Figures 2 (d), (e) and (f) show that the waves lose their vertical symmetry while propagating on the bar slope and toward the shoreline.

The overall numerical results are in very good agreement with the experimental data. The proposed model well represents the spatial evolution of the incident waves propagating over the longshore bar due to the wave-structure interaction by using only three vertical layers.

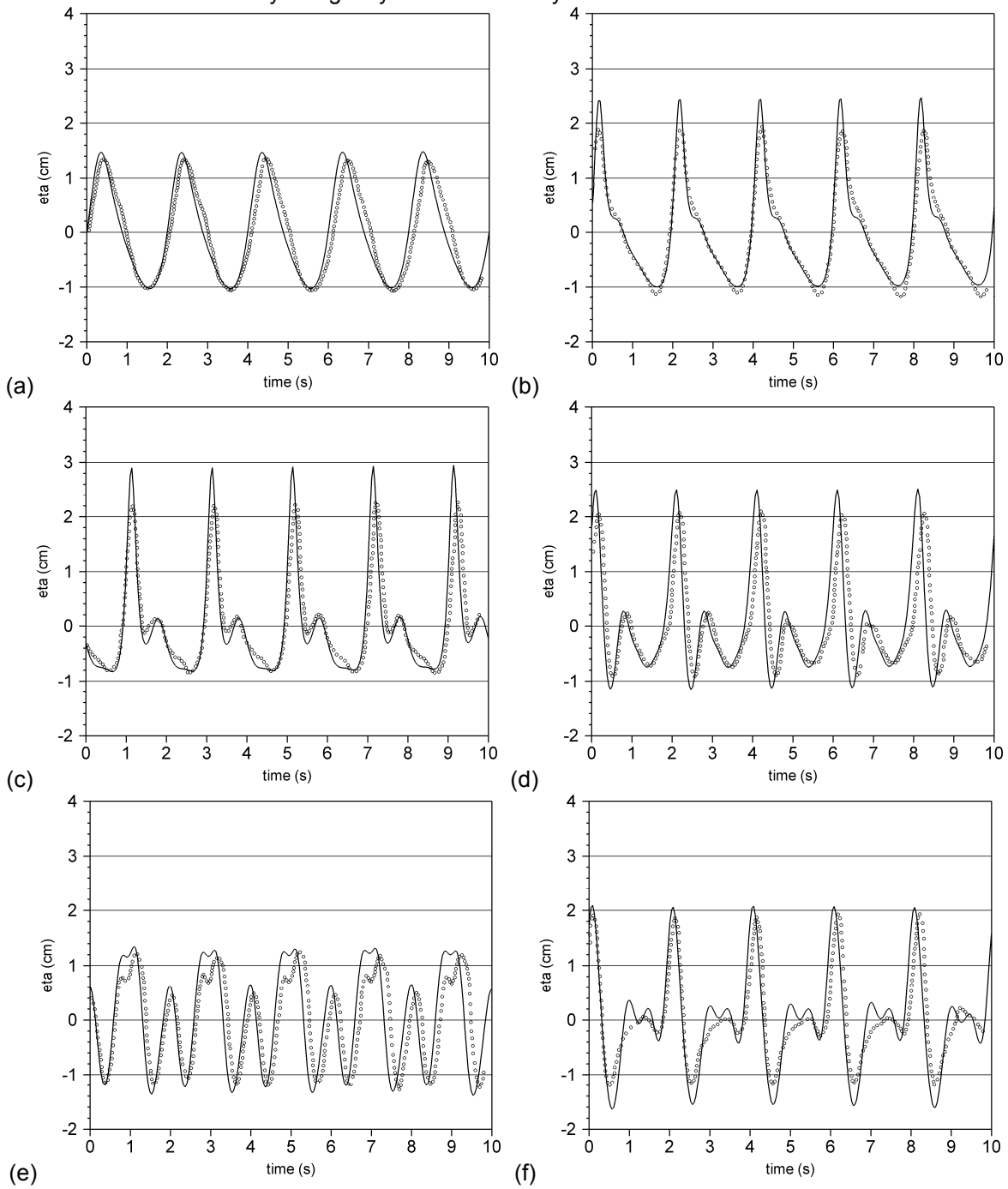


Figure 2. Spatial evolution of waves propagating over a longshore bar. Free surface elevation at the measurement stations: (a) $x(a) = 10.8\,m$; (b) $x(b) = 12.8\,m$; (c) $x(c) = 13.8\,m$; (d) $x(d) = 14.8\,m$; (e) $x(e) = 16\,m$; (f) $x(f) = 17.6\,m$. Solid line: numerical results. Circles: experimental data.

4.2 Wave propagation over an elliptic shoal.

The experimental test extracted from the "Report W. 154-VIII" of the Delft Hydraulics Laboratory and proposed by Berkhoff et al. (1982) is numerically reproduced in order to verify the ability of the proposed model to simulate the physical processes of wave propagation and wave deformation (refraction and diffraction) due to an elliptic shoal.

The experimental bottom physical domain is reproduced and shown in Figure 3. The bottom topography is characterized by a slope of $1{:}50$ above which an elliptic shoal is present. By defining the Cartesian coordinates (x, y) and (x', y') respectively related by a rotation of $-20°$, the bottom and elliptic obstacle equations are given by

$$H = 0.45 \quad if \quad y' \leq -5.484 \tag{7}$$

$$H = max\left(0.1; 0.45 - 0.02(5.484 + y')\right) \quad if \quad y' \geq -5.484 \tag{8}$$

$$\left(\frac{x'}{4}\right)^2 + \left(\frac{y}{3}\right)^2 = 1 \tag{9}$$

$$d = -0.3 + 0.5\sqrt{1 - \left(\frac{x'}{5}\right)^2 - \left(\frac{y'}{3.75}\right)^2} \tag{10}$$

where H is the water depth and d the elliptic obstacle thickness. It has to be noticed that the minimum depth is $10\ cm$: only non-breaking waves are considered.

The wave train has an amplitude of $0.0464\ m$ and a period of $1s$. The numerical simulation spatial and time discretization steps are $\Delta x = \Delta y = 0.05\ m$ and $\Delta t = 0.005\ s$. Waves are numerically generated on the eastern boundary of the computational domain while, on the western boundary, a sponge zone is placed in order to absorb the wave motion energy. Reflective conditions are set for the lateral boundaries. Lastly, four vertical layers are used in the vertical direction.

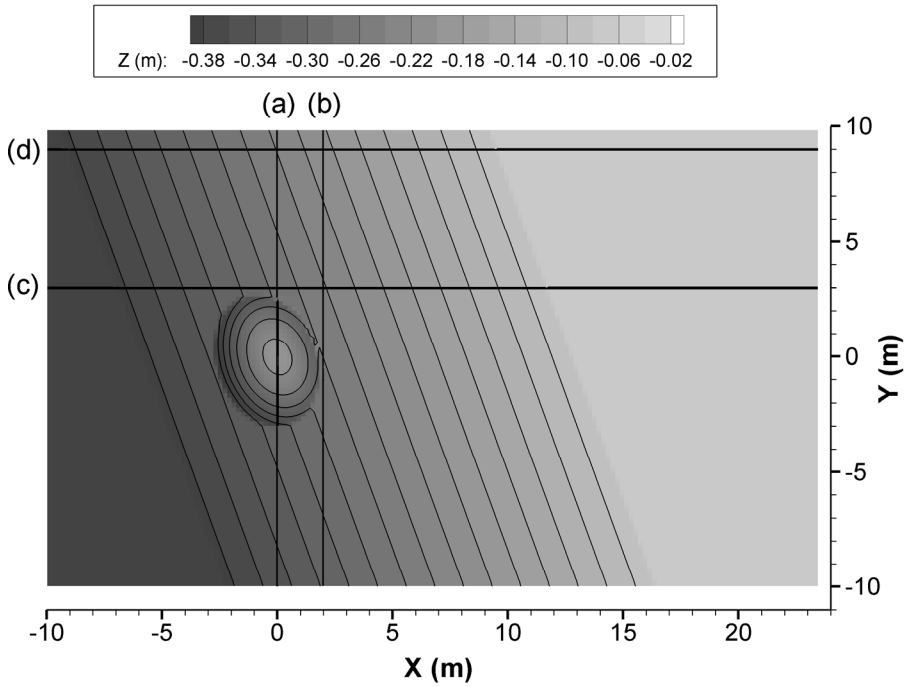


Figure 3. Wave propagation over an elliptic shoal. Bottom topography and section traces: (a) $x = 0\ m$; (b) $x = 2\ m$; (c) $y = 3\ m$ and (d) $y = 9\ m$.

Figures 4 (a) – (d) show the comparison between the numerical results and the measured data from Berkhoff et al. (1982) of the wave heights at sections $x = 0\ m$, $x = 2\ m$, $y = 3\ m$ and $y = 9\ m$ indicated in Figure 3. The computed wave height is obtained as the free surface elevation maximum and minimum difference, over a time interval in which the wave form is permanent. From Figure 4 (a) it can be seen an increase of the wave height, due to the shoaling phenomenon induced by the top of the obstacle. From Figure 4 (b) it can be seen a decrease of the wave height, due to the diffraction phenomenon induced by the obstacle boundaries. From Figures 4 (c) and (d) it can be noticed a loss of symmetry and periodicity of the wave behind the elliptic obstacle due to diffraction and refraction phenomena.

From these results, it is possible to deduce how the bottom elliptic obstacle influences and modifies the wave motion by means of diffraction, shoaling and refraction phenomena. The presence of such phenomena can also be deduced by observing Figure 5, where an instantaneous free-surface wave field is shown.

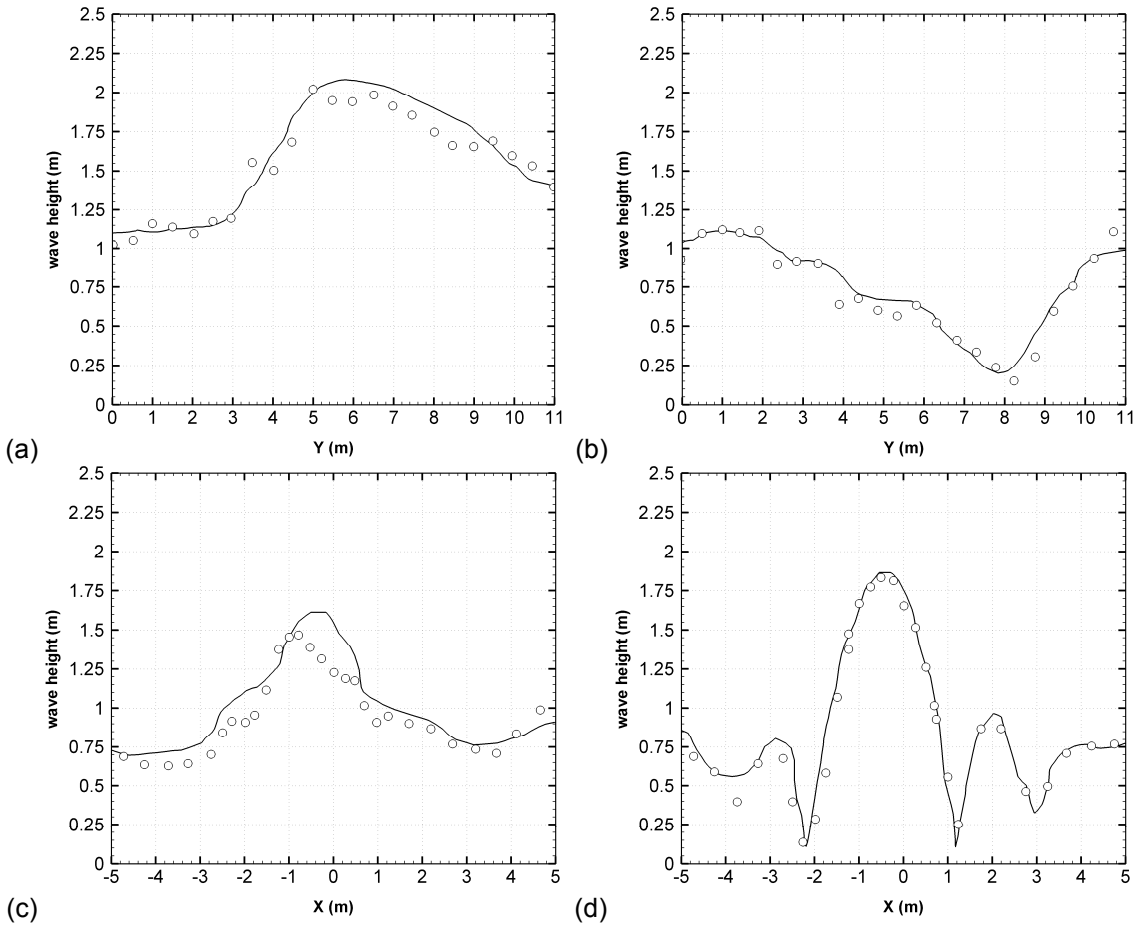


Figure 4. Wave propagation over an elliptic shoal. Wave height along sections: (a) $x = 0\ m$; (b) $x = 2\ m$; (c) $y = 3\ m$ and (d) $y = 9\ m$. Solid line: numerical results. Circles: experimental data.

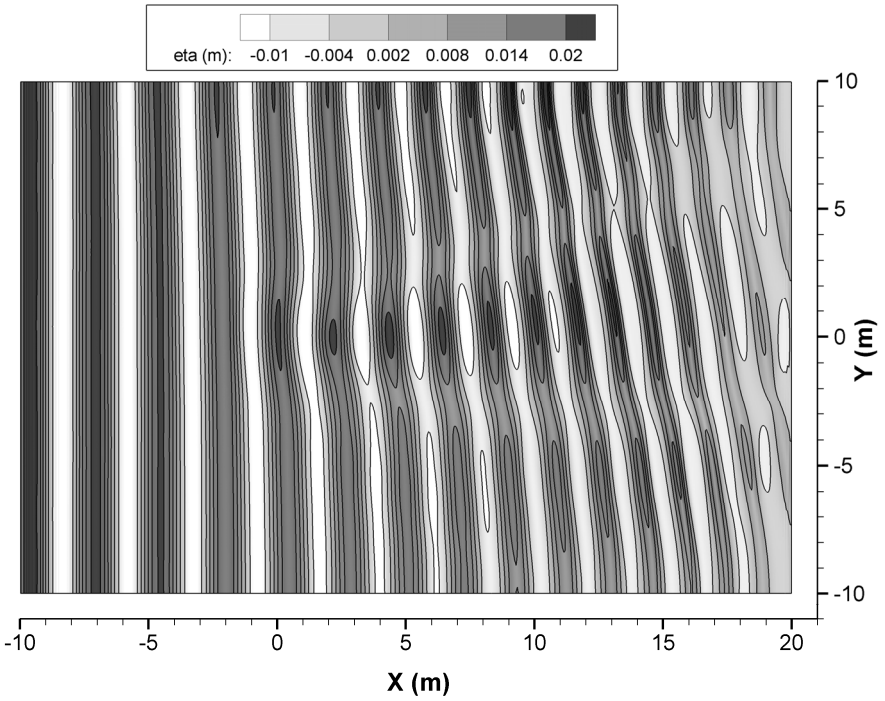


Figure 5. Wave propagation over an elliptic shoal. Instantaneous free surface elevation.

5 CONCLUSIONS

A model for non-hydrostatic free-surface flows based on an integral formulation of the Navier-Stokes equations in a time-dependent system of coordinates is proposed. The equations of motion are numerically integrated by a combined finite-difference/finite-volume scheme. The time integration is performed by a third order strong-stability preserving Runge-Kutta method. A fractional-step, pressure-correction method is used at each stage in order to produce a diverge-free non-hydrostatic velocity field. The intermediate hydrostatic velocity field is calculated by a shock-capturing upwind WENO scheme and an HLL Riemann solver. The pressure Poisson equation used in the corrector step is integrated by a four-colour Zebra line Gauss-Siedel alternate method and by adopting a multigrid strategy.

REFERENCES

Beji, S. & Batties, J.A. (1993). Experimental Investigation of Wave Propagation Over a Bar. *Coastal Engineering*, 19, 151-162.

Berkhoff, J.C.W., Booy, N. & Radder, A.C. (1982) Verification of Numerical Wave Propagation Models for Simple Harmonic Linear Water Waves. *Coastal Engineering*, 6, 255-279.

Cioffi, F. & Gallerano, F. (2006). From Rooted to Floating Vegetal Species in Lagoons as A Consequence of The Increases of External Nutrient Load: An Analysis by Model of The Species Selection Mechanism. *Applied Mathematical Modelling*, 30(1), 10-37.

Gallerano, F. & Cannata, G. (2011). Central WENO Scheme for The Integral Form of Contravariant Shallow-Water Equations. *International Journal of Numerical Methods in Fluids*, 67, 939-959.

Gallerano, F., Cannata, G. & Tamburrino, M. (2012). Upwind WENO Scheme for Shallow Water Equations in Contravariant Formulation. *Computers & Fluids*, 62, 1-12.

Gallerano, F., Cannata, G. & Villani, M. (2014). An Integral Contravariant Formulation of The Fully Non-Linear Boussinesq Equations. *Coastal Engineering*, 83, 119-136.

Gallerano, F., Cannata, G. & Lasaponara, F. (2016). A New Numerical Model for Simulations of Wave Transformation, Breaking and Longshore Currents in Complex Coastal Regions. *International Journal of Numerical Methods in Fluids*, 80, 571-613.

Gottlieb, S., Ketcheson, D.I. & Shu, C.W. (2009). High Order Strong Stability Preserving Time Discretization. *Journal of Scientific Computing*, 38, 251-289.

Harten, A., Lax, P. & Van Leer, B. (1983). On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Review*, 25-35.

Oka, S. & Ishihara, T. (2009). Numerical Study of Aerodynamic Characteristics of a Square Prism in a Uniform Flow. *Journal of Wind Engineering and Industrial Aerodynamics*, 97, 548-559.

Stelling, G. & Zijlema, M. (2003). An Accurate and Efficient Finite-Difference Algorithm for Non-Hydrostatic Free-Surface Flow with Application to Wave Propagation. *International Journal of Numerical Methods in Fluids*, 43, 1-23.

Toro, E. (2001). *Shock-Capturing Methods for Free-Surface Shallow Flows.* Ed. John Wiley and Sons, Manchester.

Trottemberg, U., Oosterlee, C.W. & Schuller, A. (2001). *Multigrid*. Academic Press, New York.

Young, C.-C. & Wu, C.-H. (2010). A σ-coordinate Non-Hydrostatic Model with Embedded Boussinesq-Type-Like Equations for Modeling Deep-Water Waves. *International Journal of Numerical Methods in Fluids*, 63, 1448-1470

Yuan, H. & Wu, C.-H. (2004a). A Two-Dimensional Vertical Non-Hydrostatic Sigma Model with An Implicit Method for Free-Surface Flows. *International Journal of Numerical Methods in Fluids*, 44, 811-835.

Yuan, H. & Wu, C.-H. (2004b). An Implicit Three-Dimensional Fully Non-Hydrostatic Model for Free-Surface Flows. *International Journal of Numerical Methods in Fluids*, 46, 709-733.

LATTICE BOLTZMANN SIMULATION OF SHALLOW WATER EQUATIONS IN CHARACTERISTIC GALERKIN FINITE ELEMENT FRAMEWORK

HAEGYUN LEE[(1)] & NAMJOO LEE[(2)]

[(1)]Dankook University, Yongin, Republic of Korea,
haegyun@dankook.ac.kr
[(2)]Kyungsung University, Pusan, Republic of Korea,
njlee@ks.ac.kr

ABSTRACT

The characteristic Galerkin finite element method for the discrete Boltzmann equation is developed for the numerical simulation of depth-integrated shallow water equations. Due to the inherent geometric flexibility of the unstructured mesh in finite element method, the mesh clustering near large gradients and the numerical treatments of complex geometries become possible. It is believed that the characteristic Galerkin procedure with appropriate boundary conditions results in accurate solutions with little numerical diffusion. The bend flow is simulated and the numerical results are in good agreement with previous experimental results.

Keywords: Lattice Boltzmann method; shallow water equations; characteristic Galerkin method; unstructured mesh; bend flow.

1 INTRODUCTION

The lattice Boltzmann method (hereafter, LBM) simulates microscopic collision and streaming of fluid particles and evaluates the macroscopic velocity, pressure, etc. from the computed particle distribution. Since LBM uses a regular lattice, it seems difficult to apply the LBM to the complex flow domain. As a solution to this issue, some methods applicable to irregular lattice have been proposed recent years. These can be classified into two groups. The first category is the discretization-based LBM with conventional discretization method (e.g., FDM, FVM, FEM, etc.). The second is the interpolation-supplemented LBM (Tateishi and Kashiyama, 2004). The latter is very effective in applications. However, in the interpolation-based schemes, an accurate interpolation scheme is required for the interpolation of the particle distribution function in the streaming step. Therefore, the discretization-based scheme has become a popular choice.

The LBM known to be a discrete computational method based upon the lattice gas automata, which is a simplified, fictitious molecular model. This consists of three basic tasks: lattice Boltzmann equation, lattice pattern and local equilibrium distribution function. The former two are standard, which is the same for fluid flows. The latter determines what flow equations are solved by the LBM model, which is often derived for certain flow equations such as the equations for incompressible Navier-Stokes equations or shallow water equations.

In order to solve the LBE in complex geometries while preserving the advantages of the conventional LBM such as data locality and little numerical diffusion, we employed the strategy of Lee and Lin (2001), which adopted the characteristic Galerkin finite element method (Zienkiewicz and Codina, 1995; Lin et al., 2005) for solving the discrete Boltzmann equation. In this study, the application of LBM to shallow water equations is briefly introduced and the flow in the bend is simulated.

2 LBM EQUATIONS AND FINITE ELEMENT FORMULATION

The equations for particle distribution are briefly described. The boundary conditions and turbulence modeling are also discussed.

2.1 LBM equations

The kinetic equation for the distribution function (e.g., distribution of passive scalar), $f_\alpha(x_i,t)$ can be written as:

$$\frac{\partial f_\alpha}{\partial t} + e_{\alpha i}\frac{\partial f_\alpha}{\partial x_i} = \Omega_\alpha \qquad [1]$$

where α = 0 ~ 8 for D2Q9 model (Figure 1). The left-hand side terms represent the streaming process, where the distribution function streams (advects) along the lattice link with velocity $\mathbf{e}_\alpha$ (=$\Delta\mathbf{x}/\Delta$t). The right-hand side term, $\mathbf{\Omega}_\alpha$ represents the rate of change of distribution function in the collision process. BGK approximation for the collision operator can be approximated as,

$$\Omega_\alpha = -\frac{f_\alpha - f_\alpha^{eq}}{\lambda} + \frac{1}{6} e_{\alpha i} F_i \quad [2]$$

where λ represents a relaxation time toward the equilibrium distribution (f_α^{eq}) and F_i is the external force given by

$$F_i = -gh\frac{\partial z_b}{\partial x_i} + \frac{\tau_{wi}}{\rho} \quad [3]$$

where z_b is the bottom elevation from the reference level and τ_{wi} is the bottom friction (usually represented by Manning or Chezy equation) in x_i direction. This relaxation term, λ is related to the diffusion coefficient on the macroscopic scale. The equilibrium distribution function for the shallow water equation is given by (Zhou, 2004)

$$f_\alpha^{eq} = \begin{cases} h - \dfrac{5gh^2}{6} - \dfrac{2h}{3} u_i u_i & \text{when } \alpha = 0 \\ \dfrac{gh^2}{6} + \dfrac{h}{3} e_{\alpha i} u_i + \dfrac{h}{2} e_{\alpha i} e_{\alpha j} u_i u_j - \dfrac{h}{6} u_i u_i & \text{when } \alpha = 1,3,5,7 \\ \dfrac{gh^2}{24} + \dfrac{h}{12} e_{\alpha i} u_i + \dfrac{h}{8} e_{\alpha i} e_{\alpha j} u_i u_j - \dfrac{h}{24} u_i u_i & \text{when } \alpha = 2,4,6,8 \end{cases} \quad [4]$$

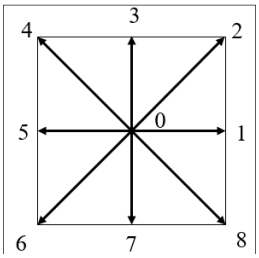


Figure 1. All possible velocities for the 9-bit lattice BGK model on square lattice.

The macroscopic variables, depth and depth-averaged velocity, are recovered by

$$h(\mathbf{x},t) = \sum_a f_a(\mathbf{x},t) \quad [5]$$

$$u(\mathbf{x},t) = \frac{1}{h(\mathbf{x},t)} \sum_\alpha e_{\alpha i} f_\alpha \quad [6]$$

2.2 Finite element formulations

In this study, we followed the approaches suggested by Lee and Lin (2001) for Navier-Stokes equations. The predictor-corrector method is expressed as follows:

(1) Predictor step

$$\begin{aligned} \hat{f}_\alpha - f_\alpha^n = & -\Delta t e_{\alpha r} \frac{\partial f_\alpha^n}{\partial x_r} - \frac{1}{\tau}\left(\hat{f}_\alpha - f_\alpha^{eq,n}\right) + \frac{\Delta t}{6} e_{\alpha i} F_i \\ & + \frac{(\Delta t)^2}{2} e_{\alpha s} \frac{\partial}{\partial x_s}\left(e_{\alpha r} \frac{\partial f_\alpha^n}{\partial x_r} + \frac{1}{6} e_{\alpha i} F_i \right) \\ & + \frac{\Delta t}{2} e_{\alpha s} \frac{\partial}{\partial x_s}\left[\frac{1}{\tau}\left(f_\alpha^n - f_\alpha^{eq,n} \right) \right] \end{aligned} \quad [7]$$

where $\tau = \lambda / \Delta t$ is the (non-dimensional) relaxation time.

(2) Corrector step

$$f_\alpha^{n+1} - \hat{f}_\alpha = -\frac{1}{\tau}\left(f_\alpha^{eq,n} - \hat{f}_\alpha^{eq} \right) \quad [8]$$

The applications of characteristic-Galerkin formulation of the two-steps lead to the following formulations. First, The predictor step is

$$\begin{aligned}(1+1/\tau)\mathbf{M}\hat{\mathbf{f}}_\alpha &= \mathbf{M}\mathbf{f}_\alpha^n - \Delta t\mathbf{C}_\alpha\mathbf{f}_\alpha^n + \Delta t\mathbf{E}_\alpha\mathbf{s}_\alpha^n - (\Delta t)^2\mathbf{D}_\alpha\mathbf{f}_\alpha^n - \Delta t\mathbf{Q}_\alpha\mathbf{f}_\alpha^n \\ &+ (1/\tau)\mathbf{M}\mathbf{f}_\alpha^{eq,n} + \Delta t\mathbf{Q}_\alpha\mathbf{f}_\alpha^{eq,n} - (\Delta t)^2\mathbf{S}_\alpha\mathbf{s}_\alpha^n\end{aligned} \quad [9]$$

where

$$\mathbf{M} = \int_{\Omega_e} \mathbf{N}\mathbf{N}^T d\Omega \quad [10]$$

$$\mathbf{C}_\alpha = \int_{\Omega_e} \mathbf{N}e_{\alpha r}\frac{\partial \mathbf{N}^T}{\partial x_r} d\Omega \quad [11$$

$$\mathbf{D}_\alpha = \frac{1}{2}\int_{\Omega_e}\frac{\partial \mathbf{N}}{\partial x_s}e_{\alpha s}e_{\alpha r}\frac{\partial \mathbf{N}^T}{\partial x_s}d\Omega - \frac{1}{2}\int_{\Gamma_e}\mathbf{N}n_s e_{\alpha s}e_{\alpha r}\frac{\partial \mathbf{N}^T}{\partial x_r}d\Gamma \quad [12]$$

$$\mathbf{Q}_\alpha = -\frac{1}{2\tau}\int_{\Omega_e}\mathbf{N}e_{\alpha r}\frac{\partial \mathbf{N}^T}{\partial x_r}d\Omega \quad [13]$$

$$\mathbf{E}_\alpha = \int_{\Omega_e}\mathbf{N}d\Omega \quad [14]$$

$$\mathbf{S}_\alpha = \frac{1}{2}\int_{\Omega_e}\mathbf{N}e_{\alpha r}\frac{\partial \mathbf{N}^T}{\partial x_r}d\Omega \quad [15]$$

$$\mathbf{s}_\alpha = \frac{1}{6}\mathbf{e}_{\alpha i}\cdot\mathbf{F}_i \quad [16]$$

where Γ_e denotes the boundary of each element and n_s is the s-direction component of the unit outward normal vector on Γ_e of the element. For the above spatial discretization, $\mathbf{D}_\alpha$ and $\mathbf{Q}_\alpha$ are needed for discretization along the characteristics and contribute to the stabilization of Eq. [9] (Lee and Lin, 2001). The fourth term on the right-hand side of Eq. [9] is the surface integral, which is needed on the domain boundary since they cancel out in the interior of the domain Ω. Lee and Lin (2001) cautioned that, with none of these terms, the scheme is likely to be unstable. Secondly, the corrector step is

$$\mathbf{f}_\alpha^{n+1} = \hat{\mathbf{f}}_\alpha - (1/\tau)\left(\mathbf{f}_\alpha^{eq,n} - \hat{\mathbf{f}}_\alpha^{eq}\right) \quad [17]$$

2.3 Turbulence modeling

Modeling of turbulence in LBM is executed by redefining the new (non-dimensional) relaxation time, τ_t. Zhou (2004) proposed the following form

$$\tau_t = \frac{1}{2}\left[\tau + \sqrt{\tau^2 + 18\sqrt{\Pi_{ij}\Pi_{ij}}\,C_s^2/\left(e^2 h\right)}\right] \quad [18]$$

where C_s is the Smagorinsky constant and Π_{ij} is defined as

$$\Pi_{ij} = \sum_\alpha e_{\alpha i}e_{\alpha j}\left(f - f_\alpha^{eq}\right) \quad [19]$$

The new relaxation time, τ_t, replaces τ in Eq. [9] and [17].

