THREE-DIMENSIONAL NUMERICAL MODELLING OF
HYDRODYNAMICS AND MORPHODYNAMICS AROUND
IN-STREAM STRUCTURES

A Dissertation in
Civil Engineering

by
Yuncheng Xu

© 2019 Yuncheng Xu

Submitted in Partial Fulfillment
of the Requirements
for the Degree of
Doctor of Philosophy

May 2019
The dissertation of Yuncheng Xu was reviewed and approved* by the following:

Xiaofeng Liu
Assistant Professor of Civil Engineering
Dissertation Advisor, Chair of Committee

Peggy A. Johnson
Professor of Civil Engineering

Roman A. DiBiase
Assistant Professor of Geosciences

Chaopeng Shen
Assistant Professor of Civil Engineering

Patrick J. Fox
Head of the Department of Civil and Environmental Engineering

*Signatures are on file in the Graduate School.
Abstract

In-stream structures include both engineered structures and naturally formed ones. Examples of engineered in-stream structures are bridge piers, abutments, dams, engineered log jams (ELJ), rock vanes, J-hook vanes, among many others. Naturally formed in-stream structures mainly refer to large woody debris (LWD), which consist of fallen trees, logs, stumps, root wads, and piles of branches along the river course. These in-stream structures play a very significant role in flow resistance, sediment transport, invertebrate habitats and other aspects of fluvial ecosystem. For example, LWDs can provide food sources for aquatic insects and create refuge and habitat for fishes. They also create hydraulic diversity and roughness along river banks. Due to the geometrical complexity of these in-stream structures, the surrounding stream flow is extremely complicated and turbulent. The flow around and through complex in-stream structures can also result in local scour, sedimentation, and other morphodynamic changes. Thus, the overarching goal of this thesis research is to model and understand the flow and sediment transport processes associated with in-stream structures.

This thesis is organized as three parts: (a) high resolution numerical investigations on the three-dimensional (3D) hydraulics of LWDs, with a focus on the importance of how to represent porosity in computational models, (b) a new immersed boundary (IB) method designed for the accurate prediction of local bed shear stress, which is the driver for sediment motion, and (c) development and application of a coupled hydro-morphodynamics model for complex in-stream structures.

The first part tackles the problem of how to represent the porosity of in-stream structures. In many existing literature, the geometry of LWDs are simplified as simple cylinders or solid blocks, which are far from the reality of their complex and irregular shapes. This research tries to understand how much geometric details are needed in the numerical studies of in-stream structures. Three different representations, fully resolved geometry, porosity approximation, and solid barrier
simplification, were tested and compared. It is found that the porous media model and the solid barrier model, which are computationally economic, can describe the flow dynamics only to some extent. From the calibration of drag force and wake length, it is found that the equivalent grain size \(d_{50}\) in the porosity model should scale as the key element diameter for the simulated ELJ. A wake length scale analysis was performed for the semi-bounded flow around this in-stream structure near the bank. The length estimator in the literature for unbounded vegetation patches can be used with modifications. The results also show that the flow passing through the porous in-stream structure has a significant impact on mean velocity, turbulence kinetic energy, sediment transport capacity and integral wake length. Since geometrically-fully-resolved simulations are not currently feasible for engineering practices, the following suggestions are made based on this study. If the near-field and wake are important for the purpose of the structure, the well-calibrated porosity model seems to perform better than the solid barrier model. However, care needs to be taken when interpreting the results because this work also identified substantial loss of physical information with the porosity model. When the emphasis is the far field away from the structure, both the porosity model and the solid barrier model give comparable results.

The second part focuses on the development of a versatile computational fluid dynamics (CFD) code which can be used to track and model the dynamic evolution of the sediment bed as the scour hole develops. The immersed boundary methodology was adopted because it can deal with large and arbitrary bed deformation. More importantly, IB method can easily deal with the interaction between evolving sediment bed and in-stream structures. One technical difficulty with IB method is that in the literature focus was not on the wall shear stress. The use of the IB methods in the literature gave very poor wall shear stress, which is important for sediment transport. The root of the problem is that the original wall functions for turbulent boundary layer flow lack smoothness due to the nonlinearity and discontinuity between the log-law layer and the laminar layer. In IB method, the wall function is enforced through IB cells. However, for complex and evolving surfaces, there is no control on where the IB cells will be located in the boundary layer. The IB cells located in the log-law layer and the laminar layer follow different functions and thus will give non-smooth wall shear distribution. To remedy this, this research introduces a new IB method with a \(y^+\)-adaptation wall function. The basic idea is that when an IB cell is too close to the immersed boundary, it is automatically replaced by cells in the fluid region further away from the boundary. Thus, all IB cells are in the log-law layer and they use the same function to evaluate turbulent flow quantities. As a result, the wall shear stress is much smoother. In the new IB method, the enforcement of boundary conditions is through IB cells on which the variables are reconstructed (interpolated) from their neighbouring cells.
with an explicit, iterative scheme. Three interpolation schemes are provided, i.e., quadratic, linear and mixed. Example cases in 1D, 2D, and 3D show the new IB method together with the $y^+$-adaptation wall function produces results compare well with theory and experiments.

The third part of the thesis is the utilization of the IB method developed above and the development of a three-dimensional local scour model. The bed is treated as immersed boundary. The major components of the 3D scour model are the CFD part for turbulent flow field and the sediment transport part for updating the bed location. During the simulation, a robust and parallel interpolation scheme between 3D background mesh and 2D immersed boundary mesh is implemented. An edge-center storage method is used to address the divergence calculation problem in the Exner equation. This problem is caused by mesh non-orthogonality. One unique feature of the model that that a diffusion-based sand-slide algorithm is adopted. The relationship between sand slide and the augmented angle of repose is analyzed inside the scour. The model is validated against experimental measurements and its capability is demonstrated with a case where local scour occurs around a bridge pier with complex geometry. The demonstration shows that the model has the capability of simulating the exposure of in-stream structure foundation, which is extremely difficult if other approaches, such as dynamic mesh, are adopted.
# Table of Contents

List of Figures ix

List of Tables xiv

Acknowledgments xv

Chapter 1

Introduction 1

1.1 Motivations and methodology 1

1.2 Objective and research questions 7

1.3 Outline 8

Chapter 2

Hydrodynamics modeling of complex in-stream structures 12

2.1 Introduction 12

2.2 Methodology 17

2.2.1 Governing Equations 18

2.2.1.1 Hydrodynamic Model 18

2.2.1.2 Porosity Model 20

2.2.1.3 Drag Force Calculation 21

2.2.2 Geometry Preparation and Mesh Generation 22

2.2.3 Boundary Conditions 25

2.2.4 Configuration of the Porous Media Model 27

2.2.5 Computational Platform 29

2.3 Results and Discussions 30

2.3.1 Validation of the Fully-Resolved Configuration 30

2.3.2 Calibration of Porous Media Model 34

2.3.3 Evaluation of Different Representations in the CFD Model 37

2.3.3.1 Flow Adjustment 38

2.3.3.2 Near-Surface Velocity 43
2.3.3.3 Near-Bed Turbulence ................. 44
2.3.3.4 Flow Structure ....................... 45
2.4 Conclusions ................................ 49

Chapter 3
An immersed boundary method with smooth wall shear for
morphodynamics modeling ............ 55
3.1 Introduction ............................... 55
3.2 Governing equations and their discretization ................. 61
  3.2.1 Governing equations for flow and turbulence model .......... 61
  3.2.2 Discretization method .................. 62
  3.2.3 Iterative solution algorithm for pressure-velocity coupling . 64
3.3 Treatment of immersed boundaries ......................... 67
  3.3.1 Representation of immersed boundaries .................. 67
  3.3.2 Reconstruction algorithm .................. 69
    3.3.2.1 Quadratic interpolation scheme for reconstruction . 70
    3.3.2.2 Alternative linear interpolation scheme for recon-
             struction .................................. 73
  3.3.3 Coefficient matrix manipulation .................... 76
3.4 \( y^+ \)-adaptation immersed wall model ................. 79
  3.4.1 Basic wall function for immersed boundary method .......... 79
  3.4.2 \( y^+ \)-adaptation wall model for immersed boundary method . 81
  3.4.3 Rough wall function for immersed boundary method .......... 85
3.5 Test cases ................................ 86
  3.5.1 Pitz-Daily 2D case ........................ 86
  3.5.2 Flow over a 3D dune ..................... 92
3.6 Summary and conclusions ....................... 101

Chapter 4
Development and applications of a robust 3D local scour model
based on immersed boundary method 107
4.1 Introduction ................................ 107
4.2 Hydrodynamic model .......................... 112
  4.2.1 Governing equations and turbulence model ................ 112
  4.2.2 Rough wall models .......................... 114
4.3 Bed morphodynamics model ...................... 120
  4.3.1 Exner equation and bed-load transport .................. 120
  4.3.2 Critical shear stress ...................... 121
  4.3.3 Sand-slide algorithm ...................... 125
Chapter 4

4.4 Immersed boundary implementation ........................................ 128
  4.4.1 Sharp interface method ...................................................... 128
  4.4.2 Immersed boundary wall function ........................................ 134
  4.4.3 Coupled hydrodynamic and morphodynamic models ................. 136
  4.4.4 Interpolation between 3D hydro mesh and 2D morpho mesh .......... 138
  4.4.5 Effect of mesh non-orthogonality on divergence calculation ....... 140
  4.4.6 Intersection between bed and object ..................................... 144
  4.4.7 Computational speed ......................................................... 151
    4.4.7.1 Test cases description ................................................... 151
    4.4.7.2 Effect of mesh resolution .............................................. 152
    4.4.7.3 Effects of 3D-2D interpolation stencil size ......................... 157

4.5 Validations and discussions .................................................. 158
  4.5.1 Hydrodynamics modeling .................................................... 161
  4.5.2 Scour modeling ............................................................... 167
  4.5.3 A test case of complex bridge piers .................................... 173

4.6 Conclusions ................................................................. 178

Chapter 5

Summary and future work ..................................................... 184
  5.1 Summary ................................................................. 184
  5.2 Future work ............................................................... 187

Appendix A

FV discretization schemes .................................................. 189

Appendix B

Estimation of shear velocity in fully developed open channel flow ... 191
## List of Figures

2.1 The engineered log jam (ELJ) structure in the flume. In the 3D view, a fake free surface was added. ................................. 24
2.2 Zoom-in view of the fully-resolved ELJ mesh generated by snappy-HexMesh. .......................................................... 25
2.3 Time averaged $y^+$ distribution for the fully-resolved ELJ and the solid barrier. ........................................................... 26
2.4 Comparison of the cross-sectional distributions of mean streamwise velocity and turbulent kinetic energy (TKE) between the numerical and experimental results for the background case without ELJ. The variables are made dimensionless with channel width $B$, water depth $h$ and mean velocity $u_0$. ........................................... 32
2.5 Comparison of the cross-sectional distributions of the mean streamwise velocity between the experiment and the fully-resolved case with ELJ. ................................................................. 33
2.6 Comparison of the cross-sectional distributions of TKE between the experiment and the fully-resolved case with ELJ. ........... 34
2.7 Error distribution on the upstream and downstream cross-sections between the porosity cases and the fully-resolved case. The white blank region means that the values exceed the range of the colorbar. 37
2.8 Transverse distribution of the mean streamwise velocity and TKE on the upstream and downstream cross-sections at $z/h = 0.4$. .... 38
2.9 Flow domain partition, colored by time-averaged velocity on the water surface. ................................................................. 40
2.10 Longitudinal profiles of cross-sectionally-averaged streamwise velocity and TKE within the adjustment and wake regions. The $x$ coordinate is normalized by the length of ELJ ($L$). The gray area denotes ELJ. ................................................................. 41
2.11 Time-averaged streamwise velocity distribution at 5 mm below the free surface. .............................................................. 44
2.12 Time-averaged TKE distribution in the plane at 15 mm above the bed. ................................................................. 46
2.13 Instantaneous flow field represented by streamlines in the wake zone. The streamlines are colored by the instantaneous velocity. 47
2.14 Instantaneous flow structure represented by the isosurfaces of $\lambda_2 = 3$. 48
2.15 Zoom-in view of the instantaneous flow structure for the fully-resolved case with the isosurfaces of $\lambda_2 = 3$. 49

3.1 Schematic view of a control volume and its neighbour. The blue volume represents the control volume $P$. The red volume represents a neighbouring cell $N$. The grey face represents the shared face between $P$ and $N$. $S_f$ is the area vector of the shared face. 63
3.2 Schematic view of IB representation: IB cell centers (black cross), hit points (black dots), image points (black circles), IB cells (red filled), live cells (green filled), and dead cells (white filled). Immeresed interface $\Gamma_{IB}$ is represent by the blue curve. 68
3.3 Effect of IB cell wall distance on the simulated shear velocity. $y^+_IB$ and $y^+_IP$ denote the dimensionless wall distance of IB cell center and image point, respectively. The immersed boundary $\Gamma_{IB}$ can be moved vertically in the yellow region to change the IB cell wall distance. 83
3.4 Effect of IB $y^+$ adaptation on calculated shear velocity. 84
3.5 Simulated velocity distributions in 1D channel with different roughness height using the proposed rough IB wall function. Here, $u^+ = u/u_T$. 86
3.6 The mesh for immersed boundary methods (upper) and the body-fitted mesh (lower). The red cells in upper mesh are the IB cells. 87
3.7 Comparison of results among the present IB method, the IB method in (Jasak et al., 2014), and simulation with a body-fitted mesh, and experiments in Pitz and Daily (1981). The flow is from left to right. 90
3.8 Wall shear stress distribution along the lower and upper walls. 92
3.9 Schematic view of step IB cells, which are highlighted by yellow frames. IB cells (red filled), live cells (green filled), and dead cells (white filled). Immeresed interface $\Gamma_{IB}$ is represent by the blue curve. 92
3.10 The bathymetry of the 3D dunes and the two meshes. One mesh (left) is the background for IB method and the other (right) is the body-fitted mesh for comparison. The flow comes from left to right. 94
3.11 Comparison of the velocity profiles among experiments (Maddux et al., 2003a) and two numerical simulations using the present IB method and a body-fitted mesh. Here, IBM denotes the present IB method (with $y^+$ adaptation scheme) and BFM denotes the body-fitted mesh. The profiles are sampled on two vertical slices at two different span-wise locations. RMSE denotes the root-mean-square error. The vertical dash line denotes the measurement location.

3.12 Comparison of $\tau_{sf1}$ distribution among experiments (Maddux et al., 2003a) and two numerical simulations using the present IB method (with and without $y^+$ adaptation scheme). $\tau_{sf1}$ is derived from near bed velocity. Dahsed curved lines denote the elevation contour.

3.13 Comparison of $\tau_{sf2}$ distribution among experiments (Maddux et al., 2003a) and two numerical simulations using the present IB method (with and without $y^+$ adaptation scheme). $\tau_{sf2}$ is derived from Reynolds stress. Dahsed curved lines denote the elevation contour.

3.14 Effect of $y^+$ adaption scheme on $\tau_{sf1}$ and $\tau_{sf2}$ distribution over vertical solid line shown in Fig. 3.12 and Fig. 3.13.

4.1 2D diagram of moving mesh technique.

4.2 Comparison among the two-layer model, the Spalding and van Driest velocity profiles for hydraulically smooth wall and rough wall at $k_s^+ = 30$.

4.3 3D diagram of $\beta$ and $\phi$.

4.4 Modified Shields diagram.

4.5 Countours of $\tau_c^*/\tau_{c0}^*$ over different $\phi$ and $\beta$. A angle of repose $\beta_0 = 33^\circ$ is used. The vertical red line represents the location of $\beta_0$. The solid blue line represents the corrected threshold $\beta'_0$ using Equation (4.24). The colored region to the right of the solid blue line represents the invalid region.

4.6 Smooth function shown in Equation (4.28). $R_{lower} = 0.8$, and $R_{upper} = 1.2$.

4.7 2D schematic representation of immersed boundary method and IB wall distance correction. The blue curve $\Gamma_{IB}$ is an immersed boundary. Upper green cells are live cells. Middle red cells are IB cells. Lower white cells are dead cells. Closed black dots are hit points. Crosses are centers of IB cells. Open dots are image points. The yellow highlighted IB cell is corrected due to close distance to IB wall.
4.8 Algorithm diagram of the immersed boundary method within a predictor-corrector algorithm for solving fluid flow. Key steps of IB algorithm implementation are highlighted by light green background and dashed blue boundaries. * Variables are frozen when solving these equations. ** Velocity, eddy viscosity, and other turbulence variables such as turbulent kinetic energy are updated in IB cells, usually based on wall functions.

4.9 Diagram of immersed boundary method coupling between hydrodynamics and morphodynamics.

4.10 Interpolation stencil of hit points for each cell centroid or edge center on non-uniform triangulated mesh. Cell centroid or edge center is represented by star (colored in red). Hit points inside stencil are represented by cross (colored in blue). Hit points outside stencil are represented by full dot (colored in black).

4.11 Comparison of $\delta V$ between using cell value and edge value on uniform and non-uniform triangulated meshes. The colormap is in logarithmic scale.

4.12 A flat bed intersected by a vertical cylinder. The thick circle (blue) represents the intersection between bed and vertical cylinder. Green cells are active cells on the 2D morpho mesh. Red cells are boundary cells. White cells are inactive cells.

4.13 A flat bed intersected by multiple objects and computational boundaries. Blue vertical walls in 3D view and dashed blue rectangle in 2D schematic represent the boundaries of background computational mesh.

4.14 Intersection on 2D morpho mesh for parallel computation. Green cells represent active cells. Red cells represent boundary cells. Pink cells represent internal boundary cells.

4.15 Flow over a sand dune case, which is used to test the computational speed.

4.16 The effect of 2D surface mesh size on the computational cost for different part in a morphological time step. Colored contours show the computational time. Dashed and solid lines show the percentage of computational time.

4.17 The effects of 3D background mesh size on the computational cost for different part in one time step. Colored contours show the computational time. Dashed and solid lines show the percentage in total computational time.

4.18 The effects of 3D-2D interpolation stencil size. Colored contours represent the computational time. Solid line represents the percentage in total computational time.
4.19 Flow around a vertical circular cylinder. Blue grid is the 3D hydrodynamics background mesh. ........................................... 159
4.20 Comparing streamwise velocity for smooth wall function ........ 163
4.21 Comparing vertical velocity for smooth wall function .............. 164
4.22 Comparing streamwise velocity for rough wall function .......... 165
4.23 Comparing vertical velocity for rough wall function .............. 166
4.24 Time evolution of flow structure around the cylinder with the instantaneous \( \lambda_2 \) criterion. ......................................................... 167
4.25 Distribution of bed elevation at different time. Flow is from left to right. ................................................................. 169
4.26 Time evolution of scour depth at the upstream edge and the downstream edge of cylinder. Scaling represents the scaling in physical acceleration. .......................................................... 171
4.27 Probability distribution of different combination of \( \phi \) and \( \beta \) within the area three diameter away from cylinder at different moments. The vertical solid red line represents the location of \( \beta_0 \). The curved solid blue line represents the valid threshold of Equation (4.24). ... 173
4.28 A set of complex bridge piers. The pier length of field scale is 13.26 m. 174
4.29 Time evolution of the scour development around complex bridge piers. The flow direction is from left to right. ....................... 176
4.30 Probability distribution of different combination of \( \phi \) and \( \beta \) near the complex bridge piers at different moments. The vertical red line represents the location of \( \beta_0 \). The curved blue line represents the valid threshold of Equation (4.24), noted as \( \beta'_0 \). ................. 178
List of Tables

2.1 Drag force for the fully-resolved case, porosity cases and the solid barrier case. ........................................ 36
2.2 Measured wake length $L_1$ and $L_w$. ......................................................... 42
2.3 Comparison between the porosity model and the solid barrier model. 51

4.1 Effect of mesh non-orthogonality on divergence calculation . . . . . . 143
4.2 Setup for the effect of 2D surface mesh size. Cell size and cell number of 3D mesh are constant at 0.05 m and 48000. . . . . . . . . 153
4.3 Effects of 3D background mesh size. Cell size and cell number of 2D mesh are set constant at 0.075 m and 6063, respectively. . . . . 155
4.4 Test conditions and meshes for scour around pier cases. ............ 160
Acknowledgments

The acknowledgement is the only place in a thesis to express something non-academic, emotional and human. Given this opportunity, I would like to briefly review the past few years of my Ph.D. study to appreciate those who supported my research and life at Penn State.

First of all, I want to express my deep gratitude to my adviser, Dr. Xiaofeng Liu. Since being inspired by the very first overseas call from him, I have been drawn to utilizing high-resolution numerical tools to study complex environment flows created by Nature and human. In the first few months, it was his encouragement and patience that assisted me with the transition from mechanical engineering to be more civil-engineering minded. It was also his guidance and enthusiasm that led me into this difficult but fascinating world of OpenFOAM development. When I was hesitating and felt lost about the meaning of this research, it was his multidisciplinary knowledge that lightened up my mind. During my most difficult days of coding and debugging, he was always there and never gave me up. His insistence, intelligence and diligence significantly influenced my academic attitude. This will be the most treasured spirit throughout my career. As to the thesis, my many discussions on the methodology and conceptual framework with him help formulate the key ideas of this thesis.

I would like to thank the other members of my doctoral committee, Dr. Peggy Johnson, Dr. Chaopeng Shen and Dr. Roman DiBiase, for their advice on my research framework. And I also want to thank Dr. Li Li and Dr. Larry Brannaka for having been members of my committee during the comprehensive examination. Additionally, I am grateful to Dr. Yong Lai, a collaborator from U.S. Bureau of Reclamation, who gave me great opportunities to enhance my programming skills and also helped funding my thesis research. I am also grateful for his encouragement for me to pursue a future academic career.
I want to thank all my group-mates for their support and help over the years: Yunxiang Chen, Mahdad Talebpour, Detian Liu, Bin Sui, Youbiao Wang, Yalan Song, Yi-xuan Zeng, Zilong Wang, Hassan Ismail, and Bing Li. I am also indebted to many other student colleagues from the fourth floor of Sackett Building. I will remember the emotional beer time with Kuai Fang and Nuan Sun, both of whom are also my roommates, the bragging lunchtime with Xinye Ji and Dapeng Feng, and of course the wonderful moments of picnic, canoeing, and hotpot with all the officemates. Especially, I want to thank Yunxiang who started PhD at the same time and in the same group as me. We shared the most meaningful memories in research and life decisions during those unforgettable years at Penn State.

Nobody else has been more important to me in the pursuit of doctoral degree than my family. I am extremely grateful to my parents, who gave me life, money and thoughts, and to my elder sister, who treats me as her own kid. I cannot finish this acknowledgment without paying special gratitude to my dear wife, Dr. Yu Ru. As a Dr. Wife in public finance, she broads my minds in all kinds of ways, and I own my deepest debt to her best patience and understanding despite the five-year-long and 7,000-mile away separate life.
Chapter 1

Introduction

1.1 Motivations and methodology

In the practice of stream restoration, the term “in-stream structure” usually refers to the engineered construction which are designed to change the physical characteristics of a channel. Furthermore, their existence will increase the stream habitat complexity. Natural in-stream structures are part of fluvial processes and they inspire the design and construction of stream restoration measures. One example of engineered in-stream structure is the large woody debris (LWD). LWD usually refers to trees, stumps, root wads and logs that fall into and are stored along the streams. It often acts as an important eco-hydraulic element in natural streams. Its influence ranges from very localized effects, such as local scour, to watershed scale such as floodplain morphology (Abbe and Brooks, 2011). The ecological significance of LWD-like in-stream structures has been emphasized by many researchers since
1980s. One important effect is associated with the particulate organic matter (Bilby and Likens, 1980), providing food and habitats for invertebrates (Benke et al., 1985) and fishes (Shirvell, 1990; Dolloff and Warren, 2003). According to Tullos and Walter (2014), for the benefit of fish habitat, in-stream structures can provide shelters, abundant foraging, and macroinvertebrate biomass, among many others. In a much broader sense, in-stream structures also include anything that human placed in rivers for various purposes, e.g, bridge piers, abutments, dams, engineered log jams, rock vanes, and J-hook vanes. Their existence also alters the flow field, geomorphology, and ecology.

It is well accepted that local micro-scale and meso-scale flow structures induced by obstacles in streams have a significant impact on the physical habitat (Shen and Diplas, 2008). For example, through flume experiment, Cullen (1991) found the local vortex might promote the growth of fish population. The linkage between the flow and any ecological model is based on the premise that the detailed flow field can be either accurately measured or simulated. However, measurements around complex in-stream structure are difficult. In addition, due to the vast variations of natural in-stream structures, it is extremely difficult to make any generalization based on limited measurement data. Computational models are thus very attractive because they can potentially model any configuration.

Depending on the assumptions on the hydraulics around in-stream structures, computational models can be categorized into one-dimensional (1D), two-
dimensional (2D), and three-dimensional (3D) models. Currently, most habitat assessment models, such as the PHABSIM model (Waddle, 2001), simplify the hydraulics around in-stream structures as 1D, which is able to capture the bulk, global flow features but not detailed, local features. It is in fact those local flow features that have the most profound and direct impact to the environment. He et al. (2009) used a 2D depth-averaged model to analyze the effects of LWD on channel morphology. The LWDs in their model are simply represented as porous media and the flow experiences extra drag force in the LWD zone.

In comparison with 1D and 2D models, 3D computational fluid dynamics (CFD) technique might be the most suitable tool to obtain high-resolution, local flow features. Up until now, there are very limited studies using CFD for complex in-stream structures, such as LWDs. Smith et al. (2011) emphasized the difficulty of implementing the complex geometric roughness elements like LWD into CFD models. Another study is reported in Allen and Smith (2012) where a URANS turbulence model was used to evaluate the impact of the geometric simplification on LWDs. Neither of the above two researches considered morphological impact.

Besides LWDs, there are some computational simulation studies on other stream restoration structures in the literature. Kang et al. (2011), Kang and Sotiropoulos (2012), Kang and Sotiropoulos (2015a), and Kang and Sotiropoulos (2015b) developed a very high accuracy 3D computational model to simulate the turbulent free surface flow interacting with a cross vane. They used Curvilinear
**Immersed Boundary** (CURVIB) method (Gilmanov and Sotiropoulos, 2005; Ge and Sotiropoulos, 2007; Borazjani et al., 2008) to treat the complex geometry and a level set method to capture the free surface. For morphodynamic modelling, Khosronejad et al. (2011) developed a CURVIB model coupled with sediment transport. With these computational modelling techniques, they investigated large dunes in meandering streams and the morphodynamic impact of J-hook vanes (Khosronejad et al., 2012, 2014, 2015b,a). Although limited, CFD studies of stream restoration projects have grown in recent years and computational models have been proven to be a very promising tool to investigate the local hydraulics and morphodynamic features around complex structures in streams.

On the other hand, in-stream structures also include man-made hydraulic infrastructures, such as bridges, gates, weirs, dams and so forth. Sometimes, their geometry can also be very complex. Like LWDs, these structures can also induce scour. Indeed, scour is one major cause for bridge failures and other structural damages. It was reported that 53% of all the bridge failures in United States were due to flood and scour (Wardhana and Hadipriono, 2003). The exact cause of scour and erosion may vary under different conditions (Richardson and Richardson, 2008). A recent review in Wang et al. (2017) classified scour events into three major forms, i.e., contraction scour, general scour, and local scour. Among the three, contraction scour and local scour are directly linked the 3D turbulent flow around the structure, which highlights the need to consider the geometric details.
Regardless it is man-made hydraulic infrastructure, river restoration structure, or a natural woody debris, they share similar hydrodynamic and morphological characteristics to some extent. The overarching goal of this thesis work is to develop and utilize 3D computational models for the study and design of these in-stream structures. To be applicable for engineering practice, such models have to balance between accuracy and computational cost. In addition, in the course of this thesis research, some unexpected technical problems emerged during the development of the models. For example, the numerical method adopted in this work, the immersed boundary method, was invented in the field of biofluids and widely adopted in many other fields later on. Its use in the field of hydraulics and sediment transport is relatively limited. One major problem encountered is that the wall shear stress was not of as great interest in other fields as in hydraulics. The immersed boundary methods reported in the literature do not work well in terms producing smooth wall shear. Non-smooth and ill-behaving wall shear stress jeopardize the sediment transport calculation and consequently make the simulated scour result useless. To overcome this, a solution is proposed and tested in this work.

Another technical difficulty when simulating 3D scour process around a structure with complex geometry is how to deal with the dynamic interaction of the evolving bed and the structure. During the erosion process, a structure may be dynamic buried or exhumed depending on the flow and sediment transport condition. It
is the result of the coupled process between flow and sediment. There are some existing 3D scour models reported in the literature for flow and sediment transport around simple vertical cylinders (Olsen and Kjellesvig, 1998; Roulund et al., 2005; Liu and García, 2008; Escauriaza and Sotiropoulos, 2011; Jacobsen and Fredsoe, 2014; Baykal et al., 2017). Most of the scour models use moving mesh technique to simulate the bed morphological changes. The mesh moves based on the deformation of the scoured bed. However, this moving mesh method is limited to simple geometries such as vertical cylinders. Even for the case of an inclined cylinder, it will be very difficult to simply moving grid points as the scour hole evolves. In recent years, a promising alternative is to use the immersed boundary method (Escauriaza and Sotiropoulos, 2011; Khosronejad et al., 2011, 2012). This method treats both the bed and the in-stream structure as immersed boundary. It has been applied in the morphodynamic modeling of stream restoration structures (Khosronejad et al., 2014, 2015b, 2018). Although their model can consider complex geometry such as rock vanes, which is a great improvement, their algorithm limits the bed motion only in the vertical direction. In other words, the bed surface in their model can not dynamically cut and slice through the structure, and thus can not simulate real burial or exhume. Such capability is important for the failure study of those complex hydraulic infrastructures. In addition, their immersed boundary method is limited by curvilinear mesh. In this work, a unstructured mesh is used and thus greatly improve the easiness of use for practice.
1.2 Objective and research questions

Based on the background and motivations described above, the objectives of this thesis research are (1) to better understand and quantify the hydrodynamics around and through in-stream structures with computational modeling, in particular the effects of how porosity is modeled, (2) to develop robust and computational affordable computational model for the simulation of scour process around in-stream structures. With these models, the specific research questions to be addressed are as follows:

(1) How necessary is it to fully resolve the geometric complexity of in-stream structures? What are the implications if the structures are modeled with porous media or solid?

(2) How to improve the smoothness of the wall shear stress simulated with immersed boundary method? What is the cause of non-smooth wall shear and how to balance accuracy and smoothness?

(3) How to incorporate the morphodynamics into immersed boundary method? And how to improve the efficiency of the scour modeling of complex in-stream structure such that it is affordable in practice?
1.3 Outline

This thesis is organized as follows:

Chapter 2 aims to study how much geometric details are needed in the numerical simulations of complex in-stream structures. Three different representations, i.e., fully resolved geometry, porosity approximation and solid barrier simplification, are tested and compared. The example of engineered log jam was used for the comparison, which focuses on the effect in the near- and far-fields.

Chapter 3 describes the development of a three-dimensional numerical model with immersed boundary method technique. This immersed boundary method is designed for high Reynolds number flows. A $y^+$-adaptation wall function is proposed to improve the smoothness of the wall shear stress computed with IB method. Example cases in 1D, 2D, and 3D will be shown to demonstrate that the $y^+$-adaptation wall function produces much improved wall shear results.

Chapter 4 introduces the develop and application of a three-dimensional local scour numerical model based on the immersed boundary method proposed in Chapter 3. This model is validated against experimental measurement and tested with a case of complex bridge pier. It has the capability of simulating the exposure of the bridge’s foundation which was initially buried. The main features of the 3D scour model to be introduced include:

- a robust and parallel-able interpolation scheme between 3D background mesh
and 2D surface mesh,

- an edge-center storage method to address the bedload flux divergence calculation problem in the Exner equation, which is caused by mesh non-orthogonality,

- a numerical acceleration method related to morphological time step,

- a physical acceleration method based on morphological scaling.

Chapter 5 summarizes the achievements and findings of this thesis research. It also discusses some future research needs and directions.

References


Chapter 2  
Hydrodynamics modeling of complex in-stream structures

2.1 Introduction

In-stream structures have been used widely in engineering practice for various purposes, e.g., river training, channel stabilization, erosion control and habitat restoration. In the past several decades, the science and engineering communities have gradually realized that Mother Nature provides great examples of good in-stream structures, which serve multiple purposes. One such natural structure is large woody debris (LWD), which consists of fallen trees, stumps and root-wads, with irregular and complex geometries. Their ecological significance has been recognized since the 1980s (Shields and Nunnally, 1984; Bisson et al., 1987; Sullivan et al., 1987). Studies have found that LWD greatly increases stream habitat
complexity (Gorman and Karr, 1978). Organic matter can easily accumulate in-between the debris of snags, leaves and root-wads, which provides nutrients and food for living organisms. The debris structures were reported to contain more than half of the organic matter in lower-order streams (Bilby and Likens, 1980), which provides abundant food sources for invertebrates, fishes and macro-invertebrate biomass. The structures can also function as shelters for fishes and prevent them from being attacked by predators (Tullos and Walter, 2014). The slow flow region in the wake of in-stream structures provides a resting area and helps fish regain energy (Fausch, 1984). The ecological benefits of in-stream structures highly rely on the hydrodynamics within and around them. Recognizing their importance to rivers, in-stream structures have been widely used in stream restoration projects (Bernhardt et al., 2005). Examples of these structures include single log vane, log cross vane, mud sill, multi-log vane deflector, water jack and engineered log jams (ELJ) (Jarrett et al., 2011), among which ELJ is a very attractive option due to its abundant eco-hydraulic benefits (Bennett et al., 2015).

The proper design of in-stream structures depends on the understanding of the associated hydraulics. These in-stream structures can be regarded as roughness elements in rivers. They alter the flow conditions by slowing down surrounding flow and changing local water depths (Gippel, 1995). In the measurements of Cultus River, Manga et al. (Manga and Kirchner, 2000) found that LWD, which only covered 2% of flow area, contributed half of the total flow resistance in the stream.
A recent flume study also showed that if these structures are fixed, they can delay the flooding time and significantly lower the runoff coefficient, which helps reduce erosion and suppress flooding damage (Wenzel et al., 2014). Thus, one important design parameter is the flow resistance due to LWDs. On the other hand, the drag force experienced by the structures is an important factor for their stability.

There are two major methods to quantify the flow resistance due to in-stream structures. The first one is to use the concept of stress partitioning, which was first introduced for the study of fluvial dynamics (Einstein and Banks, 1950). This method separates the flow resistance due to debris from the total. For example, some researchers created new Darcy–Weisbach friction factors for the prediction of flow resistance in the stream, which is constituted of bed shear, bend curvature and drag from debris (Shields and Gippel, 1995; Manga and Kirchner, 2000; Wilcox et al., 2006). In this method, it is critical yet difficult to estimate the debris drag coefficient. It is often observed that LWDs contain much sediment and many leaves inside the gaps between snags and root-wads. Therefore, these structures are treated as solid barriers, and as a result, simple theoretical models can be applied to calculate the drag force exerted on them (Shields and Gippel, 1995). In early studies, a very common approximation is to use a single cylinder as a proxy for a log in the streams. Flume experiments were conducted to investigate how the side-walls, bed bottom and orientation would affect the drag and lift coefficients for a single log (Young, 1991; Gippel et al., 1992, 1996).
Another method is to use the one-dimensional (1D) momentum equation to compute the drag force exerted on in-stream structures. For example, Turcotte et al. (Turcotte et al., 2015) found that the drag forces on large cylinders calculated from the momentum equation matched well with the measured values. They also found that the classic drag force formulas underestimate the force if the cylinder diameter is much larger than critical water depth. Compared with the stress partitioning method, the momentum analysis method is more applicable to complex and porous accumulations, like debris jams. For example, Manners et al. (Manners et al., 2007) measured detailed velocity around natural woody debris jams and calculated drag coefficients through the moment equation.

Numerous studies have shown the complexity of in-stream structures and the deviation of their hydraulic behavior from the predictions made by the two methods mentioned above. For example, Manners et al.’s (Manners et al., 2007) results show the drag coefficient for the debris jams ranges from 2.6 to 9.0. In contrast, the drag coefficient is about 0.8 for a cylinder and one for a cube. Manners et al.’s (Manners and Doyle, 2008) field study indicates that for debris jams, the drag decreases as the porosity increases. A more recent study of Shields and Alonso (Shields and Alonso, 2012) shows that the roughness on the surface of cylinder increases lift, but reduces wave force (which is the difference between actual drag force and theoretical drag force). They also found that the effect of surface roughness is not as significant as the bed proximity, blockage rate and submergence. However, under certain
conditions, the increase of complexity may reduce the drag coefficient. Gippel et al.’s (Gippel et al., 1996) flume study showed that branches added to a single trunk would lower the drag coefficient because the increase of drag force was less than the percentage increase of the frontal area.

For the solid barrier model in hydraulics, a very classic application of CFD is in the groyne fields McCoy et al. (2007); Constantinescu et al. (2009). The use of CFD to study the porous in-stream structures is rare in the fluvial hydraulics community. As an infrequent case in this point, in order to evaluate the geometric complexity of woody debris, computational simulations were performed by Allen et al. (Allen and Smith, 2012). They developed a shape complexity factor to quantify the geometric complexity of LWD. Their simulations showed that both the drag coefficient and the integrated turbulence kinetic energy decrease with the complexity factor. However, for the complex and porous structures, it is very difficult to compute the shape complexity factor. There are more CFD studies in coastal engineering for coastal structures, which often consist of amour rocks (Gent, 1995; Jensen, 2014). In order to simplify the simulations, these coastal engineering studies usually adopt a porous media flow model by adding resistance source terms, which are calculated with an extended Darcy–Forchheimer equation, into the Navier–Stokes equations. Jensen et al. (Jensen et al., 2014) calibrated the empirical resistance coefficients for coastal structures. Their simulation results had a good match with the experimental data. A similar porosity model will be
evaluated in this study.

This paper presents the findings on the effects of different in-stream structure representations in 3D computational models. In particular, the implications for flow resistance, wake length scale and turbulence characteristics have been analyzed. Three different representations of the selected in-stream structure, i.e., ELJ, were used and compared. The three geometric representations include the fully-resolved model (no simplification in geometry), the porosity model and solid barrier (i.e., treating the ELJ as an impervious box). The simulation case was based on the flume experiments for ELJ in Gallisdorfer et al. (2013); Bennett et al. (2015). In the following, the computational models will be introduced first. Calibrations of all models are then presented, followed by detailed analysis on the simulation results and comparisons. This paper then ends with discussions and conclusions.

2.2 Methodology

This section describes the 3D computational model used in this study. The description includes the governing equations for the flow, turbulence and porosity model, as well as the computational platform. The methods to calculate drag force are also introduced.
2.2.1 Governing Equations

Hydrodynamic Model

The stream flow is governed by the Navier–Stokes equations. To resolve the turbulent eddies generated by the complex geometry, large eddy simulation (LES) was performed. With spatial filtering, the velocity and pressure can be decomposed into resolved and unresolved parts, i.e., \( u_i = \tilde{u}_i + u'_i \) and \( p = \tilde{p} + p' \). \( \tilde{\cdot} \) and \( \cdot' \) indicate the resolved and unresolved field, respectively. The subscript \( i \) denotes the component in the \( i \)-th direction. The filtered Navier–Stokes equations have the form:

\[
\frac{\partial \tilde{u}_j}{\partial x_j} = 0 \tag{2.1}
\]

\[
\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \tilde{p}}{\partial x_j} \delta_{ij} + \frac{\partial}{\partial x_j} \left( \nu \frac{\partial \tilde{u}_i}{\partial x_j} \right) - \frac{\partial \tau_{ij}^{SGS}}{\partial x_j} \tag{2.2}
\]

where \( \tilde{u}, \tilde{p}, \rho \) and \( \delta_{ij} \) denote resolved velocity, resolved pressure, fluid density and the Kronecker delta, respectively. \( \tau_{ij}^{SGS} = \tilde{u}_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j \) denotes the unresolved subgrid-scale (SGS) stress tensor, which is closed by the dynamic Smagorinsky model (Smagorinsky, 1963; Deardorff, 1970; Lilly, 1992), which has the form of:

\[
\tau_{ij}^{SGS} = \frac{1}{3} \tau_{kk}^{SGS} \delta_{ij} - 2 \nu_t^{SGS} \tilde{S}_{ij} \tag{2.3}
\]
where \( \tilde{S}_{ij} \) denotes the SGS strain tensor and \( \nu_t^{SGS} \) denotes the SGS turbulent eddy viscosity, which can be calculated respectively as:

\[
\tilde{S}_{ij} = \frac{1}{2} \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \tag{2.4}
\]

\[
\nu_t^{SGS} = (C_s \Delta_g)^2 \sqrt{2 \tilde{S}_{ij} \tilde{S}_{ij}} = (C_s \Delta_g)^2 |\tilde{S}_{ij}| \tag{2.5}
\]

where \( \Delta_g \) is the local grid size and \( C_s \) is the Smagorinsky coefficient, constant for the original Smagorinsky model. However, this paper uses the dynamic Smagorinsky model by Lilly (Lilly, 1992), which was modified from Germano et al.’s (Germano et al., 1991) SGS Smagorinsky closure. Here,

\[
C_s^2 = \frac{L_{ij} M_{ij}}{M_{ij} M_{ij}} \tag{2.6}
\]

where \( L_{ij} = \tilde{u}_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j \) and \( M_{ij} = \Delta_g^2 |\tilde{S}| \tilde{S}_{ij} - \tilde{\Delta}_g^2 |\tilde{S}| \tilde{S}_{ij} \). The symbol \( \tilde{\cdot} \) represents the test-scale filtering, which is specified as twice the grid size in this study, i.e., \( \tilde{\Delta}_g = 2 \Delta_g \). \( L_{ij} \) is in fact the resolved stress tensor related to the eddy scales between \( \tilde{\Delta}_g \) and \( \Delta_g \). With the above, Equation (2.2) can be written as:

\[
\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} \delta_{ij} + \frac{\partial}{\partial x_j} \left[ \left( \nu + \nu_t^{SGS} \right) \frac{\partial \tilde{u}_i}{\partial x_j} \right] \tag{2.7}
\]
Porosity Model

One model to be evaluated is the use of porous media to approximate the real complex in-stream structures. The effect of porous media on the flow is modeled as a resistance force, i.e., an extra drag term $S_i$ in the momentum Equation (2.7):

$$\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \tilde{p}}{\partial x_j} \delta_{ij} + \frac{\partial}{\partial x_j} \left[ \left( \nu + \nu_i^{SGS} \right) \frac{\partial \tilde{u}_i}{\partial x_j} \right] + S_i \tag{2.8}$$

where the drag term $S_i$ is calculated using the Darcy–Forchheimer equation:

$$S_i = - \left( \nu_i D + \frac{1}{2} |\tilde{u}_{jj}| F \right) \tilde{u}_i \tag{2.9}$$

where the turbulence eddy viscosity $\nu_i = \nu + \nu_i^{SGS}$. The Darcy coefficient $D$ and the Forchheimer coefficient $F$ are calculated with the following formulas (Gent, 1995):

$$D = \alpha \frac{(1 - n)^2}{n^3} \frac{1}{d_{50}^2} \tag{2.10}$$

$$F = 2\beta \frac{1 - n}{n^3} \frac{1}{d_{50}} \tag{2.11}$$

where $\alpha$ and $\beta$ are model constants to be calibrated for a specific porous media.

For example, Gent (Gent, 1995) used $\alpha = 1000$ and $\beta = 1.1$, while Jensen et al. (Jensen et al., 2014) used $\alpha = 200$ and $\beta = 2.0$ in their respective research. $d_{50}$ denotes the grain diameter in the original model by Gent (Gent, 1995) and Jensen et al. (Jensen et al., 2014). In this study, it should characterize the debris size...
in the porous in-stream structure. \( n \) denotes porosity. Two derivative parameters related to the porosity model in this work are the intrinsic permeability \( K \) and inertial permeability \( K_i \), which are defined as follows:

\[
K = \frac{1}{D}; \quad K_i = 2\frac{\nu}{F}
\]  

(2.12)

Both parameters depend on the geometrical details of the complex structure and will be quantified in the simulation cases.

**Drag Force Calculation**

The drag force \( (F_d) \) refers to the streamwise force exerted on the structure due to fluid motion. For simple geometries like cylinders, it is common to use the non-dimensional drag coefficient \( (C_d) \) defined as:

\[
C_d = \frac{F_d}{\frac{1}{2}\rho U^2 A}
\]

(2.13)

where \( U \) is the mean streamwise velocity of the incoming flow and \( A \) is the frontal area of the structure. For complex structures like LWD and ELJ, it is very difficult to determine the frontal area. Thus, in this study, the dimensional drag force \( F_d \) is used instead.

Two methods were used to calculate the drag in the simulations. One is to directly integrate the pressure and shear stress on the surface of the in-stream
structures in the fully-resolved case. The other one is to integrate the pressure and shear stress on a surface enclosing the in-stream structure. The integration to get the drag force in the second method can be expressed as:

\[
F_d = \left[ \int_{CV} (-p \mathbf{I} + \boldsymbol{\tau})dA \cdot \mathbf{n} \right] \cdot \mathbf{n}_x
\]

(2.14)

Here, \(CV\) denotes the control volume tightly enclosing the structure, which is marked by the red dashed line in Figure 2.1b. \(\boldsymbol{\tau}\) denotes the Reynolds stress tensor. \(\mathbf{n}\) is the surface normal vector, and \(\mathbf{n}_x = (1, 0, 0)\) denotes the streamwise direction.

In this study, both methods were applied to the fully-resolved case and solid barrier case. For the porous media case, since there was no physical structure, only the second method was used. In fact, according to Equation (2.8), at steady state, the pressure and shear stress on the enclosing box should be balanced by the volume integration of \(S_i\) within \(CV\) if we assume other forces are small. This is also the foundation for the momentum analysis in previous studies (Shields and Gippel, 1995; Manners and Doyle, 2008; Turcotte et al., 2015).

### 2.2.2 Geometry Preparation and Mesh Generation

In this study, the ELJ flume experiments in Gallisdorfer et al. (2013); Bennett et al. (2015) were simulated and analyzed. The water depth was \(H = 114\) mm, and the flume width was \(B = 1900\) mm. For numerical simulations, the 3D geometry of the ELJ structure was drawn in FreeCAD, as shown in Figure 2.1. The ELJ structure
consisted of 18 key elements and four cross-spanning elements. The elements were cylinders with a disk on one head. The disks were 64 mm in diameter and 14 mm in thickness. For the key elements, the cylinder was 32 mm in diameter and was oriented towards the streamwise direction with the disk upstream. For cross-spanning elements, the cylinders were 19 mm in diameter and were oriented towards the spanwise direction with the disk in the center of the stream. The ELJ structure was fixed to the right bank. The total length of ELJ was $L = 450$ mm, and the width was $W = 400$ mm. The upstream length of the flume in the computational model was 5 $L$, and the downstream length was 20 $L$, as shown in Figure 2.1b.

For the fully-resolved case, the body-fitted mesh was generated by utilizing the snappyHexMesh mesh generator in the open source platform OpenFOAM (ESI-OpenCFD, 2016). The basic idea of snappyHexMesh is to remove the cells inside the geometry from the background mesh and then move the nearby points onto the ELJ boundary to fit the geometry. Sophisticated geometric operations, such as splitting, refining and merging, are used during the process. Here, the background mesh domain was 9450 mm long, 1900 mm wide and 114 mm deep. The background grid size was 10 mm in the $x$, $y$ and $z$ directions. In order to reduce the computational cost, for the downstream part (11$L$ distance from the outlet), the background grid size was 20 mm. The total cell number of the background mesh was about 2.2 million. Then, in order to obtain better resolution, the near-structure meshes were refined during the mesh generation process. The final mesh for the
Figure 2.1: The engineered log jam (ELJ) structure in the flume. In the 3D view, a fake free surface was added.

The fully-resolved case had a total cell number of about 5.5 million. The average value of $y^+$ (cell size in wall unit) is one on the surface of ELJ and 10 on the channel bottom and banks. The distribution of average $y^+$ can be found in Figure 2.3a, and the final mesh for the fully-resolved case can be seen in Figure 2.2. For the porosity model cases, the background mesh was adopted, and the porous region was marked according to the location of ELJ in the flume. For the solid barrier case, the mesh
was generated by the blockMesh mesh generator in OpenFOAM with the same resolution as the background mesh. The $y^+$ distribution can be found in Figure 2.3b.

![Zoom-in view of the fully-resolved ELJ mesh generated by snappy-HexMesh.](image)

**Figure 2.2**: Zoom-in view of the fully-resolved ELJ mesh generated by snappy-HexMesh.

### 2.2.3 Boundary Conditions

The boundaries of the channel bottom, sidewalls and the ELJ structure were set as walls. Due to the geometric complexity of the ELJ, $y^+$ (near-wall grid size) has a wide range (0 < $y^+$ < 100). To properly model the near-wall flow dynamics, the Spalding wall function (Spalding, 1961) was applied on these wall boundaries. It fits well with the velocity distribution in laminar sublayer, buffer region and outer layer. As a result, it imposes the correct wall boundary condition regardless of the near-wall grid resolution. The free surface was treated as a free-slip rigid lid.
The channel outlet was set as a far field, and no back flow was allowed. The inlet boundary condition is more difficult to impose in order to reproduce the physical experiment in Bennett et al. (Bennett et al., 2015). The mean velocity of the incoming flow was $u_0 \approx 0.124$ m/s. The Froude number ($Fr$) was 0.124, and the Reynolds number ($Re = u_0H/\nu$) was around $10^4$. Thus, the flow was subcritical.
and turbulent.