2.4 Boundary conditions

For the LBM shallow water equation model in this study, neither Dirichlet nor natural boundary conditions seem to appropriately work since usually boundary conditions are given for macroscopic variables, such as

fluid velocities, pressure, or their gradients, but never for particle distribution functions, f_α. In the solution procedure described by Eq. [9] and [17], only the streaming step requires boundary conditions since there is no spatial derivative of fa in the two collision steps (Lee and Lin, 2001). The streaming step given by Eq. [7] is hyperbolic and therefore, boundary conditions must be provided if $\mathbf{e}_\alpha \cdot \mathbf{n} < 0$ where $\mathbf{n}$ is a unit outward normal vector at the local boundary surface. On the other hand, if $\mathbf{e}_\alpha \cdot \mathbf{n} > 0$, boundary conditions are not necessary. The preferred choice is the bounce-back rule of the nonequilibrium distribution proposed by He et al. (1998), who suggested

$$\left(f_\alpha - f_\alpha^{eq}\right) - \left(f_\beta - f_\beta^{eq}\right) = 0 \qquad [20]$$

where $\mathbf{e}_\beta = -\mathbf{e}_\alpha$. In this study, the approach of Mei and Shyy (1998) was used, which regards f_α^{n+1} at the domain boundary as part of the solution and imposes macroscopic physical boundary conditions through f_α^{eq}. The free-slip boundary condition also worked reasonably well with this approach.

3 APPLICATION TO BEND FLOW

The case of 180° bend curved channel flow (Run No. 8 of Rozovskii (1957)) is selected for the validation of the developed scheme. A flow in a strongly curved channel is believed to represent one of the most complex and challenging flows encountered in a natural meandering river. As shown in Figure 2, the channel width is 0.8 m; the internal radius is 0.4 m and there is no bed slope in the channel. The flow conditions are: (1) flow discharge is 0.0123 m^3/s (2) the entrance depth is 0.063 m; and (3) the channel bed is rough with Chezy coefficient, C_z = 32 $m^{1/2}/s$. At the upstream boundary, the gradient of depth in flow direction is set to zero; velocity u is accordingly adjusted to retain the constant discharge and v = 0. As the downstream boundary condition, the depth is specified as 0.05 m. A steady-state solution was reached after 2000 iterations with Δt= 0.005 s. The Smagorinsky constant, C_s, is set to 0.25.

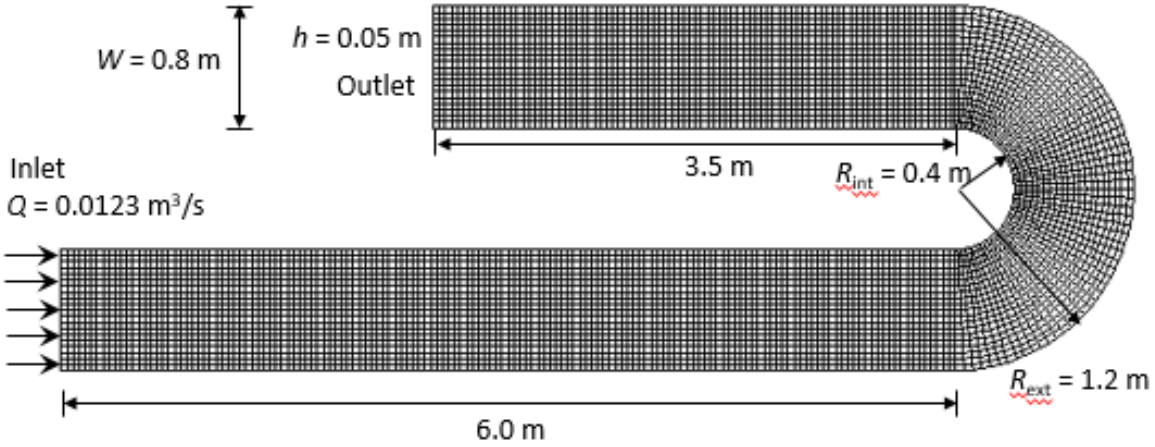
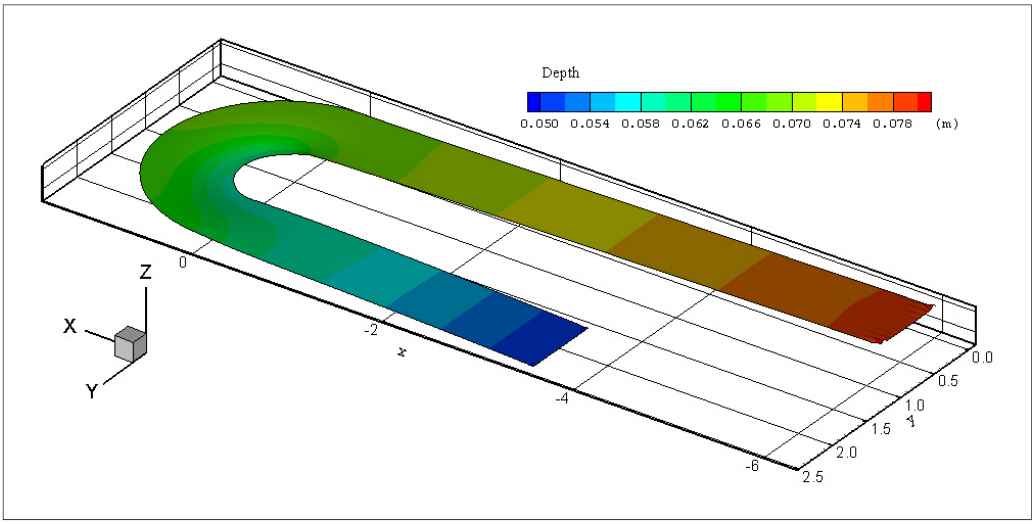
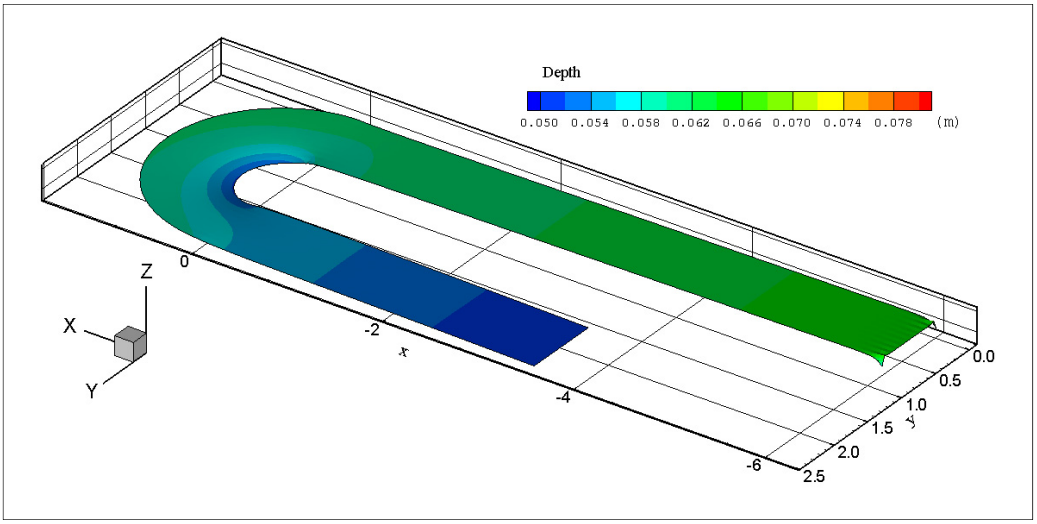


Figure 2. Schematic description of 180° curved channel and finite element mesh.

Figure 3 shows the surface plot of simulations with no-slip boundary condition for the side wall and free-slip boundary condition for the wall. Figure 4 is the comparison of streamwise velocities at 9 sections. As expected the simulation with free-slip boundary condition gives much better agreement with the experimental measurements of Rozovskii (1957). A little-bit-poor agreement at section of S = 8.0 m manifests the difficulty of curved channel flow due to the secondary flow.



(a) No-slip condition on the wall boundary



(b) Free-slip condition on the wall boundary

Figure 3. Plot of free surface in the 180^{o} bend channel.

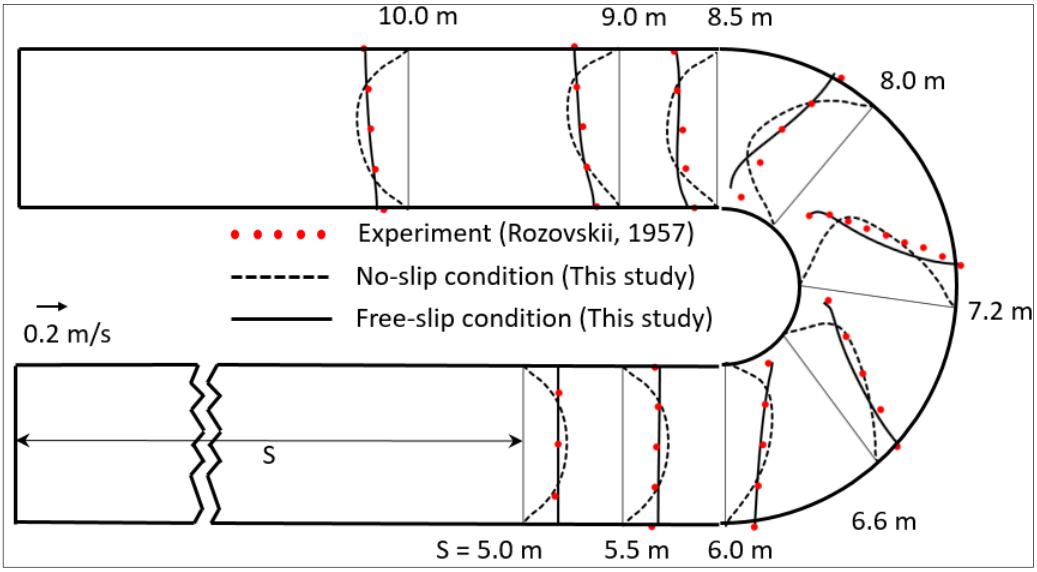


Figure 4. Comparisons of the tangential velocities (S is the distance between each cross-section and the entrance along the channel central line).

4 CONCLUSIONS

The characteristic Galerkin finite element method has been successfully applied to solve the discrete Boltzmann equation for shallow water equations. Due to the inherent geometrical flexibility of the finite element method, flows in complex geometries can be easily simulated. In addition, the use of unstructured meshes increases the numerical accuracy while reducing the computational cost. In order to circumvent the stability limit arising from the relaxation term, the predictor-corrector method of Lee and Lin (2001) is employed. The curved channel flow is tested with the developed model and favorable results were obtained, confirming the applicability of the proposed method.

Less satisfactory agreement at one section is believed to be the secondary motion at the bend and VAM-like model (Song et al., 2012) can be considered as a future study. The noted strength of the LBM is that, by defining the equilibrium function, it can represent other equations with ease. The developed model can be combined with contaminant spreading or sediment transport.

ACKNOWLEDGEMENTS

This research was supported by a grant (12-TI-C02) from Advanced Water Management Research Program funded by Ministry of Land, Infrastructure and Transport of Korean government.

REFERENCES

He, S., Chen, S. & Doolen, G.D. (1998). A Novel Thermal Model for the Lattice Boltzmann Method in Incompressible Limit. *Journal of Computational Physics*, 146, 282-300.

Lee, T. & Lin, C.-L. (2001). A characteristic Galerkin Method for Discrete Boltzmann Equation. *Journal of Computational Physics*, 171, 336-356.

Lin, C-L., Lee, H., Lee, T. & Weber, L.J. (1995). A Level Set Characteristic Galerkin Finite Element Method for Free Surface Flows. *International Journal of Numerical Methods in Fluids*, 49, 521-547.

Mei, R. & Shyy, W. (1998). On the Finite Difference based Lattice Boltzmann Method in Curvilinear Coordinates. *Journal of Computational Physics,* 143, 426-448.

Rozovskii, I.L. (1957). *Flow of Water in Bends of Open Channels.* Academy of Sciences of the Ukrainian SSR, 233.

Song, C.S., Seo, I.W. & Kim, Y.D. (2012). Analysis of Secondary current Effect in the Modeling of Shallow Flow in Open Channels, *Advances in Water Resources,* 41, 29-48.

Tateishi, J. & Kashiyama, K. (2004). A Lattice Boltzmann-CIVA Method for Incompressible Flow Simulation based on Unstructured Mesh. *Theoretical and Applied Mechanics Japan,* 53, 265-272.

Zhou, J.G. (2004). *Lattice Boltzmann Methods for Shallow Water Flows.* Springer, 112.

Zienkiewicz, O.C. & Codina, R. (1995). A General Algorithm for Compressible and Incompressible Flow - Part I. The Split, Characteristic based Scheme. *International Journal of Numerical Methods in Fluids*, 20, 869-885.

ROBUST PARAMETER SET SELECTION OF HYDRODYNAMIC MODEL FOR MULTI-SITES USING MINIMAX REGRET APPROACH)

LI LI(1), EUN SUNG CHUNG(2) & KYUNG SOO JUN(3

(1,3) Graduate School of Water Resources, Sungkyunkwan University, Suwon, Republic of Korea
lili0809@skku.edu
(2) Department of Civil Engineering, Seoul National University of Science and Technology, Seoul, Republic of Korea
eschung@seoultech.ac.kr

ABSTRACT

A robust parameter set (ROPS) selection method for a hydrodynamic flow model is developed by combining Pareto optimums obtained by outcomes of multi-site calibration problem based on the observations of multi-sites with the minimax regret approach (MRA). The multi-site calibration problem which is a multi-objective problem is solved by using an aggregation approach which aggregates the weighted criteria related to different sites into one measure, and then performs a large number of individual optimization runs with different weight combinations. Roughness parameter structure which can describe the variation of Manning's *n* with discharges and sub-reaches was developed and the related coefficients are optimized as model parameters by minimizing the sum of the squares of the weighted residuals between the computed and observed water levels at two gauging stations. Different combinations of weights on the observations at two sites are considered for the aggregation approach to solve multi-objective function, and the corresponding optimized parameter sets (Pareto optimums) are assumed as the ROPS candidates. One flood event is selected for calibration and another one is chosen for validation, where the two events differ significantly in their rainfall distributions. In the application of MRA, which is a decision criterion, the Pareto solutions are ranked based on the obtained regrets related to each Pareto solution, and the top-rated one due to the lowest aggregated regrets of both calibration and validation is determined as the only ROPS. It is found that the determination of variable roughness and the corresponding standardized RMSEs at the two gauging stations vary considerably depending on the combinations of weights on the two sites. This method can provide the robust parameter set for the multi-site calibration problems in hydrologic and hydraulic models.

Keywords: Minimax regret approach; pareto optimum; robust parameter set selection; unsteady flow model; variable roughness coefficients.

1 INTRODUCTION

Flooding is one of the most severe disasters that can affect human lives and society, and the frequency and magnitude of severe floods have probably increased due to climate change. Numerous numerical models for flood simulation have been developed and successfully applied to many water resources engineering, flood risk assessment and planning of flood mitigation measures. Apart from the numerical scheme of flood routing models, the hydraulic roughness as a model parameter is one of the most important factors that influences the estimation accuracy of water levels and discharges in channels. Manning's roughness coefficient is a key parameter used in flood routing models based on the one-dimensional Saint-Venant equations, where it represents the flow resistance from different sources. It considers all aspects of resistance, including the skin friction related to the grain size of bed materials, bed form roughness, and channel resistance due to river bends and gradual changes in cross-sections. Transitions in the flow level and discharge also lead to variations in Manning's *n* (Coon, 1998; Rouse, 1965; Chow, 1959). Therefore, the temporal and spatial changes in Manning's *n* make its determination difficult. Many studies have identified Manning's roughness coefficients (Ishii, 2000; Ramesh et al, 2000; Atanov et al., 1999); however, most of these studies have only dealt with a lumped parameter model in which Manning's *n* is regarded as temporally and/or spatially constant.

Model parameters are typically determined by using either manual or automatic calibration. In contrast to hydrological modelling, where many sophisticated approaches of automatic calibration dominate these days, the common procedure for hydrodynamic models is a manual calibration to fit the observations (Dung et al., 2011). Because of the high computational demands of hydrodynamic model, particularly for two-dimensional flood inundation models, only a few studies have dealt with automatic calibration of hydrodynamic models.

Observations should be used for model calibration and validation to optimize Manning's roughness coefficient in a hydrodynamic flow model. Calibration of a distributed hydrodynamic model using multi-site data can derive better-defined model parameter estimates by constraining the calibration process (Hunter et al., 2005; Horritt, 2000; Seibert et al., 2000), which can be considered as a multi-site calibration using a multi-objective optimization framework. This calibration problem can be defined by several objective functions that measure errors at different

sites. Calibration with multiple observations by applying multiple objective functions or multiple performance criteria has been widely employed in hydrologic models (Saeidifarzad et al., 2014; Wang, 2012; Li et al., 2010; Khu et al., 2008; Khu and Madsen, 2005) but rarely in hydraulic models.

McCabe et al. (2005) noted that multi-objective calibration allows a set of Pareto optimal solutions, or Pareto approximate solutions, whereas a single objective model calibration only identifies a set of model parameters according to a single observation set. If two or more objective functions are included in the calibration, the number of Pareto optimal or near Pareto optimal parameter sets will increase exponentially. Therefore, this calibration problem can be changed into a decision-making problem that can select a set of preferred model parameters among a number of Pareto sets (Khu and Madsen, 2005).

The purpose of this study is to develop a ROPS selection method for a hydrodynamic flow model by combining a multi-objective calibration based on multi-site measurements with MRA. A crude approximation of the Pareto solution set was obtained by aggregating the weighted criteria related to different sites into one measure which is used in the optimization, and then by performing a number of individual optimization runs with 11 different combinations of weights on two sites in terms of one decimal place. A ROPS which is a compromise solution as it can provide a good overall performance was selected from the obtained Pareto solution set by using MRA. The minimax regret approach (MRA), which is a useful decision-making technique under complete uncertainty (Kim et al., 2015a; Kim and Chung, 2014), was used to select the ROPS for variable roughness of a hydrodynamic flow model by ranking all optimized Pareto optimal sets as shown in Kim et al. (2015b). The selected parameter set can be regarded as robust because the weights assigned to the multiple objectives are not subjective as many previous studies (Dung et al., 2011) but objective. It is demonstrated in this study that the MRA can be an analytical and objective approach to select the only proper parameter set among optimized solutions obtained with all available weight combinations. There has been no literature testing the MRA as a decision criterion for multi-site calibration problems in hydrology and hydraulics. The regrets from calibration and validation were aggregated to determine the only ROPS. Roughness parameter structure which can describe the variation of Manning's *n* with discharge and sub-reaches was proposed for a hydrodynamic flow model and the related coefficients were calibrated as model parameters.

2 CASE STUDY AND DATA

2.1 Study River

The river reached in this study, which is the 69 km between Paldang Dam and Jeonryu gauging station, is the main stream of the Han River in South Korea (Figure 1). The Han River, which is 494 km long and >1 km wide, is the fourth longest river on the Korean Peninsula. It is formed by the confluence of two major branches, namely the South and North Han Rivers, which come together at Yangpyeong in Gyeonggi Province. The Han River flows through Seoul, which is the capital of South Korea, and finally flows into the Yellow Sea. The water levels were measured at the Paldang Bridge, Banpo Bridge, Hangang Bridge, Heangju Bridge, and Jeonryu gauging stations (Figure 3). Two submerged weirs in the channel, namely Jamsil and Singok submerged weirs (Figure 1), are partly gated structures. The gates of movable weirs are fully opened during flood periods, thus two different flows occur at the fixed weir and the movable weir.

Figure 1. Study area: Han River.

2.2 Data preparation

Two flood events were selected from the monsoon season (June-August), as shown in Table 1. The first and second flood events were used for calibration and validation, respectively. The rainfall causing the first event showed spatially homogeneous distribution as the total precipitation at the two stations was similar,

whereas for the second event, the total rainfall in the upstream region of the river was similar to the average of the first event but it decreased by half in the downstream region of the river. The peak discharge at Paldang Dam in the first flood event was over 1.5 times that in the second flood event. Songjeong station is near the Jungrang Stream junction, and Gimpo station is near the Jeonryu gauging station (Figure 3).

Table 1. Descriptions of two events used in this study.

Event	Period (mm/dd/yyyy hh:mm)	Total precipitation (mm)			Peak time (mm/dd/yyyy hh:mm)	Peak runoff (m^3/s)
		Paldang	Songjeong	Gimpo		
1	07/24/2008 10:00 to 07/27/2008 04:00	124	101	111	07/25/2008 00:10	16,146
2	07/12/2004 10:00 to 07/15/2004 00:00	100	80	47	07/13/2008 10:00	9,909

The collected data include the discharges at Paldang Dam as an upstream boundary condition of the simulation model, discharges of tributary inflow, observed stages at two gauging stations (Paldang Bridge, and Banpo Bridge) for model calibration, observed data at Junryu gauging station as the downstream boundary for the model, and topographic data.

3 NUMERICAL FLOOD MODELLING

3.1 Hydrodynamic model

A looped-network unsteady flow model was developed and applied for the Han River. The governing equations comprise of link and node equations. Either fluvial or weir-type flow could occur at links, and these are described by different equations. The equations for fluvial links are as follows:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \quad [1]$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\alpha \frac{Q^2}{A}\right) + gA\frac{\partial y}{\partial x} + gA\frac{Q|Q|}{K^2} = 0 \quad [2]$$

where x and t are variables in space and time, respectively; $y(x, t)$ is the water surface elevation; $Q(x, t)$ is the discharge; $A(x, y)$ is the cross-sectional area of water; $K(x, y)$ is the conveyance; α is an energy correction factor, which is assumed to be unity; and g is the acceleration due to gravity. The conveyance is described as

$$K = \frac{1}{n}AR^{2/3} \quad [3]$$

where R is the hydraulic radius and n is Manning's roughness coefficient.

The governing equations for weir-type links and nodes can be found in Kim and Jun (2004).

The Preissmann's four-point implicit scheme was used to discretize the partial differential equations [1] and [2] (e.g., Cunge et al., 1980; Liggett and Cunge, 1975). The discretized equations combined with equations of weir-type links and nodes comprise of the nonlinear system that can be solved by the Newton-Raphson method using the looped double-sweep algorithm. Details of the algorithm were provided by Holly et al. (1990).

The distributed hydrodynamic model in this study includes 156 computational points with respectively possible different roughness parameter. To perform parameter optimization, it is required to reduce the number of adjustable parameters. This is achieved by developing variable parameter structure in this study.

In Eq. [3], Manning's n can be generally expressed as (Yen, 2002; Coon, 1998; Chow, 1959)

$$n = n\big(x, Q(x,t)\big) \quad [4]$$

where Manning's n at each computational point not only changes with the sub-reach to which it belongs, but also with the discharge at that point. In this study, a power function which includes two parameters (Eq. [5]) was adopted as a functional relationship between Manning's roughness coefficient and the resulting discharge (Figure 2). The whole reach is divided into two sub-reaches which are upstream and downstream of the Wangsook stream junction. This is because the upstream reach is a natural river in which the cross sections are highly irregular, whereas the downstream reach is a channelized river in which the cross sections are relatively regular. Therefore, two different power functions were considered in each sub-reach and a total of four parameters (α_1, α_2, β_1, and β_2) should be optimized for the whole reach.

$$n_i = \alpha_i Q^{\beta_i}, \quad i = 1, \ldots, I \quad [5]$$

where α_i and β_i are the coefficients of a power function in *i*th sub-reach and *i* is the total number of sub-reaches which is equal to two in this study.

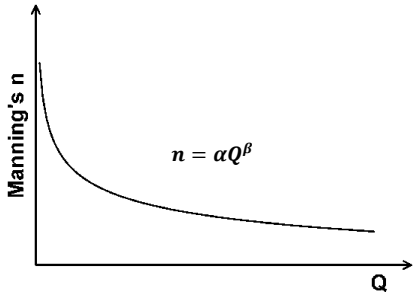


Figure 2. Relationship between Manning's *n* and discharge (power function).

3.2 Model set up

The model consisted of 10 nodes (one at either end of the river reach, an upstream/downstream pair at each of the two submerged weirs, and one at each of the four tributary-inflow junction points) and 11 links (Figure 3). As stated earlier, the submerged weirs at Jamsil and Singok were composed of a fixed and a movable weir; the gates of the movable weir were fully opened during flood periods. Therefore, channel and weir-type flows occurred separately on either side. To simulate these two flows, a fluvial and a weir-type link were used for the channel and weir-type flows, respectively, in the model. These two links will connect the upstream and downstream nodes of the submerged weir by forming a loop (Figure 3).

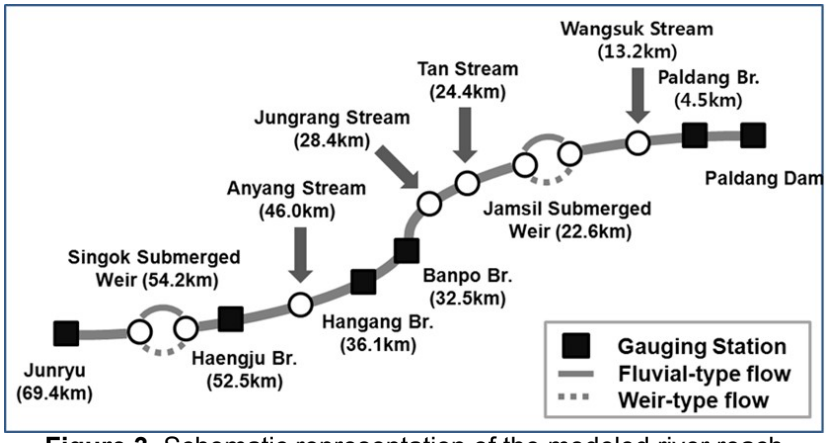


Figure 3. Schematic representation of the modeled river reach.

4 CALIBRATION FRAMEWORK

4.1 Procedure of ROPS selection

In this study, the aggregation approach was applied to solve the multi-site objective problem by aggregating performance indices from individual sites into one objective function. The main difficulty of using aggregation approaches is the need to assign reasonable weight to each objective function (Dung et al., 2011). The weights for different sites affect directly the ROPS selection and hence should be determined carefully. Shinma and Reis (2014) indicated that the equal-weighted approach is both simple and reasonable in most cases, but is not appropriate for multi-event and multi-site optimization problems. In this study, we used the MRA to select the ROPS by considering all combinations of available weights in terms of one decimal place. A flowchart illustrating the proposed methodology is presented in Figure 4.

The main steps of the methodology are as follows:

i. Formulating the multi-site optimization problem for the hydrodynamic flow model,
ii. Optimizing multi-site objective functions for calibration with available combinations of weights at two sites in terms of one decimal place,
iii. Computing regrets based on the performance measures of different Pareto optimums for two events of calibration and validation, and
iv. Deciding the ROPS based on the aggregated regrets of calibration and validation.

Before performing the MRA, the performance values should be linearly normalized so that the effect of different magnitudes of water stages of two events at two sites can be removed. The linear normalization technique applied in this study uses the maximum and minimum feasible values to ensure that the performance values range from 0 to 1.

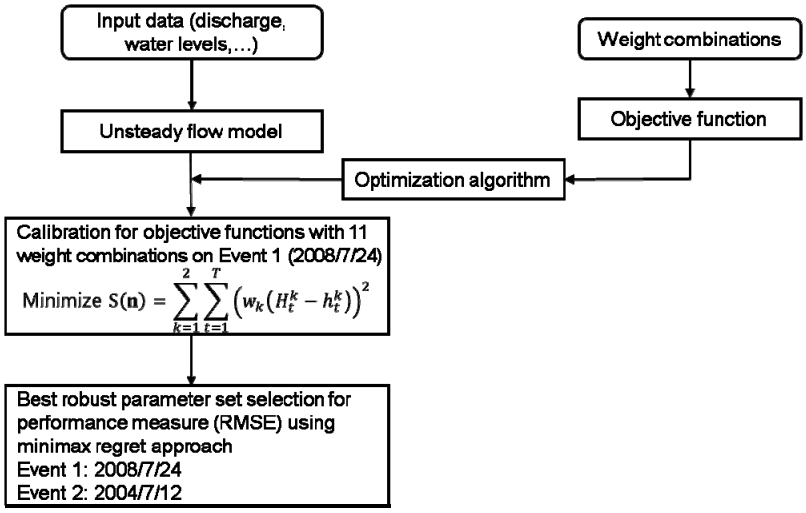


Figure 4. Flowchart illustrating the proposed methodology.