For the background case without ELJ, an internal cyclic boundary condition was used at the inlet to create a fully-developed open channel flow. If there is no structure in the open channel, the internal cyclic boundary condition works well. However, if there is a structure such as ELJ inside this subcritical open channel flow, the disturbance due to the ELJ can interfere with the internal cyclic boundary condition at the inlet. In order to solve this problem, a turbulent inlet library of instantaneous velocity, SGS turbulent kinetic energy (TKE) and SGS viscosity were pre-computed in the background case at a time interval of 0.06 s. Then, for the fully-resolved case, the pre-computed instantaneous flow field was mapped onto the inlet boundary.

2.2.4 Configuration of the Porous Media Model

In the porosity model, $D$ and $F$ need to be specified in the zone occupied by ELJ. From Equations (2.10) and (2.11), $\alpha$, $\beta$ and $d_{50}$ are the key parameters to determine the values of $D$ and $F$. Different combinations of these parameters may lead to different results (Jensen et al., 2014). The flow regime within ELJ is classified by the pore Reynolds number (Burcharth and Anderson, 1995; Jensen et al., 2014):

$$Re_p = \frac{\langle u \rangle d_{50}}{n\nu}$$  \hspace{1cm} (2.15)
where $\langle \bar{u} \rangle$ denotes the mean flow velocity within the porous zone. This study assumes a similar definition of different flow regimes as in coastal engineering (Jensen et al., 2014), i.e., the Forchheimer flow regime ($10 < Re_p < 150$), the transitional flow regime ($150 < Re_p < 300$) and the fully-turbulent flow regime ($Re_p > 300$).

In this study, $d_{50}$, an equivalent grain size, needs to be specified in Equation (2.15). According to the geometric information of ELJ, two important length scales were identified as the possible $d_{50}$ values, i.e., key element diameter (32 mm) and cross-spanning elements diameter (19 mm). For comparison, a small value of $d_{50} = 1$ mm was also tested. For the first two pore sizes, $Re_p > 300$, and the flow within ELJ is fully turbulent. For the small equivalent pore size of 1 mm, the flow is in the transition regime. Regarding $\alpha$ and $\beta$, two combinations from Gent (Gent, 1995) and Jensen et al. (Jensen et al., 2014) were tested. In summary, six combinations of $\alpha$, $\beta$ and $d_{50}$ were tested (see Table 2.1), and the results were compared with the experiments.

Theoretically, in the solid barrier case, $n = 0$ in Equations (2.10) and (2.11); thus, $S_i$ becomes infinity in Equation (2.9). However, to avoid numerical instability caused by the infinity $S_i$, the ELJ was simply modeled as an impervious box.
2.2.5 Computational Platform

The 3D computational modeling was performed with OpenFOAM, an open-source computational physics platform (ESI-OpenCFD, 2016). The mesh generation tool snappyHexMesh in OpenFOAM was used to generate the body-fitted mesh for ELJ. For the porous media case, a special solver was written to solve the fluid equations with the porous media-induced drag term. The equations were solved using a finite volume method. The time derivative terms were discretized with a second-order implicit backward scheme. The gradient terms and divergence terms were discretized using the Gauss theorem, and a second-order linear interpolation was used to interpolate the values at cell centers to faces. The porosity model introduced in this paper has already been implemented in OpenFOAM. The details of the model can be found in its manual and the source code.

The computational efficiency of a model depends on many factors, including parallel computing method, flow field initialization, and so forth. In this study, based on the core-hours used in each simulation, the porosity model and the solid barrier case used about 1/200 and 1/50, respectively, of the computational time by the fully-resolved case.
2.3 Results and Discussions

In the following, the fully-resolved ELJ model is first validated with experimental data. Then, the results from simplified representations, i.e., the porosity model and solid barrier, are presented and analyzed.

2.3.1 Validation of the Fully-Resolved Configuration

For the fully-resolved case, which serves as the baseline for the evaluation of the other two simplifications, we found that the turbulent inlet condition was critical for better comparison with the experiments in Bennett et al. Bennett et al. (2015). In this study, a background case without ELJ was first run to get a fully-developed open channel flow, which was then used at the inlet. Figure 2.4 compares the cross-sectional distributions of mean streamwise velocity $u$ and turbulent kinetic energy (TKE) $k$ between the numerical and experimental results for the background case. The simulated mean fields were based on averages over 140 seconds, while in Bennett et al.’s (Bennett et al., 2015) experiment, the averaging time was 180 s. The results from the 140 s averaging time have little difference from those with 100 s averaging time. Therefore, the 140 s averaging time was used in this work. From the comparison in the figure, a similar pattern and comparable magnitude can be observed for both velocity and TKE. The velocity core is slightly stretched in the simulation than in the experiment.
Bennett et al. (Bennett et al., 2015) measured the flow on two cross-sections located at the immediate upstream and downstream sides of ELJ at $x = -95$ mm and $x = 545$ mm, respectively. Figure 2.5 presents the comparison of the upstream and downstream cross-sectional distribution of the mean streamwise velocity. ELJ was located adjacent to the right sidewall (looking downstream). The red lines indicate the edge of ELJ. On the upstream cross-section, reduced velocity can be observed in the front of ELJ. The pattern and magnitude are quite similar in both experiment and the simulation. However, the high flow region from computational model slightly deviates from the experiment, which might be due to the flattened high velocity core in the simulated incoming flow. For the downstream cross-section, a large velocity gradient can be observed on the eddy of ELJ in both cases. The velocity in the upper main channel flow adjacent to ELJ is slightly larger in LES than in the experiment. However, they are very comparable in general.

The fully-resolved simulation case also shows good overall agreement on TKE with the experiment (Figure 2.6). In the computational model, the TKE $k$ was calculated as:

$$k = k^{SGS} + \frac{1}{2} \left( u'_x^2 + u'_y^2 + u'_z^2 \right)$$

(2.16)

where $u'$ denotes the resolved velocity fluctuation in three directions and $k^{SGS}$ denotes the unresolved sub-grid scale kinetic energy. The simulated upstream distribution of TKE has a noticeable difference from the experiment because of the overall small magnitude. For the downstream cross-section, the shear layer induced
Figure 2.4: Comparison of the cross-sectional distributions of mean streamwise velocity and turbulent kinetic energy (TKE) between the numerical and experimental results for the background case without ELJ. The variables are made dimensionless with channel width $B$, water depth $h$ and mean velocity $u_0$.

by ELJ created a high TKE zone, which was captured well by the computational model.

In the fully-resolved case, the drag force on ELJ can be directly calculated with the integration of pressure and shear stress on the structure surface, namely the first method, which gives $F_d = 0.592 \pm 0.042 \, \text{N}$. The second method of using the momentum equation with the simulated flow field gives $F_d = 0.596 \, \text{N}$, which is very close to the result from the first method. For comparison, the measured drag force in Bennett et al. (Bennett et al., 2015) was $F_d = 0.851 \pm 0.060 \, \text{N}$, which indicates that the simulation underestimated the drag force. This underestimation might be attributed to several factors, including the incoming flow condition, the mismatch between the fabricated
ELJ in experiment and the design, etc. In particular, the incoming flow cannot be exactly reproduced as seen in Figure 2.4, which leads to slower impingement onto ELJ and lower drag force. In the experiment, it will be ideal, but very time consuming, to measure in great detail the inflow conditions to be used at the inlet in the simulations. In this way, the discrepancy could be greatly reduced.

From the above comparison on mean velocity, TKE and drag force, the fully-resolved case reproduced the experimental condition well considering the complexity of the problem. In the following, the fully-resolved case will be used as the base for comparison with the porous media and solid barrier simplifications.
2.3.2 Calibration of Porous Media Model

The calibration of the porous media parameters ($\alpha$, $\beta$ and $d_{50}$) is based on the fully-resolved case, which itself has been validated with experiments in previous section. As summarized in Burcharth et al. (Burcharth and Anderson, 1995) and Jensen et al. (Jensen et al., 2014), the selection of $\alpha$ and $\beta$ varies for different porous materials and packing modes. Six combinations of $\alpha$, $\beta$ and $d_{50}$ were tested for the ELJ structure. Table 2.1 lists the parameters, including the intrinsic and inertial permeability, of the six different combinations.

To quantify the difference between the porosity cases and the fully-resolved case, the cross-sectional error distributions for mean velocity and mean TKE are plotted in Figure 2.7. The error is defined as the porous media model results minus
the fully-resolved simulation values. Figure 2.8 shows the transverse distribution of mean streamwise velocity and TKE on the upstream and downstream cross-sections at $z/z_0 = 0.4$. To reduce the length of this paper, we only plot the results from the case of $\alpha = 1000$ and $\beta = 1.1$, which gives better results than the case of $\alpha = 200$ and $\beta = 2$. The red dashed rectangle represents the location of ELJ. As shown in Figures 2.7a and 2.8, for the upstream cross-section, large velocity errors mainly occur near the bed for small $d_{50}$. When $d_{50}$ is small (1 mm), the porosity model underestimates the streamwise velocity at the upstream side of ELJ, which is caused by the lower permeability suppressing the flow. The velocity suppression is more significant near the bank and free surface. In contrast, larger $d_{50}$ leads to higher permeability and thus overestimates the streamwise velocity in the upstream side of ELJ. For the downstream cross-sections, error occurs mainly in the shear layer. All porosity model cases underestimate the streamwise velocity in the shear layer, with slight overestimation in the near bed region. This near-bed overestimation is less significant when $d_{50}$ has the value of the key element diameter. For the TKE distribution shown in Figures 2.7b and 2.8, the error distribution is different from velocity. With $d_{50} = 32$ mm and 19 mm, TKE was under-predicted in the shear layer. However, if $d_{50}$ is very small (1 mm), TKE was dramatically over-predicted in the shear layer. This is caused by the fact that lower permeability of the porous media region forces more flow to the main channel and generates a larger velocity gradient in the shear layer.
In addition to the overall difference of velocity and TKE, an integral error over a cross-sectional plane can be defined as:

\[
\varepsilon = \frac{1}{A_{\text{Surf}}} \sum_{i=0}^{N} \left[ (\eta_{\text{porosity},i} - \eta_{\text{full},i})^2 \Delta A_i \right]
\]  
(2.17)

where \(A_{\text{Surf}}\) denotes the total area of the surface, \(\Delta A_i\) denotes the area for the \(i\)-th sub-area and \(\eta_i\) denotes the flow variable on the \(i\)-th sub-area. As shown in Table 2.1, the general trend is that smaller \(d_{50}\) resulted in smaller permeability, and the error \(\varepsilon\) became larger for both upstream and downstream velocity and TKE. However, the drag force error does not always follow this trend. For all six calibration cases, the results also show that the downstream error \(\varepsilon\) is always larger than the upstream due to the rich turbulent dynamics behind the ELJ. As seen in Table 2.1, the solid barrier case also over-predicted the drag force, and the error is much larger than that of the porosity case. The solid barrier case is closer to the \(d_{50} = 1\) mm porosity case in terms of drag force and spacial integral error.

Table 2.1: Drag force for the fully-resolved case, porosity cases and the solid barrier case.

<table>
<thead>
<tr>
<th>(\alpha)</th>
<th>(\beta)</th>
<th>(d_{50}) (mm)</th>
<th>(K) (m²)</th>
<th>(K_i) (m²)</th>
<th>(F_d/F_{\text{fully}})</th>
<th>(\varepsilon_{u/u_0,\text{up}})</th>
<th>(\varepsilon_{u/u_0,\text{down}})</th>
<th>(\varepsilon_{k/u_0^2,\text{up}})</th>
<th>(\varepsilon_{k/u_0^2,\text{down}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1.1</td>
<td>32</td>
<td>(2.34 \times 10^{-6})</td>
<td>(2.31 \times 10^{-8})</td>
<td>1.111</td>
<td>0.065</td>
<td>0.117</td>
<td>0.341</td>
<td>1.295</td>
</tr>
<tr>
<td>1000</td>
<td>1.1</td>
<td>19</td>
<td>(8.24 \times 10^{-7})</td>
<td>(1.37 \times 10^{-8})</td>
<td>1.157</td>
<td>0.075</td>
<td>0.146</td>
<td>0.337</td>
<td>1.273</td>
</tr>
<tr>
<td>1000</td>
<td>1.1</td>
<td>1</td>
<td>(2.28 \times 10^{-9})</td>
<td>(7.22 \times 10^{-10})</td>
<td>1.113</td>
<td>0.131</td>
<td>0.162</td>
<td>0.673</td>
<td>2.570</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
<td>32</td>
<td>(1.17 \times 10^{-5})</td>
<td>(1.27 \times 10^{-8})</td>
<td>1.134</td>
<td>0.070</td>
<td>0.129</td>
<td>0.347</td>
<td>1.267</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
<td>19</td>
<td>(4.12 \times 10^{-6})</td>
<td>(7.55 \times 10^{-8})</td>
<td>1.153</td>
<td>0.079</td>
<td>0.146</td>
<td>0.369</td>
<td>1.252</td>
</tr>
<tr>
<td>200</td>
<td>2</td>
<td>1</td>
<td>(1.14 \times 10^{-8})</td>
<td>(3.97 \times 10^{-10})</td>
<td>1.148</td>
<td>0.126</td>
<td>0.168</td>
<td>0.649</td>
<td>2.584</td>
</tr>
<tr>
<td>solid barrier</td>
<td>0</td>
<td>0</td>
<td>(1.246)</td>
<td>(0.145)</td>
<td>(0.238)</td>
<td>(0.636)</td>
<td>(3.123)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overall, based on the results for drag force $F_d$ and error $\varepsilon$, the combination of $\alpha = 1000$, $\beta = 1.1$ and $d_{50} = 32$ mm gave the best results and, thus, was selected for further analysis.

Figure 2.7: Error distribution on the upstream and downstream cross-sections between the porosity cases and the fully-resolved case. The white blank region means that the values exceed the range of the colorbar.

### 2.3.3 Evaluation of Different Representations in the CFD Model

The effects of ELJ representation were investigated with three cases, including the validated fully-resolved case, the calibrated porous media case and the solid barrier case. The aspects of the ELJ hydraulics evaluated included flow adjustment, near-surface and near-bottom flow and turbulence and flow structure.
Figure 2.8: Transverse distribution of the mean streamwise velocity and TKE on the upstream and downstream cross-sections at $z/h = 0.4$.

Flow Adjustment

To the best of our knowledge, there is no quantitative study on how flow will adjust when it passes an in-stream structure, which is usually bounded by stream banks. However, some recent studies investigated unbounded flow around finite and semi-finite (porous) vegetation patches (White and Nepf, 2003; Rominger and Nepf, 2011; Chen et al., 2012). They revealed the relationship between the porous media properties and the flow adjustment in terms of streamwise velocity and wake length. We used similar analysis with the aim to find comparable relationships. As in previous studies (Rominger and Nepf, 2011; Chen et al., 2012), the flow domain can be partitioned into three regions as shown in Figure 2.9, namely the upstream adjustment region, the interior adjustment region and the wake region. For unbounded wake flows, two wake length scales ($L_1$ and $L_w$) can be defined. $L_1$ denotes the length scale of the steady wake region where the velocity remains
almost constant. $L_w$ denotes the length scale of the distance between the structure and the von Kármán vortex street. However, Kim et al. (Kim et al., 2015) showed that the von Kármán vortex street cannot form in semi-bounded flows, which is consistent with the numerical results in this paper. These two lengths also manifest themselves in the signature of turbulence. $L_1$ is approximately located at the first peak of turbulence intensity in the wake region, while $L_w$ is located at the second peak. Chen et al. (Chen et al., 2012) proposed the following empirical wake length equations:

$$\frac{L_1}{L} = \begin{cases} \frac{2.5 \left[ 8 - \frac{C_d a W}{C_d a W} \right]}{C_d a W} & \text{if } C_d a W < 4 \\ 2.5 & \text{if } C_d a W > 4 \end{cases}$$

(2.18)

$$\frac{L_w}{L} = \begin{cases} 1.2 & \text{if } C_d a W > 20 \\ 1.2 + (25 \pm 10)C_d a W^{-0.9 \pm 0.2} & \text{if } C_d a W < 20 \end{cases}$$

(2.19)

where $C_d$ can be calculated with Equation (2.13). $a = m d$, where $m$ is the element number per unit planar area and $d$ is the diameter of the cylindrical element used in Chen et al. (Chen et al., 2012). $C_d a W$ denotes the flow blockage due to the obstruction. For ELJ, we use the key elements to approximate $n$ and $d$. As a result, $d = 0.032$ m and $m = 14/(0.4 \text{ m} \times 0.5 \text{ m})$.

Chen et al. (Chen et al., 2012) measured the centerline velocity at half depth in flume experiments to represent the depth-averaged velocity for their unbounded
flow around a porous vegetation patch. In this study, since we have the simulated velocity distribution around ELJ, local cross-sectional averaging was performed within the adjustment and wake regions. Local regional averaging is more suitable for this semi-bounded flow. Figure 2.10 shows the longitudinal profiles of cross-sectionally-averaged streamwise velocity and TKE within the adjustment and wake regions. For the velocity profile, due to the semi-boundedness (wall effect), the incoming normalized streamwise velocity is only about 0.43. The trend of the velocity profiles is in accordance with that in Chen et al. (Chen et al., 2012). In the steady wake region, the larger pore size \(d_{50}\) case tends to match better with the fully-resolved case. The velocity profiles of solid barrier and small pore size case are similar, which makes sense because the small pore size reduces the permeability. They both show negative streamwise velocity inside the steady wake region. This is induced by the strong boundary layer separation and vortex generated at the edge of the object. With the increase of porosity, more passing-through flow reduces the negative pressure gradient, pushes the wake flow further downstream and thus enlarges the wake. For TKE, it has a relatively large value at the upstream side of ELJ edge and a small value at the immediate downstream side where the area
is shielded. After that, TKE increases rapidly in the steady wake region. We also found that the growth rate and asymptotic TKE value are affected by the permeability and the resulted flow passing through ELJ. Reduced pass-through flow will induce higher shear, more eddies and, thus, higher TKE in the steady wake region. Neither the porosity model cases, nor the solid barrier case reproduced a good match on TKE in the whole of the wake zones. The porosity model with a large pore size matched with the fully-resolved case in the steady wake zone, while the solid barrier case matched better further downstream. For all porosity model cases, they tend to maintain high turbulence even after the steady wake region.

Figure 2.10: Longitudinal profiles of cross-sectionally-averaged streamwise velocity and TKE within the adjustment and wake regions. The $x$ coordinate is normalized by the length of ELJ ($L$). The gray area denotes ELJ.

Table 2.2 shows the simulated $L_1$ and $L_w$ from the cases shown in Figure 2.10. The length scales were determined by the peak of turbulent intensity. However, we
found that for the porosity cases, it was very difficult to determine the exact value of $L_w$ due to the lack of clear peaks, as shown in Figure 2.10. Thus, only the range of $L_w$ is presented. The results indicate that the porosity model with $d_{50} = 19$ mm produced a better match on $L_1$ with the fully-resolved case. Again, the increase of $d_{50}$, i.e., porosity, will increase the wake length $L_1$. According to Equations (2.13), (2.18) and (2.19), if we use the channel mean velocity in the equations, we obtain $C_d a W = 1.7$, $L_1 = 20.1 L$ and $L_w = 36.5 L$, which are much larger than the values in Table 2.2. However, if the mean velocity of the whole channel was replaced by the local mean value experienced by the ELJ, i.e., $U = 0.43 u_0$, we obtain $C_d a W = 9.4$, $L_1 = 5.6 L$ and $L_w = 10.1 L$. Obviously, the latter is closer to the simulated values. The replacement of incoming flow velocity with the local mean is reasonable and necessary because of the side wall effects.

Table 2.2: Measured wake length $L_1$ and $L_w$.

<table>
<thead>
<tr>
<th></th>
<th>$L_1$</th>
<th>$L_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully resolved</td>
<td>6.1</td>
<td>11.3</td>
</tr>
<tr>
<td>Porosity $d_{50} = 32$ mm</td>
<td>7.8</td>
<td>$10 \sim 13$</td>
</tr>
<tr>
<td>Porosity $d_{50} = 19$ mm</td>
<td>6.3</td>
<td>$10 \sim 13$</td>
</tr>
<tr>
<td>Porosity $d_{50} = 1$ mm</td>
<td>4.6</td>
<td>$10 \sim 13$</td>
</tr>
<tr>
<td>Solid barrier</td>
<td>4.5</td>
<td>7.6</td>
</tr>
</tbody>
</table>
Near-Surface Velocity

Another important metric is the near-surface velocity around the in-stream structure, which impacts a host of processes, such as aeration and heat transfer. Figure 2.11 shows the time-averaged streamwise velocity distribution at 5 mm below the free surface for different cases. For the porosity model, only the case with $\alpha = 1000$, $\beta = 1.1$ and $d_{50} = 32$ mm is shown. The comparison shows that the surface velocity is not so sensitive to how ELJ is modeled. The only noticeable difference is in the steady wake region. For the fully-resolved case and the porosity case, there is only a small region showing negative streamwise velocity. However, for the solid barrier case, a much larger region has negative streamwise velocity, which is due to the fact that no passing-through flow causes much stronger recirculation behind the solid barrier. This result is consistent with Figure 2.10.
Near-Bed Turbulence

In-stream structures also affect river morphology by changing sediment transport. Thus, near-bed turbulence, the driver of sediment motion, is evaluated. The time-averaged TKE near the bed is used as a proxy for sediment entrainment capacity (Yang et al., 2016). The near-bed TKE distributions are plotted in Figure 2.12. The data were sampled at a height of 15 mm above the bed. From the comparison, it is observed that the porosity model gave a better match with the fully-resolved case. The porosity model underestimated TKE in the shear layer, and the shear layer itself was delayed. The solid barrier case greatly overestimated TKE in the
shear layer. Thus, from the sediment transport point of view, the solid barrier approximation to the ELJ will over-predict the scour hole around the structure.

**Flow Structure**

The effects of ELJ representation in models can be evaluated at the simulated flow features. Figure 2.13 shows an instantaneous flow field (streamlines) in the wake zone, which is colored by the magnitude of instantaneous velocity. These streamlines were sourced from a horizontal line parallel to the downstream side of ELJ. The source line was located at 50 mm above the bed. As can be seen in the figure, significant discrepancy exists in the steady wake zone in different model representations. For the fully-resolved case, small and irregular flow curvatures can be observed in the steady wake zone, most of which stem from the element surface in ELJ. The incoming flow has to go through the irregular pores within ELJ, and it results in very rich flow dynamics. In the porous media case, such dynamics are totally lost, and the streamlines are much smoother than the other two cases in the steady wake zone. Low permeability in the porous media region only slows down the flow through ELJ. The only similarity between the porous media case and the fully-resolved case is that the passing-through flow causes the delay of the shear layer. In the solid barrier case, a larger recirculation zone and a strong shear layer can be seen at the lee side of the block.
Figure 2.12: Time-averaged TKE distribution in the plane at 15 mm above the bed.

Figure 2.14 shows the instantaneous isosurfaces of $\lambda_2 = 3$. $\lambda_2$ is a commonly-used flow quantity to visualize the vortex structures (Jeong and Hussaini, 1995). Substantial differences can be observed among the three cases. The fully-resolved case shows strong coherent flow structures inside the ELJ, also shown in Figure 2.15. In the shear layer, it takes a long distance for the vortex to shed behind the ELJ to re-attach to the right bank. However, in the porous media case, no such coherent structure was observed inside the porous media domain at the same value of $\lambda_2$. The vortex structure is also extremely weak in the near-field compared with the fully-resolved case. However, it shows similar vortex structure in the far-field.
as the fully-resolved case. In the solid barrier case, the contour of $\lambda_2 = 3$ shows comparable strong coherent structures as the fully-resolved case in the near-field shear layer. However, the shear layer of the solid barrier case is more diffusive in the far field than the other two cases. In addition, the reattachment distance is much smaller in the solid barrier case. This reduction of reattachment distance is caused by the lack of passing-through flow in the solid barrier case. Overall, in terms of flow structure, the porous media case models well in the far field, while the solid barrier case predicts better in the near field. Nevertheless, neither of them are able to resolve the flow structure inside the ELJ.

Figure 2.13: Instantaneous flow field represented by streamlines in the wake zone. The streamlines are colored by the instantaneous velocity.
Fully resolved

Porosity \( \alpha = 1000 \quad \beta = 1.1 \quad d_{50} = 32 \text{ mm} \)

Solid barrier

Figure 2.14: Instantaneous flow structure represented by the isosurfaces of \( \lambda_2 = 3 \).
Figure 2.15: Zoom-in view of the instantaneous flow structure for the fully-resolved case with the isosurfaces of $\lambda_2 = 3$.

2.4 Conclusions

Using three-dimensional computational fluid dynamics modeling, we evaluated the effects of different modeling representations of ELJ, a popular in-stream structure for river restoration and erosion control. Three different modeling cases, i.e., a fully-resolved case, a porosity model case and a solid barrier case, were simulated. The computational model was validated with experimental data in a flume where a scaled model of ELJ was placed. The simulated results were examined and analyzed on various aspects related to the functionality and stability of ELJ. The
applicabilities and limitations of each simplified representation of complex ELJ geometry were discussed.

Full resolution of the ELJ geometry is not feasible for wide use in practice because of the prohibitive computational cost. In this work, it served as the baseline for comparison with other methods. The fully-resolved case reproduced the velocity and TKE distributions well at the upstream and downstream cross-sections near the ELJ. When the porosity model was used, the porosity coefficients were first calibrated to get $\alpha = 1000$, $\beta = 1.1$, $d_{50} = 32$ mm. We found that the equivalent grain size $d_{50}$ should scale as the key element diameter in this particular ELJ. The calibrated porosity model only showed satisfactory performance on drag force, velocity and TKE. The solid barrier case generally gave the least accurate results.

Further comparison showed that the modeling of the passing-through flow has a significant impact on the wake and the shear layer, which is summarized in Table 2.3. The porosity model approximates the near-surface velocity and near-bed TKE in the steady wake region well, while the solid barrier model performed slightly better further downstream. According to the near-bed TKE, the shear layer is under-predicted in the porosity model and over-predicted in the solid barrier model. The solid barrier model can predict well the strong flow structure in the near-field shear layer, while the porosity model performs better in the far-field shear layer. However, neither the porosity model, nor the solid barrier model can capture the flow structure inside the ELJ. The flow adjustment analysis shows that patches
with higher flow blockage ($C_d a W$) have larger wake length. The wake lengths $L_1$ and $L_w$ from Equations (2.18) and (2.19) agree well with simulated results only if the local velocity, instead of channel mean velocity, is used.

Table 2.3: Comparison between the porosity model and the solid barrier model.

<table>
<thead>
<tr>
<th></th>
<th>Porosity</th>
<th>Solid Barrier</th>
</tr>
</thead>
<tbody>
<tr>
<td>near-bed TKE</td>
<td>under-predicted</td>
<td>over-predicted</td>
</tr>
<tr>
<td>near-field flow structure</td>
<td>well-predicted</td>
<td>extremely weak</td>
</tr>
<tr>
<td>far-field flow structure</td>
<td>too diffusive</td>
<td>well predicted</td>
</tr>
<tr>
<td>flow structure inside ELJ</td>
<td>unpredictable</td>
<td>unpredictable</td>
</tr>
<tr>
<td>wake length $L_1$ and $L_w$</td>
<td>under-predicted</td>
<td>over-predicted</td>
</tr>
<tr>
<td>computational efforts (of fully resolved)</td>
<td>about 1/200</td>
<td>about 1/50</td>
</tr>
<tr>
<td>drag force (of fully resolved)</td>
<td>about 1.11</td>
<td>about 1.25</td>
</tr>
</tbody>
</table>

In conclusion, when the porosity model or solid barrier representation is used for ELJ, the benefits of the significantly reduced computational cost come with the price of missing physical information that might be important for the intended purpose of in-stream structures. The porosity model, after careful calibration, seems to perform better than the solid barrier model, especially the drag force. However, this work also identified substantial loss of accuracy in terms of velocity, turbulent shear, sediment transport capacity and coherent structure in the vicinity and the wake zone of ELJ. Thus, care needs to be taken to interpret and use the porosity
model results in practice. If the emphasis is in the far field, then the simulation results using both the porosity model and the solid barrier representation did not show dramatic difference from the fully-resolved case. Therefore, the use of such simplified models may be justified.

References


ESI-OpenCFD (2016). OpenFOAM.


Chapter 3

An immersed boundary method with smooth wall shear for morphodynamics modeling

3.1 Introduction

Since its inception in Peskin (1972), the immersed boundary (IB) method has been widely used in many disciplines for the modeling of fluid flow bounded by rigid or flexible walls. Great progress has been made in the past several decades on various aspects of IB method. The contribution of this paper is an adaptive method for accurate and smooth wall shear distribution in high Reynolds number flows. In many applications, wall shear is a key parameter for other processes, such as erosion and sedimentation. In many previous researches on IB method,
focuses have been on the flow field around object, resulted drag and lift forces, and other flow quantities away from the immersed boundary (Kang et al., 2009; Seo and Mittal, 2011; Sotiropoulos and Yang, 2014; Bernardini et al., 2016). In drag and lift forces calculation, most cases in previous researches are pressure dominant with wall shear stress only as a secondary contributor (Capizzano, 2016; Constant et al., 2017; Posa et al., 2017). In addition, the drag and lift forces are the integral of pressure and shear on the surface. They do not necessarily reflect the accuracy and smoothness of pressure and shear distributions. There is a need to accurately model the turbulent flow process near the immersed boundary such that the wall shear behaves properly. For example, in the field of sediment transport, the sediment mass transport rate along the bed surface is a function of the wall shear (Liu and García, 2008; García, 2008). The change of bed elevation due to sediment transport is proportional to the divergence of the sediment flux. Thus, the bed surface evolution is related to the divergence of the wall shear stress (to some power). It is of paramount importance to have a smooth distribution of wall shear. Mathematically, the application of divergence operator on a not-so-smooth wall shear field will further reduce the smoothness (Barsky, 2013; Stoer and Bulirsch, 2013) and may cause nonphysical spikes in the erosion rate.

The method presented in this paper is designed for the specific application of hydrodynamics and morphodynamics around complex structures in rivers and coastal areas. However, the new IB model is general and can be used in any other
applications. The basic idea of IB method is to account for the effect of walls to the fluid motion with special treatment near the surface. Comprehensive reviews on IB method can be found in for example Peskin (2002), Mittal and Iaccarino (2005), Hou et al. (2012), and Sotiropoulos and Yang (2014). IB methods can be classified into two broad categories, namely continuous forcing approach and discrete forcing approach. For turbulent flow, the treatment near an immersed boundary is often tricky and not all IB methods are suitable for the implementation of traditional wall functions developed for body-fitted meshes.

The continuous forcing approach utilizes an extra forcing term in the governing equations before discretization to model the effect of wall. This forcing term is usually based on physical laws other than the Navier-Stokes equations, such as the Hooke’s law for the elastic material (Peskin, 1972) and the porous media flow assumption (Angot et al., 1999; Khadra et al., 2000). In terms of implementation scheme, the continuous forcing approach is independent of spatial discretization. In this approach, smooth distribution functions are often used to overcome the stiffness problems at fluid-structure interface. Though tremendous efforts have been devoted to improve the accuracy and efficiency of this approach (Beyer and Leveque, 1992; Saiki and Biringen, 1996; Lai and Peskin, 2000; Bao et al., 2016, 2017; Stein et al., 2016, 2017), the application of the continuous forcing in high Reynolds number flows is still limited due to the smearing problem near boundary. Partially thanks to this limitation of failing to provide sharp representation, wall
function is rarely used in this approach.

In comparison to the continuous forcing approach, the discrete forcing approach adds an extra forcing term into the discretized governing equations. In this approach, the forcing term is usually calculated from the available flow field and desired boundary conditions. It helps solve the smearing problem near the immersed boundary in the continuous forcing approach. As a result, it is also called the sharp interface method (Sotiropoulos and Yang, 2014). For example, an indirect imposition of boundary condition in the discretized momentum equation through a forcing term \( f \) was firstly developed by Mohd-Yusof (1997) as

\[
\frac{u^{n+1} - u^n}{\Delta t} = \text{rhs} + f
\]  

(3.1)

where \( \text{rhs} \) is the right-hand-side term of the incompressible Navier-Stokes momentum equation upon discretization, which includes advection term, diffusion term and viscous term. \( u^{n+1} \) is a desired velocity which considers the effect of immersed boundary and \( u^n \) is the velocity from previous time step. The forcing term \( f \) can be employed in either near-wall fluid cells (Fadlun et al., 2000; Verzicco et al., 2000; Uhlmann, 2005; Roman et al., 2009a) or ghost cells (Tseng and Ferziger, 2003; Iaccarino and Verzicco, 2003; Mittal et al., 2008). For the application in near-wall fluid cells, \( u^{n+1} \) is calculated through interpolation with surrounding fluid cell values and wall velocity. If the application is in ghost cells, the desired velocity \( u^{n+1} \) is usually calculated through extrapolation.
In the discrete forcing approach, a boundary condition on immersed boundary can also be imposed on the discretized momentum equation by modifying the flow variables near the immersed boundary. In general, velocity and pressure at IB cells or IB nodes are modified. IB cells and IB nodes are computational cells and notes located inside the fluid domain but very close to the immersed boundary. A typical example of this direct imposition method is the one developed in Gilmanov et al. (2003) and Gilmanov and Sotiropoulos (2005). In their method, the velocity, pressure and pressure gradient at an IB node are reconstructed through linear interpolation.

Another option to enforce condition on an immersed boundary is the cut-cell method (Clarke et al., 1986; Ye et al., 1999; Udaykumar et al., 2001; Meinke et al., 2013; Schneiders et al., 2016), which satisfies the mass and momentum conservation in the vicinity of an immersed boundary. However, this category of method usually involves sophisticated geometric operations and may be extremely difficult for 3D unstructured mesh (Mittal et al., 2008; Sotiropoulos and Yang, 2014).

In many engineering applications, the geometries of boundaries and obstacles in the turbulent flow field are very complex and may be moving. For example, the geometries of river bed are usually very irregular and often dynamically evolve under the influence of flow. The use of IB methods for the simulation of flow and bed evolution is very attractive. For typical engineering applications, the simulation domain is usually large and wall functions have to be used to reduce computational
cost. For accuracy and efficiency, the direct imposition method described above is a suitable choice. However, the literature on turbulence wall functions for IB method is very limited (Capizzano, 2011; Takahashi and Imamura, 2014; Capizzano, 2016; Bernardini et al., 2016; Tamaki et al., 2017). Another finding of our literature survey is that most of previous researches are not particularly aimed at better prediction of wall shear stress distribution, which is important if other processes depend on its value, gradient, divergence, or any other form of differential operations. We also implemented the wall functions based on those in the literature. However, the wall shear stress results were not good. It is found that the wall shear stress from the wall function near IB surface is heavily dependent on the near-wall distance, i.e., the distance of IB cell center to the immersed boundary. The sensitivity of wall shear to near-wall distance and the resulted nonuniform distribution motivated the authors to propose an alternative solution for this problem which is presented in this paper.

In this work, a finite-volume method (FVM) based IB method with direct imposition of boundary conditions is developed with the open source CFD platform OpenFOAM (The OpenFOAM Foundation, 2018). It is noted that there is an immersed boundary implementation in the existing versions of OpenFOAM, which is very briefly described in Jasak et al. (2014) and Jasak and Tuković (2015). The existing implementation suffers from the same problem of nonuniform and sometime nonphysical wall shear stress distribution described above. Based on the existing
IB method implementation in OpenFOAM (code structure and functionalities), we implemented our new algorithm. For the sake of completeness, this paper describes in detail the mathematical formulations and their implementation.

The paper is organized as follows. In Section 3.2, the governing equations and the $k - \epsilon$ turbulence model are introduced, along with their FVM discretization. Section 3.3 introduces the details of the new IB method, including the algorithms for different boundary conditions. In Section 3.4, an IB wall model for the $k - \epsilon$ turbulence model in conjunction with a new $y^+$-adaptation scheme is described. In Section 3.5, simulation cases will be shown to demonstrate the performance of the new IB method.

3.2 Governing equations and their discretization

3.2.1 Governing equations for flow and turbulence model

The Reynolds-averaged Navier-Stokes (RANS) equations for incompressible flows have the form of

$$\nabla \cdot \mathbf{u} = 0 \tag{3.2}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu + \nu_T) \nabla \mathbf{u} \tag{3.3}$$

where $\rho$ denotes constant fluid density, $\mathbf{u}$ and $p$ denote Reynolds-averaged velocity and pressure, respectively. $\nu_T$ is turbulent eddy viscosity.
For simplicity, the classic $k - \epsilon$ model is used for turbulence closure. The algorithms presented in this paper can indeed be applied to other RANS models. In the $k - \epsilon$ model, the turbulent eddy viscosity $\nu_T$ can be calculated as

$$\nu_T = C_\mu k^2 / \epsilon$$  \hspace{1cm} (3.4)

where $k$ is the turbulent kinetic energy (TKE), $\epsilon$ is the TKE dissipation rate, and $C_\mu = 0.09$. For incompressible turbulent flows, $k$ and $\epsilon$ are governed by the following transport equations (Launder and Spalding, 1974)

$$\frac{\partial k}{\partial t} + \nabla \cdot (k \mathbf{u}) = \nabla \cdot \left[ \left( \nu + \frac{\nu_T}{\sigma_k} \right) \nabla k \right] - \epsilon + P_k$$  \hspace{1cm} (3.5)

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (\epsilon \mathbf{u}) = \nabla \cdot \left[ \left( \nu + \frac{\nu_T}{\sigma_\epsilon} \right) \nabla \epsilon \right] + C_{\epsilon 1} P_k \frac{\epsilon}{k} - C_{\epsilon 2} \frac{\epsilon^2}{k}$$  \hspace{1cm} (3.6)

where $C_{\epsilon 1} = 1.44$, $C_{\epsilon 2} = 1.92$, $\sigma_k = 1.0$, $\sigma_\epsilon = 1.3$. $P_k$ denotes the production term of $k$, which is calculated as

$$P_k = 2\nu_t |\nabla \mathbf{u}|^2.$$  \hspace{1cm} (3.7)

### 3.2.2 Discretization method

A finite-volume (FV) method is adopted to discretize the governing equations (Jasak, 1996). The simulation domain is discretized into cells or control volumes. On each control volume, the volume integral of most derivative terms in the governing
equations is converted to surface integral with Gauss’s theorem as

$$\int_V \nabla \star \phi dV = \int_S dS \star \phi = \sum_f S_f \star \phi_f$$

(3.8)

where \( V \) is the control volume, \( S \) is the bounding surface of \( V \), \( \phi \) is a generic solution variable, which can be scalar or vector. The star notation \( \star \) represents any tensor product including inner, outer, and cross. The subscript \( f \) denotes each face of a control volume. \( S_f \) denotes the area vector of each face (Fig. 3.1). \( \phi_f \) denotes the value of \( \phi \) on a face \( f \), which is typically obtained through interpolation. A control volume \( P \) may have multiple neighbouring cells \( N \) and each neighbouring cell share one face with cell \( P \). The summation in Eq. (3.8) loops through all faces of cell \( P \).

Figure 3.1: Schematic view of a control volume and its neighbour. The blue volume represents the control volume \( P \). The red volume represents a neighbouring cell \( N \). The grey face represents the shared face between \( P \) and \( N \). \( S_f \) is the area vector of the shared face.

The detailed FV discretizations of temporal and spatial derivatives are listed in A. As an example and for the sake of immersed boundary method derivation, the
discretized momentum equation (Eq. (3.3)) has the form of

$$\frac{u^n - u^{n-1}}{\Delta t} + \sum_f \Phi_f u_f = -\frac{1}{\rho} \sum_f S_f p_f + \sum_f (\nu + \nu_T) f S_f \cdot (\nabla u)_f$$ (3.9)

where $\Phi_f$ denotes flow flux on face $f$, $u_f$, $p_f$, $(\nu + \nu_T)_f$, and $(\nabla u)_f$ denotes interpolated velocity, pressure, viscosity and velocity gradient respectively on face $f$.

### 3.2.3 Iterative solution algorithm for pressure-velocity coupling

For the derivation of the immersed boundary method, the pressure-velocity coupling method is briefly described here. There are numerous coupling algorithms, for example the SIMPLE algorithm for steady problems and the PISO algorithm for unsteady problem (Patankar and Spalding, 1972; Issa, 1986). The major difference between steady and unsteady algorithms is the inclusion of the temporal derivative term which contributes to the diagonal of the resulted linear system matrix and the right hand side source term. Both algorithms use somewhat similar predictor and correct steps.

If the previous iteration value and current iteration value are denoted by superscripts $n-1$ and $n$, respectively, the momentum predictor step can be written
\[
\begin{align*}
\frac{\mathbf{u}^n - \mathbf{u}^{n-1}}{\Delta t} + \sum_f \Phi_f^{n-1} \mathbf{u}^n_f &= -\frac{1}{\rho} \sum_f \mathbf{S}_f p_f^{n-1} + \sum_f (\nu + \nu_T)^{n-1} \mathbf{S}_f \cdot (\nabla \mathbf{u})^n_f \quad (3.10)
\end{align*}
\]

Since value at a shared face can be interpolated with values at the centers of cells \( P \) and \( N \), the discretized Eq. (3.10) can be expressed in a linearized form:

\[
\begin{align*}
a_P \mathbf{u}_P^n + \sum_N a_N \mathbf{u}_N^n &= -\frac{1}{\rho} \sum_f \mathbf{S}_f p_f^{n-1} + \frac{1}{\Delta t} \mathbf{u}_P^{n-1} \quad (3.11)
\end{align*}
\]

where \( a_P \) and \( a_N \), respectively, denote the diagonal and off-diagonal elements of the coefficient matrix \( A_u \). Denote the right hand side of the system as \( b_u \), the matrix form of the resulted linear system is

\[
A_u \mathbf{u} = b_u. \quad (3.12)
\]

In the corrector step, a pressure Poisson equation is solved to satisfy the continuity equation, which has the form of

\[
\begin{align*}
\sum_f (a_P)^{-1} \frac{1}{\rho} \mathbf{S}_f (\nabla p)^n_f &= \sum_f (a_P)^{-1} \mathbf{S}_f \cdot (-a_N \mathbf{u}_N^n)_f + \frac{1}{\Delta t} (a_P)^{-1} \mathbf{u}_P^n \quad (3.13)
\end{align*}
\]

where the values of the variables except the pressure are from the predictor step.
The linear equation for the pressure can be written as

\[ a'_p p^a_p + \sum_N a'_N p^a_N = \sum_f (a_p)^{-1} S_f \cdot (-a_N u^a_N)_f + \frac{1}{\Delta t} (a_p)^{-1} u^a_p \]  (3.14)

where \( a'_p \) and \( a'_N \) denotes the corresponding diagonal and off-diagonal components.

The linear system form can be expressed as

\[ A_p p = b_p \]  (3.15)

For the turbulence model, it is a common practice to separate the solution of the turbulence governing equations from the solution of the Navier-Stokes equations. In other words, the pressure-velocity coupling iteration is solved first to get the flow field. Then, the turbulence model is solved sequentially. In the new immersed boundary method presented in this work, since the algorithm changes the forcing term in the Navier-Stokes equation at every iteration, it is better to embed the solution of turbulence model within the velocity-pressure coupling iteration. For the \( k - \epsilon \) turbulence model, the governing equations are solved as follows. With the available \( k \) and \( \epsilon \), which are usually from last time step or iteration, the discretization of their governing equations will result the following linear systems

\[ A_k k = b_k \]  (3.16)
The solutions of $k$ and $\epsilon$ are then used to update the turbulence eddy viscosity $\nu_T$ with Eq. (3.4).

### 3.3 Treatment of immersed boundaries

A discrete forcing approach by directly imposing boundary condition is used. In this approach, velocity, pressure and other flow variables in the fluid cells near an immersed boundary are reconstructed through interpolation over a computational stencil. The specific implementations of this IB method are presented in the following subsections.

#### 3.3.1 Representation of immersed boundaries

Similar to other IB methods in the literature (Gilmanov and Sotiropoulos, 2005; Jasak et al., 2014; Jasak and Tuković, 2015; Gilmanov et al., 2015), the present method classifies the cells in a computational domain into three categories: IB cells, live cells, and dead cells (see Fig. 3.2). IB cells are the cells closest to the immersed surface $\Gamma_{IB}$ in the fluid region. They are used to impose conditions on immersed boundaries. The algorithm to determine the flags of IB, live and dead cells is as follows:

Step 1: find all the cells intersecting with immersed surface $\Gamma_{IB}$. 

$$A_\epsilon \epsilon = b_\epsilon \tag{3.17}$$
Step 2: label the cells, excluding the intersecting cells, in fluid and solid regions as live and dead cells, respectively.

Step 3: for each intersecting cell:

- if the cell center is in the fluid region, it is labeled as IB cell;
- otherwise, it is labeled as dead cell, and re-label its adjacent live cells (sharing a boundary) as IB cells.

For each IB cell, a “hit point” (black dot in Fig. 3.2) is then calculated as the normal projection of the IB cell center (black cross in Fig. 3.2) to the immersed boundary. An “image point” (black circle in Fig. 3.2) is the image of a hit point with respect to the IB cell center. These different points are used in the immersed boundary method.

![Figure 3.2: Schematic view of IB representation: IB cell centers (black cross), hit points (black dots), image points (black circles), IB cells (red filled), live cells (green filled), and dead cells (white filled). Immeresed interface $\Gamma_{IB}$ is represented by the blue curve.](image-url)
3.3.2 Reconstruction algorithm

To enforce the correct flow behavior near an immersed boundary $\Gamma_{IB}$, flow variables at IB cell centers are reconstructed based on their required boundary conditions. The reconstruction requires a proper interpolation stencil which depends on specific interpolation scheme. In this work, the interpolation stencil only contains live and IB cells and the building process of interpolation stencil is as follows:

Step 1: find the cell that are enclosing the image point, namely image cell. If image point is located in dead cells or outside of the computational domain, find the nearest IB/live cell as its image cell.

Step 2: search and select the neighbouring live and IB cells which share face with this image cell. These neighbouring cells are named as the first-row cells.

Step 3: search and select the neighbouring live and IB cells of the first-row cells. These neighbouring cells are named as the second-row cells, which should exclude those already in the first-row cell list.

Step 4: repeat Step 2 until the total number of the neighbouring cells for each IB cell is larger than a specified value, which may depend on the interpolation scheme.

Step 5: sort all the neighbouring cells based on the distance to image point.
The interpolation stencil searching algorithm as well as the interpolation scheme may be very expensive if the stencil is too large. It is more efficient and cheaper to use image point, instead of IB cell center, as the center of the interpolation stencil. In parallel computation, if cells in the interpolation stencil are located in other processors, it will incur additional communication cost.

**Quadratic interpolation scheme for reconstruction**

As suggested in Mittal and Iaccarino (2005), higher-order interpolation needs to be used for high Reynolds number problems. In this work, a second-order (quadratic) polynomial interpolation scheme is implemented as follows (Seo and Mittal, 2011; Jasak et al., 2014; Jasak and Tuković, 2015; Singh et al., 2015):

\[
\tilde{\phi}(\tilde{x}, \tilde{y}, \tilde{z}) = \sum_{i=0}^{2} \sum_{j=0}^{2} \sum_{k=0}^{2} C_{ijk} \tilde{x}^i \tilde{y}^j \tilde{z}^k, \quad i + j + k \leq 2
\]

(3.18)

where \(\tilde{x} = x - x_{\text{HP}}\), \(\tilde{y} = y - y_{\text{HP}}\), and \(\tilde{z} = z - z_{\text{HP}}\) are local (relative) coordinates of a stencil cell center \((x, y, z)\). Subscript HP denotes hit point. \(\tilde{\phi}(\tilde{x}, \tilde{y}, \tilde{z})\) is a polynomial function to approximate \(\phi(x, y, z)\), a generic flow variable at \((x, y, z)\). Considering the restriction on \(i, j,\) and \(k\), this polynomial has 10 unknown coefficients \(C_{ijk}\) in total. For a given IB cell, it is assumed that there are \(M\) cells in its interpolation stencil. For this selected stencil, Eq. (3.18) can be written in the form of matrix-vector product

\[
\phi = Ac
\]

(3.19)
where \( \phi \) denotes a vector of size \( M \). \( c \) denotes a vector consisting of the 10 unknown coefficients \( C_{ijk} \). \( A \) denotes a \( M \times 10 \) matrix, which can be computed by the local coordinates of each stencil cell center. A weighted least squares approach is adopted to determine the unknown coefficients (Li, 1998; Seo and Mittal, 2011). For each cell in the interpolation stencil, the weight function \( w \) can be written as

\[
w = \frac{1}{2} \left[ 1 + \cos \left( \frac{\delta}{1.1\delta_{\text{max}}} \pi \right) \right]
\] (3.20)

where \( \delta \) is the distance between a stencil cell center and the corresponding hit point. \( \delta_{\text{max}} \) is the maximum value of \( \delta \). Applying the weight function to both sides of Eq. (3.19) and solving for \( c \), one can get

\[
W\phi = WA\,c
\] (3.21)

\[
c = (WA)^{-1} W\phi
\] (3.22)

where \( W \) is a \( M \times M \) matrix with \( w \) in Eq. (3.20) as its elements. This weighted least squares technique has been proven to have the advantages of robustness, smooth error distribution and locally-supported second-order accuracy (Li, 1998).