4.2 Multi-site objective function

The general multi-site objective function can be expressed as

$$\min\{F_1(\boldsymbol{\theta}),\ F_2(\boldsymbol{\theta}), \ldots, F_n(\boldsymbol{\theta})\} \quad \boldsymbol{\theta} \in \boldsymbol{\Theta} \qquad [6]$$

where $F_i(\boldsymbol{\theta}), i = 1, 2, \ldots, n$ are the n number of different objective functions that needs to simultaneously be minimized with respect to the parameter vector $\boldsymbol{\theta}$, and $\boldsymbol{\Theta}$ is the feasible parameter space.
The objective of model calibration aims to obtain optimal roughness parameters ($\mathbf{n}$) by minimizing the sum of squares for the weighted residuals between the computed and observed water levels at three sites, thus Eq. [6] can be specified as

$$\text{Minimize } S(\mathbf{n}) = \sum_{k=1}^{2}\sum_{t=1}^{T}\left(w_k(H_t^k - h_t^k)\right)^2 \qquad [7]$$

where H_t^k and h_t^k are the observed and computed water levels, respectively, at the kth station at time level t; the superscripts, k = 1, and 2 denote Paldang Bridge, and Banpo Bridge, respectively; $\mathbf{n}$ is a parameter vector that represents a set of four coefficients of two power functions; and w_k is the weight for the kth site ($w_1 + w_2 = 1$). In total, 11 combinations of weights in terms of one decimal place (1.0, and 0.0; 0.9, and 0.1; … 0.0, and 1.0) were considered in this study.

The Gauss-Marquardt-Levenberg optimization algorithm was used to find the optimal model parameters. This algorithm can converge toward the optimal solution more efficiently and faster by combining the advantages of the inverse Hessian method and the steepest descent method (Liu et al., 2005).

4.3 Minimax regret approach

Many studies have noted that the MRA is a reliable criterion that can be used to make reasonable decisions when the likelihoods of possible outcomes a not clearly known and it is not possible to use the expected utility criteria or the classical expected value (Chung and Kim, 2014; Loulou and Kanudia, 1999).

For a cost minimization problem, the regret $R(\boldsymbol{\theta})$ is the difference between the cost with the parameter set $\boldsymbol{\theta}$ and the least cost at the kth site, as follows:

$$R(\boldsymbol{\theta}) = C(\boldsymbol{\theta}) - \min C(\boldsymbol{\theta}) \quad \boldsymbol{\theta} \in \boldsymbol{\Theta} \qquad [8]$$

where $C(\boldsymbol{\theta})$ is the performance values when the parameter set $\boldsymbol{\theta}$ is considered. $C(\boldsymbol{\theta})$ was described by the normalized RMSE in this study.

The minimax regret is

$$\text{MMR} = \min\{\max R(\boldsymbol{\theta})\} \qquad [9]$$

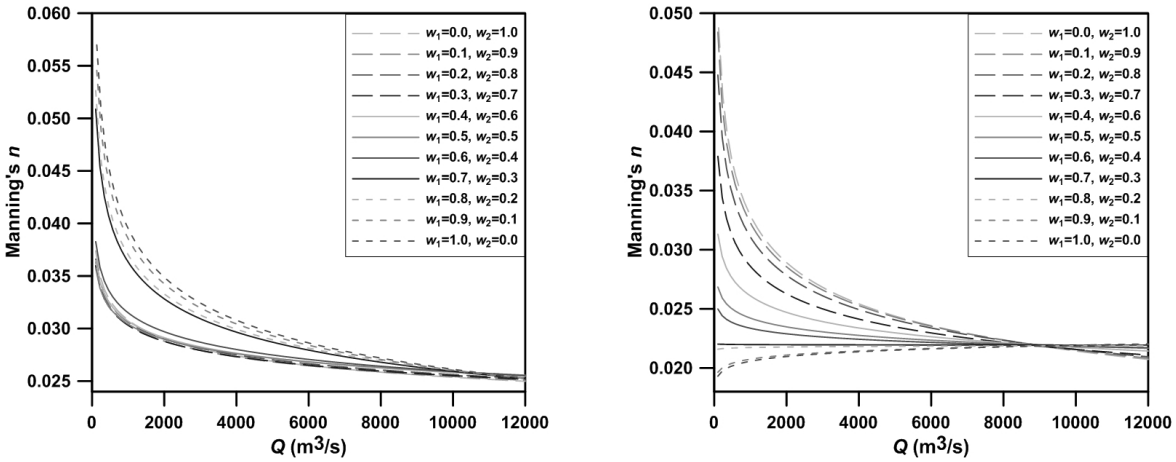
The corresponding parameter set for the minimax regret is

$$\boldsymbol{\theta}^* \in \text{Argmin}\{\max R(\boldsymbol{\theta})\} \qquad [10]$$

5 RESULTS AND DISCUSSION

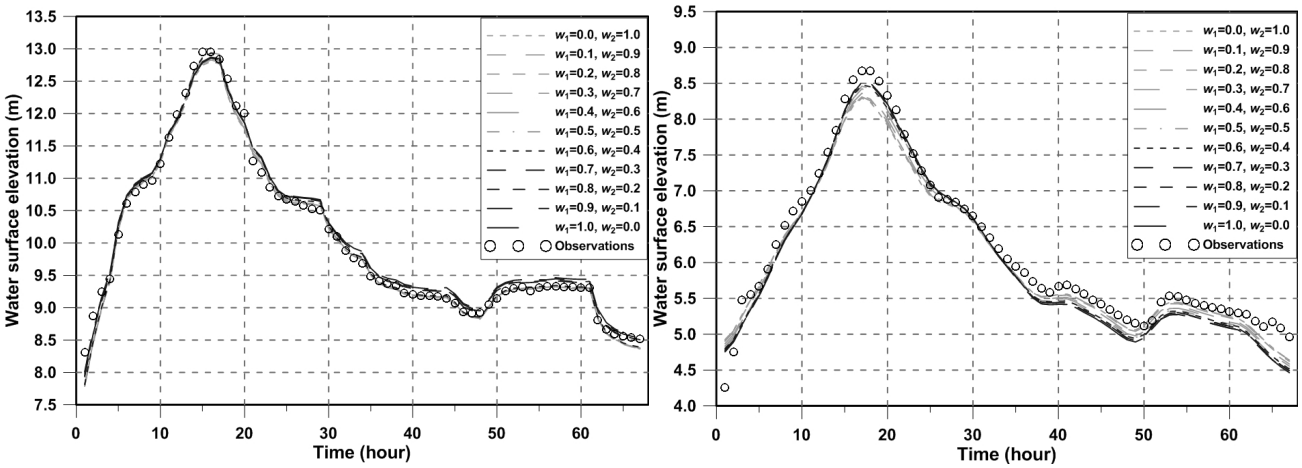
5.1 Calibration and validation of ROPS candidates

Multi-site objective functions with 11 combinations of weights were optimized for calibration. Four coefficients of two power functions that describe the functional relationship between Manning's *n* and discharge were optimally estimated for both the upstream and the downstream reaches of the Wangsook stream junction. Figure 5 shows the relationships determined by considering 11 combinations of weights at two sites. The results showed that Manning's *n* decreases as discharge increases in both reaches in most cases, and the Manning's *n* in the upstream reach is greater than that in the downstream reach when the values of discharge is larger than 2000 m^3/s.



(a) (b)

Figure 5. Estimated relationships between Manning's *n* and discharge: (a) upstream reach; (b) downstream reach.

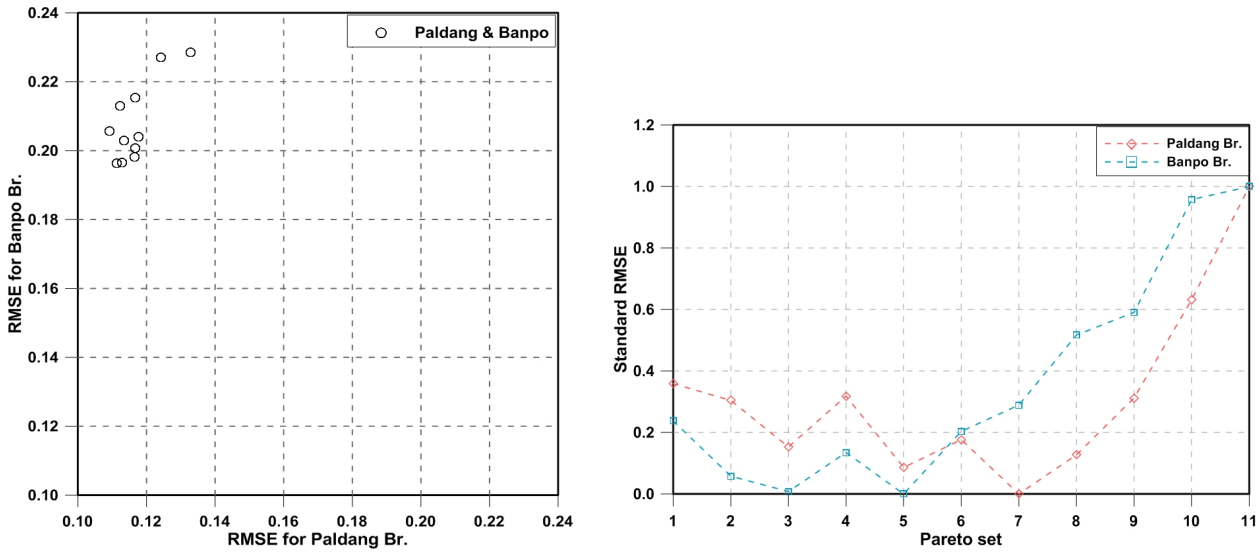


(a) (b)

Figure 6. Calibrated hydrographs (lines) using 11 optimized parameter sets: (a) Paldang Bridge; (b) Banpo Bridge.

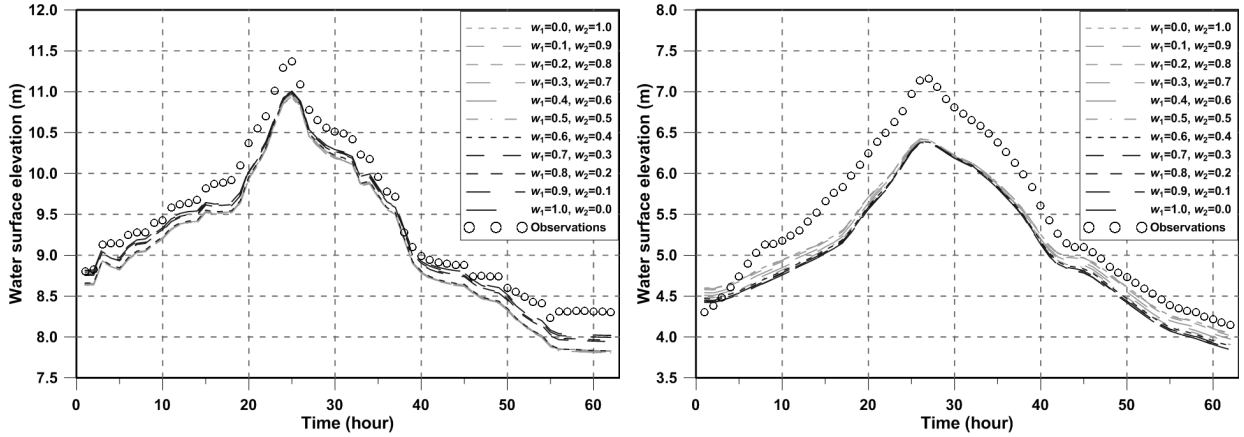
For the calibration, Figure 6 shows good overall agreement between the observations (black circles) and the hydrographs computed with the 11 weight combinations. According to the RMSEs in Figure 7(a), most values at Banpo Bridge were greater than that at the other site. The estimated coefficients of power functions from the calibration were used for the second flood event as a validation procedure.

The Figure 8 shows that the overall difference between observed (black circles) and computed water levels produced from validation is greater than that from calibration, especially for the Banpo Bridge. The RMSEs at Banpo Bridge produced during validation were obviously greater than those at the other site, and the standardized RMSEs at both two sites varied significantly with different combinations of the weights on two sites (Figure 9). The standardized sums of squares of the weighted residuals for multi-site calibration and validation are shown in Figure 10.



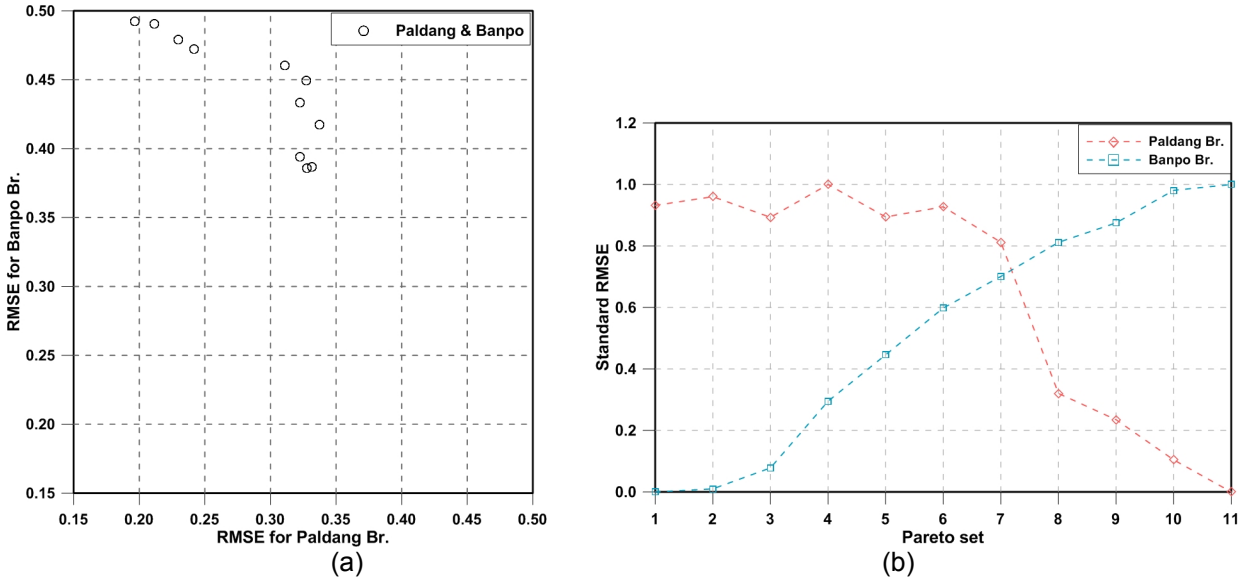
(a) (b)

Figure 7. RMSEs for two sites with 11 parameter sets.
(a) scatter plot of RMSEs; (b) standardized RMSEs.



(a) (b)

Figure 8. Validated hydrographs (lines) with 11 optimized parameter sets:
(a) Paldang Bridge; (b) Banpo Bridge.



(a) (b)

Figure 9. RMSEs for two sites with 11 parameter sets.
(a) scatter plot of RMSEs; (b) standardized RMSEs.

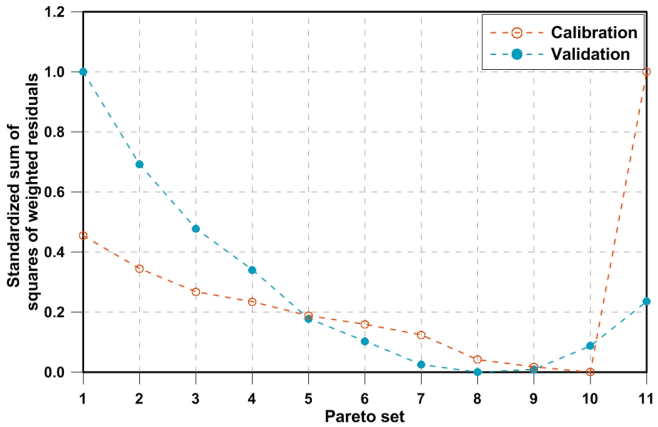


Figure 10. Standardized sum of squares of weighted residuals from calibration and validation.

The rankings of the combinations of weights are summarized in Tables 2 and 3. Table 2 lists the results obtained from the calibration, and Table 3 presents those from the validation. The best combinations of weights based on the MRA were (0.4, 0.6) and (0.7, 0.3) for the first and second flood events, respectively.

Table 2. Robust ranking based on the minimax regret approach (MRA) for the calibration.

Combinations of weights		Calibration	
w_1	w_2	Regret	Rank
0.4	0.6	0.086	1
0.2	0.8	0.153	2
0.5	0.5	0.204	3
0.6	0.4	0.289	4
0.1	0.9	0.304	5
0.3	0.7	0.318	6
0.0	1.0	0.359	7
0.7	0.3	0.517	8
0.8	0.2	0.591	9
0.9	0.1	0.957	10
1.0	0.0	1.000	11

Table 3. Robust ranking based on the MRA for the validation.

Combinations of weights		Validation	
w_1	w_2	Regret	Rank
0.7	0.3	0.812	1
0.6	0.4	0.812	2
0.8	0.2	0.875	3
0.2	0.8	0.893	4
0.4	0.6	0.894	5
0.5	0.5	0.928	6
0.0	1.0	0.932	7
0.1	0.9	0.960	8
0.9	0.1	0.981	9
0.3	0.7	1.000	10
1.0	0.0	1.000	11

5.2 Selection of ROPS

In this study, both the calibration and the validation results were used to select the ROPS using MRA. As listed in Tables 2 and 3, most of the weight combinations and their rankings obtained from the calibration were different from those produced by the validation. Table 4 lists the rank of parameter sets when the two regrets obtained from calibration and validation were aggregated into a composite index by averaging. It can be found from the table that the parameter set with weights on two sites, (0.4, 0.6) which ranked the first in the calibration results was also the first in the final ranking, and that with weights on two sites, (0.7, 0.3) which was the first in the validation ranked the 8th in the final ranking.

Table 4. Robust ranking based on the aggregated regrets obtained from the calibration and validation.

Combinations of weights		Calibration		Validation		Aggregated	Final
w_1	w_2	Regret	Rank	Regret	Rank	regret	rank
0.4	0.6	0.086	1	0.894	5	0.490	1
0.2	0.8	0.153	2	0.893	4	0.523	2
0.6	0.4	0.289	4	0.812	2	0.551	3
0.5	0.5	0.204	3	0.928	6	0.566	4
0.1	0.9	0.304	5	0.960	8	0.632	5
0.0	1.0	0.359	7	0.932	7	0.646	6
0.3	0.7	0.318	6	1.000	10	0.659	7
0.7	0.3	0.517	8	0.812	1	0.664	8
0.8	0.2	0.591	9	0.875	3	0.733	9
0.9	0.1	0.957	10	0.981	9	0.969	10
1.0	0.0	1.000	11	1.000	11	1.000	11

6 CONCLUSIONS

This study developed a ROPS selection framework by solving the multiple objective problem for the determination of a reasonable parameter set of a hydrodynamic flow model based on multi-sites using the MRA. The candidate ROPS were derived by optimizing the model parameters using all available 11 combinations of weights for two gauging stations. Two different flood events were selected for calibration and validation, respectively. Based on these ROPS candidates, the most suitable parameter set among 11 Pareto optimums was determined by using the MRA.

This study showed that when the regrets obtained from the calibration and validation were aggregated by weight-averaging to select ROPS, relatively smaller weight was given to the first site as shown by the upper ranking of weight combinations for the two sites (Table 4). The best parameter set was related to the combination of weights (0.4, 0.6) that had the greater weight for the second site. The optimized model parameters and the corresponding sum of squares of the weighted residuals for the model generally varied with the combinations of weights for multiple sites. This study showed that ROPS selection based on the MRA can determine the most suitable ROPS from among many optimized sets by considering multiple sites. Therefore, the users of automatic calibration approaches do not have to face the task of selecting a preferred and suitable solution from the numerous Pareto-optimal sets.

ACKNOWLEDGEMENTS

This research was supported by a grant (13AWMP-B066744-01) from Advanced Water Management Research Program by Ministry of Land, Infrastructure and Transport.

REFERENCES

Atanov, G.A., Evseeva, E.G. & Meselhe, E.A. (1999). Estimation of Roughness Profile in Trapezoidal Open Channel. *Journal of Hydraulic Engineering*, 125(3), 309-12.

Chow, V.T. (1959). *Open Channel Hydraulics*. McGraw-Hill, New York.

Chung, E.S. & Kim, Y. (2014). Development of Fuzzy Multi Criteria approach to Prioritize Locations of treated Wastewater use Considering Climate Change Scenarios. *Journal of Environmental Engineering*, 146, 505–516.

Coon, W.F. (1998). *Estimation of Roughness Coefficients for Natural Stream Channels with Vegetated Banks*, U.S. Geological Survey Water-Supply Paper 2441, 133.

Cunge, J.A., Holly, F.M. & Verwey, A. (1980). *Practical Aspects of Computational River Hydraulics*. Pitman.

Doherty, J. (2000). *Visual PEST: Model-Independent Parameter Estimation*. Watermark Computing & Waterloo Hydrogeologic, Waterloo, Ontario, Canada.

Doherty, J. & Johnston, J.M. (2003). Methodologies for Calibration and Predictive Analysis of a Watershed Model. *Journal of the American Water Resources Association,* 39(2), 254-265.

Dung, N.V., Merz, B., Bardossy, A., Thang, T.D. & Ape, l.H. (2011). Multi Objective Automatic Calibration of Hydrodynamic Models Utilizing Inundation Maps and Gauge Data. *Hydrology and Earth System Sciences*, 15, 1339-1354.

Holly, F.M., Yang, J.C., Schwarz, P., Schaefer, J., Hsu, S.H. & Einhellig, R. (1990). *Numerical Simulation of Unsteady Water and Sediment Movement in Multiply Connected Networks of Mobile-bed Channels*, IIHR Report No. 343, The Iowa Institute of Hydraulic. Research, Iowa.

Horritt, M.S. (2000). Calibration of a two Dimensional Finite Element Flood Flow Model using Satellite Imagery. *Water Resources Research,* 36(11), 3279–3291.

Hunter, N.M., Bates, P.D., Horritt, M.S, De Roo, A.P.J. & Werner, M.G.F. (2005). Utility of Different Data Types for Calibrating Flood Inundation Models within a GLUE Framework. *Hydrology and Earth System Sciences*, 9(4), 412–430.

Ishii, A. (2000). *Parameter Identification of Manning Roughness Coefficient using Analysis of Hydraulic Jump with Sediment Transport*, Kawahara Group Research Report, Chuo University, Japan. http://www.civil.chuo-u.ac.jp/lab/ouyou/02-8/paper/PDF/2000/a-ki.pdf; 2000.

Khu, S.T. & Madsen, H. (2005). Multi Objective Calibration with Pareto Preference Ordering: An Application to Rainfall-Runoff Model Calibration. *Water Resources Research,* 41, W03004.

Khu, S.T., Madsen, H. & Di Pierro, F. (2008). Incorporating Multiple Observations for Distributed Hydrologic Model Calibration: An approach using a Multi Objective Evolutionary Algorithm and Clustering. *Advances in Water Resources,* 31(10), 1387-1398.

Kim, H.J. & Jun, K.S. (2004). Unsteady Flow Model with Variable Roughness Coefficient. *Korea Water Resources Association*, 37(12), 1055-1063.

Kim, Y. & Chung, E.S. (2014). An Index based robust Decision making Framework for Watershed Management in a Changing Climate. *Science of the Total Environment,* 473, 88-102.

Kim, Y., Chung, E.S. & Jun, S. (2015a). Iterative Framework for robust reclaimed Wastewater Allocation in a Changing Environment using Multi Criteria Decision Making. *Water Resources Management*, 29(2), 295-311.

Kim, Y., Chung, E.S., Won, K.J. & Gil, K.I. (2015b). Robust Parameter Estimation Framework of a Rainfall-Runoff Model using Pareto Optimum and Minimax Regret Approach. *Water,* 7, 1246-1263.

Li, X., Weller, D.E. & Jordan, T.E. (2010). Watershed Model Calibration using Multi Objective Optimization and Multi Site Averaging. *Journal of Hydrology,* 380, 277–288.

Liggett, J.A. & Cunge, J.A. (1975). *Numerical Methods of Solution of the Unsteady Flow Equations, Unsteady Flow in Open Channels, Water Resources.* Publications, Fort Collins, Colo., 89-182.

Liu, Y.B., Batelaan, O., De Smedt, F., Poórová, J. & Velcická, L. (2005). Automated Calibration Applied to a GIS based Flood Simulation Model using PEST, *Floods from Defence to Management*, Taylor-Francis Group, London, 317-326.

Loulou, R. & Kanudia, A. (1999). Minimax Regret Strategies for Green House has Abatement: Methodology and application. *Operations Research Lett*ers, 25, 219-230.

Madsen, H. (2003). Parameter Estimation in Distributed Hydrological Catchment Modelling using Automatic Calibration with Multiple Objectives. *Advances in Water Resour*ces, 26, 205–216.

Marquardt, D.W. (1963). An Algorithm for Least Square Estimation of Nonlinear Parameters. *Journal of the Society for Industry and Applied Mathematics*, 11(2), 431-441.

McCabe, M.F., Franks, S.W. & Kalma, J.D. (2005). Calibration of a Land Surface Model using Multiple Data Sets. *Journal of Hydrology*, 302, 209–222.

Ramesh, R., Datta, B., Bhallamudi, S.M. & Narayana, A. (2000). Optimal Estimation of Roughness in Open-Channel Flows. *Journal of Hydraulic Engineering*, 126(4), 299-303.

Rouse, H. (1965). Critical Analysis of Open Channel Resistance. Journal of the Hydraulic Division, 91(HY4), 1-25.

Saeidifarzad, B., Nourani, V., Aalami, M.T. & Chau, K.W. (2014). Multisite Calibration of Linear Reservoir based Geomorphologic Rainfall-Runoff Model. *Water*, 6, 2690–2716.

Seibert, J., Uhlenbrook, S., Leibundgut, C. & Haldin, S. (2000). Multiscale Calibration and Validation of a Conceptual Rainfall-Runoff Model. *Physics and Chemistry of the Earth,* 25(1), 59–64.

Shinma, T.A. & Reis, L.F.R. (2014). Incorporating Multi-event and Multi-site Data in the Calibration of SWMM. *Procedia Engineering*, 79, 75–84.

Wang, W. (2012). Multi-site Calibration, Validation and Sensitivity Analysis of the MIKE SHE Model for a Large Watershed in Northern China. *Hydrology and Earth System Sciences*, 16, 4621–4632.

Yen, B.C. (2002). Open Channel Flow Resistance. *Journal of Hydraulic Engineering,* 128(1), 20-39.

EXPERIMENTAL BENCHMARKING OF MESH SIZE AND TIME-STEP CONVERGENCE FOR A 1ST AND 2ND ORDER SWE FINITE VOLUME SCHEME

GREG COLLECUTT[1] & BILL SYME[2]

[1,2] BMT WBM, Brisbane Australia,
greg.collecutt@bmtwbm.com.au

ABSTRACT

High Performance Computing (HPC) hardware is moving towards computation of accelerator devices such as GPU and Xeon Phi. These devices now enable multi-million cell hydraulic models to be solved in reasonable times, and have brought supercomputer performance to the office desktop. TUFLOW's new HPC 2D Shallow Water Equation (SWE) solver was developed with the GPU accelerator target in mind, and utilises an explicit finite volume scheme that parallelises effectively across thousands of computational cores. Along with conservation of volume, momentum, and energy, mesh size convergence is an important feature of any numerical solution scheme, and it is important for the modeller to understand how the model results change with mesh resolution. The modeller can then choose the mesh size for production runs, knowing the likely magnitude of the residual discrepancy between the production results and fully converged results. The new TUFLOW scheme is detailed, and two examples for which experimental results are available to benchmark the numerical results are presented. The first example is steady state sub-critical shallow water flow around a 90° bend, Malone et al. (2008). The second example is the United Kingdom Environment Agency (UK EA) Test 6a, UK Environment Agency (2013), in which a reservoir breaches into a confined channel with an obstruction. This model is highly transient with supercritical flow and moving hydraulic shocks. The convergence of the model results against mesh resolution and time-step is examined for these test cases, using 1st and 2nd order spatial interpolation options. Comparisons of model results against experimental data are made, and conclusions regarding "best practice" mesh resolution are drawn.

Keywords: Shallow water equations; finite volume schemes; high performance computing; model convergence; TUFLOW.

1 INTRODUCTION

Nearly every real world application for flood modelling involves estimating head losses where flows change direction. The ability of any numerical scheme to accurately compute such head losses will depend on the mathematics of the scheme, the resolution with which the geometry is modelled, and the size of the time step used for transient solutions.

In an ideal world, the key results from any model should converge asymptotically to limiting values as the model resolution is increased. However, this can be more difficult to realise in practice than at first might be thought. For one, increasing model resolution may admit new flow details to the solution that are real and expected, while for another numerical rounding error may accumulate and lead to artefact divergence in the results at very fine mesh resolutions. This may deter a modeller from performing mesh-size sensitivity studies, but there are at least three strong arguments in favour of such investigations.

Firstly, grossly changing results with mesh size may well indicate a flawed solution scheme, incorrect computer code, or poorly defined model boundary conditions (such as point sources or sinks). Mesh-size sensitivity studies should be performed if for no other reason than to build confidence in both model and software.

Secondly, mesh-size sensitivity studies may also aid in the calibration process in that some discrepancy between model results and calibration data may perhaps be understood and hence tolerated if it can be shown that the results are converging towards the calibration data as the mesh size is refined.

Thirdly, the mesh-size sensitivity study also provides a lower bound for the uncertainty in key model results, and should perhaps be added to the variations identified by other sensitivity studies such as variation in bed-friction coefficients and/or inflows.