Dirichlet and Neumann boundary conditions are typical for flow variables on immersed boundary. A generic quadratic function for Dirichlet boundary condition
can be derived from Eq. (3.18) as follows

\[ \tilde{\phi}(\tilde{x}, \tilde{y}, \tilde{z}) = \phi_{HP} + \sum_{i=0}^{2} \sum_{j=0}^{2} \sum_{k=0}^{2} C_{ijk} \tilde{x}^i \tilde{y}^j \tilde{z}^k, \quad 0 < i + j + k \leq 2 \]  

(3.23)

where the number of unknown coefficients \( C_{ijk} \) is reduced to 9. \( \mathbf{A} \) then becomes a \( M \times 9 \) matrix, and the vector \( \mathbf{c} \) has a length of 9:

\[
\mathbf{A} = \begin{bmatrix}
x_1 & y_1 & z_1 & x_1y_1 & x_1z_1 & y_1z_1 & x_1^2 & y_1^2 & z_1^2 \\
x_2 & y_2 & z_2 & x_2y_2 & x_2z_2 & y_2z_2 & x_2^2 & y_2^2 & z_2^2 \\
\vdots & & & & & & & & \\
x_M & y_M & z_M & x_My_M & x_Mz_M & y_Mz_M & x_M^2 & y_M^2 & z_M^2
\end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix}
c_{100} \\
c_{010} \\
\vdots \\
c_{002}
\end{bmatrix}
\]  

(3.24)

Similarly, for the Neumann boundary condition with zero gradient on the immersed boundary, the mathematical formula can be written as

\[ \nabla \phi_{HP} \cdot \mathbf{n}_{HP} = 0 \]  

(3.25)

where \( \mathbf{n}_{HP} \) is the normal direction to the immersed boundary \( \Gamma_{IB} \) at the hit point.
The number of unknown coefficients is 10 and the matrix system becomes:

\[
A = \begin{bmatrix}
1 & x_1 & y_1 & z_1 & x_1 y_1 & x_1 z_1 & y_1 z_1 & x_1^2 & y_1^2 & z_1^2 \\
1 & x_2 & y_2 & z_2 & x_2 y_2 & x_2 z_2 & y_2 z_2 & x_2^2 & y_2^2 & z_2^2 \\
& & & & \vdots & & & & & \\
1 & x_M & y_M & z_M & x_M y_M & x_M z_M & y_M z_M & x_M^2 & y_M^2 & z_M^2 \\
0 & n_{HP,x} & n_{HP,y} & n_{HP,z} & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
c = \begin{bmatrix}
c_{000} \\
c_{100} \\
c_{010} \\
& \vdots & \\
c_{002}
\end{bmatrix}
\]

(3.26)

where \(n_{HP,x}, n_{HP,y}, n_{HP,z}\) denote the \(x, y, z\) components of wall normal vector \(n_{HP}\). The matrix \(A\) becomes a \((M+1) \times (M+1)\) matrix. The extra row in Eq. (3.26) is obtained by incorporating boundary condition in Eq. (3.25) into Eq. (3.19). Once the matrix systems are constructed, the coefficients \(C_{ijk}\) for Dirichlet and Neumann boundary conditions can be obtained by solving Eqs. (3.24) and (3.26).

**Alternative linear interpolation scheme for reconstruction**

Higher-order interpolation such as the quadratic interpolation has requirement on the minimum number of cells in the interpolation stencil. However, this requirement may not always be satisfied, for example in the case of small gap or sharp wall corner. Thus, an alternative to the high-order interpolation is need for these cases. In this work, the linear interpolation scheme is used as an alternative. Instead of reducing the order in Eq. (3.18), the interpolation method proposed in Gilmanov et al. (2003) is used where the flow variable distribution is assumed to be linear.
between an IB cell center and its image point.

For either Dirichlet or Neumann boundary condition, the value at image point is calculated based on inverse-distance weighted interpolation over the stencil:

\[
\tilde{\phi}_{IP} = \frac{\sum_{i=1}^{M} \phi_i \delta_i^{-1}}{\sum_{i=1}^{M} \delta_i^{-1}}
\]  

(3.27)

where subscript IP denotes image point. \( \phi_i \) is the value of the \( i \)-th neighboring cell in the stencil. \( \delta_i \) is the distance between the \( i \)-th neighboring cell in the stencil and the image point.

For the Dirichlet boundary condition, the value at a hit point (\( \phi_{HP} \)) is known and prescribed. Thus, a simple linear relationship can be used to calculate the value at IB cell center

\[
\tilde{\phi}_{IB} = \phi_{HP} + \frac{\tilde{\phi}_{IP} - \phi_{HP}}{\delta_{IP}} \delta_{IB}
\]  

(3.28)

where \( \delta_{IP} \) and \( \delta_{IB} \) are the distance from hit point to image point and IB cell center, respectively.

For the Neumann boundary condition, the gradient at a hit point (\( \Delta \phi_{HP} \)) is known and prescribed. Since mesh is typically refined near immersed boundary, it is reasonable to assume that gradient does not change between hit point and image point. Thus, one can approximate the value at each IB cell center as

\[
\tilde{\phi}_{IB} = \tilde{\phi}_{IP} - \Delta \phi_{HP} (\delta_{IP} - \delta_{IB}).
\]  

(3.29)
Interpolation scheme selection and iterative method

The selection of interpolation scheme depends on the required accuracy, mesh resolution, and number cells in the interpolation stencil for each IB cell. To automate the selection of interpolation scheme, a decision algorithm is proposed. A user-specified threshold is set for the required number of cells \( M \) in the interpolation stencil. For example, for 3D problems, the threshold is set as 20 in this work. For each IB cell, if \( M \) is larger than 20, the quadratic interpolation is used. Otherwise, the linear interpolation is chosen. The mixed use of two interpolation schemes not only solves the problem of insufficient interpolation cells around small gaps and sharp corners, but also can help balance computational cost and accuracy. A user can tune the threshold value to achieve desired accuracy with reasonable computational cost.

Interpolation cells in the stencil may contain other IB cells which theoretically requires implicit solution of coefficients for all IB cells at once. However, such strategy requires the assembly of a much larger linear system from the sub linear system of each IB cell (Eqs. (3.24) and (3.26)). Instead, this work uses an explicit method where Eqs. (3.24) and (3.26) are solved iteratively. For all test cases presented in this paper, this iterative method is very efficient and only up to 5 iterations are needed to converge with a relative error criterion of \( 10^{-4} \).
3.3.3 Coefficient matrix manipulation

Another important immersed boundary treatment is to manipulate the linear system equations to enforce the boundary conditions on IB cells during computation, such as the SIMPLE algorithm and turbulence model in Eqs. (3.12), (3.15), (3.17) and (3.16). These equations can be in a general form

\[ A\phi = b \]  \hspace{1cm} (3.30)

where \( A \) denotes the coefficient matrix, \( b \) denotes the source term, and \( \phi \) denotes the unknown variable. The off-diagonal part in \( A \) represents the relationship between current cell and neighboring cells, and the diagonal part is resulted from the current cell itself.

For the Dirichlet boundary condition on a body-fitted mesh (BFM), at a matrix level, its effect is added to the source term \( b \). Generally, when using the Gauss’s theorem in Eq. (3.8), if face \( f \) is a boundary face, \( \phi_f \) uses the prescribed value instead of an interpolated value. For example, in the step of momentum predictor in Eq. (3.10), the boundary face flux is independent of either \( u_P \) or \( u_N \), which means the boundary face flux \( \Phi_f \) will not contribute to either \( a_P \) or \( a_N \) in Eq. (3.11). The effect of the Dirichlet condition on this face is an additional source term in the form of boundary face flux in the right-hand-side of Eq. (3.11), i.e., the source term \( b_u \) in Eq. (3.12).
However, in IB method, it is impossible to impose the exact Dirichlet condition on a boundary face as with a body-fitted mesh. In the present work, instead of using boundary faces, the values of the IB cells are “fixed” to approximate the boundary effect. Again, taking the momentum predictor in Eq. (3.10) as an example, velocity $\mathbf{u}$ is the unknown to be solved. The values of $\mathbf{u}$ at IB cell centers are determined through reconstruction outlined in previous section. For any IB cell, its original relationship with neighboring cells can be described with Eq. (3.11). To fix $u_p$, all the off-diagonal elements $a_N$ are set to zero and the diagonal term $a_P$ is changed based on

$$a_P = -\frac{1}{\rho} \sum_f S_f p_f^{n-1} / \tilde{u}_P^{n-1}$$  (3.31)

where $\tilde{u}_P^{n-1}$ denotes the reconstructed velocity from previous time step or iteration. In addition, the values in dead cells are also fixed. The exact values for dead cells do not matter since they have no influence on the solution in the fluid region.

For the Neumann boundary condition, the gradient at a boundary needs to be fixed. In body-fitted mesh, the flow variable on a boundary face, $\phi_f$, is extrapolated based on cell center value and the prescribed gradient. For example, in pressure Poisson equation (Eq. (3.13)), $(\nabla p)_f$ is fixed with a prescribed gradient. Similar to the Dirichlet boundary condition, this term also contributes to the source term on the right hand side of Eq. (3.15).

However, in IB method, the Neumann boundary condition is more complicated than the Dirichlet boundary condition. In Eq. (3.14), for any live cells, if one of
the neighboring cells \((N)\) is an IB cell, the corresponding \(p_{N, IB}\) can be written as a function of \(p_P\)

\[
p_{N, IB} = p_P + \delta_{N,P} \left( \nabla p_{f, IB} \cdot \frac{S_{f, IB}}{|S_{f, IB}|} \right)
\]  

(3.32)

where \(\delta_{N,P}\) denotes the distance between the live cell and its adjacent IB cell. \(S_{f, IB}\) denotes the area vector on the IB face. \(\nabla p_{f, IB}\) denotes the pressure gradient on the corresponding IB face, which is interpolated from \(\nabla p_P\) and \(\nabla p_{N, IB}\) from previous time step or iteration. Since during IB reconstruction, the Neumann boundary condition at an immersed interface has already been incorporated to get \(\nabla p_{N, IB}\), the prescribed \(\nabla p_{f, IB}\) then implicitly includes the Neumann boundary condition.

At a wall, the pressure \(p\) has zero normal gradient. Thus, substituting Eq. (3.32) into Eq. (3.14), one can get

\[
\left( a_P' + \sum_{N, IB} a'_{N, IB} \right) p^n_P + \sum_{N,w/o IB} a'_{N} p^n_N = rhs - \sum_{N,IB} \delta_{N,P} \left( \nabla p_{f, IB} \cdot \frac{S_{f, IB}}{|S_{f, IB}|} \right) \]  

(3.33)

For the \(k-\epsilon\) turbulence model equations, the Dirichlet and Neumann boundary conditions are both used in the solution of Eqs. (3.5) and (3.6), where the value of \(k\) and \(\epsilon\) on the immersed boundary, \(k_{IB}\) and \(\epsilon_{IB}\), are calculated with the new IB wall function to be introduced in Section 3.4.
3.4 \(y^+\)-adaptation immersed wall model

3.4.1 Basic wall function for immersed boundary method

The IB wall function in this work is based on the two-layer wall model proposed in Roman et al. (2009b). One of the key parameters for wall-bounded turbulent flow is the shear velocity \(u_\tau\). Here, the shear velocity is calculated with the flow information at image point, which is more representative than the constructed values in IB cell. Usually, it is assumed that local equilibrium exists in the near wall region flow, meaning \(P_k \approx \epsilon\) at image point. Thus, for a given IB cell and its corresponding image point, \(u_\tau\) can be estimated as \(C_1^{1/4}\sqrt{k_{IP}}\), and the dimensionless wall distance \(y^+ = u_\tau y/\nu\) is calculated as

\[
y^+_{IP} = \frac{C_1^{1/4}\sqrt{k_{IP}}}{\nu}, \quad y^+_{IB} = y^+_{IP} y_{IB}/y_{IP}
\]  

(3.34)

where \(y_{IP}\) denotes the distance from image point to immersed boundary, \(y_{IB}\) denotes the distance from IB cell center to immersed boundary. The ratio of \(y_{IP}/y_{IB}\) is set to be 2 in this work.

According to the two-layer wall model, the shear velocity can be recalculated as

\[
u_\tau = \begin{cases} 
C_1^{1/4}\sqrt{k_{IP}} & \text{if } y^+_{IP} > y^+_{Laminar} \\
\sqrt{\nu|u_{tan,IP}^{old}|y_{IP}} & \text{if } y^+_{IP} \leq y^+_{Laminar}
\end{cases}
\]  

(3.35)
where \( u_{\text{tan,IP}}^{\text{old}} \) denotes the interpolated tangential velocity at image point from previous time step or iteration, and \(||\) denotes its magnitude. The threshold \( y_{Laminar}^+ \) has a value of 11. With the shear velocity, the wall shear stress can be calculated as

\[
\tau_w = \rho u_\tau^2
\]  
(3.36)

For the two points (image point and IB cell center) along the same line perpendicular to the immersed boundary, their velocity should follow the same log-law, i.e., with the same shear velocity. Therefore, the tangential velocity at IB cell center can be calculated as

\[
u_{\text{tan,IB}}^{\text{new}} = \begin{cases} 
\frac{u_\tau \kappa}{\log (E y_{IB}^+)} & \text{if } y_{IB}^+ > y_{Laminar}^+ \\
u_\tau y_{IB}^+ & \text{if } y_{IB}^+ \leq y_{Laminar}^+ 
\end{cases}
\]  
(3.37)

where \( E \) is an empirical coefficient with a value of 9.8 for smooth walls. Following this, the new values of eddy viscosity \( \nu_t \), \( k \), and \( \epsilon \) at IB cell center can be calculated as

\[
u_{T} = \begin{cases} 
\frac{y_{IB}^+ \kappa}{\log (E y_{IB}^+)} \nu_\tau & \text{if } y_{IB}^+ > y_{Laminar}^+ \\
0 & \text{if } y_{IB}^+ \leq y_{Laminar}^+ 
\end{cases}
\]  
(3.38)

\[
k_{IB}^{\text{new}} = \begin{cases} 
(\nu_{T} + \nu) \frac{u_{\text{tan,IP}}^{\text{old}}}{y_{IP}} C_{\mu}^{-0.5} & \text{if } y_{IB}^+ > y_{Laminar}^+ \\
k_{IP} & \text{if } y_{IB}^+ \leq y_{Laminar}^+ 
\end{cases}
\]  
(3.39)
\( \epsilon_{\text{new IB}} = \begin{cases} 
C_{0.75}^{0.75}(k_{\text{IB}}^{\text{new}})^{1.5} & \text{if } y_{\text{IB}}^+ > y_{\text{Laminar}}^+ \\
\kappa y_{\text{IB}} & \text{if } y_{\text{IB}}^+ \leq y_{\text{Laminar}}^+ 
\end{cases} \)  

(3.40)

3.4.2 \( y^+ \)-adaptation wall model for immersed boundary method

The basic wall function described in the previous section often produces non-smooth wall shear stress distribution. In many applications such as sediment transport and erosion where the physical process depends on the wall shear, the smoothness of wall shear is of critical importance. In this section, the original wall function and its ill behavior when used with immersed boundary method are firstly analyzed. Then, a new \( y^+ \)-adaptation method is proposed. For easy argument, Fig. 3.3a shows a schematic view of 1D boundary layer flow near an immersed boundary. It defines wall distances, \( y_{\text{IB}} \) and \( y_{\text{IP}} \), which are the wall distance from IB cell center and image point, respectively. The location of immersed boundary \( \Gamma_{\text{IB}} \) can be varied to change the wall distances \( y_{\text{IB}} \) and \( y_{\text{IP}} \).

For many wall functions used with RANS models, it is often required that the near-wall cell center is in the log-law layer, i.e., the non-dimensional wall distance \( y^+ \) is larger than 30. However, this requirement may not always be satisfied in IB method. The reason is that the background grid is fixed and the immersed boundary geometry is arbitrary. As a result, there is no direct control on the wall distances \( y_{\text{IB}} \) and \( y_{\text{IP}} \). In addition, the immersed boundary may move and consequently wall distances may dynamically change. It is very likely very small
wall distances will result and thus violate the $y^+$ requirement for wall functions. When the wall distances are small, i.e., the IB cell and IP point are within the viscous or buffer layer, the resulted wall shear is very different. The cause of this difference is that the wall function stipulates a nonlinear and discontinuous behavior depending on whether $y_{IB}^+$ is larger or smaller than $y_{Laminar}^+$.

To prove this point, a simple 1D turbulent channel flow with $Re = UH/\nu = 5 \times 10^5$ is simulated using the setup in Fig. 3.3a where the immersed boundary can be shifted vertically to vary the wall distances. Here, the mean velocity $U = 0.5$ m/s, the channel depth $H = 1$ m, and the kinematic viscosity $\nu = 10^{-6}$ m$^2$/s. The boundary is smooth. Assuming fully developed open channel flow, the calculated theoretical shear velocity is 1.84 cm/s (see B for details). The simulated shear velocity and corresponding IB cell wall distance are plotted in Fig. 3.3b. It is observed that when the IB cell center is in the log-law layer, i.e., $y_{IB}^+ > 30$, the simulated shear velocity approaches the theoretical value. However, when the IB cell is located closer to the immersed boundary and not in the log-law layer, the simulated shear velocity has significant error and the error increases when the wall distance decreases.
Figure 3.3: Effect of IB cell wall distance on the simulated shear velocity. \( y_{IB}^+ \) and \( y_{IP}^+ \) denote the dimensionless wall distance of IB cell center and image point, respectively. The immersed boundary \( \Gamma_{IB} \) can be moved vertically in the yellow region to change the IB cell wall distance.

The root of the ill behavior of wall shear is the lack of control on wall distance in immersed boundary method. To overcome this problem, a \( y^+ \)-adaptation algorithm is developed. The basic idea is to adjust the IB cells if their wall distance is too small. Specifically, if \( y_{IP}^+ < 30 \) or \( y_{IB}^+ < 11 \), the corresponding IB cell will be set as dead cell and a new IB cell will be searched for among its neighbouring live cells in the direction away from the immersed boundary (see Fig. 3.4a). At every time
step, this adaptation of IB cell will continue until all IB cells are located in the log-law layer. Fig. 3.4b shows the same 1D channel flow case with the adaptation treatment. It is found that regardless the initial IB cell wall distance, the algorithm automatically adjusts and makes sure that the IB cell is located in the log-law layer ($y_{IB}^+ > 11$ and $y_{IP}^+ > 30$). In addition, the resulted shear velocity is almost constant ($\approx 1.8$ cm/s) for any given initial IB cell wall distance.

It is worth to note that the adaptation ensures well-behaved wall shear stress with the cost of slight reduction in how accurately the immersed boundary is represented by the labeling of IB, fluid, and dead cells. However, this reduction in accuracy is not significant if mesh is sufficiently refined near the immersed boundary.

![1D schematic of the $y^+$ adaptation process](image)

(a) 1D schematic of the $y^+$ adaptation process

![Graph showing shear velocity vs $y^+$](image)

(b) after $y^+$ adaptation

Figure 3.4: Effect of IB $y^+$ adaptation on calculated shear velocity.
3.4.3 Rough wall function for immersed boundary method

Many applications in natural and built environments involve the flow and transport over rough boundaries, for example the river bed and ocean floor where sediment transport occurs. To cope with that, the present wall function for immersed boundary method is also designed to include the roughness effect. The rough wall function changes the value of coefficient \( E \) as a function of a non-dimensional roughness height \( k_s^+ \). The functional form of the empirical formula is

\[
E = \begin{cases} 
0.9 \left( \frac{k_s^+ - 2.25}{87.75} + 0.5k_s^+ \right)^{-\sin[0.4258(\log k_s^+)-0.811]} & \text{if } k_s^+ > 90 \\
0.9 (1 + 0.5k_s^+)^{-1} & \text{if } k_s^+ \leq 90
\end{cases}
\]  

(3.41)

where \( k_s^+ = u_s k_s / \nu \). A simple 1D channel flow case is used to test the proposed rough IB wall function. The mean velocity \( U \) is 0.5 m/s, channel depth \( H \) is 1 m, and the Reynolds number \( Re = UH / \nu = 5 \times 10^5 \). Figure 3.5 shows the velocity distributions in the channel with different roughness and the comparison with theoretical distribution. Good agreement can be observed.
The new immersed boundary method has been implemented in the *pimpleFoam* solver in OpenFOAM v5.x (The OpenFOAM Foundation, 2018). In general, for unsteady problems, temporal discretization uses the first-order implicit Euler scheme, or more accurate second-order schemes such as “Backward” in OpenFOAM. Spatial discretization uses second-order linear schemes for interpolation.

### 3.5 Test cases

#### 3.5.1 Pitz-Daily 2D case

The 2D Pitz-Daily case is a classic backward facing step case commonly used to evaluate the performance of turbulence models (Pitz and Daily, 1981). The flow is assumed as incompressible. The averaging inlet velocity is \( U = 13.3 \) m/s, step height \( H = 0.0254 \) m, kinematic viscosity is \( \nu = 1.5 \times 10^{-5} \) m/s\(^2\), where Reynolds number is \( Re = U H / \nu = 2.2 \times 10^4 \). It is used here as a validation case to compare with results from experiments and simulation with a body-fitted mesh. The case
was also simulated with the IB method reported in Jasak et al. (2014) and Jasak and Tuković (2015). There are two major differences between the present IB method and theirs. One is that there is no $y^+$-adaptation process in their method. The second is that in their $k - \epsilon$ turbulence model, the gradient of both $k$ and $\epsilon$ are fixed as zero on the IB faces.

The mesh used with immersed boundary methods has a total of 15000 cells and the body-fitted mesh has 12225 cells. The resolutions of both meshes are comparable and they are plotted in Fig. 3.6. The mesh for the immersed boundary methods uses a uniform background mesh with no local refinement. The red cells are the IB cells for the upper and lower walls at the end of simulation, which has already been modified by the $y^+$-adaptation algorithm. The contraction of the outlet leads to the stair-like arrangement of IB cells, which is very common in the use of immersed boundary method. For the body-fitted mesh, there is some refinement near the walls and in the zone of shear layer downstream of the step.

Figure 3.6: The mesh for immersed boundary methods (upper) and the body-fitted mesh (lower). The red cells in upper mesh are the IB cells.
At the inlet, the boundary conditions for $u$, $k$ and $\epsilon$ are prescribed using experimental data from (Pitz and Daily, 1981). At the outlet, pressure is fixed as zero and the boundary conditions for all other variables are zero gradient. Wall functions are applied to the upper and lower walls.

Figure 3.7 shows the comparison between the experiment and simulations. Overall, the present IB method gives satisfactory results. Specifically, as seen in Fig. 3.7a for streamwise velocity, though the present method is slightly off in comparison with experimental data, it gives almost the same result as the simulation with a body-fitted mesh. The IB method in Jasak et al. (2014) gives less ideal result where the re-circulation in the wake zone is weaker than the body-fitted mesh and present IB methods. This might be induced by the zero gradient condition applied to IB faces in Jasak et al. (2014). In Fig. 3.7b and Fig. 3.7c for $k$ and $\epsilon$, the IB method in Jasak et al. (2014) overestimated their value in the wake, especially in the shear layer. The contour, which is not presented here, shows that there are some cells with big values of $k$ at the corner of the step where shear layer starts to form. These big values are not observed in the results with a body-fitted mesh or the present IB method. This might again be caused by the zero gradient condition for $k$ applied on IB faces in Jasak et al. (2014). In Fig. 3.7d for the eddy viscosity, the present IB model agrees with body-fitted mesh simulation while the IB method in Jasak et al. (2014) overestimated the level of eddy viscosity.

It is also noted that in Fig. 3.7c for $\epsilon$, in some near wall region, $\epsilon$ appears
to be smaller in IB method than body-fitted mesh. This is caused by different calculation locations of boundary value in these two methods. In the wall function with body-fitted mesh, $\epsilon$ is evaluated at boundary cell center. However, in IB method, $\epsilon$ is evaluated at IB cell center. According to Eq. (3.40), the calculated $\epsilon$ at a boundary is very sensitive to the wall distance $y$ (smaller $y$ leads to larger numerical value of $\epsilon$). Due to the $y^+$-adaptation scheme, the IB cell centers are in general far away from the solid wall, thus the IB cells’ $\epsilon$ values are smaller. Despite the visual difference in the $\epsilon$ distribution, the bulk flow behavior from the IB method is comparable with the body-fitted mesh method.
Figure 3.7: Comparison of results among the present IB method, the IB method in (Jasak et al., 2014), and simulation with a body-fitted mesh, and experiments in Pitz and Daily (1981). The flow is from left to right.