2 TUFLOW HPC FINITE VOLUME SCHEME

The new TUFLOW HPC solver (currently branded TUFLOW GPU) evolves the 2D Shallow Water Equations (SWE) on a uniform Cartesian grid. Cell averaged depth and velocity (i.e. volume and momentum) were tracked with a finite volume scheme, and propagated forward in time using the classic fourth order Runge-Kutta method. Figure 1 illustrates the notation adopted in which the cell averaged quantities for the cell

in question are denoted with the subscript cc, the neighbouring cells are denoted n1 through to n4, the cell wall volume fluxes Φ_1 to Φ_4. The cell width and height are Δx and Δy respectively.

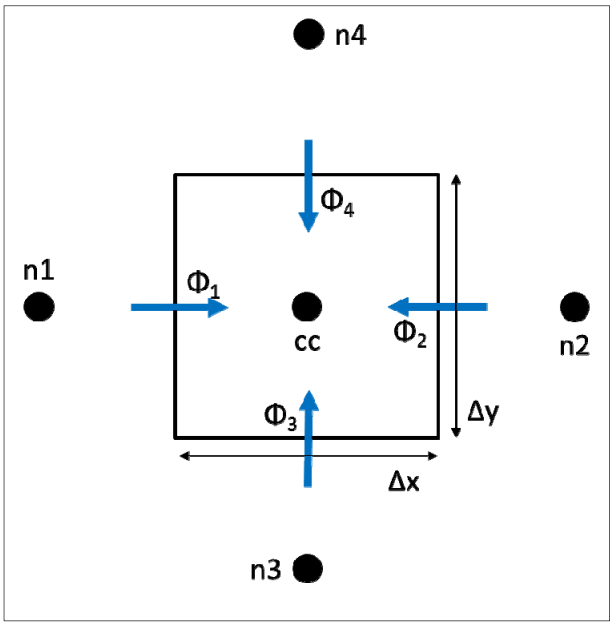


Figure 1. Finite volume notation.

The time derivative of the water depth was computed by considering conservation of volume, as expressed in Equation 1 in which *A* is the cell area, *h* is the water depth, and S_Q is any volume source or sink connected to the cell. The cell area *A* is considered to be time invariant. It may be modified by a 'storage reduction factor' if water displacing structures were present within the cell.

$$\frac{A\partial h}{\partial t} = \sum \Phi_i + S_Q \qquad [1]$$

The volume fluxes across each face are computed using the face normal velocities, the water depth at the face, and the relevant face width. The latter may also be modified by a 'flow reduction factor' where impervious structures impede the flow. The water depth at a face was taken to be the water depth of the cell that the water was flowing from (known in computational fluid dynamics as 'upwinding'). In the case of h_1 for example, if $(u_{cc} + u_{n1},) > 0$, then h_{n1} is used, otherwise h_{cc} is used. In the case that the bed elevation of the downwind cell was higher than the bed elevation of the upwind cell, the water depth at the face was reduced by the difference in bed elevations, and limited to be non-negative.

$$\Phi_1 = \Delta y \frac{u_{cc}+u_{n1}}{2} h_1 \qquad \Phi_2 = -\Delta y \frac{u_{cc}+u_{n2}}{2} h_2 \qquad \Phi_3 = \Delta x \frac{v_{cc}+v_{n3}}{2} h_3 \qquad \Phi_4 = -\Delta x \frac{v_{cc}+v_{n4}}{2} h_4 \qquad [2]$$

In a similar manner the time derivatives of the *u* and *v* (x-dir and y-dir respectively) velocities are also computed by considering conservation of momentum in each direction, as shown in Equation 3 with the viscosity terms omitted. Here u_i, is the face interpolated *u* velocities, *n* is the Manning's bed friction coefficient, *z* the bed elevation, *g* gravity. The right most forcing term of Equation 3, forcing due to gradients in surface elevation, requires care in its discretised implementation as it can become a net source of energy within the model if incorrectly posed.

$$\frac{A\partial(hu)}{\partial t} = \sum \Phi_i u_i - Ag \frac{n^2 (u^2+v^2)^{\frac{1}{2}} u}{h^{\frac{1}{3}}} - Agh \frac{\partial(z+h)}{\partial x} \qquad [3]$$

Equation 3 requires very small time-steps to solve for *u* when the water depth is shallow. A more efficient implementation can be achieved by separating out the bed friction term in Equation 3, expressing it in terms of the end-of-time-step *u'* and *v'*, and solving for the end-of-time-step *u'* and *v'* implicitly using the assumption that the other forcing terms remain constant. By doing this, Equation 3 can be rearranged to Equation 4 (and similarly for *v*):

$$u' = \frac{u+\Delta t u_{dot}}{1+a(u'^2+v'^2)^{\frac{1}{2}}} \qquad u_{dot} = \frac{1}{h}\left(\frac{\sum \Phi_i u_i}{A} - u\frac{\partial h}{dt}\right) - g\frac{\partial(z+h)}{\partial x} \qquad a = \Delta t g \frac{n^2}{h^{\frac{4}{3}}} \qquad [4]$$

u' and *v'* are solved iteratively together, and the effective derivatives for *u* and *v* calculated and passed to the explicit integrator.

Turbulent viscosity was modelled using the Smagorinsky formulation as shown in Equation 5, and the terms shown in Equation 6 added to the right side of Equation 3 (and similarly incorporated into Equation 4). Note that the turbulent viscosity ν_T is calculated separately at the four faces.

$$\nu_T = mA\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2\right]^{\frac{1}{2}} + c \qquad [5]$$

$$\Delta y\left[h_2\nu_{T2}\frac{\partial u}{\partial x}\right)_2 - h_1\nu_{T1}\frac{\partial u}{\partial x}\right)_1\Big] + \Delta x\left[h_4\nu_{T4}\frac{\partial u}{\partial y}\right)_4 - h_3\nu_{T3}\frac{\partial u}{\partial y}\right)_3\Big] \qquad [6]$$

The interpolation of *u* and *v* to the cell faces can be performed either using a first order technique in which the cell average values from the upwind cell are used, or by using a second order method. The difficulty with using a pure second order method such as $u_1 = (u_{n1} + u_{cc})/2$, is that it is not total variation diminishing (TVD) and is known to be unstable. Here a hybrid technique had been adopted. To illustrate this method, consider the left face of the cell with positive flux Φ_1. The cell average *u* velocity is u_{cc}, the upwind cell average *u* is u_{n1}, and now also consider the cell on the opposite (far) side of the upwind cell with an average *u* velocity of u_{11}. The hybrid technique looks at the form of the solution for *u* over these three cells. In the limit that the solution shows no curvature, pure second order interpolation was used, and in the limit that the solution shows reversal (i.e. $u_{n1}$is outside the range spanned by u_{11} and u_{cc}), then the first order scheme (up-winding) is adopted. For solutions in-between, the method blends smoothly from one to the other. Mathematically, for the case illustrated, the blend is expressed in Equation 7. In the event that flux Φ_1 is negative, u_{n2}, u_{cc}, u_{n1} are used in place of u_{11}, u_{n1}, u_{cc} respectively.

$$u_i = (1-\xi)u_{n1} + \xi u_{cc} \qquad \xi = 2x(1-x) \qquad x = \frac{u_{n1}-u_{11}}{u_{cc}-u_{11}} \qquad 0 \le x \le 1 \qquad [7]$$

The computed derivatives for *h*, *u* and *v*, are local in time and may be computed for each cell independently of other cells. This approach lends itself to massively parallel computation where threads were mapped to cells and each executes the same code and the order of execution is unimportant. There were a handful of data that were globally reduced (either summed or the maximum located) at each time step but efficient techniques exist for performing these reductions in a massively parallel environment.

The time evolution of the water depth and velocities was computed explicitly with the classic Runge-Kutta fourth order method. Adaptive time step size was typically employed to maintain stability of the model. The three time step control criteria were courant number N_u, celerity number N_c, and diffusion number N_d as shown in Equation 8. The time step was adjusted to maintain these at or below 1.0, 1.6, and 0.3 respectively.

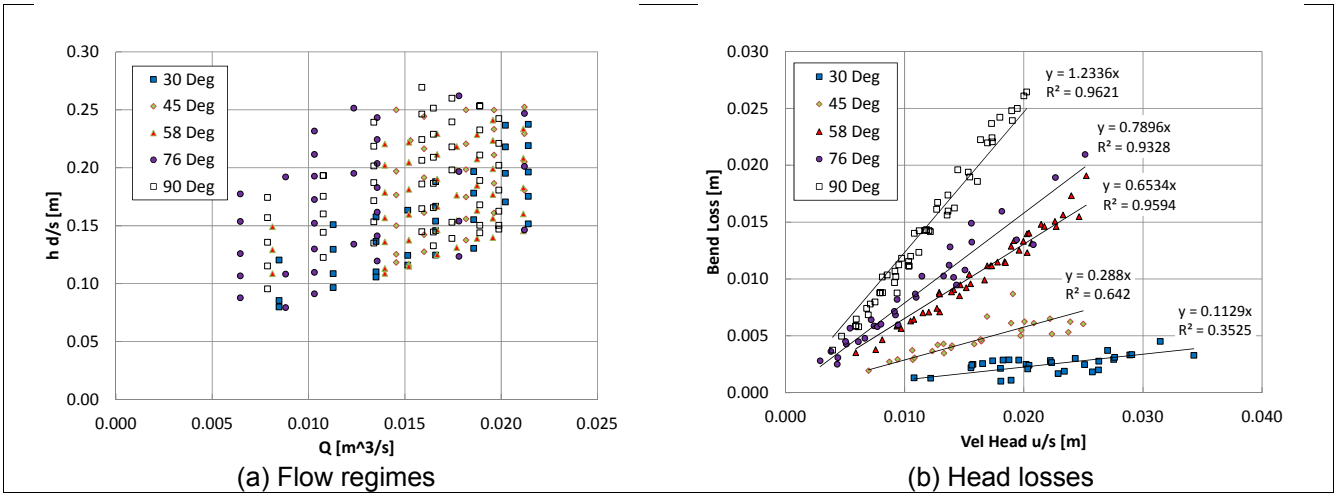
$$N_u = \max\left(\frac{|u|\Delta t}{\Delta x}, \frac{|v|\Delta t}{\Delta y}\right) \qquad N_c = \max\left(\frac{\sqrt{gh}\Delta t}{\Delta x}, \frac{\sqrt{gh}\Delta t}{\Delta y}\right) \qquad N_d = \max\left(\frac{\nu_T\Delta t}{\Delta x^2}, \frac{\nu_T\Delta t}{\Delta y^2}\right) \qquad [8]$$

2 ABRUPT ANGLED BEND TEST CASE

The TUFLOW GPU solver had been extensively tested with a wide variety of models, but there are only a few available with test data available for benchmarking. The first benchmark test considered was that of sub-critical flow around an abrupt 90 degree angled bend, as performed by Malone et. al. (2008).

Their test setup consisted of two flat bottomed flumes connected at an angle, with the downstream depth and volume flow rate able to be varied. Malone and colleagues measured the head loss attributable to the bend over a range of flow rates, water depths, bend angles, and performed a detailed analysis. The flow regimes (downstream water depth vs volume flow rate) tested are shown in Figure 2 along with a linear fit for head loss as a function of velocity head. The slope of this linear fit is the head loss factor, which represents the ratio of total energy loss around the bend to upstream kinetic energy. For the 90 degree abrupt bend, a loss factor of 1.23 was obtained.

For the majority of the tests performed, the water depths within the flumes were comparable to the flume width. This casts some doubt on the validity of modelling these tests with the SWE, but as will be shown the results agree surprisingly well.



(a) Flow regimes (b) Head losses

Figure 2. Abrupt angled bend test, reproduced from Malone et. al. (2008).

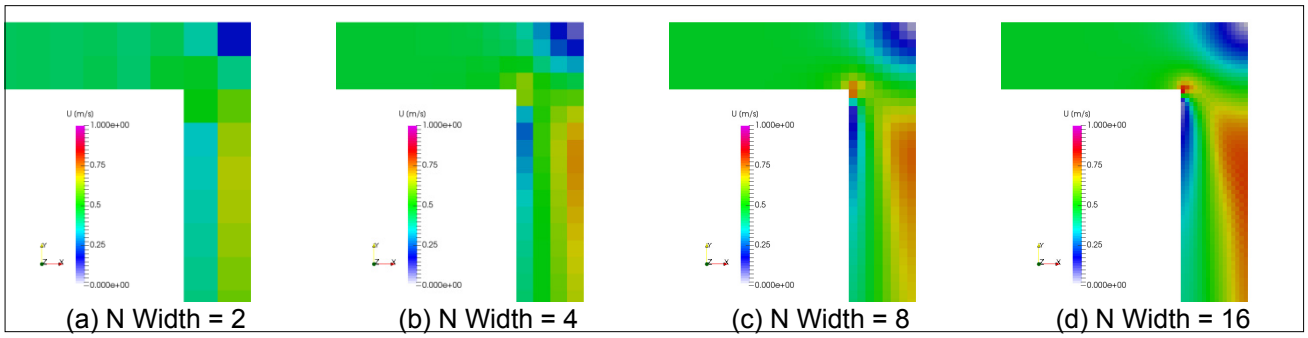
The first numerical study considers flow around the 90 degree abrupt bend, with flow rate of 0.008m^3/s, and downstream water level of 0.123m. The flume width was 0.15m, and the upstream and downstream arms were 18m widths long. The bed friction (Manning's) coefficient was set to 0.01, and both arms had a downhill gradient of 0.2%. The numerical solution was also expected to depend on mesh size, viscosity coefficients, spatial interpolation scheme, and time step. Figure 3 illustrates the flow solutions at the bend for mesh sizes ranging from 2, 4, 8 and 16 cells across the flume width resulting in mesh cell sizes less than 1 cm. Flow enters from left and exits from the bottom.

The solution was found to reach a steady state by approximately 60s from model start. Each flume arm was divided into tiles of one flume width square, and the total head in each tile was spatially and temporally averaged over 1s intervals from 60s to 120s. The total head in each tile was plotted as a function of distance from the bend, and a linear fit was derived. The head loss attributable to the bend was calculated by extrapolating the upstream and downstream linear that fits to the bend, and computing the difference.

Figure 4 plots the loss factor attributable to the bend as a function of mesh size, for the 1st and 2nd order spatial schemes and for constant viscosity values (i.e. *c* of Equation 5, in units m^2/s) and for proportional viscosity values (i.e. *m* of Equation 5, dimensionless). From these results it can be seen that both 1st and 2nd order schemes tend to over-predict the head loss, and that both 1st and 2nd order schemes with non-zero constant viscosity appear to asymptote to limiting values with decreasing cell size.

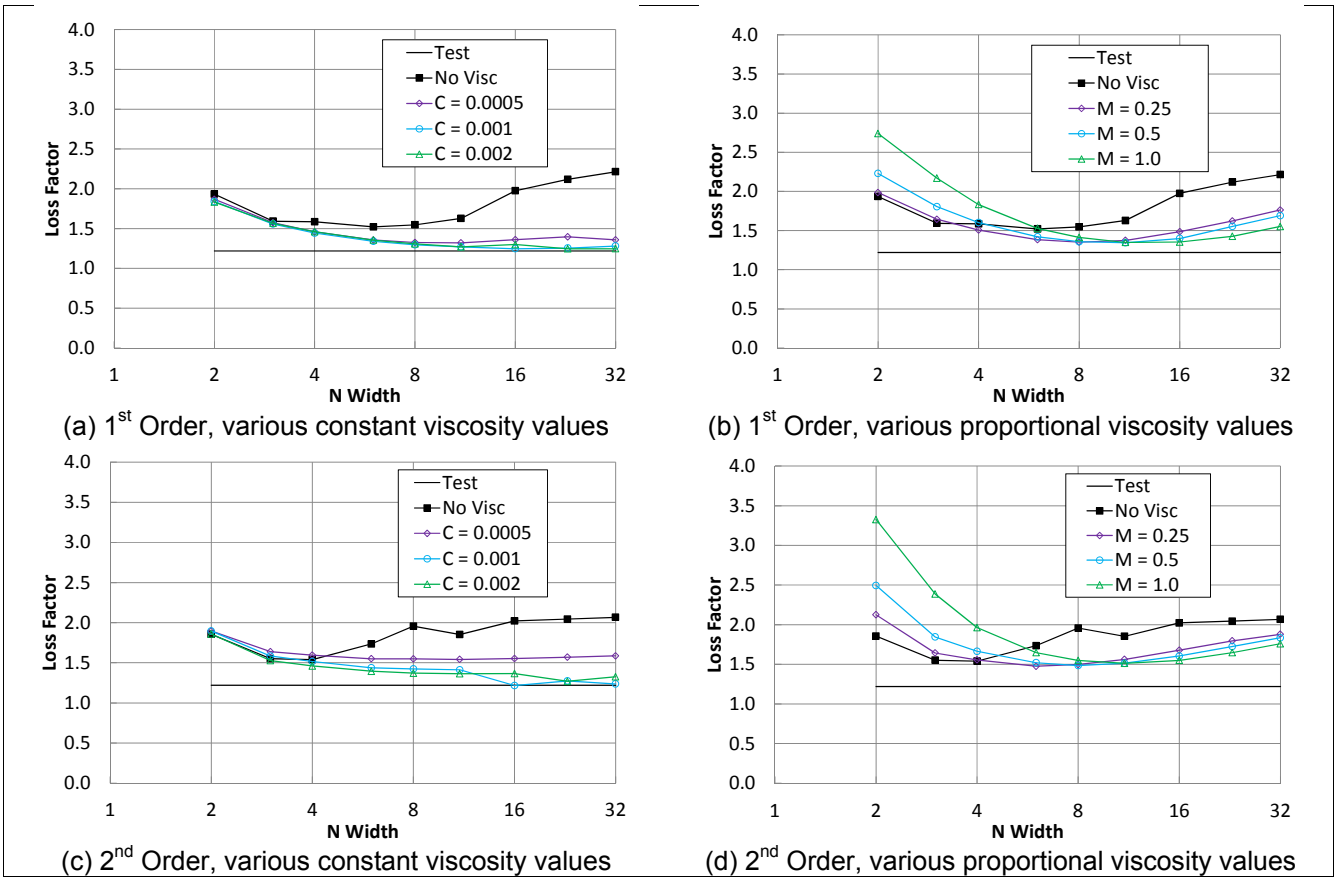
For both 1st and 2nd order schemes, the zero-viscosity case does not show convergence. A possible reason for this becomes apparent when a zero viscosity flow solution for a finer mesh size (in this case N Width = 16) was visualised, as shown in Figure 5, in which vigorous eddy structures were apparent in the flow. The introduction of some models of viscosity suppresses the intensity of these structures. Interestingly for both 1st and 2nd order schemes, non-convergence was apparent for the various proportional viscosity values trialled. A possible explanation for this behaviour is that, as the proportional viscosity term is also proportional to cell area, reducing cell size leading to zero viscosity in its limit (provided the velocity gradients within large eddies are determined by the flume width, not the mesh size).

In summary, it appears that excellent solutions were obtained with both 1st and 2nd order spatial schemes with just enough constant viscosity to prevent eddy structures in the flow. Reasonable solutions are found utilising 3 or more cells across the flume width.



(a) N Width = 2 (b) N Width = 4 (c) N Width = 8 (d) N Width = 16

Figure 3. Flow solutions at various mesh resolutions.



(a) 1st Order, various constant viscosity values

(b) 1st Order, various proportional viscosity values

(c) 2nd Order, various constant viscosity values

(d) 2nd Order, various proportional viscosity values

Figure 4. TUFLOW HPC Abrupt 90deg bend head loss factor.

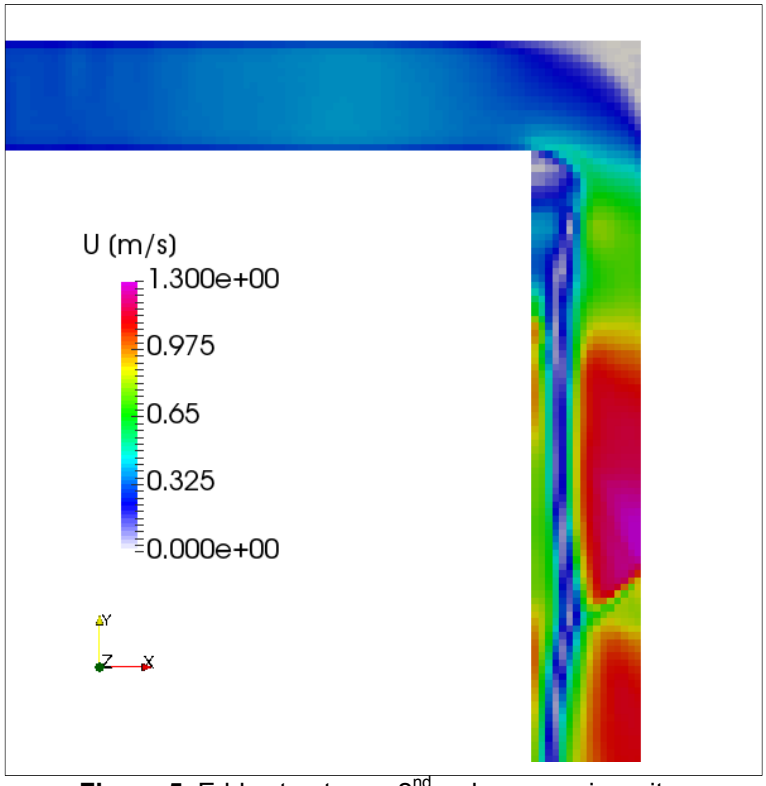
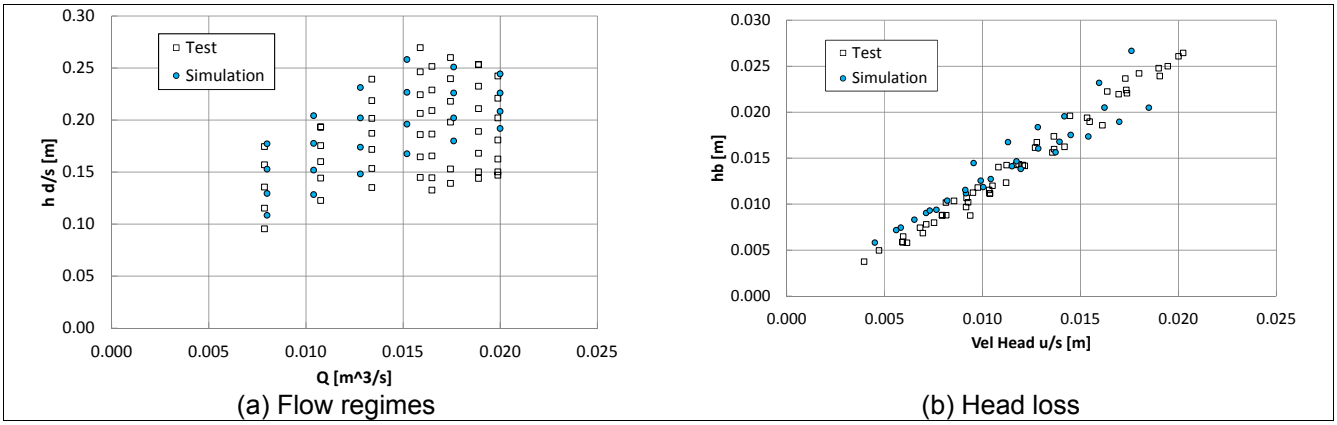


Figure 5. Eddy structures, 2nd order, zero viscosity.

Solutions for other flow rates and downstream depths were also investigated. Figure 6 shows the flow regimes considered and the resulting head losses using the 2nd order spatial scheme and a constant viscosity of 0.002m^2/s.



(a) Flow regimes (b) Head loss

Figure 6. Abrupt 90 degree angled bend.

Malone et. al. (2008) also considered other bend angles: 30, 45, 58, and 76 degrees. Again using the same flow parameters as were used for the mesh size study, bend angles from 0 to 90 degrees in 2 degree increments were studied. The results are presented in Figure 7 using a constant viscosity of 0.002m^2/s, and 8 cells across the flume width. There appears to be some scatter in the results, most likely caused by the process of discretising an angled smooth walled channel onto a Cartesian grid, but otherwise the form is promising. In general, the loss factors tend to be similar to the results obtained from the test data. One of the issues noted here is that when discretising an angled smooth walled channel onto a Cartesian grid, the apparent steps in the wall cause additional head loss in this scheme, as the flow must change direction around each step. The approach adopted within the scheme for solving this issue is to progressively modify the effective widths of the faces adjacent and perpendicular to the wall leading into and out of steps, thus allowing the flow in the cells adjacent to the wall to maintain a velocity vector parallel to the average wall line.

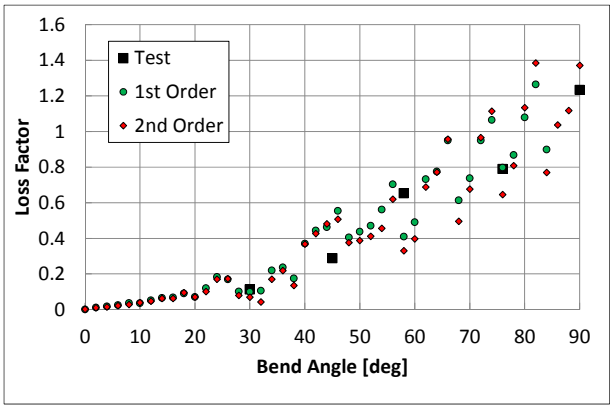


Figure 7. Head loss factor vs abrupt bend angle.

3 UK ENVIRONMENT AGENCY TEST 06A

The other test case studied for which test data were available was the dam breach model which forms part of the UK Environment Agency's test suite, UK Environment Agency (2013). In this scaled test a reservoir was released into a long channel containing a building which obstructs the flow. This is a demanding case for comparison as the flow was highly transient and at times supercritical with moving hydraulic jumps. Figure 8 is a schematic of the test setup and a snapshot of the simulation flow velocity at time t = 7s is shown in Figure 9.

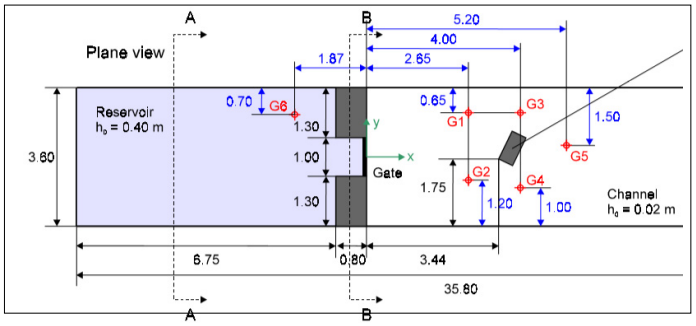


Figure 8. UK EA Test 06A, schematic.

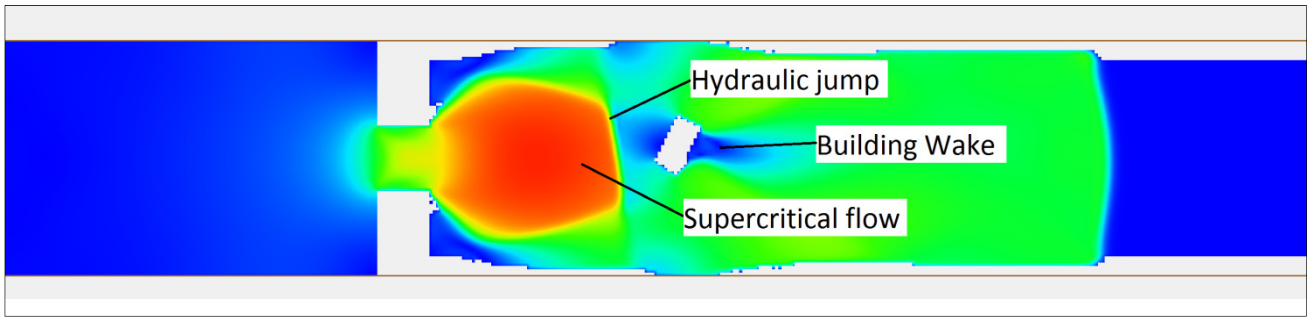


Figure 9. UK EA Test 06A, flow velocity magnitude at t = 7s.

More details regarding this test can be found in the UK EA test notes. The solution showed in Figure 9 was computed using the 2nd order spatial scheme, a mesh size of 0.05m, and constant viscosity of 0.05m^2/s. Water depth and velocity were recorded at a number of gauge locations. For brevity the depth and velocity at the first three gauges are compared in Figure 10, where it can be seen that the 2nd order solution offers a better to the test results.