The two oblique wall boundaries near the outlet are the perfect location to show
the capability of the proposed $y^+$-adaptation scheme. For comparison, another case without the $y^+$-adaptation was also simulated. The resulted distributions of wall shear stress are shown in Fig. 3.8. The wall shear stress distribution is plotted using line or scatters in which their distance to the wall is proportional to the magnitude of wall shear. The sign of the wall shear (positive in the $x$-direction) is represented by whether the line or scatters are located inside (positive) or outside (negative) of the domain. It is noted that the magnitude of wall shear from the IB method in Jasak et al. (2014) is divided by $10^3$ so the spikes can be plotted within the figure.

The insert in Fig. 3.8 is a close-up view of the shear stress along the lower wall. It can be observed that without $y^+$-adaptation, the wall shear distribution is not as smooth and has a semi-periodic abnormality. The abnormality is located at IB cells whose wall distances are small. The periodicity is due to the fact that the background mesh is uniform and the lower wall is an oblique line. As a result, the IB cell distribution without $y^+$-adaptation along the lower wall is stair-like (see IB cells bounded by yellow lines in Fig. 3.9). On the hand hand, the $y^+$-adaptation method significantly reduces the abnormality and thus increases the smoothness in wall shear.
3.5.2 Flow over a 3D dune

To further show the performance of the new IB method, this test case simulates the turbulent open channel flow over 3D dunes and compares with the experimental study in Maddux et al. (2003b,a). In their experiment, there are 14 dunes in a flume, 2 of which are used in this work. The shape of the 2 dunes is plotted in Fig. 3.10. The mean velocity is 0.261 m/s. The flume width and mean water depth are 0.9 m and 0.561 m, respectively. The maximum height of the dunes is 0.06 m.
The Reynolds number is around $1.5 \times 10^5$. More details about the experiment can be found in Maddux et al. (2003b).

For comparison, simulation with a body-fitted mesh was also performed. The body-fitted mesh and the background mesh for the IB method are plotted in Fig. 3.10. The body-fitted mesh was generated with the *snappyHexMesh* tool in OpenFOAM, with an average cell size of 0.02 m and a total of 98,000 cells. The mesh was refined near the bottom. The flow was from left to right and a periodic boundary condition was used at inlet and outlet since we only simulated 2 dunes in the middle. To achieve the same mean velocity as in the experiment, a pressure gradient was dynamically adjusted and applied to the whole computational domain. According to the experiment, the free surface change was insignificant ($\sim 10^{-3}$ m). Thus, the free surface was replaced with a shear-free rigid lid in the simulations. The mesh for the IB method has a uniform cell size of 0.02 m and the total number of cells is 111,6000 ($=80 \times 45 \times 31$). The IB method simulation used the same boundary conditions as the body-fitted mesh and the flow was also driven by a dynamically adjusted pressure gradient. However, the pressure gradient was only applied to IB cells and live cells.
Figure 3.10: The bathymetry of the 3D dunes and the two meshes. One mesh (left) is the background for IB method and the other (right) is the body-fitted mesh for comparison. The flow comes from left to right.

The comparison for the velocity profiles among the experiment, and the two simulations with the present IB method (with and without $y^+$-adaptation), and the simulation with body-fitted mesh is shown in Fig. 3.11. Since the dune is 3D, there is a variation of flow field in the span-wise direction. Therefore, the velocity profiles are compared at two slices along the flume at different span-wise locations (the center at $y = 0$ m and the off-center at $y = 0.225$ m). Qualitatively, the comparison is good and both IB method and body-fitted mesh simulations capture the variations of the flow field over the dune. The present IB method even has a better prediction for the flow field away from the bed. Near the dune surface, the prediction from the body-fitted mesh simulation is slightly better. Quantitatively, the root-mean-square error (RMSE) of results from both simulations was calculated. On both slices, the RMSE of the body-fitted mesh simulation is around 0.02 - 0.04 m/s, which is comparable with the RANS simulation results from Chen et al.
(2015) with about 2 million cells. For comparison, the RMSE of the IB method simulation is about 0.03 - 0.05 m/s, which are mainly contributed by the shear layer and re-circulation zone behind the dune. Although the RMSE of the IB method simulation is slightly higher than the body-fitted mesh simulation, the difference is small. When applying the IB method, one should keep in mind that it is an approximation of the wall effect after all.

Figure 3.11: Comparison of the velocity profiles among experiments (Maddux et al., 2003a) and two numerical simulations using the present IB method and a body-fitted mesh. Here, IBM denotes the present IB method (with $y^+$ adaptation scheme) and BFM denotes the body-fitted mesh. The profiles are sampled on two vertical slices at two different span-wise locations. RMSE denotes the root-mean-square error. The vertical dash line denotes the measurement location.

Maddux et al. (2003a) provides two different sets of bed shear stress (skin friction), both of which will be compared against the present IB method. One set is derived from near-bed mean velocities, which is calculated based on a rough
boundary condition as

$$\tau_{sf1} = \frac{\kappa U_1}{\ln (30y_1/k_s)} \frac{\kappa U_1}{\ln (30y_1/k_s)}$$  \hspace{1cm} (3.42)$$

where $U_1$ and $y_1$ denotes streamwise velocity and wall distance at measurement points, respectively. Correspondingly, in the present IB method, we replaced them with $u_{\text{tan,IP}}$ and $y_{\text{IP}}$ for the calculation of wall shear. Here, $u_{\text{tan,IP}}$ is the tangential velocity at image point. The roughness height $k_s = 1$ mm. The other set is calculated from the near-bed Reynolds shear stress

$$\tau_{sf2} = -\bar{u}'w'$$  \hspace{1cm} (3.43)$$

In the present IB method, $-\bar{u}'w'$ is estimated by $u_r^2$, which is calculated from Eq. (3.35).

In the results of Maddux et al. (2003a), the magnitudes of both $\tau_{sf1}$ and $\tau_{sf2}$ were normalized by total bed shear stress $\tau_T = 0.46 \times 10^{-3} m^2/s^2$, which was derived from spatially-averaged momentum balance. In order to compare with Maddux et al. (2003a), we used the same normalization and the distributions of simulated wall shear $\tau_{sf1}$ and $\tau_{sf2}$ over the 3D dune are plotted in Fig. 3.12 and Fig. 3.13. The simulated results with and without $y^+$-adaptation are both shown.

Figure 3.12a shows the experimental result of $\tau_{sf1}$ over the 3D dunes. The dashed contour lines represent the elevation of bed which consists of a pair of dunes.
The crest of the right dune is located in the middle and the left dune has two crests located on the two sides in the $y$ direction. From experiment, $\tau_{sf1}$ is very small in the trough and the lee sides of the two dunes. The wall shear is large on the stoss side of the dunes. Fig. 3.12b and Fig. 3.12c show the simulated wall shear using the present IB method with and without $y^+$-adaptation, respectively. Both simulated results have similar distribution as the experiment. It can be observed that $\tau_{sf1}$ is much smoother across the contour lines on the stoss side of the dunes if $y^+$-adaptation is used. In comparison with experiment, both simulation results show larger value of $\tau_{sf1}$ in the lower part and the lee side of the dunes and smaller $\tau_{sf1}$ in the stoss side of the dunes (about 20% smaller). Some features on the crests were not captured accurately. This might be caused by the use of a single roughness height $k_s$ for the whole dune.

Figure 3.13 shows the experimental and numerical results of $\tau_{sf2}$ over the dunes, which are different from $\tau_{sf1}$. Maddux et al. (2003a) explained that it might be caused by the large sampling volume of ADV and the vortex shedding over the 3D dunes. On the other hand, $\tau_{sf1}$ is derived from time-averaged mean flow velocity while $\tau_{sf2}$ is derived from flow turbulence. From this point of view, the distribution difference between $\tau_{sf1}$ and $\tau_{sf2}$ is reasonable because mean velocity is high on stoss side and low on lee side as shown in Fig. 3.11, while turbulence is low on stoss side and high on lee side due to vortex shedding. Due to this uncertainty, the comparison of $\tau_{sf2}$ with experiment is in general acceptable but less ideal than $\tau_{sf1}$.
The simulated $\tau_{sf2}$ distribution pattern matches with experiment. In addition, the use of $y^+$-adaptation made the wall shear distribution smoother.

Figure 3.14 further shows the effect of $y^+$-adaptation on the distributions of $\tau_{sf1}$ and $\tau_{sf2}$ along the solid blue lines in Fig. 3.12 and Fig. 3.13. It can been observed that with $y^+$-adaptation the wall shear is smoother, especially for $\tau_{sf2}$, which is calculated with TKE.
Figure 3.12: Comparison of $\tau_{sf_1}$ distribution among experiments (Maddux et al., 2003a) and two numerical simulations using the present IB method (with and without $y^+$ adaptation scheme). $\tau_{sf_1}$ is derived from near bed velocity. Dashed curved lines denote the elevation contour.
Figure 3.13: Comparison of $\tau_{sf2}$ distribution among experiments (Maddux et al., 2003a) and two numerical simulations using the present IB method (with and without $y^+$ adaptation scheme). $\tau_{sf2}$ is derived from Reynolds stress. Dashed curved lines denote the elevation contour.
Figure 3.14: Effect of $y^+$ adaption scheme on $\tau_{sf1}$ and $\tau_{sf2}$ distribution over vertical solid line shown in Fig. 3.12 and Fig. 3.13.

### 3.6 Summary and conclusions

Based on the analysis of existing immersed boundary methods and the associated wall functions for turbulent flows, this paper introduces a new immersed boundary method and a $y^+$-adaptation strategy for turbulent flow simulations with RANS models. One distinctive feature of the new model is that the resulted wall shear stress is smoother which is important for processes on the immersed boundary driven by wall shear, for example heat transfer and erosion. The model is developed
with unstructured mesh and equipped with efficient geometric operation algorithms. As a result, it can be used to simulate flow and transport around surfaces with complex shapes.

In the present IB method, the effect of the immersed boundary is enforced through the IB cells on which the flow variables are reconstructed from their neighboring cells with an explicit and iterative interpolation scheme. Three choices of interpolation schemes are implemented, i.e., quadratic, linear, and mixed. The boundary conditions (Dirichlet or Neumann) are enforced through linear system matrix manipulation once the governing equations have been discretized with a finite-volume method.

Both smooth and rough wall functions on the immersed boundary are developed for the standard $k-\epsilon$ turbulence model. The original wall function lacks smoothness due to the nonlinearity and discontinuity between the log-law layer and laminar layer. However, in immersed boundary method, it is hard to control where the IB cell centers will be located in a boundary layer. When an IB cell’s wall distance is too small, i.e., located in the laminar layer, the resulted wall shear stress is not correct and thus the overall wall shear distribution on the immersed boundary is not smooth. To remedy this, a $y^+$-adaptation algorithm is developed to automatically adjust IB cell designation such that all IB cells are in the log-law layer. As demonstrated in all test cases, the $y^+$-adaptation algorithm is robust and produces reasonably accurate and smooth wall shear stress.
References


Chapter 4  
Development and applications of a robust 3D local scour model based on immersed boundary method

4.1 Introduction

Scour and erosion are common in both natural and built environments. They are one major cause of bridge failure and other infrastructure damages. Wardhana and Hadipriono (2003) reported that 53% of all the bridge failures in United States were due to flood and scour. Scour and erosion can happen under several different conditions. But they all share the same feature, which is the flow moves
the sediment at different rate in different areas around a structure. Simple mass imbalance for a area can cause either erosion or sedimentation. For example, the existence of a bridge will cause local acceleration and deceleration of the flow field around its foundation. The sediment transport capacity of flow is proportional to water velocity. Thus, sediment may be moved away from areas of flow acceleration and deposited in areas of flow deceleration (Richardson and Richardson, 2008; Wang et al., 2017). In general, there are three different forms of scour, i.e., general scour, contraction scour, and local scour. General scour is caused by the sediment transport regime in a river reach. This may be due to overall increase or reduction of sediment yield from the watershed. Contraction scour occurs when the flow is accelerated by the contracting effect of in-stream structures. The placement of structures in river and streams reduces the effective flow area and thus increase the flow speed and sediment transport rate. Local scour is mainly resulted from local flow structures like horseshoe vortex around in-stream structures. To properly model local scour, it is important to resolve the local flow structures.

Due to its importance for the safety of infrastructures placed in river and stream, computational models have been used widely for the study of flow and sediment transport around these structures. In the early years when computational fluid dynamics just started to be used in hydraulics, the structures studied were very simple, for example a cylinder. Scour around a simple vertical cylinder has been numerically investigated by many researchers (Olsen and Kjellesvig, 1998; Roulund
et al., 2005; Liu and García, 2008; Zhao et al., 2010; Escauriaza and Sotiropoulos, 2011; Jacobsen and Fredsoe, 2014; Baykal et al., 2017). To tracking the bed morphological change, most of the scour models used the moving mesh technique where the mesh grid points are moved according to the scour hole development. However, the mesh moving and deformation method is limited to simple geometries, like Figure 4.1a. As the scour hole develops, the foundation may experience burial or exposure, such as shown in Figure 4.1b. The use of grid point movement to track these dynamics changes around a structure with complex shape is almost impossible.

![Figure 4.1: 2D diagram of moving mesh technique.](image)

An alternative for tracking the evolution of bed is the immersed boundary method (Escauriaza and Sotiropoulos, 2011; Khosronejad et al., 2011, 2012). In this method, instead of moving the grid points, the computational mesh is fixed. A surface, which represents a wall boundary such as a bed, is “immersed” in the
background mesh. Special treatment is performed to reflect the flow behavior near the surface, which may be fixed or moving. This method has been applied in the morphodynamics modeling of stream restoration structures (Khosronejad et al., 2014, 2015, 2018).

In-stream structures, regardless natural or man-made, may have great geometric complexity. Even for the simple case of bridges, it is very often for a bridge pier to have multiple columns, footings, and piles. In the literature, very limited work has been reported to study the hydraulics and sediment transport around complex bridge piers (Beheshti and Ataie-Ashtiani, 2016). For example, Alemi and Maia (2018) used the sediment scour model developed in Flow-3D to investigate a bridge pier with multiple columns (Wei et al., 2014). However, their study only considered the interaction of the bed with the vertical piles, which essentially simplified the case. For in-stream structures used for river restoration, e.g., large woody debris, the geometry might be much more complex. To advance our understanding of the physical transport processes and to help engineering design, a more sophisticated, robust, and yet computationally affordable model is needed.

In this work, a novel 3D local scour model is proposed based on an immersed boundary method designed for unstructured mesh. The scour model is developed in the open source CFD platform OpenFOAM (ESI-OpenCFD, 2018). In the present scour model, the in-stream structure is captured with a body-fitted mesh. On the other hand, the bed is treated as an immersed boundary, which is represented
by a 2D surface. The scour model has two parts: hydrodynamics and morphodynamics. The hydrodynamic part simulates the 3D turbulent flow field on the fixed background mesh using an immersed boundary method for the bed. The hydrodynamic part results in wall shear stress distribution on the bed surface. In the morphodynamic part, sediment transport and bed evolution are calculated on the 2D immersed boundary surface. The 2D bed surface deforms according to the scour and erosion.

The chapter is organized as follows. In Section 4.2, the hydrodynamics part, in particular the SST-SAS turbulence model and wall functions with roughness, is introduced. Section 4.3 describes the bed morphodynamics part, with discussions on critical shear stress and a new sand-slide algorithm. In Section 4.4, the coupling between immersed boundary method and the morphodynamics model is described, along with the immersed boundary wall function implementation. In Section 4.5, the scour model is validated against experimental data and an application on bridge pier scour is shown.
4.2 Hydrodynamic model

4.2.1 Governing equations and turbulence model

The incompressible Reynolds-averaged Navier-Stokes (RANS) equations have the form of

$$\frac{\partial \bar{u}}{\partial t} + \nabla \cdot (\bar{u}\bar{u}) = \frac{\nabla \bar{p}}{\rho} + \nabla \cdot (\nu + \nu_t)\bar{S}$$  (4.1)

where $\bar{u}$ denotes the mean velocity, $\bar{p}$ denotes the mean pressure, $\nu$ denotes the kinematic viscosity, and $\nu_t$ denotes the turbulent eddy viscosity. $\bar{S} = 1/2(\nabla \bar{u} + \bar{u}^T)$ is the mean rate of strain tensor.

The turbulent eddy viscosity in Equation (4.1) is estimated by the SST(Shear-Stress Transport)-based Scale-Adaptive Simulation (SST-SAS) model (Egorov and Menter, 2008; Menter and Egorov, 2010; Egorov et al., 2010). It is a unsteady RANS turbulence model based on the $k-\omega$ SST model. The governing equations for the adopted turbulence model are

$$\frac{\partial k}{\partial t} + \nabla \cdot (\bar{u}k) = \nabla \cdot [(\nu + \sigma_k \nu_t) \nabla k] + P_k - C_\mu k \omega$$  (4.2)

$$\frac{\partial \omega}{\partial t} + \nabla \cdot (\bar{u}\omega) = \nabla \cdot [(\nu + \sigma_\omega \nu_t) \nabla \omega] + P_\omega - \beta \omega + 2(1 - F_1)\sigma_{\omega 2} \frac{1}{\omega} \nabla k \nabla \omega + Q_{SAS}$$  (4.3)

where $k$ denotes the turbulence kinetic energy, $\omega$ denotes the specific dissipation
rate. $P_k = \nu_t |\vec{S}|$ is the production term of $k$, the minimal value of which is limited by $c_1 \beta^* k \omega$. $P_\omega = \gamma |\vec{S}|$ is the production term of $\omega$, the minimal value of which is limited by $c_1 \gamma \beta^* \omega^2$. $F_1$ is calculated by

$$F_1 = \tanh \left\{ \min \left\{ \max \left( \frac{\sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{y^2 \omega} \right), \frac{4 \sigma \omega k}{C_{Dkw} y^2} \right\} \right\}^4 \quad (4.4)$$

where $C_{Dkw} = 2 \sigma \omega^2 - 1 \nabla k \nabla \omega$, whose minimal value is limited by $10^{-10}$. $y$ is the wall distance. $Q_{SAS}$ is the additional SAS source term:

$$Q_{SAS} = \max \left[ \zeta_2 \kappa |\vec{S}|^2 \left( \frac{L}{L_{\nu K}} \right)^2 - 2 C k \max \left( \left| \nabla \omega \right|^2, \left| \nabla k \right|^2 \right), 0 \right] \quad (4.5)$$

where $L = \sqrt{k}/(c_{25} \omega)$ denotes the modeled length scale. $L_{\nu K} = \kappa |\vec{S}|/|\nabla^2 \vec{u}|$ is a three-dimensional generalization of the classic one-dimensional boundary layer definition, whose minimal value is limited by $C_S \delta \sqrt{\zeta_2 k/(\beta/\beta^* - \gamma)}$.

With the solutions of $k$ and $\omega$, $\nu_t$ can be computed as

$$\nu_t = \max \left\{ \frac{k}{\omega}, \frac{a_1}{b_1 F_2 |\vec{S}|} \right\} \quad (4.6)$$

where $F_2$ is calculated by

$$F_2 = \tanh \left\{ \max \left( \frac{\sqrt{k}}{\beta^* \omega y}, \frac{500 \nu}{y^2 \omega} \right) \right\}^2 \quad (4.7)$$

The coefficients used in the above turbulence formulas are listed as follows:
The behaviour of SAS model is similar to the Detached Eddy Simulation (DES) models. But it is less sensitive to grid size (Egorov and Menter, 2008). The SAS model resolves the turbulence spectrum at grid level for unsteady flow by introducing the length scale $L_{\nu,K}$. Its turbulence-resolving capability has been validated in flows with massive separation, especially for those cases with high wavenumber dissipation (Egorov et al., 2010; Zheng et al., 2016). The complex geometries of in-stream structures usually result in small eddies, which have high wavenumbers.

### 4.2.2 Rough wall models

Either Einsteinian formulation or Bagnoldean formulation indicates that near-bed particle motion is tightly connected with bed shear stress (Garcia, 2006). Therefore, accurate prediction of bed shear stress is of critical significance to sediment transport simulations. In a typical turbulent boundary layer, the flow can be roughly divided into three regions: inner layer, outer layer, and a transitional region inbetween. The mean flow velocity distribution in the inner layer can be further divided into three sublayers, namely, viscous (laminar) sublayer, buffer sublayer, and logarithmic
sublayer. Based on this division in the inner layer, the mean velocity at a certain distance from wall can be written as a function of shear velocity, kinematic viscosity, and roughness height. In the outer layer, the mean velocity is determined by edge velocity of boundary layer, distance to edge of boundary layer, boundary layer thickness, and shear velocity. For sediment transport, the shear velocity can be calculated through the flow information in the inner layer.

In the logarithmic sublayer, the mean velocity profile follows the log-law, in a non-dimensional form as

\[ u^+ = \frac{1}{\kappa} \ln y^+ + B - \Delta u^+ \]  

(4.8)

where \( u^+ = u/u_\tau \), \( y^+ = y u_\tau / \nu \). \( u_\tau \) denotes shear velocity, \( y \) is wall distance, \( \nu \) is kinematic viscosity. \( \kappa \approx 0.4 \) is the von Karman constant, \( B \approx 5.5 \) is the intercept for hydraulically smooth wall, \( \Delta u^+ \) is a roughness-related constant. Equation (4.8) can be also written as

\[ u^+ = \frac{1}{\kappa} \ln E_k y^+ \]  

(4.9)

where \( E_k \) is roughness parameter, which can be calculated as (Cebeci and Chang,
1978)

\[
\frac{E}{E_k} = \begin{cases} 
1 & k_s^+ \leq 2.25 \\
\frac{[(k_s^+ - 2.25)/87.75 + C_{k_s}k_s^+]^{0.4528}}{\sin(0.4528(\ln k_s^+ - 0.811))} & 2.25 < k_s^+ \leq 90.0 \\
1 + C_{k_s}k_s^+ & k_s^+ > 90.0
\end{cases}
\] (4.10)

where \( E \approx 9.8, C_{k_s} \approx 0.5 \). The turbulent Reynolds stress dominates in the logarithmic sublayer, while the viscous stress dominates in the laminar sublayer. In the laminar sublayer, usually Reynolds stress is very small and negligible. Based on scale analysis, one can write the following in the laminar sublayer

\[
u^+ = y^+
\] (4.11)

For the transition between laminar and logarithmic sublayers, a buffer sublayer exists and is dominated by neither viscous stress nor Reynolds stress. Thus, neither linear approximation or logarithmic approximation can be applied in the buffer sublayer. In the practice of implementing wall function in CFD codes, a common approach is to use a two-layer wall model, neglecting the buffer sublayer. The value of \( y^+ \) that separates the laminar and logarithmic sublayers can be solved easily from Equations (4.9) and (4.11). For smooth walls \( k_s^+ \leq 2.25 \), the \( y^+ \) value is about 11.53. However, as shown in Figure 4.2, because of this separation, the functional behavior of the flow in the two sublayers does not follow the same functions and
thus the transition is not smooth. In the literature, there are some other analytical
solutions (Örlü et al., 2010) of mean velocity profile blending the two sublayers,
including van Driest (1956), Spalding (1961), (Musker, 1979), Nickels (2004) et
al. Since roughness is common in sediment transport, this work only considers
those velocity profiles for rough walls, which include the Spalding and van Driest
formulas.

The Spalding velocity profile is a Taylor expansion on the inverse of the log
law Equation (4.9), which can be written as (Spalding, 1961; Shabbir and Turner,
2004)

$$y^+ = u^+ + e^{-\kappa D} \left[ e^{\kappa u^+} - 1 - \kappa u^+ - \frac{(\kappa u^+)^2}{2!} - \frac{(\kappa u^+)^3}{3!} \ldots \right] \quad (4.12)$$

where $D \approx 5.5$ for smooth wall, which also corresponds to $E_k$ in the form of

$$D = \frac{1}{\kappa} \ln E_k \quad (4.13)$$

The van Driest velocity profile is based on the integration of shear stress relationship
with wall distance, which can be written as (van Driest, 1956; Rotta, 1962; Cebeci
and Chang, 1978)

$$u^+ = 2 \int_0^{y^+} \frac{dy^+}{1 + \sqrt{1 + 4\kappa^2(y^+ + \Delta y^+)^2[1 - \exp(-(y^+ + \Delta y^+)/A)]^2}} \quad (4.14)$$
where $\Delta y^+$ accounts for roughness

$$\Delta y^+ = 0.9 \left[ \sqrt{k_s^+ - k_s^+ \exp \left( -\frac{k_s^+}{6} \right)} \right] \quad 5 < k_s^+ < 2000 \quad (4.15)$$

All three velocity profiles are compared with the same shear velocity in Figure 4.2 for both smooth wall and rough wall with $k_s^+ = 30$. The figure shows that the Spalding and van Driest velocity profiles match very well for the smooth wall case. For the rough wall case, the van Driest profile deviates slightly from the log-law in Equation (4.9). The Spalding velocity profile is located between the two-layer model and the van Driest velocity profile.