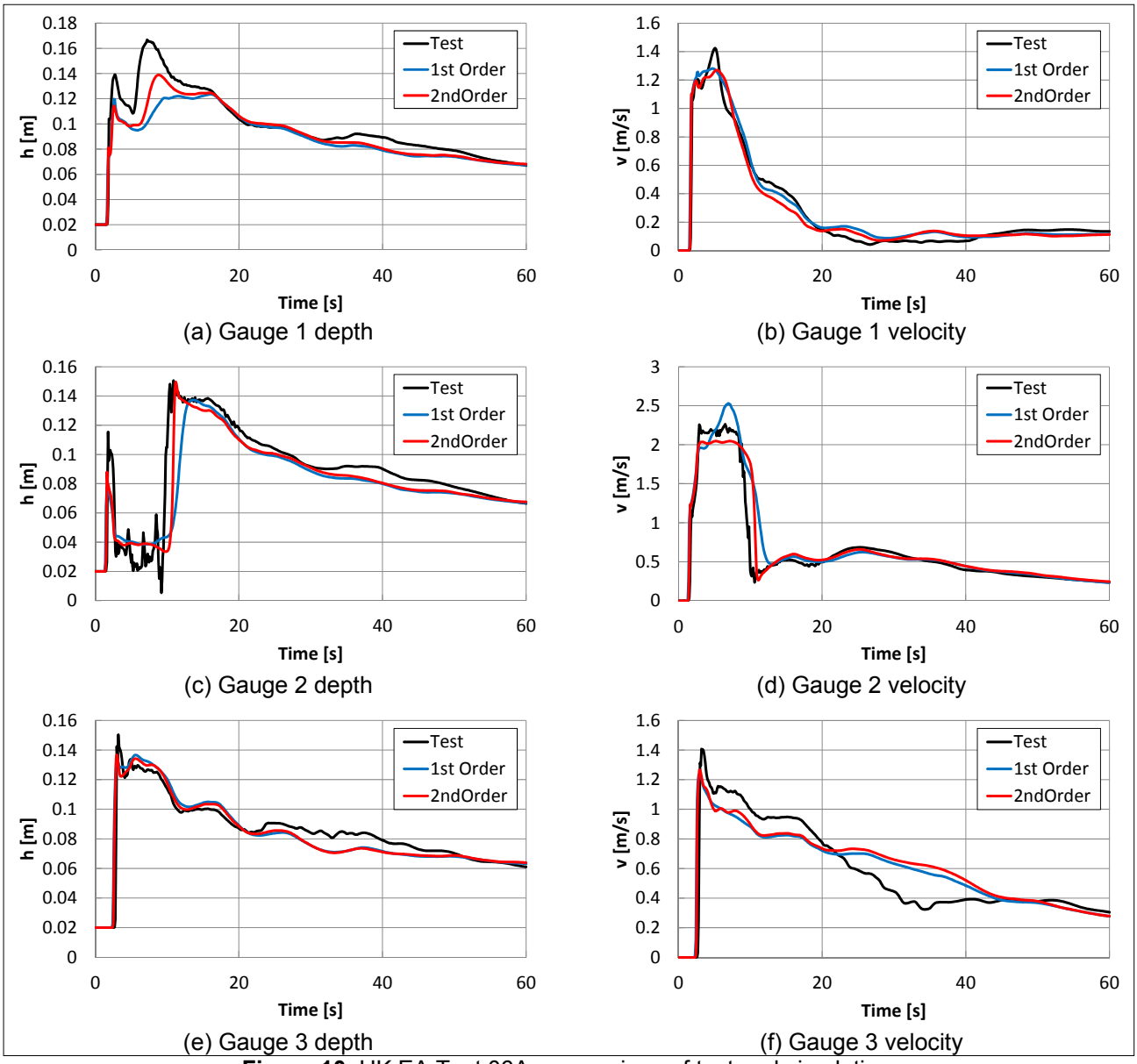
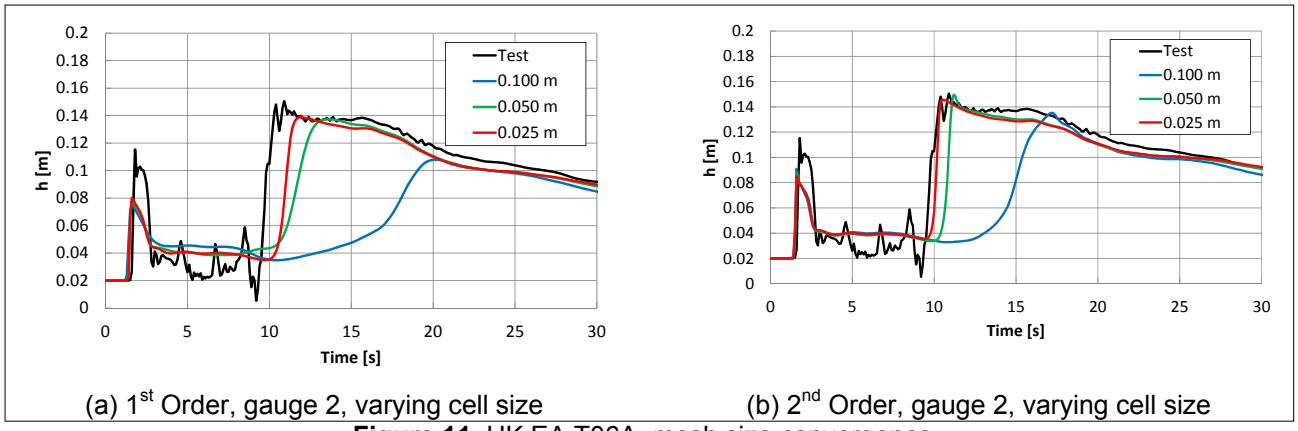


Figure 10. UK EA Test 06A, comparison of test and simulation.

Particular interest was in the results for gauge 2, in which the flow is super critical from about t = 2s to t = 10s, and then the hydraulic jump moves upstream over the gauge at approximately t = 10s and the flow becomes subcritical. The timing of this phenomenon is sensitive to the choice of viscosity coefficient and a

constant viscosity of 0.05m^2/s was found to be optimal. Both 1st and 2nd order schemes reproduce hydraulic jumps well. It is also worthy to note that a better fit with test data was obtained with Manning's bed friction coefficient of 0.012 in place of the suggested 0.01 in the UK EA test notes.

Figure 11 compares the results for the depth at gauge 2 using both 1st and 2nd schemes and for the three mesh sizes 0.100m, 0.050m, and 0.025m. These results indicate that for both 1st and 2nd order schemes there was very little difference in the results between 0.050m and 0.025m cell size. The building width was 0.4m, and a mesh size of 0.05m represents 8 cells across the building.



(a) 1st Order, gauge 2, varying cell size (b) 2nd Order, gauge 2, varying cell size

Figure 11. UK EA T06A, mesh size convergence.

As noted earlier, adaptive time-stepping was used to maintain the three control numbers N_u, N_c, N_d (Equation 8) below target limits. With many simulations, as the model evolves, the number controlling the time step may change. The scheme makes provision to run at a time step that was some fraction of the target time step, effectively causingN_u, N_c, N_d to be controlled to values that were some fraction of the default targets. The UK EA Test 06A model was re-run with a lower constant viscosity of 0.02m^2/s, so as to cause the simulation to be courant number, controlled initially before becoming controlled diffusion number. Figure 12 represents the depth at gauge 1 using the 2nd order solver and the full target time step (Co = 1.0), 0.9, 0.8 and 0.7 of the target time step. Firstly note that the results for Co = 0.7 lie on top of the results for Co = 0.8 which indicates that time-step convergence has been achieved at Co = 0.8. Secondly note that there were small deviations from the converged results at Co = 0.9 and 1.0. The scheme incorporates NaN checking, and where a time step has failed the solution is wound back and the time-step repeated with a half-sized step. On successful completion of a repeated time step, the time step size was gradually increased towards the target value over successive time steps. This feature has kept both of these larger time step simulations running, even though their results in this case are of questionable quality. It is always recommended that a modeller check for time step convergence of the results, as the required time step fraction may vary with the exact particulars of the model.

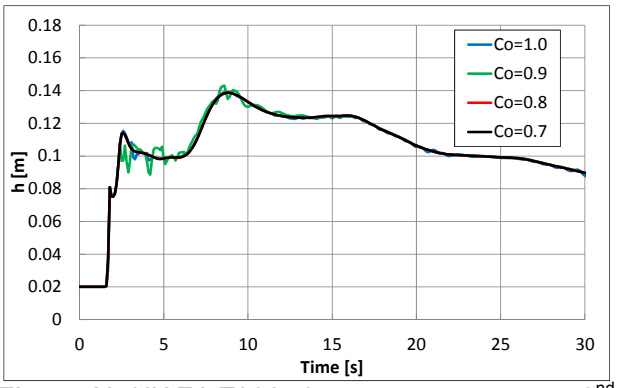


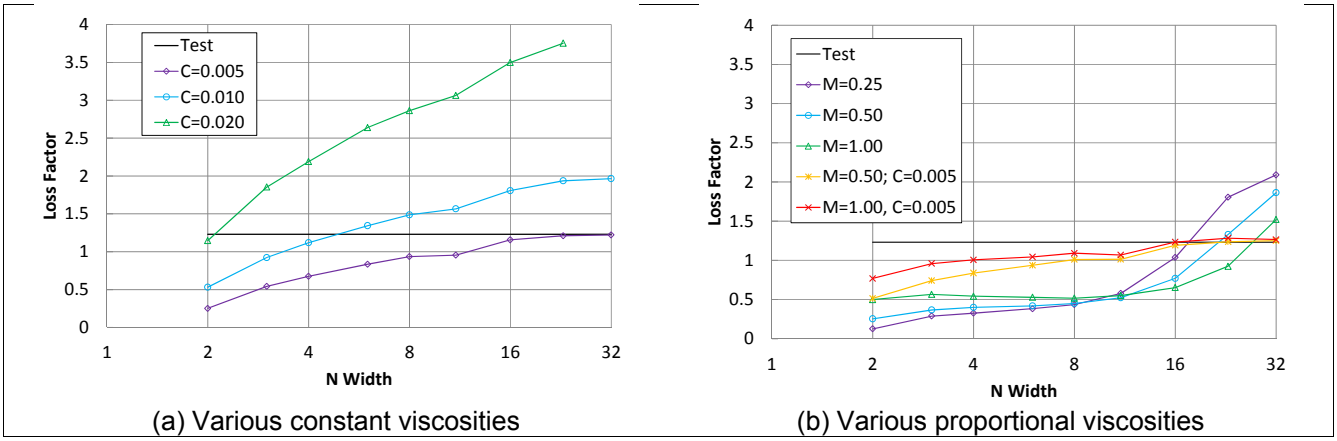
Figure 12. UK EA T06A, time step convergence, 2nd order spatial scheme (Note that results for Co = 0.7 overlay those for Co = 0.8).

4 COMPARISON WITH TUFLOW STELLING/SYME SCHEME

The abrupt 90 deg bend benchmark was solved using the established TUFLOW ADI finite difference implicit scheme as developed by Stelling and modified by Syme (1991) with the results shown in Figure 13. The ADI scheme shows some convergence with finer mesh sizes with constant only viscosity, but not with proportional only viscosity. As highlighted earlier, the proportional viscosity term becomes small in the limit of

small cell sizes and if the velocity gradients were determined by problem geometry and not mesh size. Significant interest is that with constant viscosity the Stelling/Syme scheme converges to the asymptotic limit from below, rather than from above as was the case with the new HPC scheme. This was consistent with applications of the Stelling/Syme in practice where a small amount of additional energy loss is typically required at sharp bends and flow constrictions when benchmarking and calibrating models, Syme (2001).

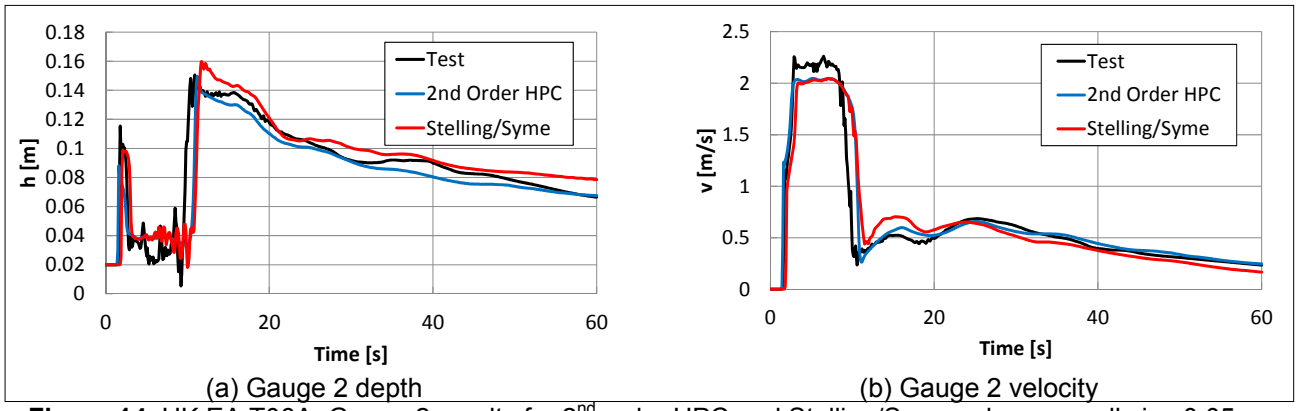
Of even more interest, as shown in Figure 13b, the combination of both proportional and constant viscosity terms (M=1.00, C=0.005) can generate a head-loss vs mesh size curve that is nearly flat (i.e. mesh size independent) for meshes 3 cells across the flume width and finer.



(a) Various constant viscosities (b) Various proportional viscosities

Figure 13. TUFLOW Stelling/Syme scheme, abrupt 90 deg bend head loss.

The UK EA Test 06A benchmark was also solved with the implicit Stelling/Syme scheme using the default eddy viscosity coefficients of M=0.5 and C=0.05, with the results for gauge 2 shown in Figure 14. The agreement between the two schemes is excellent, and the results for the other gauges show similar fidelity.



(a) Gauge 2 depth (b) Gauge 2 velocity

Figure 14. UK EA T06A, Gauge 2 results for 2nd order HPC and Stelling/Syme schemes, cell size 0.05m.

5 IMPLICATIONS FOR REAL WORLD APPLICATION

As shown above, the choice of Smagorinski and constant viscosity formulations has a profound effect on bend head loss estimation for the cases examined. For both the new explicit HPC scheme and the standard Stelling/Syme implicit scheme, using proportional viscosity only does not lead to mesh-size convergence and may also allow chaotic solutions to develop in the limit of very small cell sizes. For the new explicit HPC scheme, either 1st or 2nd order interpolation in combination with a small constant viscosity appears to yield excellent results. However, the optimal constant viscosity depends on the problem scale of 0.002m^2/s for the laboratory scale flume tests, and approximately 0.05m^2/s for the UK EA Test 06A. This is to be expected since the term is dimensional, and therefore its optimum value is likely to change with problem scale (and it also must be supplied in correct model units). Further research into the scale dependence of this term is needed, as the results would indicate a reducing constant viscosity coefficient with reducing problem scale.

For 90 degree bend losses, the new HPC scheme yields results that are not strongly dependent on mesh size provided the flow channels are represented by at least 3 cells across their width. In practice, it may not be possible to always have this level of mesh resolution; therefore, the impact of mesh size and choice of viscosity on key results should be investigated by the modeller prior to production runs.

It was always useful to bear in mind that a model's utility was not determined by whether it was a perfect representation of reality, but by how well it allows investigators to confidently consider as previously unseen

flow scenarios and determination of the potential impacts of material changes. Such confidence in a model's utility is derived through both calibrations against available data and sensitivity studies.

6 SUMMARY

The finite volume scheme utilised in the new TUFLOW 2D HPC solver was presented and benchmarked against test results for two distinctly different test cases.

The solver demonstrated mesh size convergence for both test cases in both 1st and 2nd order spatial interpolation schemes with a constant viscosity model. Convergence typically required 8 cells or more across feature widths, however meshes as coarse as three cells across feature widths still appear to perform well with regard to predicting head loss.

The solver demonstrated time step convergence. For the highly transient test case 'UK EA Test 06A', convergence was obtained using time steps of 0.8 of the limiting time step targets.

The solver reproduced the test results from UK EA Test 06A well including supercritical flow and hydraulic shocks/jumps.

Using only the proportional term in the Smagorinsky viscosity model, mesh size convergence is not demonstrated for either spatial scheme, possibly due to the model viscosity diminishing at smaller mesh sizes.

In the absence of viscosity, the 2nd order spatial scheme admitted eddy structures in the solution, while the 1st order solution demonstrated a dampening of eddy formation most likely due to numerical diffusion.

In terms of best practice modelling with the new HPC scheme it is recommended that

(1) as far as practical meshes utilise three or more cells across any feature of interest;
(2) the 2nd order spatial scheme can be selected, unless the 1st order scheme produces sufficiently similar results to the 2nd order scheme;
(3) the overall time step control to be set to 0.8 of default targets; and
(4) sufficient (and only sufficient) constant viscosity is used to prevent multitudes of small eddies forming in the solution.

REFERENCES

Malone, T.R. & Parr, A.D. (2008). *Bend Losses in Rectangular Culverts*, Kansas Department of Transportation, (http://ntl.bts.gov/lib/30000/30900/30935/KU-05-5_Final_Report.pdf).

Syme, W.J. (1991). Dynamically Linked Two-Dimensional / One-Dimensional Hydrodynamic Modelling Program for Rivers, Estuaries & Coastal Waters. *PhD Thesis*, Dept. of Civil Engineering, The University of Queensland, May 1991.

Syme, W.J. (2001). Modelling of Bends and Hydraulic Structures in a Two-Dimensional Scheme. *The Institution of Engineers Australia Conference on Hydraulics in Civil Engineering*, Hobart.

UK Environment Agency (2013). *Benchmarking the Latest Generation of 2D Hydraulic Modelling Packages*, Environment Agency, Horison House, Deanery Road, Bristol, BS1 9A United Kingdom, (http://evidence.environment-agency.gov.uk/FCERM/Libraries/FCERM_Project_Documents/SC120002_Benchmarking_2D_hydraulic_models_Report.sflb.ashx).

PARALLEL COMPUTATION OF A DAM-BREAK FLOW MODEL USING OPENACC AND OPENMP APPLICATIONS

SHANGHONG ZHANG[(1)], WENDA LI[(2)] & ZHU JING[(3)]

[(1,2,3)] Renewable Energy School, North China Electric Power University, Beijing, China,
zhangsh928@126.com; 546038591@qq.com; 839137938@qq.com

ABSTRACT

High performance calculations are of great importance to the simulation of dam-break events, as accuracy and accelerated speed are the key factors in the process of dam-break flow modeling. Based on the finite volume method, this paper established a high performance dam-break flow simulation model. An explicit scheme was used to discretize the 2D shallow water control equations and Roe's approximate Riemann solution of the finite volume method was adopted for the interface flux of the grid cells. A graphics processing unit (GPU)-based parallel method, OpenACC, and Open Multi-Processing (OpenMP) parallel mode are used to realize parallel computing, respectively. Because an explicit discrete technique is used to solve the governing equations, and there is no correlation between the grid calculations in a single time step, the parallel dam-break model can be easily realized by adding OpenACC and OpenMP directives to the loop structure of the grid calculations. To analyze the performance of the model, we considered the Pangtoupao flood storage area in China using a multi-core computer with an Nvidia Tesla K20c card and two different grid division schemes. By carefully studying the implementation method and optimization of data transportation in the parallel algorithm, a speedup factor of 17.76 can be achieved with OpenACC and a speedup factor of 8.6 was achieved with the conventional OpenMP parallel mode on a 16-kernel computer. The results demonstrate that the optimized feature settings greatly influence the degree of speedup and models involving larger numbers of calculations exhibit greater efficiency and higher speedup factors. In addition, both of the OpenACC and OpenMP parallel modes are found to have good portability, making it easy to implement parallel computation from the original serial model. And, it is possible to simulate dam-break flows in large-scale watersheds on a single computer with parallel computing.

Keywords: Dam-break flow; parallel computation; graphics processing unit; OpenACC; OpenMP.

1 INTRODUCTION

Dam projects not only bring about great benefits to human beings, but also give rise to potential dangers. Although the possibility of a dam-break is very small, should one occur, it will bring great loss of life and property (Kocaman, 2012; Erpicum, 2010). Therefore, there is a need to develop high performance mathematical models of the flooding that would result from a dam-break event to provide the theoretical and technical foundation of a risk analysis and disaster evaluation system.

Dam-break flows, whose evolution occurs in dry riverbeds, are a type of rapidly varying flow. Potential models need to solve discontinuous flow calculations with wet/dry boundaries (Brufau et al., 2004). To improve the simulation accuracy, an appropriate solution and a fine mesh are usually required. For problems involving a discontinuous flow and wet/dry boundaries, the finite volume method is generally applicable. If the explicit form is adopted in the discretization of the friction source, the calculation of boundary fluxes uses only previous values and there is no correlation between grid calculations in the same time step. This latter point makes the model algorithm simpler and more effective (Zoppou and Roberts, 2002). Therefore, the explicit finite volume method is widely applied in the simulation of water flow (Brodtkorb et al., 2012; De La Asunción et al., 2011; Lastra et al., 2009).

To improve the computational efficiency of mathematical models, parallel algorithms have been introduced (Neal et al., 2010; Zhang et al., 2013). In terms of their realization, parallel computation algorithms fall into three categories: message-passing interface (MPI) models, shared memory models (OpenMP), and GPU general computing models (Ghosh et al., 2012; Leandro et al., 2014; Reyes et al., 2012). In a shared memory model, communication between threads is achieved by reading and writing directly to share memory. Using the OpenMP standard, it is easy to solve the traditional programming problem of load balancing. Models developed under this standard have the advantages of simplicity, good extendibility, and portability (Chapman et al., 2008). Yu (2012) researched the transport processes of non-uniform sediment in the Yellow River by embedding OpenMP parallel instructions into a loop structure and obtained 1.55× of speedup in a dual-core server. Neal et al. (2010) also added OpenMP instructions to the major cycle of a flood inundation model, acquiring 5.8× of speedup on eight-core servers. In mainstream parallel algorithms, GPU models obtain good acceleration (Grillo et al., 2013). GPUs have multi-streaming processors and powerful computing capabilities,

which are of vital significance for efficient parallel computation with large amounts of data and high precision requirements (Wang et al., 2014). The OpenACC application programming interface is one of the GPU acceleration methods, it is a collection of compiler directives and runtime routines that use the C/C++ or FORTRAN language to compile the specified cycle and code. Pickering et al. (2015) analyzed the popular OpenACC programming standard, as implemented by the PGI compiler suite, to evaluate its utility and performance potential in computational fluid dynamics applications. Herdman et al. (2012) developed an explicit hydrodynamics scheme in the CloverLeaf mini-application based on OpenACC directives and Hart et al. (2012) used the OpenACC programming model to accelerate the Himeno benchmark code and analyze the resulting performance.

The main goals of this study are as follows: (1) to establish a two-dimensional hydrodynamic model based on the finite volume method that can simulate the dam-break flow routing process of the Pangtoupao flood storage area; (2) to develop parallel dam-break models using OpenACC and OpenMP, respectively, and investigate the performance of parallel models under different mesh sizes, and (3) to analyze the main factors influencing the computation time and efficiency.

2 STUDY AREA

The Pangtoupao flood storage area is located on the left bank of the Nen and Songhua Rivers, in the upper reaches of Zhaoyuan County, northwest of Harbin City, China (Figure 1). The length of the storage area is 46 km from east to west and 58 km from north to south, giving a total area of 1994 km^2 and a volume of about 55 × 10^8 m^3. The dam in Pangtoupao was breached on August 15th, 1998, and multiple levees were breached by the flooding in the Nen and Songhua Rivers.

The dam-break flood hydrodynamic model was built with a digital elevation model (resolution 10 m) of Pangtoupao and the flood data observed in Songhua River in 1998. The OpenACC platform used to evaluate the model's performance consisted of an Intel® Xeon® E5-2603 CPU with an Nvidia Tesla K20 GPU card, CPU frequency of 1.80 GHz, and memory capacity of 4 GB. The software development environment combined the Windows 7 64-bit operating system with the PGI PVF version 14.9 compiler. The OpenMP platform is used to evaluate the model's performance consisted of a Lenovo T350 G7 server containing 16 kernels, with a CPU frequency of 2.4 GHz and memory capacity of 8 GB. Visual Studio 2008 and Intel Visual Fortran11.0 were used to build the model.

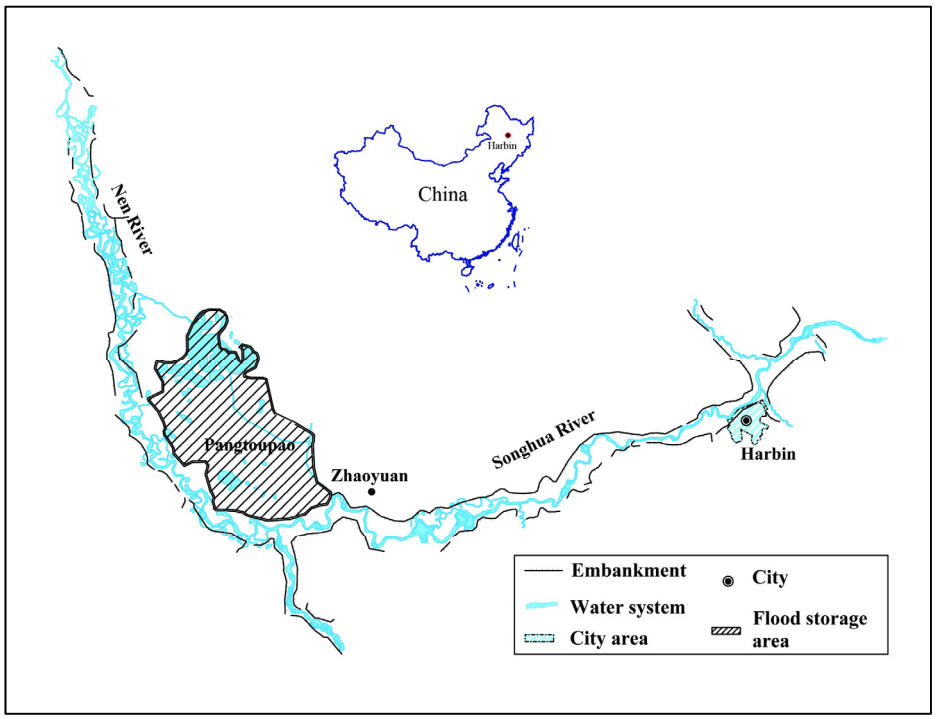


Figure 1. Sketch of the study area.

3 METHOD

3.1 Dam break flow model

3.1.1 Control equation

Dam-break flows have the approximate properties of plane flows. The conservative form of the planar 2D shallow water equations is as follows:

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=S \qquad [1]$$

$$U=\begin{bmatrix} h \\ hu \\ hv \end{bmatrix},\ F=\begin{bmatrix} hu \\ hu^2+\frac{gh^2}{2} \\ huv \end{bmatrix},\ G=\begin{bmatrix} hv \\ huv \\ hv^2+\frac{gh^2}{2} \end{bmatrix},\ S=\begin{bmatrix} 0 \\ gh(S_{ox}+S_{fx}) \\ gh(S_{oy}+S_{fy}) \end{bmatrix}$$

where h is water depth (m) and u, v are the water velocities in the x and y directions, respectively. $S_{Ox}=-\partial Z_b/\partial x$ and $S_{Oy}=-\partial Z_b/\partial y$ describe the bottom slope in the x and y directions, respectively, and $S_{fx}=n^2u\sqrt{u^2+v^2}h^{-4/3}$ and $S_{fy}=n^2v\sqrt{u^2+v^2}h^{-4/3}$ give the friction gradient in the x and y directions, respectively. Z_b is the bottom elevation and n is the Manning coefficient.

The equation does not consider the influence of the Coriolis force or wind.

3.1.2 Computational grid

The Pangtoupao flood storage area has an irregular complex shape. To ensure the precision of the calculations, a quadrilateral unstructured mesh is selected to cover the whole computation area. Two meshes are studied in this paper, with average side-lengths of 350 and 150 m. The total number of quadrilateral mesh cells is 18,109 and 98,991.

3.1.3 Flux calculation and source discretization

A cell-centered format is adopted in the calculations, where the physical variables are defined in the grid center, as shown in Figure 2. Using Gauss's formula, we can integrate Eq. [1] on a grid to give:

$$\int_{\Omega_i}\frac{\partial U}{\partial t}d\Omega+\int_{\Omega_i}\nabla\cdot E d\Omega=\int_{\Omega_i}S d\Omega \qquad [2]$$

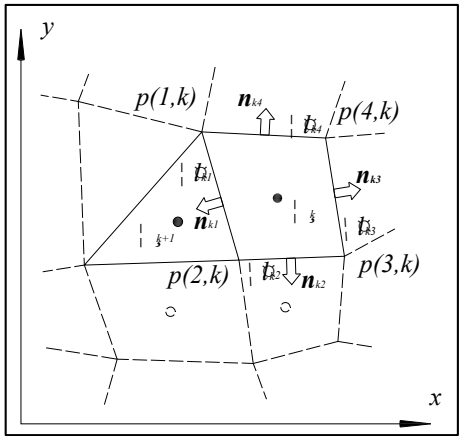


Figure 2. Schematic diagram of the finite volume discretization.

The line integral along the grid is transformed to the volume integral:

$$\Delta U=-\frac{\Delta t}{\Delta S_k}\sum_{i=1}^{4}(E_{ki}\cdot n_{ki})\Delta l_{ki}+\frac{\Delta t}{\Delta S_k}\int_{\Omega}S d\Omega \qquad [3]$$

where i is the number of edges, Δl_{ki} is the length of each side, n_{ki} is the outward normal vector of each side, Δs_k is the area of each calculation unit, and E_{ki} is the numerical flux via edge i.

The interface flux is solved using Roe's scheme, which can effectively compute the interruption of dam-break flow:

$$E\cdot n=\frac{1}{2}\left[(F,G)_R.n+(F,G)_L.n-\left|\bar{J}\right|(U_R-U_L)\right] \qquad [4]$$

where $\bar{J}$ is the Roe average of the Jacobian matrix $J=\frac{\partial(E\cdot n)}{\partial U}$. The eigenvalue matrix of $\bar{J}$ is $\bar{\lambda_k}$ (k = 1, 2, 3), and the related eigenvectors are $\bar{e_k}$ (k = 1, 2, 3). UL and UR are left and right interface conservative form variables, respectively. The detailed process of solving a Roe scheme was described by Toro (2009).