![Figure 4.2: Comparison among the two-layer model, the Spalding and van Driest velocity profiles for hydraulically smooth wall and rough wall at $k_s^+ = 30$](image)

In turbulence wall functions, given the velocity ($u$) and wall distance ($y$) at the first cell or grid near a wall, $u_\tau$ can be solved iteratively from all the above velocity profiles. Another way to calculate is through the assumption that there exists a local turbulence equilibrium where turbulent dissipation equals production. This
will be discussed in Section 4.4.2. According to the Boussinesq approximation of eddy viscosity, the turbulent viscosity $\nu_t$ can be calculated as

$$\nu_t + \nu = \frac{\tau/\rho}{\partial u/\partial y} = \frac{u_{\tau}^2}{\partial u/\partial y}$$  \hfill (4.16)

In CFD codes, the velocity gradient $\partial u/\partial y$ is approximated by $u/y$ in the near-wall region. However, for the van Driest velocity profile, $\partial u/\partial y$ can be derived analytically from Equation (4.14)

$$\frac{\partial u}{\partial y} = \frac{u_{\tau}^2}{\nu} \frac{\partial u^+}{\partial y^+} = \frac{u_{\tau}^2}{\nu} \frac{2}{1 + \sqrt{1 + 4\kappa^2(y^+ + \Delta y^+)^2[1 - \exp(-(y^+ + \Delta y^+)/A)]^2}}$$  \hfill (4.17)

If one uses the two-layer wall model Equations (4.9) and (4.11), the velocity gradient can be written as

$$\frac{\partial u}{\partial y} = \begin{cases} \frac{u_{\tau}^2}{\nu} & \text{laminar sublayer} \\ \frac{1}{\kappa y^+} \frac{u_{\tau}^2}{\nu} & \text{log-law sublayer} \end{cases}$$  \hfill (4.18)

In the Spalding wall function, $y^+$ is represented as a function of $u^+$. Thus, $\frac{\partial y^+}{\partial u^+}$ can be calculated by the reciprocal of $\frac{\partial u^+}{\partial y^+}$. 

119
4.3 Bed morphodynamics model

4.3.1 Exner equation and bed-load transport

In the current implementation of the morphodynamics model, only bedload is considered. Future work will be done to also include suspended load. The morphological changes of the bed is governed by the Exner Equation

\[(1 - n) \frac{\partial \eta}{\partial t} = -\nabla \cdot q_b \tag{4.19}\]

where \( \eta \) denotes bed elevation, \( n \) is the porosity of bed. \( q_b \) denotes bed-load transport rate vector. In the numerical implementation, the \( q_b \) vector has two dimensions, calculated by

\[q_b = q_0 \frac{\tau_b}{|\tau_b|} \tag{4.20}\]

where \( \tau_b \) is bed shear stress on the 2D bed surface calculated from 3D flow field. It should be noted that theoretically \( \tau_b \) is a 3D vector because the bed is a surface in 3D space. Equation (4.20) only considers two dimensions in \( x \) and \( y \) (if gravity is in the \( z \) direction). This is because the Exner equation is a 2D equation in \( x \) and \( y \) directions where the elevation \( z \) is a treated as a solution variable.

The bed-load transport uses a similar formula as in Roulund et al. (2005), Liu and García (2008), and Baykal et al. (2015), which is originally from Engelund and
Fredsøe (1976)

\[ q_0 = \frac{1}{6} \pi D^3 P_{EF} u_b \]  

(4.21)

where \( D \) is the grain size, usually represented by \( D_{50} \). \( u_b \) denotes the magnitude of particle’s velocity in the bed-load layer, which is calculated as

\[ u_b = 10(1 - 0.7 \sqrt{\frac{\tau^*}{\tau^*_c}}) \]  

(4.22)

where \( \tau^* = \tau_b / (\rho g R D) \) is non-dimensional measure of bed shear stress, or termed as the Shields parameter. \( \rho \) is the density of fluid. \( g \) is the gravity magnitude. \( R \) is the submerged specific gravity of particle. \( \tau^*_c \) denotes the dimensionless critical shear stress. \( P_{EF} \) denotes the probability of particle moving in the bed-load layer, which can be calculated as

\[ P_{EF} = \left[ 1 + \left( \frac{\frac{1}{2} \pi \mu_d}{\tau^* - \tau^*_c} \right)^4 \right]^{-1/4} \]  

(4.23)

where \( \mu_d \) denotes the dynamic friction coefficient.

### 4.3.2 Critical shear stress

The critical shear stress is the threshold shear condition for the initiation of particle motion, typically expressed in the non-dimensional form \( \tau^*_c \). Most threshold conditions given in the literature are for plane bed with no sloping effect. However,
due to scour and erosion, the bed is not a plane and thus the slope effect needs to be accounted for in the threshold condition. In this work, $\tau_c^*$ is calculated based on the force analysis in Engelund and Fredsøe (1976) and has the form of

$$
\tau_c^* = \tau_{c0}^* \left( \cos \beta \sqrt{1 - \frac{\sin^2 \phi \tan^2 \beta}{\mu_s^2} - \frac{\cos \phi \sin \beta}{\mu_s}} \right)
$$

(4.24)

where $\mu_s$ denotes the static friction coefficient ($=0.63$, which also corresponds to a repose angle of $\beta_0 = 33^\circ$). $\phi$ is the angle between flow velocity vector and bed’s steepest slope direction. $\beta$ is the slope angle of bed. The 3D diagram of $\phi$ and $\beta$ can be found in Figure 4.3. $\tau_{c0}^*$ is the $\tau_c^*$ value for flat bed without slope effect, which is a function of flow and sediment properties. As shown in Figure 4.4, $\tau_{c0}^*$ can be calculated from modified Shields diagram (Brownlie, 1981; Garcia, 2006)

$$
\tau_{c0}^* = 0.11R_{ep}^{-0.6} + 0.03 \exp(-17.77R_{ep}^{-0.6})
$$

(4.25)

where $R_{ep} = \sqrt{gRDD/\nu}$. For smaller $R_{ep}$, Mantz (1977) has shown that it can be approximated as

$$
\tau_{c0}^* = 0.0625R_{ep}^{-0.261}, \quad 0.056 < R_{ep} < 3.16
$$

(4.26)
In order to visualize the effect of $\phi$ and $\beta$ on $\tau_c^*$, the distribution of $\tau_c^*/\tau_{c0}^*$ over different $\phi$-$\beta$ combinations is plotted in Figure 4.5. The vertical red line shows the angle of repose. Without considering the slope effect and if there is no flow, any particle on a bed slope to the left of the red line is stable and to the right is not stable. However, considering the effect of slope, the threshold condition is represented by the blue line. Again, the region to the left of the blue curve is stable and to the right is not stable. The contour lines and their numbers shows the ratio of $\tau_c^*/\tau_{c0}^*$. For example, the points on the contour line $\tau_c^*/\tau_{c0}^* = 1.0$ are all...
these combinations of angles $\phi$ and $\beta$ such that the threshold condition does not change. Above this contour line, the ratio is larger than 1.0, which means that the threshold condition is larger than the value on plane bed. This situation happens mostly when the flow tries to move the sediment particle up the slope (against the gravity). On the other hand, below this 1.0 contour line, the ratio is smaller than 1.0, which means that the threshold value is smaller than the value on plane bed. This happens mostly when the flow tries to move the particle down the slope (in the same direction of gravity). The negative contour values (in dashed lines) means that even without flow, the particles on these large slopes will move under gravity.

A consequence of the slope angle effect is that the bed slope angle can exceed the angle of repose if the flow is pushing the particles up the slope and thus holding a much steep angle. On the other hand, if the flow is pushing the particles down the slope, the bed slope can be lower than angle of repose. Theoretically, the modification of the repose angle should be considered in the morphological models, for example the sand slide algorithm to be introduced in Section 4.3.3. However, this work only considers the effect of slope on the threshold condition, not the angle of repose in the sand slide algorithm. Future work needs to take this into consideration.
4.3.3 Sand-slide algorithm

During scour hole development, local bed angle in certain areas may exceed the angle of repose, which is not stable. The exceedance is caused by the fact the Exner equation, which describes the conservation of bed materials, does not take into consideration of the angle of repose. When such condition occurs, if only considering the gravity effect, sediment particles will slide down the slope in the steepest descending direction until the bed reaches the angle of repose. In the literature, sand-slide algorithms have been proposed to deal with this problem (Liang et al., 2005; Liang and Cheng, 2005; Roulund et al., 2005; Khosronejad et al., 2011; Nabi et al., 2013). In most cases, the bed was treated as either triangulated or uniform mesh. Liang et al. (2005) and Liang and Cheng (2005) applied a Gaussian...
average to smooth out the bed slope larger than repose of angle on 1D bed meshes. This method is only a numerical smoothing without physical meaning. Khosronejad et al. (2011) proposed a method where, for each bed cell, the angle of repose was satisfied by correcting the cell center elevations of current cell and its neighbors based on local mass conservation. Roulund et al. (2005) based their sand slide method on the force balance of a particle on slope. In their method, a particle velocity was firstly calculated by balancing the drag force against gravity component and friction force. Then, they used particle velocity to estimate bed-load transport rate in Equations (4.20) and (4.21). Afterwards, the bed elevation would be solved in Exner equation. The final bed elevation could be obtained by repeating the above steps until all particle velocities equal to zero, where gravity component is only balanced against friction force.

In the present work, a simple sand-slide algorithm is proposed. This algorithm uses a modified diffusion equation for bed elevation ($\eta$) as follows

$$\frac{\partial \eta}{\partial t} = \nabla \cdot (K \nabla \eta)$$

(4.27)

where $K$ denotes a diffusivity coefficient, which is limited by the bed slope ($-\nabla \eta$). The time derivative term on left hand side of Equation (4.27) is used for numerical purpose. The sand diffusion process will smooth out the excessive bed angles if they are larger than the repose angle.

In the numerical implementation, $\eta$ is a cell value on a 2D bed mesh. On
the other hand, $K$ is stored at edge centers, which is used to quantify the sand slide flux between the two neighbour cells sharing this edge. The limitation is applied by enforcing zero $K$ if the bed slope angle is smaller than the angle of repose. Mathematically it means $|\nabla \eta \cdot n_f| < \mu_s$, where $n_f$ denotes the normal direction of the edge. The strict enforcement of this limit will increase the stiffness of Equation (4.27). Some solutions are proposed and implemented alleviate the stiffness problem:

- In order to mimic the sand-slide effect, a small $K_0 = 1 \times 10^{-5} \, m^2/s$ is used to account for $K$ if $|\nabla \eta \cdot n_f| > \mu_s$.
- The pseudo time step $\Delta t$ used in Equation (4.27) is limited to be small too.
- With small pseudo time step size $\Delta t$, the algorithm can give smooth $\eta$. However, it may requires large number of iterations. A balance should be sought between speed and stability.
- To further reduce the stiffness, a smooth transition is used in the place of the sharp step function. The smooth function has the form of

$$\frac{K}{K_0} = \begin{cases} 
0 & R \leq R_{\text{lower}} \\
6R^5 - 15R^4 + 10R^3 & R_{\text{lower}} < R \leq R_{\text{upper}} \\
1 & R > R_{\text{upper}}
\end{cases}$$  \hspace{1cm} (4.28)
where $R = |\nabla \eta \cdot n_f| / \mu_s$. In the present study, $R_{\text{lower}} = 0.8$, and $R_{\text{upper}} = 1.2$, as shown in Figure 4.6.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{smooth_function.png}
\caption{Smooth function shown in Equation (4.28). $R_{\text{lower}} = 0.8$, and $R_{\text{upper}} = 1.2$.}
\end{figure}

\section{4.4 Immersed boundary implementation}

\subsection{4.4.1 Sharp interface method}

The basic idea of the immersed boundary method implemented in this work is similar to Ikeno and Kajishima (2007), Gilmanov and Sotiropoulos (2005), and Ge and Sotiropoulos (2007), which is the so-called sharp interface immersed boundary method (Sotiropoulos and Yang, 2014). The implementation uses a finite volume method.

As shown in Figure 4.7(a), the blue curve denotes an immersed boundary $\Gamma_{IB}$. $\Gamma_{IB}$ divides the whole computational domain into two regions based on the location of cell center, namely fluid region and solid region. The details of the cell
classification are as follows:

**Dead cells:** cells whose centers are located in the solid region, shown as lower white cells in Figure 4.7(a). Dead cells are not included in the calculation and remain “frozen” when solving the discretized linear equation system.

**IB cells:** cells whose centers are located in the fluid region and who are cut by $\Gamma_{IB}$, shown as red cells in Figure 4.7(a). Variables, such as velocity and pressure, in IB cells are reconstructed based on no-slip boundary condition. Wall functions, as discussed in Section 4.4.2, are implemented through the IB cells.

**Live cells:** cell whose centers are located in the fluid region and who are not cut by $\Gamma_{IB}$, shown as green cells in Figure 4.7(a).
Figure 4.7: 2D schematic representation of immersed boundary method and IB wall distance correction. The blue curve $\Gamma_{IB}$ is an immersed boundary. Upper green cells are live cells. Middle red cells are IB cells. Lower white cells are dead cells. Closed black dots are hit points. Crosses are centers of IB cells. Open dots are image points. The yellow highlighted IB cell is corrected due to close distance to IB wall.

Figure 4.8 shows a diagram of immersed boundary method implementation in a typical predictor-corrector algorithm. This algorithm has been implemented in OpenFOAM (ESI-OpenCFD, 2018), which is used in this research.
Figure 4.8: Algorithm diagram of the immersed boundary method within a predictor-corrector algorithm for solving fluid flow. Key steps of IB algorithm implementation are highlighted by light green background and dashed blue boundaries.

* Variables are frozen when solving these equations.

** Velocity, eddy viscosity, and other turbulence variables such as turbulent kinetic energy are updated in IB cells, usually based on wall functions.
Key steps in IB method are highlighted in Figure 4.8. The most difficult and time-consuming step is to update IB information, such as cell classification based on the location of immersed boundary in three-dimensional space. The immersed boundary is usually geometrically represented by a surface mesh, such as in the stereolithography (STL) format. IB information also includes other variables and interpolation stencils that may be used in the following sections, such as the distance between IB cell center to the immersed boundary. To speedup the calculation, fast-searching algorithms and data structure such as octree (Meagher, 1980), are used to find the spatial relationships, for example, whether a point is inside an enclosed surface and the distance from a point to a surface.

In the predictor step, velocity in IB cells are reconstructed by surrounding flow information and desired boundary condition on the IB surface before solving the momentum equations. The velocity information at IB cells is important for the in turbulence model and wall functions, which will be introduced in Section 4.4.2. When solving the momentum equations, the velocity in IB cells and dead cells are frozen, which means they will not change during the calculation. In this way, the reconstructed velocity in IB cells, which contains the boundary condition information on the IB surface, indirectly enforce the desired condition on the same IB surface.

The last step in a predictor step is to compute the predicted fluid flux, which is used in the solution of the pressure Poisson equation. Since there is no flux
in the solid region, the flux between dead cells are forced to be zero. The faces between IB cells and dead cells are denoted as IB faces. As shown in Figure 4.7, these IB faces (edges in 2D) are usually cut by immersed boundary. Thus, their location can not be fully determined. In this case, the flux does not have to be zero on IB faces. However, in order to meet the mass conservation on an immersed boundary, the sum of the flux on the IB faces need to be adjusted to zero. The adjustment strategy can be found in Equation (4.29), where $\Phi$ denotes flux on IB cells. Subscripts $\text{in}$ and $\text{out}$ denote inward flux and outward flux, respectively.

$$
\begin{cases}
\text{each } \Phi_{\text{in}} \text{ multiplies } \frac{\sum |\Phi_{\text{in}}| - \sum |\Phi_{\text{out}}|}{\sum |\Phi_{\text{in}}|} & \text{if } \sum |\Phi_{\text{in}}| > \sum |\Phi_{\text{out}}| \\
\text{each } \Phi_{\text{out}} \text{ multiplies } \frac{\sum |\Phi_{\text{out}}| - \sum |\Phi_{\text{in}}|}{\sum |\Phi_{\text{out}}|} & \text{if } \sum |\Phi_{\text{out}}| > \sum |\Phi_{\text{in}}|
\end{cases}
$$

After solving the pressure Poisson equation, the pressure gradient in dead cells are set to be zero to ensure zero flux in solid region. In the step of turbulence model, velocity and other turbulence variables will be corrected based on corresponding boundary condition and wall functions. Before solving relative turbulence transport equation, all the face flux in IB and dead cells will be killed except for the faces between IB cells and live cells. The zero flux inside IB cells ensures non-convection which kills the turbulence production terms in IB cells.

After solving the turbulence model, if the IB geometry is changed, IB information will be updated for the next time-step. The morphodynamics model is invoked after the turbulence model solution step. The coupling between the morphodynamics
model and the immersed boundary method will be introduced in Section 4.4.3.

### 4.4.2 Immersed boundary wall function

Mathematically, the velocity profile in a rough turbulent channel can be expressed as

$$u^+ = f(y^+, k_s^+)$$ (4.30)

If this velocity profile can be calculated, the corresponding velocity gradient $\partial u/\partial y$ or $\partial u^+/\partial y$ can be determined and thus the wall shear can be calculated.

The wall model presented in this section is designed for immersed boundary method. The most important parameter in the wall function is the shear velocity $u_\tau$. At least two different ways exists for the calculation of $u_\tau$:

- Based on turbulent boundary layer theory, there is a local turbulence equilibrium in the near wall region where approximately the turbulent kinetic energy production rate $P_k = \nu_t(\partial u/\partial y)^2$ equals to the dissipation rate $\epsilon = C_\mu k^2/\nu_t$. If this equilibrium holds, $u_\tau$ can be estimated by TKE (or $k$) at image points as follows

  $$u_\tau = C_{1/4}^{1/4}\sqrt{k_{IP}}$$ (4.31)

- Given the velocity $u$ and wall distance $y$ as well as $k_s$ at an image point, the corresponding $u_\tau$ can be estimated through iterative solution of Equation (4.30).
Both methods mentioned above can predict the shear velocity. Afterwards, the velocity and eddy viscosity on IB cell centers can be calculated as

\[ u_{IB} = u_f \left( \frac{y_{IB}}{\nu}, \frac{k_s u_r}{\nu} \right) \] (4.32)

and

\[ \nu_{t,IB} = \frac{u^2}{\nu} \left. \frac{\partial y}{\partial u} \right|_{y=y_{IB}} \] (4.33)

Other turbulence variables on IB cell centers can be calculated as (Baykal et al., 2015)

\[ \omega_{IB} = \frac{u^2_{t}}{\nu} \max \left[ 96.885 \left( \frac{1}{(y^+)^2} \right), \frac{1}{\sqrt{C_{\mu} k y^+}} \right] \] (4.34)

\[ k_{IB} = \omega_{IB} \nu_{t} \] (4.35)

The immersed boundaries, e.g., the scour bed, are usually of complex shape and moving. Since the background mesh is fixed, there is no control on the distance between IB cells and the boundary. So the IB wall distance \( y_{IB} \) can be very small for certain IB cells. This creates some numerical problems because the wall function formulations involve the division by \( y_{IB} \). If \( y_{IB} \) is even smaller than roughness height \( k_s \), it then has no physical meaning to predict the velocity or turbulence variables at IB cell centers which are below roughness elements. As described in the previous chapter, to deal with this difficulty and to have smooth wall shear stress distribution, an adaptive algorithm is used. This is called the \( y^+ \) adaptation,
which is implemented in the process of updating IB information in Figure 4.8. The details on the $y^+$ adaptation can be found the previous chapter.

### 4.4.3 Coupled hydrodynamic and morphodynamic models

The coupling between hydrodynamics and morphodynamics is a typical fluid-structure interaction (FSI) problem. Usually, there are two types of FSI coupling strategies, namely loose and strong coupling (Borazjani et al., 2008). The local scour model presented in this work belongs to the loose coupling FSI category, in which the hydrodynamics and morphodynamics are modeled sequentially during each time step. Indeed, as show in Figure 4.9, the morphodynamics model is invoked after the turbulence step in the hydrodynamics model.
In the present scour model, only bed-load sediment transport is considered. The governing equation of bed sediment conservation, Equation (4.19), is solved on a 2D mesh. It is noted that the bed is a surface in 3D space (represented by an immersed boundary in this work). The 2D mesh for the Exner equation is indeed the projection of the 3D bed surface onto the horizontal plane. In this work, the 2D mesh is the same as the projection of the immersed boundary surface mesh.

As discussed in Section 4.3.1 and shown in Figure 4.9, the key of the coupling process is to map the bed shear stress $\tau_b$ from 3D hydrodynamics computational domain (3D hydro mesh) to 2D morphodynamics computational space (2D morpho...
mesh). Based on Section 4.4.2, in a 3D hydro mesh, \( \tau_b \) is calculated at each hit point for each IB cell as indicated in Figure 4.7. Usually, these hit points do not match with the cell centers of 2D morpho mesh, as shown in Figure 4.10. Therefore, proper interpolation scheme is needed during the coupling process. On the other hand, the solution of Section 4.3.1 on the 2D morpho mesh needs to calculate the divergence of bed-load transport rate \( q_b \), which is a function of \( \tau_b \). It will be discussed in Section 4.4.5

**4.4.4 Interpolation between 3D hydro mesh and 2D morpho mesh**

Figure 4.10 shows the interpolation stencil of hit points for one cell (in red) on an non-uniform triangulated mesh. The background triangulated mesh denotes the 2D morpho mesh. The interpolated variables can be either stored at cell centroid or edge center as shown in Figure 4.10a and Figure 4.10b, respectively. Crosses and full dots represent the hit points used in the immersed boundary method on the 3D hydro mesh. It should be noted that based on the definition, each hit point is located on one triangle cell. For each cell centroid or edge center, a searching algorithm is used to determine the interpolation stencil as follows:

1. determine the hit points located on each cell;

2. for each variable storage location (either cell centroid or edge center), find its nearest neighbouring cells and include their corresponding hit points.
3. if the total number of interpolation hit points is less then the specified minimum number, the search expands to the neighbors of the neighbouring cells.

4. searching stops until the total number of included hit points meets the minimum requirement for the interpolation stencil.

Different schemes may be used for the interpolation. A most common one is the inverse-distance weighted interpolation

\[
\phi = \frac{\sum_{i=1}^{M} \phi_i \delta_i^{-1}}{\sum_{i=1}^{M} \delta_i^{-1}}
\]  

(4.36)

where \( \phi_i \) is the value of the \( i \)-th hit point in the stencil. \( \delta_i \) is the distance between the \( i \)-th hit point in the stencil and the cell centroid or edge center. \( M \) is the minimum requirement for the number of hit points in an interpolation stencil.
4.4.5 Effect of mesh non-orthogonality on divergence calculation

A straightforward treatment in finite volume method is to store $\tau_b$ and calculate $\mathbf{q}_b$ on the centroid of each cell in the 2D mesh. In finite volume method, Gauss’s theorem is used to compute the divergence as

$$\int_V \nabla \cdot \mathbf{q}_b dV = \sum_f S_f \cdot \mathbf{q}_{b,f} df \quad (4.37)$$

where $V$ represents a control volume. $S_f$ represents face area vector. $\mathbf{q}_{b,f}$ represents the value of $\mathbf{q}_b$ on face $f$. It should be noted that the notation in Equation (4.37) is a general form for 3D discretization. But it has the same form for a 2D mesh,
in which case volume and face representations will be replaced by face and edge representations, respectively. As a result, in 2D morpho mesh, the actual calculation of $\nabla \cdot \mathbf{q}_b$ uses the edge value of $\mathbf{q}_b$, which is typically calculated by linear interpolation using two neighbouring cell values in finite volume method. For non-uniform mesh, especially for triangulated (tetrahedral) mesh, non-orthogonality may lead to error in this linear interpolation, which further induces error in the divergence calculation for bedload transport. In the Exner Section 4.3.1, the divergence of the bedload transport rate vector is the cause of the bed elevation change. Thus, numerical errors in the divergence calculation will degenerate the scour simulation accuracy.