Simple discretization of the source term cannot obtain harmonious calculation results, so a source term dependent on bottom topography must undergo Eigen-decomposition and upwind treatment to balance the interface flux (Hubbard et al., 2000), as shown in Eq. [5]:

$$\overline{S_0}=\sum_{j=1}^{4}\sum_{k=1}^{3}\left[\frac{1}{2}(1-\text{sign}(\bar{\lambda}^k))\alpha^k\bar{e}^k\Delta s_j\right]^j \qquad [5]$$

where sign() is the sign function, $a^1=-\frac{1}{2}\bar{c}\,\bar{z_b}$, $a^2=0$, $a^3=\frac{1}{2}\bar{c}\,\bar{z_b}$, $\bar{c}$ is the average velocity given by the Roe scheme, and $\bar{z_b}=\frac{1}{2}[(z_b)_L+(z_b)_R]$. $(z_b)_L$ and $(z_b)_R$ are the bottom elevations of the left and right sides, respectively.

To realize a dam-break model in a parallel system, an explicit scheme is used to discretize the friction source in Eq. [3], giving:

$$\Delta U=\frac{\Delta t}{A}\left[\left(-\sum_{j=1}^{4}E_j\eta_j l_j\right)+\sum_{j=1}^{4}\sum_{k=1}^{3}\left\{\frac{1}{2}\left(1-\text{sign}\left(\bar{\lambda}^k\right)\right)\alpha^k\bar{e}^k l_j\right\}^j+\Delta t S_f^n\right] \qquad [6]$$

After the discretization, calculating ΔU only requires the water level, flow, and elevation of the front time step. There is no correlation between each grid cell calculation in the current time step.

3.2 Parallel model implementation

The implementation of the dam-break model with a finite volume discretization involves three steps: data input, numerical calculation, and results output. The first part includes the specification of initial parameters, grid information, initial water level, and flow. The second part mainly includes the calculation of the flux and source terms. The third part outputs the water depth and velocity of each grid cell. The calculation of grid cells belongs to the main calculation unit in a time step, and each of these units must solve the flux and source equations in the grid cell. Because an explicit discretization scheme is used to solve the equations, each grid calculation requires only the results from the previous time step. There is no correlation between the grid calculations in the same time step. Therefore, adding parallel instructions to the circulation calculations in a time step can realize parallel computing.

3.2.1 OpenACC parallel method

OpenACC is a high-level directive-based programming model that accelerates parallel parts using the C/C++ and FORTRAN languages. In dam-break flow models, the key to a parallel implementation based on OpenACC is that the calculation-intensive region is offloaded to the GPU for parallel computation. The simulation is iterated over a series of time steps and it is most beneficial if all calculation units in a time step can be run on the GPU. The model uses an explicit scheme to calculate the flow field and the numerical mesh calculation is uncorrelated within a single time step. Thus, the cycle calculation in each grid can be executed in parallel. GPUs have numerous computation cores. Different grid cell calculations are distributed to these different cores and each core executes the grid cell flux and source calculations. In this study, an Nvidia Tesla K20c card was used to execute the parallel computations. This card has a Kepler GK110 GPU and 2496 Nvidia CUDA cores, meaning the dam-break model can be executed efficiently with the OpenACC application. The calculation results are deposited directly into the memory structure, allowing the velocity and water depth to be updated.

The model targeted by OpenACC API-enabled implementations is host-directed execution with an attached accelerator device. The host allocates memory on the accelerator and copies data and the corresponding code from CPU to GPU memory. The data and code are allocated to free resources on the GPU to accelerate the computation and back up the data once the calculations are completed. The host initiates the data transfer, allocates memory on the accelerator device, sends the code to the accelerator, passes arguments to the compute region, queues the device code, waits for completion, transfers results back to the host, and deallocates memory (see Figure 3).

3.2.2 OpenMP parallel method

The same as the OpenACC parallel method, the key to a parallel implementation based on shared memory is to determine the main loops that can be run separately in the model. OpenMP parallel instructions can then be used to allocate these uncorrelated units to different processors, decreasing the calculation time. As shown in Figure 4, adding OpenMP instructions to the circulation calculations in a time step can realize parallel computing. Different grid cell calculations are distributed to different threads, and each thread executes grid cell flux and source calculations. The calculation result will be deposited into the memory structure directly, allowing the velocity and water depth to be updated. The flowchart of the parallel model solution is the same as the serial one, with the only difference being that some OpenMP instructions are added to the time step loop. Thus, it is easy to realize parallel computing with OpenMP in this kind of model.

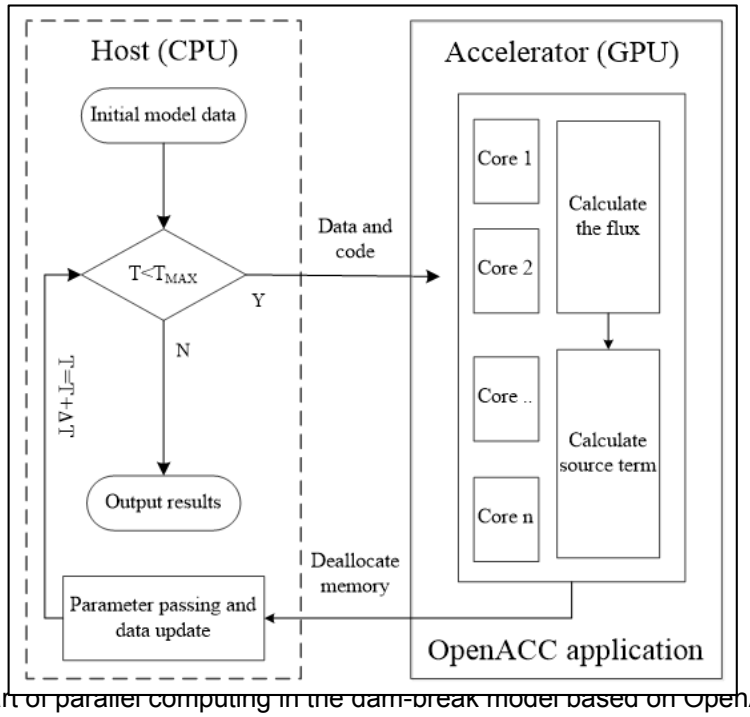


Figure 3. Flowchart of parallel computing in the dam-break model based on OpenACC application.

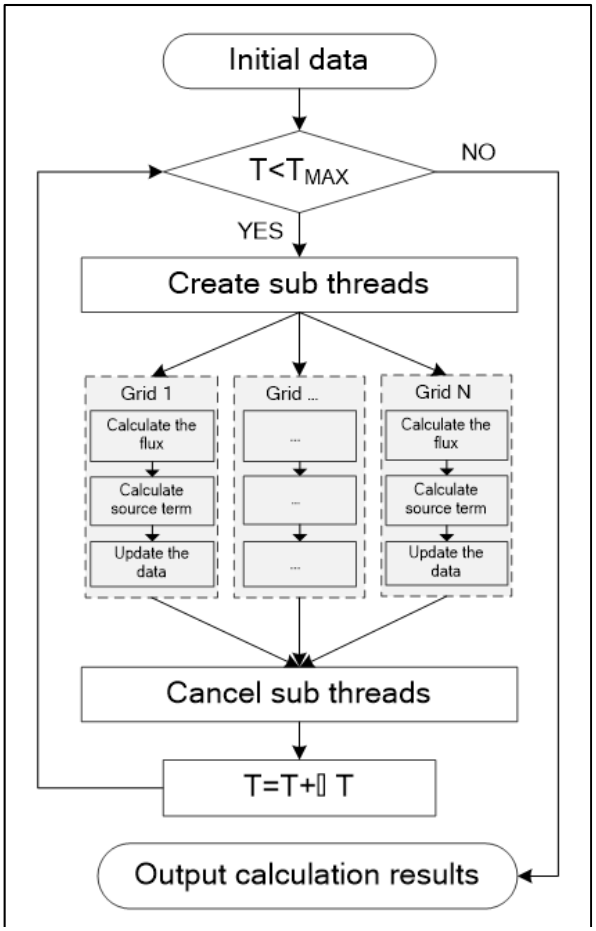


Figure 4. Flowchart of parallel computing in the dam-break model.

3.3 Model performance testing

To comprehensively analyze the model performance, we conduct the following operations: (1) verify the precision of the model calculations; (2) discuss model speedup and efficiency for different mesh sizes; (3) analyze the main factors influencing the model calculation's time and efficiency, and (4) contrast OpenACC and OpenMP, two different parallel algorithms, with a simple serial program.

To compare the results between different mesh schemes, the study area is divided in two different ways using a quadrilateral unstructured mesh.

Scheme 1: Mesh size of 350 m, giving 18,109 grid cells.

Scheme 2: Mesh size of 150 m, giving 99,046 grid cells.

The basic parameters and conditions of the model are as follows:

(1) The basic 1998 Songhua River observed flood data were used as the initial entrance flow, with a submerged area depth of 0 m;
(2) The Manning roughness value of the study area is 0.035;
(3) The minimum water depth method was used to treat the wet–dry boundary, with dry grid cells initially assigned a small depth value of 0.05 m and a velocity of 0 m/s. Cells in which the water depth is greater than 0.05 m are defined as being wet;
(4) The Pangtoupao dam break occurred on August 15th, 1998. As the breaches quickly broadened to form a stable entrance width, we adopted a flood breach width of 530 m in the calculations;
(5) At 08:00 on August 28th, 320 h after the dam break, the flood storage area reached its maximum value. To compare data on the maximum submerged area after the flood, the total runtime was set to 320 h.

4 RESULTS AND DISCUSSIONS

4.1 Model verification

Table 1 shows a comparison between the measured data and values calculated through a serial model. As shown in the table, the relative error between the calculated and measured values is less than 2.5%. This shows that the model has good calculation accuracy.

Table 1. Comparison between measured and calculated values.

	Inundated area (km^2)	Impoundage ($\times 10^8 m^3$)	Mean depth (m)
Measured value	1160	35.5	3.06
Calculated value	1187	35.2	3.01

The simulated flood routing of the Pangtoupao flood storage area during the flood event is shown in Figure 5. We can see that the flood entered a narrow valley following the breach and then rushed downstream into the low-lying areas from the mouth of the valley, forming a huge flooded area. The simulation of this submerging process agreed with the observed data (HBMWR and HBSWRC, 2002).

The OpenACC and OpenMP parallel model calculation results were the same as for the serial model, indicating the veracity of the parallel method. The results show that the parallel model has acceptable accuracy for the simulation of dam-break flows.

4.2 Parallel performance analysis

OpenMP and OpenACC are different parallel algorithms with simple directives and efficient characteristics. The OpenMP execution of the parallel model is based on multiple CPU threads, whereas OpenACC executes parallel model based on GPU computing capacity. These two different parallel mechanisms have a number of similarities in their compiler modes. The two mechanisms both use the compiler guidance mode to add directives in the parallel region without modifying the source code. OpenACC offloads code to the accelerator through directives. OpenMP is controlled by the main thread, which generates sub-threads to execute computing tasks. The control flow returns to the single main thread after the calculation has finished.

There are three performance evaluation indices for the parallel model: speedup, efficiency, and extendibility.

The speedup formula is as follows:

$$S=T/T_n \qquad [7]$$

where T is the serial computing time and T_n is the computation time with parallel method.

For the four mesh division schemes of the Pangtoupao flood storage area, the different capacity levels were compiled on both the CPU and the GPU accelerators. Because the calculation time for these entire schemes is too long, we studied the first 100 h to compare model performance. The time, speedup, and efficiency required for the calculation are presented in Table 2.

We can see that two schemes attained a dramatic reduction in execution time. For the 350 m mesh scheme, a 320 h flood process took about 12 h to calculate with the serial model, whereas only 1.2 h was required with a parallel model on the Nvidia GPU Kepler K20c card. This rapid simulation will contribute to improved flood prevention and disaster planning. Thus, the precision and speed of the model calculation can

be effectively improved by mesh refinement and parallel computing, and we can choose the most appropriate scheme according to the actual demand. The computation time, speedup, and efficiency all increased with the number of grid cells. The fineness of the mesh is the main factor affecting computation time. As the mesh became finer, the number of calculations increased, which led to the increase in simulation time. The speedup and efficiency also improved as the number of grid cells increased. A speedup factor of 10.77 was achieved with 18,109 grid cells, whereas 17.76 was achieved with 98,991 cells, demonstrating that the parallel model performs better with an increased calculation load.

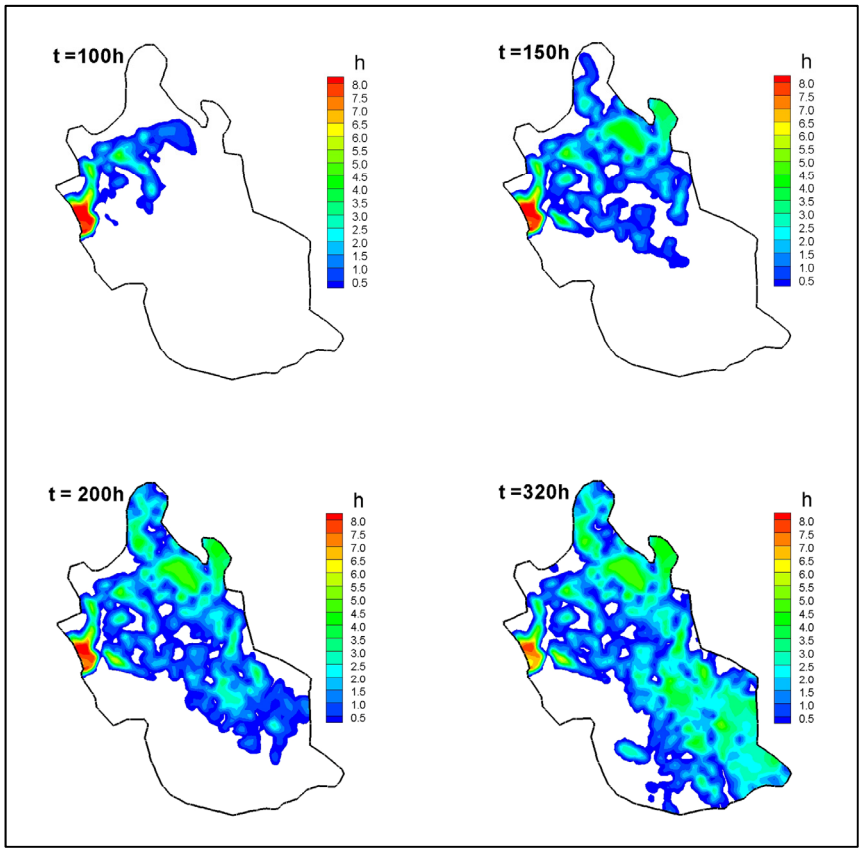


Figure 5. Simulated submerging process of the Pangtoupao flood storage area during the flood event.

Table 2. Parallel results of different calculation schemes for 100h flood process simulation.

k	A_k(m)	NELEM		Thread count						GPU
				1	2	4	8	12	16	
1	350	18109	T	13622(12h)	7327	3802	2386	1954	1792	1376(1.2h)
			S	-	1.86	3.59	5.73	6.99	7.62	10.77
			TS	-	46	72	83	86	87	91
2	150	98991	T	81685	42668	21931	11745	9942	9458	4849
			S	-	1.91	3.72	6.95	8.22	**8.64**	**17.76**
			TS	-	48	73	86	88	88	94

Notes: A_k: average grid size, NELEM: grid number, T: execute time (s), TS: time saving ratio (%), S: speedup of different calculation schemes; the data in the brackets are the time used to simulate 320h flood process.

In Scheme 1, where the grid size was 350 m, the parallel efficiency of the OpenMP parallel model increased gradually with the number of threads, achieving a speedup factor of 7.62 with 16 threads. For the same grid number, the OpenACC implementation achieved a speedup factor 10.77, which is 1.41× faster overall than the OpenMP parallel model. OpenMP attained its optimal speedup factor of 8.64 with a grid size of 150 m. The OpenACC parallel model achieved a corresponding speedup factor of 17.76. Thus, in this case study, OpenACC achieved a better acceleration effect than OpenMP. In addition, the OpenACC parallel algorithm required only one GPU to perform the intensive model calculations, which indicates good expansibility and portability.

5 CONCLUSIONS

A high-performance dam-break flow simulation model was established based on the finite volume method. An explicit scheme was used to discretize the 2D shallow water control equations and Roe's approximate Riemann solution of the finite volume method was adopted for the interface flux of the grid cells.

For the dam-break flow observed in 1998 in the Pangtoupao flood storage area, accurate results were obtained from the dam-break flow simulation.

The OpenACC and OpenMP application were used to realize parallel computing and the implementation method and key technologies of the parallel dam-break model were described. Because an explicit discrete technique is used to solve the governing equations, and there is no correlation between the grid calculations in a single time step, the parallel dam-break model can be easily realized by adding OpenACC and OpenMP directives to the loop structure of the grid calculations.

The OpenACC-annotated dam-break model was executed on the Nvidia Kepler K20c platform and two different mesh computation schemes were established to analyze the parallel performance of the dam-break model. The results revealed that models involving a larger number of calculations achieve greater efficiency and a higher rate of acceleration. The GPU executions were demonstrated to run up to 17.76 faster than the original sequential CPU implementation. For the 350 m mesh scheme, the parallel model required just 1.2 h to calculate a 320 h flood process. The OpenACC acceleration effect was better than that of the OpenMP method with a 16-kernel computer. For the 150 m mesh scheme, the OpenACC implementation was 2.06× faster than the OpenMP parallel model with 16 threads.

ACKNOWLEDGEMENTS

This study was supported by the National Key Research and Development Program of China (2016YFC0401401, 2016YFC0401407) and the National Natural Science Foundation of China (51379076).

REFERENCES

Brodtkorb, A.R., Sætra, M.L. & Altinakar, M. (2012). Efficient Shallow Water Simulations on GPUs: Implementation, Visualization, Verification, and Validation. *Comput. & Fluids*, 55, 1–12.

Brufau, P., García-Navarro, P. & Vázquez-Cendón, M.E. (2004). Zero Mass Error Using Unsteady Wetting–Drying Conditions in Shallow Flows Over Dry Irregular Topography. *International Journal for Numerical Methods in Fluids*, 45(10), 1047-1082.

Chapman, B., Jost, G. & Van Der Pas, R. (2008). *Using OpenMP: Portable Shared Memory Parallel Programming* (Vol. 10). MIT press. (Din. dok thu press format)

De La Asunción, M., Mantas, J.M. & Castro, M.J. (2011). Simulation of One-Layer Shallow Water Systems on Multicore and CUDA Architectures. *The Journal of Supercomputing*, 58(2), 206-214.

Erpicum, S., Dewals, B.J., Archambeau, P. & Pirotton, M. (2010). Dam Break Flow Computation Based on an Efficient Flux Vector Splitting. *Journal of Computational and Applied Mathematics*, 234(7), 2143-2151.

Ghosh, S., Liao, T. & Calandra, H. (2012). Experiences with OpenMP, PGI, HMPP and OpenACC Directives on ISO/TTI Kernels*. High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion,* 691-700.

Grillo, L., De Sande, F., Fumero, J.J. & Reyes, R. (2013). Programming for GPUs: The Directive-Based Approach. P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), *2013 Eighth International Conference on*, 612-617.

Hart, A., Ansaloni, R. & Gray, A. (2012). Porting and Scaling OpenACC Applications on Massively-Parallel, GPU-Accelerated Supercomputers. *The European Physical Journal-Special Topics*, 210(1), 5-16.

Herdman, J.A., Gaudin, W.P., McIntosh-Smith, S., Boulton, M., Beckingsale, D.A., Mallinson, A.C. & Jarvis, S.A. (2012). Accelerating Hydrocodes with OpenACC, OpenCL and CUDA. In *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion,* 465-471.

Hubbard, M.E. & Garcia-Navarro, P. (2000). Flux Difference Splitting and the Balancing of Source Terms and Flux Gradients. *Journal of Computational Physics*, 165(1), 89-125.

Hydrology Bureau of Ministry of Water Resources (HBMWR), and Hydrology Bureau of Songliao Water Resources Commission (HBSWRC). (2002). 1998 Songhua River flood. China Water & Power Press.

Kocaman, S. & Ozmen-Cagatay, H. (2012). The Effect of Lateral Channel Contraction on Dam Break Flows: Laboratory Experiment. *Journal of hydrology*, 432, 145-153.

Lastra, M., Mantas, J.M., Ureña, C., Castro, M.J. & García-Rodríguez, J.A. (2009). Simulation of Shallow-Water Systems Using Graphics Processing Units. *Mathematics and Computers in Simulation*, 80(3), 598-618.

Leandro, J., Chen, A.S. & Schumann, A. (2014). A 2D Parallel Diffusive Wave Model for Floodplain Inundation with Variable Time Step (P-DWave). *Journal of Hydrology*, 517, 250-259.

Neal, J.C., Fewtrell, T.J., Bates, P.D. & Wright, N.G. (2010). A Comparison of Three Parallelisation Methods for 2D Flood Inundation Models. *Environmental Modelling & Software*, 25(4), 398-411.

Pickering, B.P., Jackson, C.W., Scogland, T.R., Feng, W.C. & Roy, C.J. (2015). Directive-based GPU Programming for Computational Fluid Dynamics. *Computers & Fluids*, 114, 242-253.

Reyes, R., López-Rodríguez, I., Fumero, J.J. & de Sande, F. (2012). accULL: An OpenACC Implementation with CUDA and OpenCL Support. In *European Conference on Parallel Processing*, 871-882

Toro, E.F. (2013). Riemann Solvers & Numerical Methods for Fluid Dynamics: A Practical Introduction. *Springer Science & Business Media.*

Wang, X., Shangguan, Y., Onodera, N., Kobayashi, H. & Aoki, T. (2014). Direct Numerical Simulation and Large Eddy Simulation on a Turbulent Wall-Bounded Flow Using Lattice Boltzmann Method and Multiple GPUs. *Mathematical Problems in Engineering*, 2014, 1-10.

Yu, S.B. (2012). OpenMP Parallel Computing Model for Two-Dimensional Flow with Non-Uniform Sediment. *Adv. Sci. Tec. Water Res.* 32(2), 11-14.

Zhang, S., Xia, Z. & Wang, T. (2013). A Real-Time Interactive Simulation Framework for Watershed Decision Making Using Numerical Models and Virtual Environment. *Journal of Hydrology*, 493, 95-104.

Zhang, S., Xia, Z., Yuan, R., & Jiang, X. (2014). Parallel Computation of a Dam-Break Flow Model Using OpenMP on a Multi-Core Computer. *Journal of Hydrology*, 512, 126-133.

Zoppou, C. & Roberts, S. (2003). Explicit Schemes for Dam-Break Simulations. *Journal of Hydraulic Engineering*, 129(1), 11-34.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

TURBULENCE CHARACTERISTICS OF A SHALLOW MIXING LAYER DEVELOPING OVER 2-D DUNES

GOKHAN KIRKIL[(1)]

[(1)] Department of Industrial Engineering, Kadir Has University, 34083, Cibali, Istanbul, Turkey,
gokhan.kirkil@khas.edu.tr

ABSTRACT

Results of a high resolution Detached Eddy Simulation (DES) are used to characterize the evolution of a shallow mixing layer developing between two parallel streams in a long open channel over two-dimensional (2D) dunes (Figure 1). The study discusses the vertical non-uniformity in the mixing layer and provides a quantitative characterization of the growth of the large-scale quasi 2D coherent structures with the distance from the splitter plate. The presence of large-scale roughness elements in the form of an array of two-dimensional dunes with a maximum height of 0.25D (D is the channel depth) induces a very rapid and larger shift of the centerline of the mixing layer due to the increased influence of the bottom roughness. Results show that in streamwise sections situated after 100D (D is the channel depth) from the splitter plate, the width of the mixing layer close to the free surface stays constant. The tilting of the mixing layer interface toward the low speed stream is observed as the free surface is approached in all vertical sections. Consistent with visualizations of the mass transport of a passive scalar within the mixing layer, close to the free surface, the estimated streamwise length of the quasi 2D mixing layer eddies is about 1.5 to 2.0 times larger than the local width of the mixing layer.

Keywords: Shallow mixing layers; detached eddy simulation; dunes.

1 INTRODUCTION

The mixing layer region controls the exchange of mass and momentum between two streams of different velocities. Turbulent shallow mixing layers are observed in rivers, coastal regions and atmosphere. A typical example relevant for environmental flows is the flow downstream of a river confluence with a small angle between the two tributaries. As in most cases the flow depth is much smaller than the width of the river downstream of the confluence, the flow conditions are shallow.

The shallowness of the fluid and the bottom friction significantly affect the dynamics of a shallow mixing layer compared to that of a free (deep) mixing layer (Uijttewaal and Booij, 2000). As a result in a shallow mixing layer: a) the transverse spreading rate reduces with the distance from the origin of the mixing layer (end of splitter plate) until the growth of the mixing layer ceases; b) the velocity difference on the two sides of the mixing layer decreases in the streamwise direction; and c) the axis of the mixing layer shifts toward the low-speed side. The vertical development of the large-scale eddies in a shallow mixing layer is constrained by the bed and free surface. This makes anisotropic effects very important. Though the large-scale eddies are quasi two-dimensional (2D), the interaction of the flow with the bed generates 3D small-scale eddies.

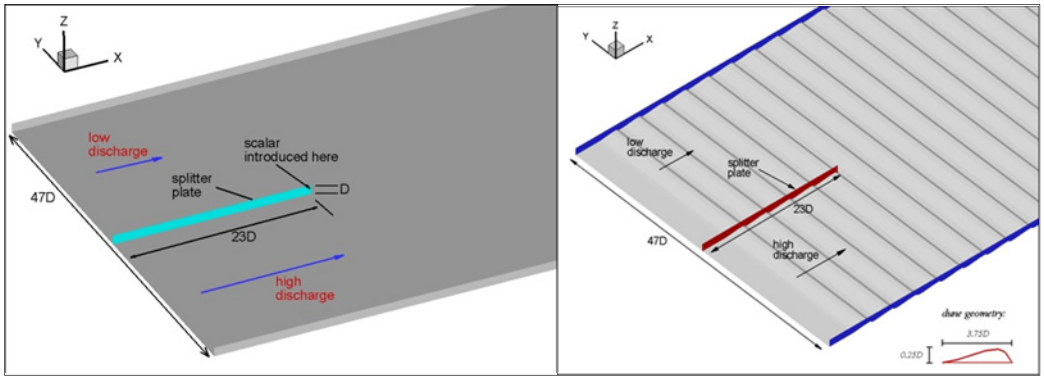


Figure 1. Sketch showing the computational domain around the splitter plate.

The development of a shallow mixing layer developing between two parallel streams with unequal bulk velocities U_{10} and U_{20} (the index '0' denotes in-coming flow values upstream of the splitter end) in a long open channel is investigated numerically using Detached Eddy Simulation (DES). Previous experimental investigations of shallow mixing layers with similar parameters (flow depth D, mean flow velocity $U_0=(U_{10}+U_{20})/2$) to the one considered in the present numerical study were reported by Chu and Babarutsi

(1987), Uijttewaal and Booij (2000) and van Prooijen and Uijttewaal (2002). In the numerical simulation, a splitter wall separates two fully-turbulent currents (see Fig. 1). As a result of the presence of the free surface and the bed, the vertical development of the large-scale turbulent structures in the mixing layer is constrained with respect to the widely studied case of a free mixing layer. The growth of the large-scale coherent structures in the mixing layer takes place mostly in the horizontal directions and is driven by the transverse shear induced by the difference in the mean velocities of the two streams.

Fully three-dimensional eddy-resolving numerical simulations have the advantage that they allow a detailed investigation of the variation of the turbulence characteristics over the depth of the mixing layer and of the vertical motions and associated vertical momentum transport. Such information is not yet available from experimental studies that concentrated on the investigation of the flow in a horizontal plane situated near or at the free surface (Chu and Babarutsi, 1988; Uijttewaal and Booij, 2000). In the numerical simulation, a splitter wall separates two fully-turbulent currents (see Fig. 1). As a result of the presence of the free surface and the bed, the vertical development of the large-scale turbulent structures in the mixing layer is constrained with respect to the widely studied case of a free mixing layer. The growth of the large-scale coherent structures in the mixing layer takes place mostly in the horizontal directions and is driven by the transverse shear induced by the difference in the mean velocities of the two streams.

Fully three-dimensional eddy-resolving numerical simulations have the advantage that they allow a detailed investigation of the variation of the turbulence characteristics over the depth of the mixing layer and of the vertical motions and associated vertical momentum transport. Such information is not yet available from experimental studies that concentrated on the investigation of the flow in a horizontal plane situated near or at the free surface (Chu and Babarutsi, 1988; Uijttewaal and Booij, 2000).

2 NUMERICAL SIMULATIONS

A general description of the DES code was given in Chang et al. (2007). The 3D incompressible Navier-Stokes equations were integrated using a fully-implicit fractional-step method. The governing equations were transformed to generalized curvilinear coordinates on a non-staggered grid. Convective terms in the momentum equations were discretized using a blend of fifth-order accurate upwind biased scheme and second-order central scheme. All other terms in the momentum and pressure-Poisson equations were approximated using second-order central differences. In the present DES simulation, the Spalart-Allmaras (SA) one-equation model was used. The time integration was done using a double time-stepping algorithm. The time discretization is second order accurate. The validation of the code including for simulations involving scalar transport was discussed in Chang et al. (2007).

The Reynolds number defined with the channel depth, D (=0.067mm), and the mean velocity of the two currents, U_0 (=0.23 m/s), is close to 15,500. The splitter length is 23D. The width of the channel is 47D and the length of the channel downstream of the splitter is 157D (=10.55 m) (Fig. 1). The channel bottom is smooth. The depth-averaged mean velocities of the two streams are $U_{10}=0.61U_0$ and $U_{20}=1.39U_0$, respectively. The mean velocities in the z/D=0.9 plane are $0.7U_0$ and $1.56U_0$, respectively. These conditions are similar to those present in the second test case studied by Uijttewaal and Booij (2000) where the streamwise velocity values at z/D~0.9 were used to estimate U_{10} and U_{20}. The bed-friction velocities non-dimensionalized by the bulk velocity in each stream are 0.052 and 0.048. The spatial development of the shallow mixing layer was analyzed using DES on a mesh containing close to 10 million cells (912*336*32 in the streamwise, spanwise and vertical directions). The viscous sub-layer was resolved in the simulation and no-wall functions were used. In a second simulation, an array of identical 2D dunes is present. The wavelength of the dunes is 3.75D and their height is 0.25D. The dunes have a generic shape that approximates dunes developing in medium-size rivers. The equivalent non-dimensional bed roughness is close to 500 wall units (fully-rough regime).

The incoming flow contains realistic turbulence fluctuations obtained from two preliminary simulations with periodic boundary conditions in the streamwise direction. The free surface was treated as a rigid lid which is justified as the channel Froude number (Fr=0.28) is much less than one. The wall surfaces are treated as no-slip boundaries. Mass exchange processes are studied by considering the transport of a passive scalar for which an advection-diffusion equation is solved. The passive scalar was introduced continuously at the end of the splitter plate over the whole depth of the channel. The Schmidt number was assumed equal to unity.

3 RESULTS

Comparison of the instantaneous concentration fields in horizontal planes situated near the bed (z/D=0.1) and the free surface (z/D=0.9) in Fig. 2 shows that the coherence of the quasi-2D eddies is decreasing as the channel bottom is approached. Also, there is a clear loss of the coherence of the large-scale eddies for x> 100D (~6.7 m, x is measured from the end of the splitter plate). This explains the observed decrease in the rate of growth of the mixing layer width in the streamwise direction (see discussion of Fig. 3) at all flow depths. The decay in the rate of growth increases with the decrease of the distance from the bed (see Kirkil and Constantinescu, 2008 for more details). The concentration contours in Fig. 2 suggest a rate of growth close to

zero at z/D=0.1 for x/D>120D. Close to the free surface the average size of the largest eddies in the transverse direction is around 10D-15D (0.67-1.00 m) in the downstream part of the channel (x/D>120).

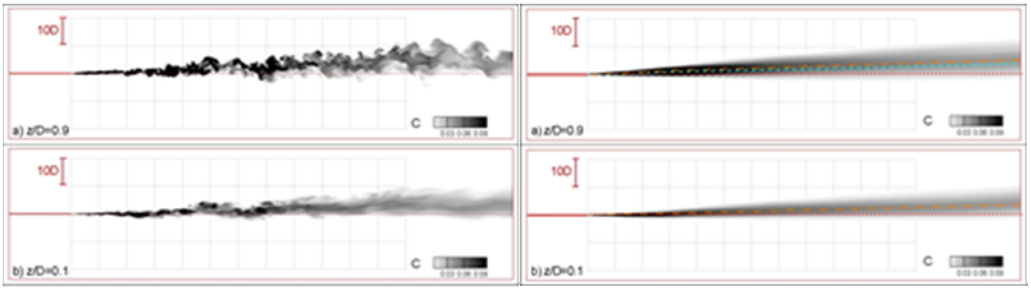


Figure 2. Contours of scalar concentration in the instantaneous (left) and mean (right) flow. a) z/D=0.9; b) z/D=0.1. The dashed-dot-dot line corresponds to the jet centerline determined from the mean concentration profiles. The dashed line corresponds to the jet centerline determined from the mean streamwise velocity profiles.

Comparison of the mean concentration profiles at z/D=0.9 and z/D=0.1 in Fig. 2 shows the positions of the centerline inferred from the mean streamwise velocity profiles and from the mean concentration fields are not identical. Based on the distribution of the mean concentration, the vertical tilt of centerline surface in the transverse direction is larger than 1.5D (~0.1 m) for x>120D. The vertical shift is slightly smaller in the case in which the velocity profiles were used to calculate the position of the centerline.

The development of the mixing-layer in the vertical direction is strongly non-uniform. This can be inferred from Fig. 2 where the width of the mixing layer inferred from the mean concentration field at a given streamwise location is significantly smaller at z/D=0.1 compared to z/D=0.9. Figure 3 allows a more quantitative comparison of the width of the mixing layer at different levels (z/D=0.1 and z/D=0.9) and in the depth-averaged flow. The width of the mixing layer in Fig. 3 was estimated based on the mean streamwise velocity field. The local width of the mixing layer is defined as the maximum slope thickness (Uijttewaal and Booij, 2000):

$$\delta = \frac{U_1 - U_2}{(\partial u/\partial y)_{max}} \qquad [1]$$

where $U_1(x)$ and $U_2(x)$ are the streamwise velocities in the two streams at a given streamwise location, y is the spanwise direction and u is the streamwise velocity. The curves in Figure 3 are obtained by using eqn. (1).

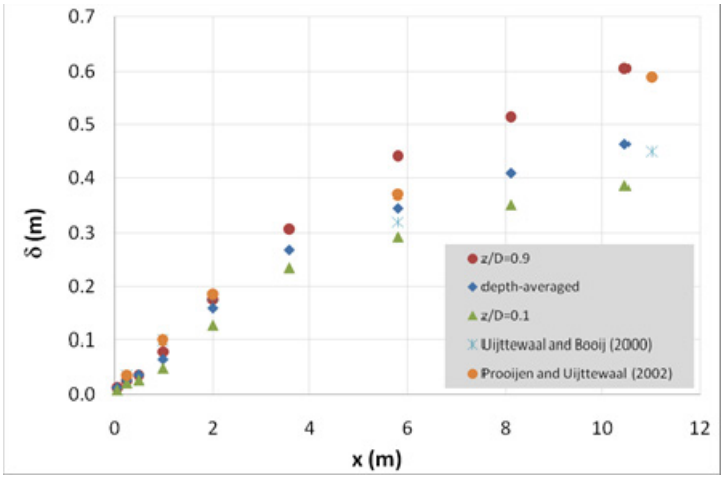


Figure 3. Mixing layer width vs. downstream distance from the splitter obtained from the profiles of the mean streamwise velocity. Also shown are two sets of experimental measurements performed close to the free surface.

The results at z/D=0.9 were compared in Fig. 3 with two sets of measurements conducted at the free surface (van Prooijen and Uijttewaal, 2002) and 1 cm below the free surface (Uijttewaal and Booij, 2000). Up to x=2 m (30D) the mixing layer width at z/D=0.9 predicted by DES is close to the widths inferred from the two sets of measurements. Farther downstream, DES overpredicts the width compared to the experimental values. For example, at x=6 m (91D) DES predicts a width of 0.44 m, van Prooijen and Uijttewaal (2002) estimated a width of 0.38 m (the model equation used by the same authors to fit a smooth curve through their

data predicted a value of 0.4 m), and Uijttewaal and Booij (2000) estimated a width of 0.32 m. At x=10.25 m (155D), DES predicts a width of 0.6 m. This is quite close to the value of 0.59 m estimated by van Prooijen and Uijttewaal (2002). The width (0.45 m) estimated by Uijttewaal and Booij (2000) is significantly lower. It is not entirely clear what is the reason for the relatively significant differences between the widths estimated from the two experiments, but DES predictions of the streamwise variation of the mixing layer width are clearly closer to the data of van Prooijen and Uijttewaal (2002).

Furthermore, the DES mean streamwise velocity fields were used to infer the streamwise variation of the mixing layer width close to the bed (z/D=0.1) and the streamwise variation of the width of the depth-averaged mixing layer. Such quantitative information on the vertical variation of the mixing layer width was not available from previous experiments. The differences with the values measured at or close to the free surface are significant for x>0.5 m (8D). For example, at x=2 m (30D) the mixing layer widths at z/D=0.9 and 0.1 are 0.185 m and 0.12 m, respectively, while the depth-averaged width is close to 0.16 m. At x=3.7 m, the depth averaged width (0.27 m) is roughly the mean of the values at z/D=0.1 (0.23 m) and z/D=0.9 (0.31). For x>4 m, the depth-averaged value is closer to the width measured close to the bed. This is because the width decreases relatively fast with the distance from the free surface in the upper layer of the channel and then much more gradually in the lower layer. For example, at x=10.25 m (155D) the width at z/D=0.1 (0.38 m) is about 30% lower than the width at z/D=0.9 (0.61 m). The width of the depth-averaged mixing layer is 0.47 m. The relative width reduction between z/D=0.9 and z/D=0.1 based on the velocity fields is consistent with the one estimated based on the mean concentration fields (e.g., see Fig. 3).

Figure 4 shows the power density spectrum of the spanwise velocity component at a point situated in the downstream part of the mixing layer (x=4.67 m), close to its centerline. A distinct -3 subrange is present at the low frequency side of the spanwise velocity spectrum. It indicates that in this region energy is in average transferred from the smaller eddies toward the larger turbulent eddies in the flow (inverse energy cascade). This process is responsible for the growth of the quasi 2D mixing layer eddies. By contrast, the spectrum of the vertical velocity (not shown) does not contain a -3 subrange and the energy at low Strouhal numbers (St=fD/U) corresponding to the large scales (St<0.1) is about one to two order of magnitude lower than the one in the spectrum for the spanwise velocity. This provides additional evidence of the fact that the large-scale eddies inside the mixing layer are horizontal quasi-2D eddies. A short -5/3 inertial subrange is observable for 0.2<St<1. The limited extent of the -5/3 subrange is due to the relatively small value of the Reynolds number of the simulation. The peak frequency in the -3 subrange of the spectrum of the spanwise velocity fluctuations measured by Uijttewaal and Booij (2000) at x=5.75 m was 0.14 Hz which corresponds to St~0.042. This is in good agreement with the corresponding peak frequency in Fig. 4 which is St=0.037.

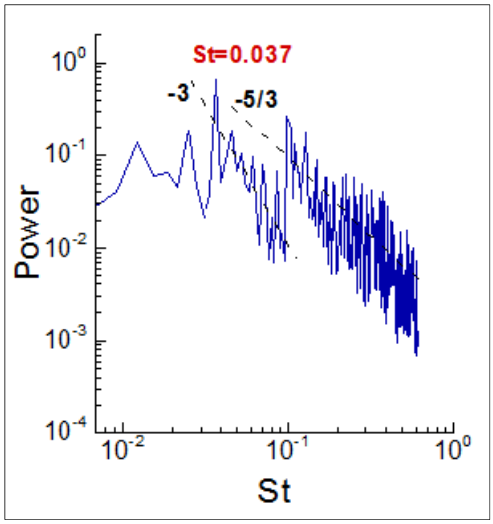


Figure 4. Power density spectra for spanwise velocity fluctuations at a point located close to the center of the mixing layer (z/D=0.9) and at 4.67 m from the splitter plate.

Figure 5 shows the autocorrelation function of the spanwise velocity fluctuations $R_{vv}(\tau)$ at two points situated close to the mixing layer centerline and the free surface (z/D=0.9). The streamwise distance from the splitter plate is 1.3 m and 10.25 m, respectively. The equation used to calculate $R_{vv}(\tau)$ is

$$R_{vv}(\tau) = \frac{< v'(t_0)v'(t_0+\tau) >}{(< v'2(t_0) >< v'2(t_0+\tau) >)} \qquad [2]$$

where the brackets < > indicate averaging over all values of t0. The autocorrelation function shows the presence of strong modulations in time at both positions. The period of these modulations is larger at x=10.25 m compared to x=1.32 m, which indicates the dimensions of the large-scale mixing layer eddies increases in the streamwise direction. The autocorrelation functions show a fast decay at small values of the displacement time τ followed by a slow oscillatory regime. The average time delay between two successive peaks within the slow oscillatory regime corresponds to the time of passage of the large-scale eddies in the z/D=0.9 plane. The fact that the correlation is still strong after several peaks (this regime covers at least 20,000 ms based on our results) indicates that coherent structures passing at a certain location are fairly well correlated with respect to each other.

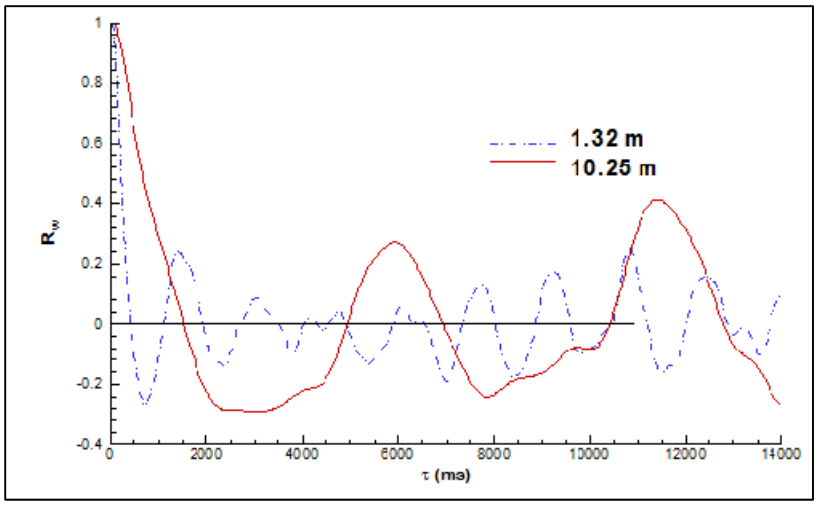


Figure 5. Autocorrelation functions of spanwise velocity fluctuations in the center of the mixing layer at 1.32 m (20D) and 10.25 m (150D) downstream of the splitter plate.

The characteristic time scale of the large-scale coherent structures estimated from Fig. 5 is 1,500 ms at x=1.32 m and 6,000 ms at x=10.25 m. These time scales are close to the time intervals between the first (τ=0 ms) and the second maximum of the oscillation, that were used by Uijttewaal and Booij (2000) to infer the characteristic time scale of the eddies. They estimated a time scale of 1,000 ms at x=0.5 m and of 2,800 ms at x=2.0 m, which suggest a time scale of 1,800 ms at x~1.3 m. At x=11 m the time scale they estimated was close to 10,000 ms which again indicates reasonable agreement between experiment and simulation. Using the average convection velocity in the z/D=0.9 plane at the two streawise locations and the time scale associated with the passage of these eddies, one can estimate the average size of the mixing layer eddies in the streamwise direction. This assumes that the eddy lifetime is much larger than the passage time of the eddy at that location. The values predicted by DES are 0.39 m at x=1.32 m and 1.56 m at x=10.25 m. This average size of the eddies in the streamwise direction is expected to correlate with the mixing layer width (0.13 m at x=1.32 m and 0.61 m at x=10.25 m). Indeed, the ratio is between 2.5 and 3. The fact that the ratio is larger than one indicates that the large-scale eddies are not circular. Rather, they are elongated in the streamwise direction, which is consistent with visualizations of the instantaneous concentration fields (e.g., see Figure 2) and their temporal evolution observed in animations.

Finally, some preliminary results from the simulation containing large-scale roughness elements in the form of an array of 2D dunes are discussed. The presence of the dunes increases significantly the total bed friction with most of the increase due to the form or pressure drag component. Figure 6 shows the instantaneous concentration contours at the free surface along with the centerline inferred from the mean concentration contours. Several differences between the instantaneous concentration distributions in the z/D=0.9 plane in Figs. 2a and Fig.6 are observed. The concentration levels are lower in the case containing dunes. This is due to the scalar that is trapped in the recirculation region past the crest of the dunes and that diffuses in the spanwise direction, predominantly toward the lower speed side of the mixing layer. More importantly, the larger equivalent bed roughness induces a faster decrease of the entrainment coefficient which translates into a faster and larger shift of the mixing layer centerline (Fig. 7). For example, in the x=58D section, s/D=2.24 in the flat bed case and s/D=3.35 in the case with dunes. These values were obtained based on the mean concentration profiles. The values in the x=148D section are 4.79D and 7.16D, respectively. In fact, preliminary results suggested that the equivalent bed roughness is high enough such that the growth rate of the mixing layer, in terms of its width and centerline shift, is almost zero for x/D>130 in the case containing dunes (Fig. 4).

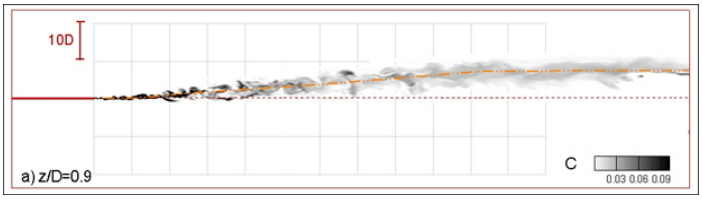


Figure 6. Instantaneous contours of scalar concentration for deformed bed case in a horizontal plane situated at 0.9D from the free surface.

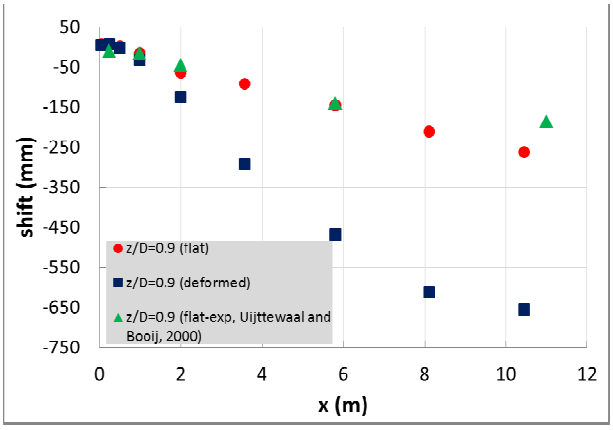


Figure 7. Variation of lateral shift of jet centreline with streamwise distance based on the analysis of flow fields at z/D=0.9

4 CONCLUSIONS

DES predictions of the shallow mixing layer development close to the free surface were found to agree reasonably well with experimental observations. Meanwhile, the numerical results showed that the vertical variations of the width and centerline position of the shallow mixing layer are significant. For example, at distances between 75D and 150D from the splitter plate, the width of the mixing layer close to the free surface is 20-30% more than the width in the near-bed region.

Power spectra of the horizontal velocity components captured the presence of a -3 subrange at streamwise locations situated more than 10D from the splitter plate. This is consistent with the presence of large-scale quasi two-dimensional horizontal eddies and the inverse energy cascade that feeds energy into the quasi two-dimensional large-scale eddies. Close to the free surface, the estimated streamwise length of the quasi 2D mixing layer eddies is about 2.5 to 3.0 times larger than the local width of the mixing layer. The larger size of the quasi-2D eddies suggested by the autocorrelation function is consistent with visualizations of the convection of a passive scalar within the mixing layer.

The length of the domain was not sufficient for the mixing layer to reach the zero-growth regime in which the bottom friction effects become so large that the large-scale horizontal eddies cannot be sustained by the lateral shear and their coherence is gradually lost. The spectral properties of the turbulence within a mixing layer that reaches this regime due to increased bed roughness associated with the presence of 2D dunes will make the object of a future paper. nd numbered consecutively, as in Eq. [1]. An alternative method is given in Eq. [2] for long sets of equations where only one referencing equation number is wanted.

REFERENCES

Chang, K., Constantinescu, G. & Park, S. O. (2007). Assessment of Predictive Capabilities of Detached Eddy Simulation to Simulate Flow and Mass Transport Past Open Cavities. *Journal of Fluids Engineering*, 129(11), 1372-1383.

Chu, V. H. & Babarutsi, S. (1988). Confinement and Bed-Friction Effects in Shallow Turbulent Mixing Layers. *Journal of hydraulic engineering*, 114(10), 1257-1274.

Constantinescu, G. & Squires, K. (2004). Numerical Investigations of Flow over a Sphere in the Subcritical and Supercritical Regimes. *Physics of fluids*, 16(5), 1449-1466.

van Prooijen, B. and Uijttewaal, W. (2002). A linear approach for the evolution of coherent structures in shallow mixing layers. *Physics of fluids*, 14(12), 4105-4114.

Uijttewaal, W.S. and Booij, R. (2000). Effects of shallowness on the development of free-surface mixing layers. *Physics of fluids*, 12(2), 392-402.

Proceedings of the 37th IAHR World Congress
August 13 – 18, 2017, Kuala Lumpur, Malaysia

URBAN FLOOD SIMULATION USING CHRE2D MODEL

JENNIFER G DUAN[(1)]

[(1)] *Dept. of Civil Engineering and Engineering Mechanics, University of Arizona, Tucson, AZ 85721. Email:gduan@email.arizona.edu*
gduan@email.arizona.edu

ABSTRACT

CHRE2D is a two-dimensional hydrodynamic and sediment transport model that simulates surface flow routing and sediment transport using numerical solutions of shallow water equations and the kinematic or diffusion wave approximation. The shallow water equations are discretized by the first-order Godunov-type finite volume method. The stability analysis showed that the friction source term increased exponentially as the flow depth was very small. This breaks the balance between the friction and the slope source terms. An approximate solution to the momentum equation, kinematic or diffusion wave approximation, are introduced to compensate this balance. This technique enables the numerical model is accurate, robust and stable for both very shallow overland (*e.g.,* 10^{-10} m) and concentrated channel flows. The resulted CHRE2D model is capable of simulating both hydrological flow (*e.g.,* surface flow routing) and hydraulic flow (*e.g.,* dam break), which has not been achieved in similar commercial software, such as FLO2D, ARM2D. Additionally, the CHRE2D model implemented the Grass-type sediment transport formula to simulate the total sediment load in both overland flow and channel flow.

Keywords: Computational hydraulics; CFD model; flooding; sediment transport; erosion.

1 INTRODUCTION

Surface flow consists of two parts: overland flow and channel flow (Beven, 2012; Eagleson, 1970; Ponce, 1989). Overland flow, also called sheet flow, is the flow of a relatively thin layer of water on land surface. Channel flow occurs when overland flow reaches rills, gullies, ditches or rivers. The most significant difference between overland and channel flow is the flow depth in channel flow is much larger than that in overland flow. The shallow water equations (SWE) and its simplifications, *i.e.,* the diffusion wave equations (DWE) and the kinematic wave equation (KWE), are often used to model the movement of surface flow (Abbott, 1979; Freeze and Harlan, 1969; Singh, 1996). Both the diffusion and kinematic wave equations are simplifications of the shallow water equations (Ferrick, 1985; Vieira, 1983). The theoretical limits and accuracies of the diffusion and kinematic wave equations have been extensively analyzed and discussed (Ponce, 1991; Singh, 2002).

Modeling surface flow over watershed needs to simulate both overland and channel flow (Beven, 2012; Borah, 2011; Freeze and Harlan, 1969; Singh and Frevert, 2006), which often requires two separate flow models. In the KINEROS model (Woolhiser et al., 1990), both overland and channel flow are modeled by solving the kinematic wave equation so that it cannot simulate the backwater effect. As to the Systeme Hydrologique Europeen (SHE) model (Abbott et al., 1986), the CASCade of planes, 2-Dimensional (CASC2D) model (Julien et al., 1995) and the Gridded Surface Subsurface Hydrologic Analysis (GSSHA) model (Downer and Ogden, 2004), the diffusion wave equation is employed for both overland and channel flow simulations. Other watershed models similar to the ones mentioned earlier include Panday and Huyakorn (2004), Ivanov et al. (2004), Qu and Duffy (2007), He et al. (2008), Bates et al. (2010) and Lopez-Barrera et al. (2012).

The CHRE2D model is a unified surface flow model based on the solutions of shallow water equations and the diffusion wave approximation (Yu and Duan, 2012; 2014; 2017). The expected flow depth of the model ranges from 10^{-10} m to 10^{3} m. And, the DEM data will be directly used in the model without removing depressions from raw data. Besides, for the proposed model, it is not necessary to couple a channel network to the computational mesh. To accurately predict the effect of flood inundation, the model is capable of capturing the sharp front of flood wave with the minimum numerical dissipation.

Additionally, surface flows, especially flash flood flows, usually carry a large amount of sediment that can induce geomorphologic changes along its path. Sediment transport module is also included in the CHRE2D model. For unsteady flows, spatial variations of sediment concentrations result in significant differences in the spatial distributions of flow densities. Since the densities are not constants but spatial functions, dam-break flows are often modeled by the variable density shallow water equations (VDSWEs) (Cao et al., 2004; Guan et al., 2014; Leighton et al., 2010; Li et al., 2013; Rosatti et al., 2008).

To account for its hyperbolic nature, the VDSWEs is often solved by the Godunov-type finite volume method (LeVeque, 2002; Toro, 2009). Being different from the SWEs, flow density in the VDSWEs is an independent variable. This difference brings significant impacts into the VDSWEs. Brufau et al. (2000) and Cao et al. (2004) reformulated the VDSWEs and moved the density-related terms to the right-hand side of the VDSWEs. After the reformulation, the left-hand side of the VDSWEs has the same form as the SWEs, while

the right-hand side has two more terms than the SWEs. Cao et al. (2004) pointed out that one term represents the effect of spatial variation of flow density and the other term describes the momentum exchange between flow and bed. Since the modified VDSWEs have the same left-hand side as the SWEs, it is common to apply the previously well-developed Godunov-type SWEs solvers to solve the VDSWEs and to discretize the source terms by a central difference scheme (Cao et al., 2004; Guan et al., 2014; Li et al., 2013).
The CHRE2D model treated sediment transport in non-equilibrium state that requires a spatial lag distance to reach its equilibrium concentration (Phillips and Sutherland, 1989). The model is the first that uses a well-balanced numerical scheme to solve the original/conservative formulation of the VDSWEs in a coupled way. The paper will present the governing equations for both surface flow and sediment transport model, but neglect the numerical schemes. Selected simulation results will be compared with laboratory and field measurements.

2 GOVERNING EQUATIONS OF SURFACE FLOW MODEL WITHOUT SEDIMENT TRANSPORT

The governing equations are the depth-averaged Navier-Stokes equations without turbulence diffusion terms in the momentum equations. Those equations include mass conservation equation and two momentum equations are also called the shallow water equations. The mass conservation equation is:

$$\frac{\partial h}{\partial t}+\frac{\partial (hu)}{\partial x}+\frac{\partial (hv)}{\partial y}=i_0 \tag{1}$$

and the momentum conservation equations are:

$$\begin{cases}\dfrac{\partial (hu)}{\partial t}+\dfrac{\partial}{\partial x}(hu^2+\dfrac{1}{2}gh^2)+\dfrac{\partial (huv)}{\partial y}=ghS_{0x}-C_f\left\|\mathbf{u}\right\|u \\ \dfrac{\partial (hv)}{\partial t}+\dfrac{\partial (huv)}{\partial x}+\dfrac{\partial}{\partial y}(hv^2+\dfrac{1}{2}gh^2)=ghS_{0y}-C_f\left\|\mathbf{u}\right\|v\end{cases} \tag{2}$$

where t is the time, x and y are the spatial coordinates, respectively, h is the flow depth, u and v are the components of flow velocity in x and y directions, respectively, i_0 is the rate of rainfall excess, g is the gravity acceleration, $S_{0x}=-\dfrac{\partial b}{\partial x}$ and $S_{0y}=-\dfrac{\partial b}{\partial y}$ are x and y components of bed slope, respectively, b is the bed elevation, $C_f=gn^2h^{-1/3}$ is the drag coefficient, n is the Manning's roughness coefficient, $\mathbf{u}=(u \quad v)^T$ is the vector of flow velocity, and $\left\|\mathbf{u}\right\|=\sqrt{u^2+v^2}$ is the modulus of flow velocity.
The shallow water equations can also be written in vectorial form as:

$$\frac{\partial \mathbf{Q}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}=\mathbf{S}_0-\mathbf{S}_f+\mathbf{S}_r$$

$$\mathbf{Q}=\begin{pmatrix}h\\hu\\hv\end{pmatrix} \qquad \mathbf{F}=\begin{pmatrix}hu\\hu^2+gh^2/2\\huv\end{pmatrix} \quad \mathbf{G}=\begin{pmatrix}hv\\huv\\hv^2+gh^2/2\end{pmatrix} \tag{3}$$

$$\mathbf{S}_0=\begin{pmatrix}0\\ghS_{0x}\\ghS_{0y}\end{pmatrix} \quad \mathbf{S}_f=\begin{pmatrix}0\\C_f\left\|\mathbf{u}\right\|u\\C_f\left\|\mathbf{u}\right\|v\end{pmatrix} \qquad \mathbf{S}_r=\begin{pmatrix}i_0\\0\\0\end{pmatrix}$$

where $\mathbf{Q}$ is the vector of conservative variables, $\mathbf{F}$ and $\mathbf{G}$ are x and y components of advective flux, respectively, $\mathbf{S}_0$ is the slope source term, $\mathbf{S}_f$ is the friction source term, and $\mathbf{S}_r$ is the rainfall source term. In addition, $\mathbf{U}=\begin{pmatrix}h & u & v\end{pmatrix}^T$ is used in the paper to represent the vector of primitive variables.

Although extensive research has been carried out on the stability of different numerical schemes (positivity preserving property and well-balanced property), few study exists that adequately covers the stability of very thin (10^{-3} m $\sim 10^{-10}$ m) flow, *i.e.,* overland flow. So, a common practice is to define a tolerance parameter: If the flow depth is lower than the tolerance, then the cell is considered to be a dry cell (Begnudelli and Sanders, 2007; Cea et al., 2010; Costabile et al., 2013). In this study, a novel correction method is proposed in this study based on a physical approximation of the shallow water equations, *i.e.,* the diffusion wave approximation (DWA). According to the theory of kinematic wave number (Vieira, 1983; Woolhiser and Liggett, 1967), the diffusion wave approximation becomes an accurate approximation of the shallow water equations as the flow depth is reduced. So, instead of updating the flow velocities by the momentum equations, when the flow depth is less than a threshold value (10^{-3} m), the flow velocities are updated implicitly by the diffusion wave approximation:

$$\begin{cases} u^{(n+1)} \left\| \mathbf{u}^{(n+1)} \right\| = -\eta_x \dfrac{(h^{(n+1)})^{4/3}}{n^2} \\ v^{(n+1)} \left\| \mathbf{u}^{(n+1)} \right\| = -\eta_y \dfrac{(h^{(n+1)})^{4/3}}{n^2} \end{cases} \tag{4}$$

Or Eq. (4) can be rewritten as:

$$u^{(n+1)} = \begin{cases} +\dfrac{1}{n} \dfrac{\sqrt{-\eta_x}}{(1+(\eta_y/\eta_x)^2)^{1/4}} (h^{(n+1)})^{2/3}, & \text{if } \eta_x < 0 \\ 0, & \text{if } \eta_x = 0 \\ -\dfrac{1}{n} \dfrac{\sqrt{\eta_x}}{(1+(\eta_y/\eta_x)^2)^{1/4}} (h^{(n+1)})^{2/3}, & \text{if } \eta_x > 0 \end{cases} \tag{5}$$

And

$$v^{(n+1)} = \begin{cases} +\dfrac{1}{n} \dfrac{\sqrt{-\eta_y}}{(1+(\eta_x/\eta_y)^2)^{1/4}} (h^{(n+1)})^{2/3}, & \text{if } \eta_y < 0 \\ 0, & \text{if } \eta_y = 0 \\ -\dfrac{1}{n} \dfrac{\sqrt{\eta_y}}{(1+(\eta_x/\eta_y)^2)^{1/4}} (h^{(n+1)})^{2/3}, & \text{if } \eta_y > 0 \end{cases} \tag{6}$$

Eq. (5) and (6) are velocities solved by the diffusion wave approximation. Both equations are implicit, which require the solutions of flow depth at $n+1$ time step. Since the flow velocities are not inversely proportional to the flow depth now, the stability of the numerical solution is maintained by the approximation. Besides, for the still water situation, zero surface gradients lead to zero velocities. So, the diffusion wave approximation also satisfies the well-balance property.

3 MATHEMATICAL MODEL OF VARIABLE DENSITY SEDIMENT LADEN FLOW

In this paper, the dam-break flows are described by the variable density shallow water equations (VDSWEs). The system of governing equations is based on the two-phase flow assumption, *i.e.,* the flow of water and sediment mixture can be treated as continuums, and both phases are moving together at the same velocities. The VDSWEs consist of the mass and momentum conservation equations for flow and sediment. For a given control volume, the bulk volume conservation equation for sediment-laden flow is written as:

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = S_b \tag{7}$$

where t is time, x and y are the spatial coordinates, respectively, h is flow depth, u and v are the depth-averaged flow velocities in x and y directions, respectively, and S_b (m/s) is the sediment flux between flow and mobile bed. For the sediment-laden flow, since the density of flow varies with sediment concentration, the mass conservation equation of sediment laden flow is expressed as:

$$\frac{\partial(\rho h)}{\partial t}+\frac{\partial(\rho hu)}{\partial x}+\frac{\partial(\rho hv)}{\partial y}=\rho_b S_b \tag{8}$$

where ρ is the density of sediment water mixture, $\rho_b=\phi\rho_w+(1-\phi)\rho_s$ is the density of saturated mobile bed, ϕ is the porosity of mobile bed material, ρ_w is the density of clear water, and ρ_s is the density of sediment. The concentration of sediment is calculated by $C=\frac{\rho-\rho_w}{\rho_s-\rho_w}$. The depth-averaged momentum conservation equations in x and y directions are given as:

$$\frac{\partial(\rho hu)}{\partial t}+\frac{\partial}{\partial x}(\rho huu+\frac{1}{2}\rho gh^2)+\frac{\partial(\rho huv)}{\partial y}=\rho ghS_{0x}-\rho C_f\|\mathbf{u}\|u \tag{9}$$

$$\frac{\partial(\rho hv)}{\partial t}+\frac{\partial(\rho hvu)}{\partial x}+\frac{\partial}{\partial y}(\rho hvv+\frac{1}{2}\rho gh^2)=\rho ghS_{0y}-\rho C_f\|\mathbf{u}\|v \tag{10}$$

where g is the gravitational acceleration, $S_{0x}=-\frac{\partial(b_0+b)}{\partial x}$ and $S_{0y}=-\frac{\partial(b_0+b)}{\partial y}$ are the x and y components of bed slope, respectively, b_0 is the elevation of immobile bed, b is the depth of mobile bed, $C_f=gn^2h^{-1/3}$ is the drag coefficient, n is Manning's roughness coefficient, $\|\mathbf{u}\|=\sqrt{u^2+v^2}$ is the magnitude of flow velocity, and $\mathbf{u}=\begin{pmatrix}u\\v\end{pmatrix}$ is the velocity vector.

The system of governing equations described above, *i.e.,* Eq. (7-10), can be written in a vectorial form as:

$$\frac{\partial\mathbf{Q}}{\partial t}+\frac{\partial\mathbf{F}_x(\mathbf{Q})}{\partial x}+\frac{\partial\mathbf{F}_y(\mathbf{Q})}{\partial y}=\mathbf{S}_0(\mathbf{U})-\mathbf{S}_f(\mathbf{U})+\mathbf{S}_b(\mathbf{U}) \tag{11}$$

where $\mathbf{U}$ and $\mathbf{Q}$ are the vectors of primitive variables and conservative variables, respectively:

$$\mathbf{U}=\begin{pmatrix}h\\\rho\\u\\v\end{pmatrix},\ \mathbf{Q}=\begin{pmatrix}h\\\rho h\\\rho hu\\\rho hv\end{pmatrix} \tag{12}$$

and $\mathbf{F}_x(\mathbf{Q})$ and $\mathbf{F}_y(\mathbf{Q})$ are the vectors of fluxes:

$$\mathbf{F}_x(\mathbf{Q})=\begin{pmatrix}hu\\\rho hu\\\rho huu+\frac{1}{2}\rho gh^2\\\rho huv\end{pmatrix},\ \mathbf{F}_y(\mathbf{Q})=\begin{pmatrix}hv\\\rho hv\\\rho hvu\\\rho hvv+\frac{1}{2}\rho gh^2\end{pmatrix} \tag{13}$$

and $\mathbf{S}_0(\mathbf{U})$, $\mathbf{S}_f(\mathbf{U})$, and $\mathbf{S}_b(\mathbf{U})$ are the bed slope, bed friction, and bed material source terms, respectively:

$$\mathbf{S}_0(\mathbf{U}) = \begin{pmatrix} 0 \\ 0 \\ \rho g h S_{0x} \\ \rho g h S_{0y} \end{pmatrix}, \mathbf{S}_f(\mathbf{U}) = \begin{pmatrix} 0 \\ 0 \\ C_f \|\mathbf{u}\| u \\ C_f \|\mathbf{u}\| v \end{pmatrix}, \mathbf{S}_b(\mathbf{U}) = \begin{pmatrix} S_b \\ \rho_b S_b \\ 0 \\ 0 \end{pmatrix} \tag{14}$$

Eq. (11) represents a system of time dependent, nonlinear hyperbolic partial differential equations. This system may result in sharp and discontinuous solutions even if starting from continuous initial conditions. When this happens, the non-conservative formulation by Cao et al. (2004) is ambiguous from the mathematical point of view (Cozzolino et al., 2014).

In addition to the governing equations for the flow simulation, the evolution of mobile bed is described by the following conservation equation of bed material:

$$\frac{\partial b}{\partial t} = -S_b \tag{15}$$

The bed material flux between flow and mobile bed is evaluated by the nonequilibrium sediment transport formula. The formula reads:

$$S_b = \frac{1}{1-\phi} \frac{(q_b - q_b^*)}{L} \tag{16}$$

where L is the nonequilibrium adaption length of sediment transport, q_b is the actual flux of sediment transport, and q_b^* is the sediment transport capacity or the equilibrium sediment transport rate. The nonequilibrium adaption length denotes the spatial lag between actual sediment transport rate and its saturation/equilibrium rate. The calculation of adaption length is:

$$L = \max\left(L_b, \frac{h\|\mathbf{u}\|}{\alpha_0 \omega_0}\right) \tag{17}$$

where L_b is the adaption length of bed-load, α_0 is the adaption coefficient of suspended-load, and ω_0 is the settling velocity of a single sediment particle. Following El Kadi Abderrezzak and Paquier (2011), $L_b = 0.25\ \mathrm{m}$ and $\alpha_0 = 2$ are employed in the proposed model.

The settling velocity of sediment particle is calculated by (*Cao*, 1999):

$$\omega_0 = \sqrt{(13.95\nu / d_{50})2 + 1.09 s g d_{50}} - 13.95\nu / d_{50} \tag{18}$$

where ν is the kinematic viscosity of water, $s = \rho_s / \rho_w - 1$ is the specific gravity of submerged sediment particle, and d_{50} is the median diameter of sediment particle. The sediment transport capacity is calculated by the modified Meyer-Peter-Müller (MPM) formula (El KAdi Abderrezzak and Paquier, 2011; Van Emelen et al., 2014):

$$q_b^* = 12\sqrt{s g d_{50}^3}(\theta - \theta_c)^{1.5} \tag{19}$$

where $\theta = u_*^2 / sgd$ is the Shields number, $u_* = C_f^{1/2}\|\mathbf{u}\|$ is the friction velocity, and $\theta_c = 0.047$ is the critical Shields number.

4 MODEL TEST

4.1 Test case 1: Santa Cruz Watershed in Tucson, AZ

The Santa Cruz River Watershed (SCRW), located in the south-central Arizona, is a transboundary watershed at an elevation ranging from 668.12 to 2,845.92 m. The Santa Cruz River (SCR) is an ephemeral river that drains into the Gila River, a tributary to the Colorado River. The SCR flows to the south and makes one 40.22-kilometer loop through Mexico before re-entering the United States at nearly five miles from Nogales, Arizona. Then, the river flows northward to its confluence with the Gila River. The study area is the upper Santa Cruz River watershed encompassing the reach of the SCR within the US, approximately 4,000 square kilometers. The largest tributary to this reach is the Rillito River in Tucson, Arizona. The topographic data was from the LIDAR survey in 2005 by the Pima County Regional Flood Control District. Although the LIDAR data has a fine grid resolution, the computational cell was obtained by aggregating the LIDAR data to $100\,\mathrm{m}\times 100\,\mathrm{m}$ cell. The simulation domain is approximately $40,000\,\mathrm{m}\times 100,000\,\mathrm{m}$, so the total number of cells is 400,000. The simulation period was from 07/27/06 to 08/04/06 totaling 216 hours, with a peak flow of 37,913.73 cfs occurring at Dodge blvd bridge at 6:50 AM on 07/31/06. The hourly gridded precipitation data with a resolution of 32 km was obtained from NLDAS (http://ldas.gsfc.nasa.gov/nldas/NLDAS2forcing.php). The channel transmission loss was treated similarly as the infiltration loss. Two infiltration rates are defined at each cell, one is for the overland flow, and the other is for the channel flow. If flow depth in a cell is greater than 0.1 m, the flow is considered as channel flow, and the transmission loss is calculated based on the conductivity of channel flow. The soil texture types in the study area are loam (92.3%) and silt (7.7%). To simplify the computation, the model assigned loam soil type to each cell. Loam soil consists of 40% sand, 40% silt, and 20% clay, and the saturated hydraulic conductivity (K_s) was calibrated as 0.0162 m/hr for overland flow and 0.0324 m/hr for channel flow, respectively. The capillary pressure head at the wetting front, H_f, is set as 11.01 cm and the calibrated one is 17 cm. The soil moisture deficit, M_d, is 0.29 in the model (Rawls *et al*., 1983). The Manning's roughness coefficient is 0.03. The grid size is 100 m by 100 m. The time step ranges from 1.0 s, and it takes about 1.5 hours to run a 24-hour storm event. The model runs on a 12-core DELL server. The simulated flow inundation maps at different times are shown in Figure 1. By using different infiltration parameters, the simulated hydrographs were compared with measurements at the Cortaro gauge (Figure 2).

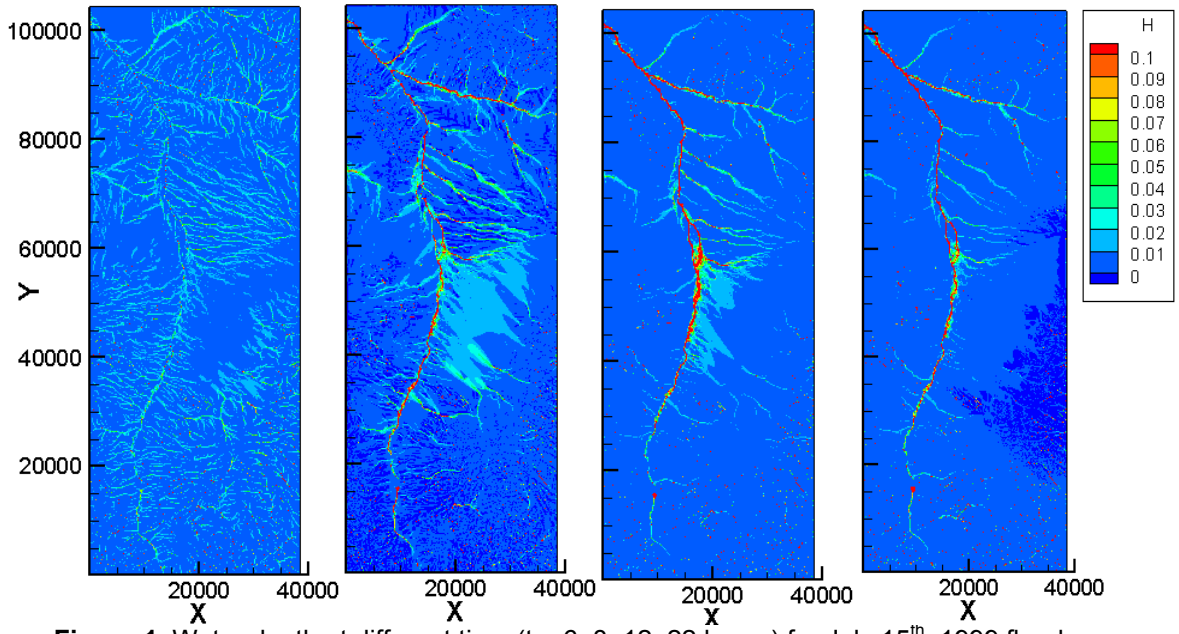


Figure 1. Water depth at different time (t = 6, 8, 12, 22 hours) for July 15th, 1999 flood.

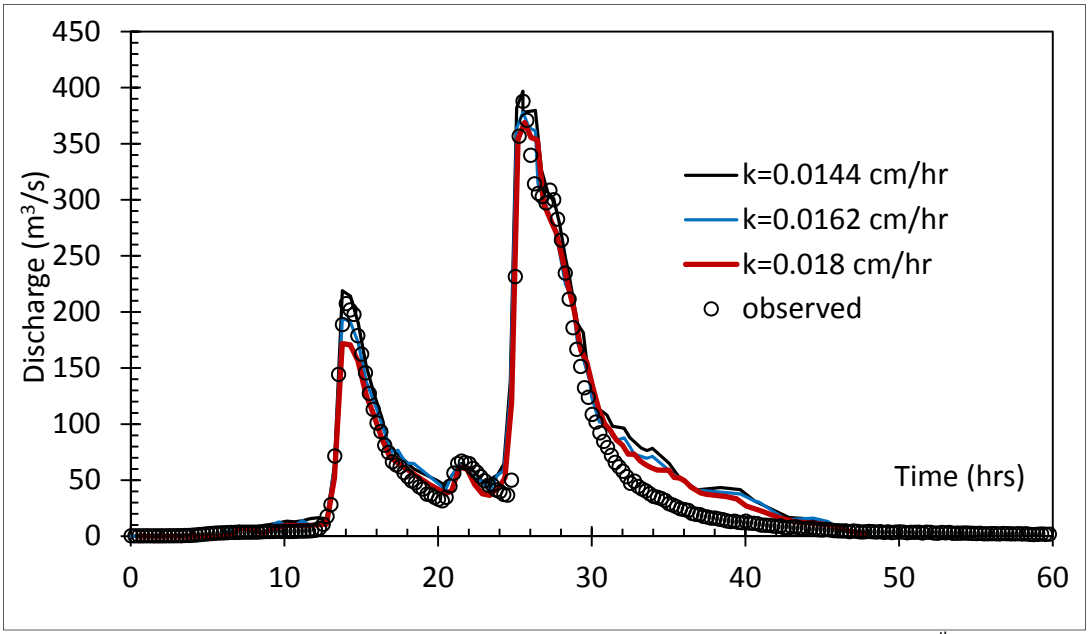


Figure 2. Measured and simulated hydrograph at Cortaro gauge of July 15th, 1999.

4.2 Test case 2: 1996 Lake Ha! Ha! catastrophic flood event

In this part, the performance of the proposed model is tested against a field case: the 1996 Lake Ha! Ha! catastrophic flood event, in the Saguenay region of Quebec, Canada. A detailed description of this flood event and extensively documented data are provided by Capart et al. (2007). Brooks and Lawrence (1999) gave a detailed geomorphic description about the flood. El Kadi Abderrezzak and Paquier (2009) carried out a 1D numerical simulation of unsteady flow and sediment transport during this event.

From July 18 to 21, 1996, an extreme precipitation event affected the Saguenay region of Quebec, Canada. At the Ha! Ha! Lake, an earthfill dyke was being overtopped by up to 0.26 m of water and a new outlet channel formed. Overall, 59×10^6 m^3 of water was estimated to have drained from the lake. The failure of the dyke resulted in a peak discharge of 8 times the 100-year flood. The Ha! Ha! River was severely damaged by the resulting flood flow (Brooks and Lawrence, 1999).

The numerical simulation started with the digital elevation model (DEM) of the Ha! Ha! River, which was surveyed in May 1994 (Capart et al., 2007). The spatial data is based on the Modified Transverse Mercator (MTM) projection, zone 7 coordinates (NAD83). The spatial ranges of the DEM data are: $275{,}000\ \text{m} \le x \le 282{,}000\ \text{m}$ on the east-west direction, and $5318000\ \text{mN} \le y \le 5354000\ \text{mN}$. In addition to the DEM data, the geometry data of evenly spaced cross sections are also provided. These 363 cross sections spaced at 100 m interval and are oriented approximately normal to the axis of the Ha! Ha! River. Each cross section is identified by its streamwise distance measured from the failed dyke. In spite of abundant data, it has to be pointed out that the average error of the preflood riverbed elevations is estimated to be about 2 m (El Kadi Abderrezzak and Paquier, 2009).

Comparisons of the selected cross sections along the river are shown in Figure 3. In the longitudinal direction, the measured and calculated thalwegs are plotted in Figure 4. The RMSE of thalweg change is 5.3 m, and the NRMSE is 20.18%. It is obvious that the model performed better in the longitudinal direction than in the cross-sectional direction. Also, the simulated thalweg profile at the middle reach is overestimated, especially at the 23 km from the channel mouth. According to El Kadi Abderrezzak and Paquier (2009), this part of the channel is controlled by outcrops of bedrocks. During the 1996 flood event, a new reach formed at the right floodplain and the bed was eroded up to 20 m there. The proposed model failed to predict this geomorphic change.

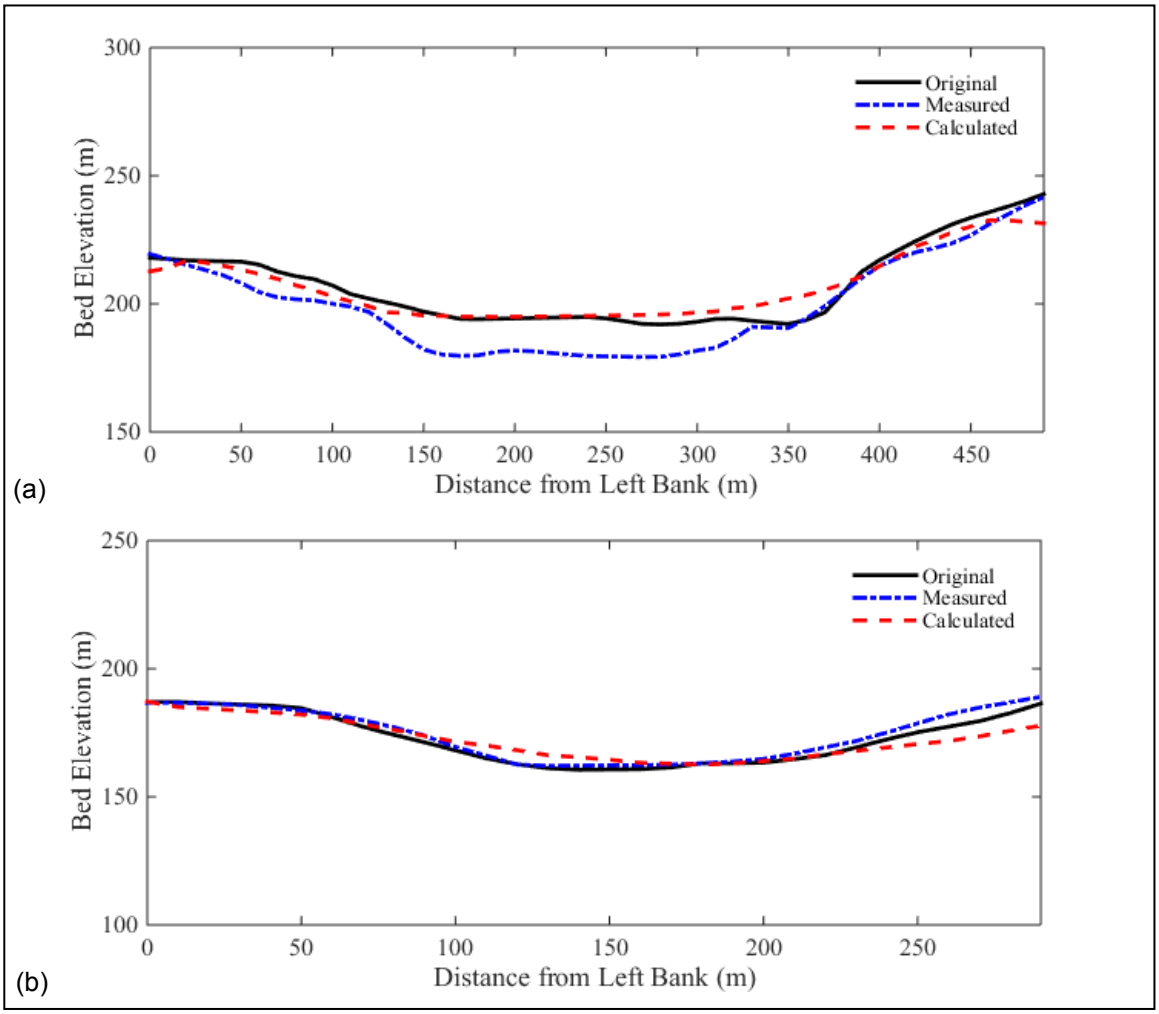


Figure 3. Comparison of selected measured and calculated bed cross sections.

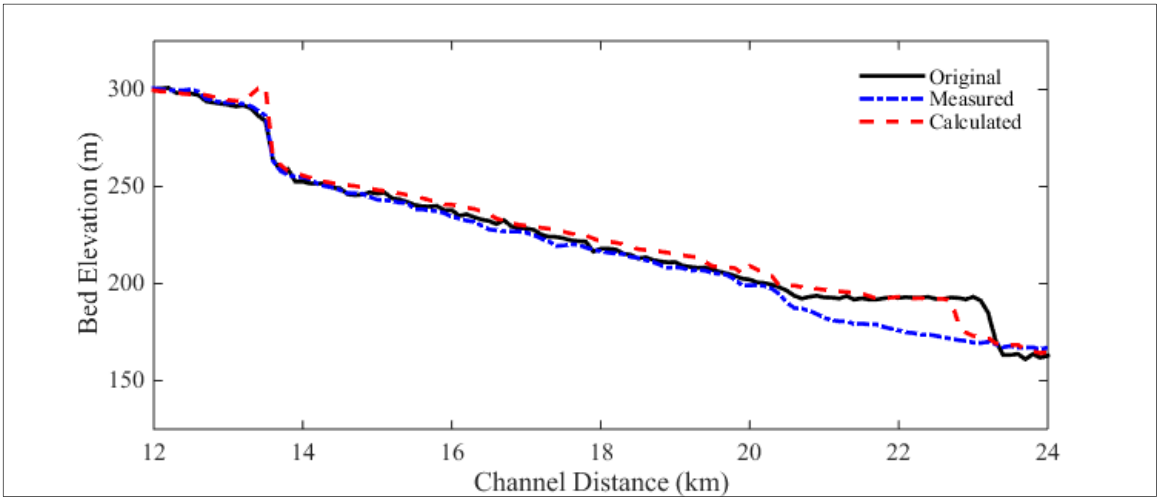


Figure 4. Measured and calculated thalwegs from 12 to 24 km.

5 DISCUSSIONS

For the Lake Ha! Ha! field test case, although the RMSEs of bed elevation (in meter scale) are two orders of magnitude greater than the laboratory cases (in centimeter scale), the NRMSEs of bed elevation (49.53% in horizontal direction and 20.18% in longitudinal direction) are at the same order of magnitude as the laboratory cases (24.68% for 1D dam-break flow case, 20.96% for 2D dam-break flow case). Since there is no consensus on which sediment transport equation is the most suitable for a specific river reach, an appropriate calibration procedure by using different bed load transport equation and varying Manning's roughness coefficient is needed to reach better matches of modeling results with observations. The focus of this paper is

the new well-balanced numerical scheme for solving the VDSWE, this calibration procedure was not performed for each testing case. Nevertheless, the simulating results of bed elevation changes satisfactorily match both the laboratory and field observations. Therefore, the developed well balanced numerical scheme for solving VDSWE is proved to be a robust and accurate method for simulating dam break flow over mobile bed.

6 CONCLUSIONS

CHRE2D model is a robust surface flow routing and sediment transport model, which is capable of simulating hydrodynamics of unsteady flow, surface flow over watershed, and sediment transport processes. The performance of the model was verified by many laboratory and field cases. For flow simulation, the model predicted accurately peak flows and low hydrographs. The sediment module predicted reasonable changes of river cross sections and thalweg caused by a realistic dam break flow. The accuracy and simplicity of the proposed model, together with the robust implementation of well-balanced numerical scheme, makes this model suitable for practical hydraulic engineering applications.

ACKNOWLEDGEMENT

The research is partially funded by NSF Career Award EAR-0846523 to the University of Arizona. The funding support is essential for authors to complete this research.

REFERENCE

Cozzolino, L., Cimorelli, L., Covelli, C., Morte, R.D. & Pianese, D. (2013). Novel Numerical Approach for 1D Variable Density Shallow Flows over Uneven Rigid and Erodible Beds. *Journal of Hydraulic Engineering*, 140(3), 254-268.

El Kadi Abderrezzak, K. & Paquier, A. (2010). Applicability of Sediment Transport Capacity Formulas to Dam-Break Flows over Movable Beds. *Journal of Hydraulic Engineering*, 137(2), 209-221.

Guan, M., Wright, N.G. & Sleigh, P.A. (2014). 2D Process-based Morphodynamic Model for Flooding by Noncohesive Dyke Breach. *Journal of Hydraulic Engineering*, 140(7), 1-12.

Leighton, F.Z., Borthwick, A.G. & Taylor, P.H. (2010). 1-D Numerical Modelling of Shallow Flows with Variable Horizontal Density. *International journal for numerical methods in fluids*, 62(11), 1209-1231.

Li, W., Vriend, H.J., Wang, Z. & Maren, D.S. (2013). Morphological Modeling using a Fully Coupled, Total Variation Diminishing Upwind-Biased Centered Scheme. *Water Resources Research*, 49(6), 3547-3565.

Van Emelen, S., Zech, Y. & Soares-Frazão, S. (2015). Impact of Sediment Transport Formulations on Breaching Modelling. *Journal of Hydraulic Research*, 53(1), 60-72.

Yu, C. & Duan, J. (2012). Two-Dimensional Depth-Averaged Finite Volume Model for Unsteady Turbulent Flow. *Journal of Hydraulic Research*, 50(6), 599-611.

Yu, C. & Duan, J. (2014). Two-Dimensional Hydrodynamic Model for Surface-Flow Routing. *Journal of Hydraulic Engineering*, 140(9), 1-13.

Yu, C. & Duan, J. (2017). Simulation of Surface Runoff Using Hydrodynamic Model. *Journal of Hydrologic Engineering*, 22(6), 1-12.