Mesh non-orthogonality is usually defined as the angle between the line connecting two neighbouring cell centers and the corresponding face(edge) normal. In OpenFOAM, non-orthogonal correction may be applied when solving the Laplacian term. Detailed discussion of non-orthogonal correction in OpenFOAM can be found in Ishigaki et al. (2017). It should be noted that the non-orthogonal correction in OpenFOAM is only implicitly applied on the Laplacian term, not on the gradient term or divergence term. But Equation (4.19) requires to solve divergence term explicitly.

In order to check the effect of mesh non-orthogonality on the calculation of divergence in the present model, two sets of triangulated mesh were used for comparison as shown in Figure 4.11. One is non-uniform and the other is uniform. The values of non-orthogonality of both meshes can be found in Table 4.1. The
computational domain for each mesh is approximately $1 \text{ m} \times 1 \text{ m}$, with grid size $\approx 8.6 \text{ cm}$. For the non-uniform mesh, although the cell shape is non-uniform, the area of each triangle is almost the same. Two types of variable storage location are considered, namely cell centroid and edge center. An non-linear 2D vector function $\mathbf{V}(x, y) = (V_x, V_y)$ is created as

$$V_x(x, y) = \sin(x) + x^2, \quad V_y(x, y) = \cos(y) + xy \quad (4.38)$$

where $V_x$ and $V_y$ are the components of $\mathbf{V}$ in two directions. The theoretical solution of divergence is

$$\nabla \cdot \mathbf{V} = \frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} = \cos(x) - \sin(y) + 3x \quad (4.39)$$

The numerical error here is defined as the absolute difference between the theoretical solution and the numerical solution

$$\delta_\mathbf{V} = \left| \nabla \cdot \mathbf{V} - \tilde{\nabla} \cdot \tilde{\mathbf{V}} \right| \quad (4.40)$$

where $\tilde{\nabla} \cdot \tilde{\mathbf{V}}$ denotes the numerical solution.
Figure 4.11: Comparison of $\delta V$ between using cell value and edge value on uniform and non-uniform triangulated meshes. The colormap is in logarithmic scale.

Table 4.1: Effect of mesh non-orthogonality on divergence calculation

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Non-orthogonality (ave, max) (°)</th>
<th>Variable storage location</th>
<th>Ave error</th>
<th>Max error</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-uniform</td>
<td>(6.2, 22.6)</td>
<td>cell centroid</td>
<td>$8.5 \times 10^{-2}$</td>
<td>$5.3 \times 10^{-1}$</td>
</tr>
<tr>
<td>non-uniform</td>
<td>(6.2, 22.6)</td>
<td>edge center</td>
<td>$1.1 \times 10^{-2}$</td>
<td>$1.3 \times 10^{-2}$</td>
</tr>
<tr>
<td>uniform</td>
<td>(0.0, 0.0)</td>
<td>cell centroid</td>
<td>$4.2 \times 10^{-3}$</td>
<td>$1.2 \times 10^{-2}$</td>
</tr>
<tr>
<td>uniform</td>
<td>(0.0, 0.0)</td>
<td>edge center</td>
<td>$5.6 \times 10^{-3}$</td>
<td>$6.3 \times 10^{-3}$</td>
</tr>
</tbody>
</table>
Figure 4.11 shows the distribution of the numerical error $\delta_V$ for the four different scenarios. The spatially averaged value and maximum value of $\delta_V$ for each case can be found in Table 4.1. It is found that the use of a uniform mesh is more accurate than the use of non-uniform mesh for divergence calculation. Meanwhile, compared with cell centroid storage, the edge center storage will greatly reduce the numerical error. Thus, it is strongly suggested that edge center storage is used for the variable of $\tau_b$ on the 2D morpho mesh when solving sediment transport equations.

4.4.6 Intersection between bed and object

In this present scour model, the river bed is treated as immersed boundary method while in-stream structures are modeled with body-fitted mesh. When the bed moves vertically as the scour hole develops, it dynamically intersects with the structures. As a result, the bed surface for sediment transport needs to be redefined at every time step due to the intersection.

The simplest case of local scour is due to flow around a fixed vertical cylinder. For an initially flat bed, the intersection between bed and cylinder is a circle as shown in Figure 4.12. All triangles intersected by this circle are labeled as boundary cells (for the solution of the Exner equation), colored in red in Figure 4.12a. All the cells outside of the circle are labeled as active cells (colored in green) on which Equation (4.19) will be discretized.
Unlike the immersed boundary method used for 3D hydrodynamics computational domain, it is not necessary to do any special treatment on boundary cells for Equation (4.19) because the mapped wall shear $\tau_b$ has already included the wall effect due to the structure. However, coarse mesh like Figure 4.12b cannot guarantee the sediment flux on the bed is conserved around object. In other words, the summation of sediment flux through the blue circle may not be guaranteed to be zero. One way to alleviate this problem is mesh refinement. As seen in Figure 4.12c, zero sediment flux through the vertical cylinder will be enforced better if a refined mesh is used.

In engineering practice, in-stream structures may be made of multiple objects. This geometric intersection algorithm developed in this work also applies to these situations. For example, as shown in Figure 4.13, three vertical cylinders are placed...
in an open channel. Figure 4.13a shows the 3D view of this example. The blue vertical walls are boundaries of the background 3D hydro computational mesh. Three vertical cylinders along with the vertical walls are intersected by the flat bed, which is treated as immersed boundary. Figure 4.13b shows the 2D schematics of the intersection between 2D bed and multiple objects. The blue circles are intersections between vertical cylinders and flat plane, while the dashed blue rectangle is the intersect between the flat plane and the blue vertical walls shown in Figure 4.13a.
Figure 4.13: A flat bed intersected by multiple objects and computational boundaries. Blue vertical walls in 3D view and dashed blue rectangle in 2D schematic represent the boundaries of background computational mesh.
So far, only boundary cells (or cut cells) on 2D morpho mesh are identified using the same method as IB method. The active cells, indicated by green color in Figure 4.12 and Figure 4.13b, however, remain to be determined. In some special condition like in Figure 4.13c, the active cells in the bed may be split by the objects into several isolated regions, which may cause some difficulties. The following searching algorithm is proposed to ensure its ability to deal with the isolated region situation:

1. all cells are marked as unlabeled at beginning

2. use the octree algorithm to find all the cut cells and label them as boundary cells. These boundary cells intersect with the solid boundaries in the body-fitted background mesh.

3. pick a random unlabeled cell, and search its connected neighbouring unlabeled cells, and neighbours’ neighbouring unlabeled cells recursively. All the searched unlabeled cells are labeled as one group of isolated cells.

4. repeat the above step until there is no unlabeled cell.

5. check one random cell of each isolated group to see if it is located inside background computational mesh. If it is inside 3D hydro mesh, all the cells of this group will be labeled as active cells. Otherwise, they will be labeled as inactive cells.
For parallel computation, the above steps will be executed in each processor within corresponding decomposed 3D background mesh domain. However, the bed surface mesh will not be decomposed. Every processor has a copy of the whole bed mesh and a complete 2D morpho mesh will be labeled based on corresponding subdomain mesh. A demonstration case can be found in Figure 4.14. As indicated in Figure 4.14, pinks cells represent the internal boundary cells, which are intersected by the internal boundaries of the decomposed mesh. These internal boundaries are used to communicate information between different processors.
In parallel computation, Equation (4.19) will be only solved on the master node. All the labeling information will be collected from slave nodes and combined.
on the master node. Afterwards, the internal boundary cells (colored in pink in Figure 4.14b) on master node will be relabeled as active cells. Therefore, in parallel mode, the efficiency of this algorithm mainly depends on the cell number of the 2D bed surface mesh.

### 4.4.7 Computational speed

The computational speed of the scour model depends on many factors. This section mainly describes the impacts of several important parameters, which include 2D surface mesh size, 3D background mesh size, number of hit points in each 3D-2D interpolation stencil and number of processors. It should be noted that this chapter focuses on the coupling between hydrodynamics and morphodynamics modeling. Therefore, this part will not cover the performance of IB method in the hydro model.

**Test cases description**

A simple case of flow over a sand dune is selected to test the computational speed. The bed bathymetry and the background computational domain can be seen in Figure 4.15. In this figure, the blue mesh is the background mesh which is slightly smaller than the bed such that it can be fully cut by the bed. In order to control the variables that may affect the computational speed, the background mesh uses a uniform grid.
Effect of mesh resolution

As discussed in Section 4.4.5, the 2D surface mesh may be non-uniform. But for simplicity, only uniform 2D surface mesh is used in the test. The 3D background mesh is fixed with a resolution (cell size) of 0.05 m. As shown in Table 4.2, nine sets of 2D mesh size are used to test the computational cost of the case shown in Figure 4.15. According to Figure 4.9, the whole computational process for a morphological time step in the present scour model can be divided into three parts: IB information update, hydrodynamics calculation, morphodynamics calculation. IB information update and morphodynamics calculation can be regarded as 3D-2D coupling process added onto hydrodynamics calculation. The average computational time for each part in one time step for each set of 2D surface mesh can be found in Table 4.2, which are also plotted in Figure 4.16.
Table 4.2: Setup for the effect of 2D surface mesh size. Cell size and cell number of 3D mesh are constant at 0.05 m and 48000.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Cell size ratio (2D/3D)</th>
<th>Cell number of 2D mesh</th>
<th>IB update</th>
<th>Hydro. calc.</th>
<th>Morpho. calc.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.50</td>
<td>1491</td>
<td>0.64</td>
<td>0.99</td>
<td>0.08</td>
<td>1.71</td>
</tr>
<tr>
<td>2</td>
<td>2.25</td>
<td>1817</td>
<td>0.66</td>
<td>0.99</td>
<td>0.08</td>
<td>1.73</td>
</tr>
<tr>
<td>3</td>
<td>2.00</td>
<td>2314</td>
<td>0.71</td>
<td>1.04</td>
<td>0.10</td>
<td>1.85</td>
</tr>
<tr>
<td>4</td>
<td>1.75</td>
<td>3030</td>
<td>0.74</td>
<td>1.02</td>
<td>0.11</td>
<td>1.87</td>
</tr>
<tr>
<td>5</td>
<td>1.50</td>
<td>4095</td>
<td>0.79</td>
<td>1.01</td>
<td>0.13</td>
<td>1.93</td>
</tr>
<tr>
<td>6</td>
<td>1.25</td>
<td>6063</td>
<td>0.93</td>
<td>1.06</td>
<td>0.17</td>
<td>2.16</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>9558</td>
<td>1.14</td>
<td>1.07</td>
<td>0.24</td>
<td>2.45</td>
</tr>
<tr>
<td>8</td>
<td>0.75</td>
<td>16920</td>
<td>1.55</td>
<td>1.06</td>
<td>0.37</td>
<td>2.98</td>
</tr>
<tr>
<td>9</td>
<td>0.50</td>
<td>38477</td>
<td>3.41</td>
<td>1.21</td>
<td>0.80</td>
<td>5.42</td>
</tr>
</tbody>
</table>

From Table 4.2 and Figure 4.16, it can be observed that the cost of both the IB information update and the morphological calculation increase almost linearly with the total cell number of 2D surface mesh for the same background mesh. However, the hydrodynamic calculation does not change as much with the increasing 2D mesh cell number. It should be noted that mesh resolution is more related to cell size than total cell number. If the 2D surface mesh is too refined, the computational cost may increase dramatically. On the other hand, if the 2D surface mesh is too
coarse, it may be not enough to describe geometrical details of the scour hole. In order to balance the computational cost and accuracy, it is recommended to make the 2D surface mesh size comparable with 3D background mesh size, so that 3D flow information will be maintained as much as possible during the interpolation process from 3D mesh to 2D mesh.

![Graph showing effect of 2D surface mesh size on computational cost](image)

Figure 4.16: The effect of 2D surface mesh size on the computational cost for different part in a morphological time step. Colored contours show the computational time. Dashed and solid lines show the percentage of computational time.

The size of the 3D background mesh has a great impact on the overall computational cost. To investigate this, for a constant 2D surface mesh size of 0.075 m, nine sets of 3D mesh size are used to test the computational cost. The setups and results can be found in Table 4.3 and Figure 4.17. Figures 4.17a and 4.17b plot the same data against different horizontal axes, i.e., the cell number and the mesh size of 3D background mesh.
Table 4.3: Effects of 3D background mesh size. Cell size and cell number of 2D mesh are set constant at 0.075 m and 6063, respectively.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Cell size ratio (2D/3D)</th>
<th>Cell number of 3D mesh</th>
<th>IB update</th>
<th>Hydro. calc.</th>
<th>Morpho. calc.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.50</td>
<td>219450</td>
<td>3.19</td>
<td>5.57</td>
<td>0.29</td>
<td>9.05</td>
</tr>
<tr>
<td>2</td>
<td>2.25</td>
<td>158400</td>
<td>2.60</td>
<td>3.78</td>
<td>0.22</td>
<td>6.60</td>
</tr>
<tr>
<td>3</td>
<td>2.00</td>
<td>112360</td>
<td>1.23</td>
<td>2.42</td>
<td>0.18</td>
<td>3.83</td>
</tr>
<tr>
<td>4</td>
<td>1.75</td>
<td>72726</td>
<td>0.94</td>
<td>1.45</td>
<td>0.15</td>
<td>2.54</td>
</tr>
<tr>
<td>5</td>
<td>1.50</td>
<td>48000</td>
<td>0.77</td>
<td>0.99</td>
<td>0.12</td>
<td>1.88</td>
</tr>
<tr>
<td>6</td>
<td>1.25</td>
<td>26136</td>
<td>0.60</td>
<td>0.55</td>
<td>0.11</td>
<td>1.26</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>13780</td>
<td>0.41</td>
<td>0.32</td>
<td>0.09</td>
<td>0.82</td>
</tr>
<tr>
<td>8</td>
<td>0.75</td>
<td>5600</td>
<td>0.25</td>
<td>0.14</td>
<td>0.08</td>
<td>0.47</td>
</tr>
<tr>
<td>9</td>
<td>0.50</td>
<td>1690</td>
<td>0.24</td>
<td>0.04</td>
<td>0.10</td>
<td>0.38</td>
</tr>
</tbody>
</table>

From Figure 4.17a, it can be observed that the average computational cost for one time step also increases linearly with the cell number. Both IB information update and morphological calculation increase with 3D mesh cell number. However, their percentage of total cost decrease with cell number. This is because the hydrodynamic part takes longer time as the total cell number increases. In the meantime, it can be seen that the cell number of 3D mesh has more influence over hydro calculation than the IB information update step. As shown in Figure 4.17b,
percentage of IB information update and morphological calculation increases almost linearly with the mesh size.

Figure 4.17: The effects of 3D background mesh size on the computational cost for different part in one time step. Colored contours show the computational time. Dashed and solid lines show the percentage in total computational time.

Here, we denote the total cell number of 3D background mesh as $N_{3D}$, the 3D mesh size as $\Delta x$, the total computational time for one morphological time step as $T$, the computational time of IB update and morphological calculation as $T_{IB}$. 
From Figure 4.17a and Figure 4.17b, we can obtain $T \sim N_{3D}$ and $T_{IB}/T \sim \Delta x$. Since the total cell number is proportional to the cube of the inverse of cell size, we can obtain $N_{3D} \sim (\Delta x)^{-3}$. Thus the relationship between $T$ and $T_{IB}$ can be derived as

$$T_{IB} \sim T^{2/3} \Rightarrow T_{IB} \sim N_{3D}^{2/3} \quad (4.41)$$

The implication of Equation (4.41) is that if 2D surface mesh does not change, the increase in the computational cost of the 3D-2D coupling part is less than the increase in the hydrodynamic part when the problem size increases. This scaling feature is desired because the morphological part will not be the bottleneck for the overall performance.

**Effects of 3D-2D interpolation stencil size**

In the process of IB information update, there is one step which is designed to find the interpolation stencil to map the wall shear from 3D fluid mesh to 2D bed mesh. To be more specific, the values at hit points for the IB cells in the 3D fluid domain will be interpolated to obtain the values at each triangle of the 2D surface mesh (see Section 4.4.4). The size of the stencil, or the number of hit points in each stencil, has some influence on the computational cost too. Figure 4.18 shows the impact of hit point number in interpolation stencil on computational time. The solid line represents the percentage in total computational time. This figure indicates that both IB update and hydro calculation almost do not change with the stencil size.
Only the computational time of morphological calculation increases linearly with the number of hit points in the interpolation stencil. It should be noted that in the figure 3D mesh cell size equals to 2D mesh cell size. Actually, the efficiency of the finding interpolation stencil is also very sensitive to this mesh cell size ratio.

Figure 4.18: The effects of 3D-2D interpolation stencil size. Colored contours represent the computational time. Solid line represents the percentage in total computational time.

4.5 Validations and Discussions

In order to validate the present scour model, the bridge pier scour case in Roulund et al. (2005) is selected. In this case, a vertical circular cylinder is placed on a flat bed in a steady flow as shown in Figure 4.19. The vertical cylinder is modeled using body-fitted mesh, while the flat bed is modeled by immersed boundary method. The blue grid in Figure 4.19 is the background mesh for the 3D hydrodynamics. The gray flat plane is the 2D surface mesh used to represent the flat bed. The 2D surface mesh is slightly larger than the background computational domain in length.
and width to ensure that the background mesh will be cut fully. The detailed information about flow, sediment, and mesh generation can be found in Table 4.4.

Figure 4.19: Flow around a vertical circular cylinder. Blue grid is the 3D hydrodynamics background mesh.

Table 4.4 shows the two test cases. One is for the testing of hydrodynamics where the bed is fixed. The other one is for the testing of the scour model where the bed is movable. The first one uses both body-fitted mesh and immersed boundary method to compare against experimental data (Roulund et al., 2005). The second one uses the immersed boundary method for scour as introduced in the previous sections. It should be noted that the rigid bed cases have two different types of bed, namely smooth and rough.
Table 4.4: Test conditions and meshes for scour around pier cases.

<table>
<thead>
<tr>
<th>Bed type</th>
<th>Rigid-bed</th>
<th>Live-bed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model selection</td>
<td>hydrodynamics only</td>
<td>scour model</td>
</tr>
<tr>
<td>Model type</td>
<td>body-fitted &amp; IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>Water depth $H$ (cm)</td>
<td>54</td>
<td>40</td>
</tr>
<tr>
<td>Boundary layer thickness $\delta$ (cm)</td>
<td>54</td>
<td>20</td>
</tr>
<tr>
<td>Mean flow velocity $U$ (cm/s)</td>
<td>32.6</td>
<td>46</td>
</tr>
<tr>
<td>Cylinder diameter $D$ (cm)</td>
<td>53.6</td>
<td>10</td>
</tr>
<tr>
<td>Roughness height $k_s$ (cm)</td>
<td>- / 1.0</td>
<td>0.55</td>
</tr>
<tr>
<td>Shear velocity $U_f$ (cm/s)</td>
<td>1.3 / 2.3</td>
<td>2.8</td>
</tr>
<tr>
<td>Grain size $d_{50}$ (mm)</td>
<td>-</td>
<td>0.26</td>
</tr>
<tr>
<td>Critical Shields number for flat bed</td>
<td>-</td>
<td>0.05</td>
</tr>
<tr>
<td>Static friction coefficient $\mu_s$</td>
<td>-</td>
<td>0.63</td>
</tr>
</tbody>
</table>

3D background mesh (IBM)

| Computation domain size ($D$) | $20 \times 16 \times 2.25$ | $40 \times 16 \times 3$ |
| Mesh resolution (cm) | $1 \sim 10$ | $0.5 \sim 2.5$ |

2D surface mesh size

| Computation domain size ($D$) | $28 \times 19$ | $44 \times 20$ |
| Mesh resolution (cm) | $2.5 \sim 17$ | $0.075 \sim 5$ |
Besides different types of bed, the other boundary conditions are the same for all test cases. Since free surface is not modeled, it is set as a symmetry plane. The side walls are set as slip boundaries. The outlet boundary uses the zero gradient condition for all flow variables. The inlet boundary uses prescribed velocity profiles such as shown in Figure 4.2, which are based on provided mean flow velocity ($U$), roughness height ($k_s$), shear velocity ($U_f$), and boundary layer thickness ($\delta$). It is assumed that the mean velocity is uniform outside the boundary layer thickness.

### 4.5.1 Hydrodynamics modeling

As shown in Table 4.4, both smooth and rough beds were tested. Figures 4.20 and 4.21 show the centerline distributions of streamwise and vertical velocity at different elevations, respectively. The cross markers in the figures represent the experimental measurement in Roulund et al. (2005). The blue lines represent the results of body-fitted mesh. The red lines represent the results of the present immersed boundary method. The solid lines represent the time-averaged velocity, while the dash lines represent an instantaneous velocity. All the simulations for the rigid bed cases were run for at least 100 s to reach equilibrium. Then, an additional 60 s was simulated to calculate the mean.

One of the key hydrodynamic feature which drives the scour process around cylinders is the horseshoe vortex (Sumer, 2007; Brandimarte et al., 2012; Wang et al., 2017). The horseshoe vortex can be examined in the velocity profiles shown
in Figure 4.20 and Figure 4.21. Both body-fitted mesh and immersed boundary method match well with the experimental measurement in the upstream section. The negative streamwise velocities near the upstream side of cylinder, which is due to the horseshoe vortex system, are well captured. After the flow is separated around the cylinder, the turbulence-resolving capability of the selected SST-SAS model leads to significant fluctuations in the downstream side, which can be seen in the instantaneous velocities. This increases the difficult in matching with the experimental measurement in the downstream side. The body-fitted mesh seems to have better results in the streamwise velocity than the immersed boundary method when it is close to the bed. However, considering the mesh resolution of the 3D background mesh in Table 4.4 for immersed boundary, the mismatch of the streamwise in lower elevations of $z = 0.5$ cm and 1 cm are acceptable.
Figure 4.20: Comparing streamwise velocity for smooth wall function
Figure 4.21: Comparing vertical velocity for smooth wall function
Figure 4.22: Comparing streamwise velocity for rough wall function
Figure 4.23: Comparing vertical velocity for rough wall function
Figures 4.22 and 4.23 show the results of rough rigid bed. The results of the rough bed is different from the results of the smooth bed, especially in the downstream side. The downstream flow is more unsteady if the bed is rough. The upstream velocity matches well between experiments and numerical simulations. However, both numerical results seem quite off in the downstream side. This is caused by the complex 3D flow structures in the wake zone, which can be seen in Figure 4.24. This figure shows the time evolution of instantaneous flow structure around the cylinder with the $\lambda_2$ criterion (Jeong and Hussaini, 1995). The ten snapshots represent ten typical moments in about one period of vortex shedding.

![Time evolution of flow structure around the cylinder](image)

**Figure 4.24:** Time evolution of flow structure around the cylinder with the instantaneous $\lambda_2$ criterion.

### 4.5.2 Scour modeling

The movable-bed case uses the full functionality of the scour model developed in this research. The flow condition and sediment properties are shown in Table 4.4. It is assumed that bedload is dominant such that suspended load is omitted in
the model. Based on the previous analysis of the computational cost, it takes at least several seconds to complete one morphological time step. However, the hydrodynamic time step is usually very small because of the Courant number limitation. For example, in this case the hydrodynamic time step is 0.005 s. In order to speedup the simulation, the morphological time step is usually set as several times the hydrodynamic time step. This is a common approach in scour modeling literature because the time scale of sediment transport is much larger than turbulent flow. In this work, the morphological time step is set as 0.05 s, which is ten times the hydrodynamic time step. This means the hydro-morpho coupling will be processed every ten hydrodynamic time steps. During this time period, the morphological changes will still be calculated and accumulated. But the 2D bed surface mesh will not be updated until the end of morphological time step.

Figure 4.25 shows the contour of bed elevation at different time. Flow comes from left to right. It can be observed that the scour hole develops from the upstream edge of the cylinder, where the horseshoe vortex exists as shown in Figure 4.24. It can also be observed that, at the beginning, the sediment is deposited at the downstream edge of the cylinder. Then the scour hole develops along the cylinder, and eventually the deposition at the downstream side will be transported away. This process has also been observed in the experiment (Roulund et al., 2005).
Figure 4.25: Distribution of bed elevation at different time. Flow is from left to right.

The maximum scour depth is of great interest for engineering practice. Figure 4.26 shows the time evolution of the scour depth at the upstream edge and the downstream edge of cylinder, which are usually regarded as the deepest position in upstream and downstream, respectively. This figure shows the time evolution of the upstream scour depth matches well with the experimental measurement. However, the downstream scour depth is slightly overestimated after a certain time period of around 50 minutes. A common issue in the 3D scour modeling is the time scale. In
scour experiments, typical duration is in the order of hours to days. However, in CFD simulations, it can only do minutes because of high computational cost. In this study, in order to resolve this issue, the morphological change during each time step is scaled up by 10 times. It should be noted that this scaling is different from the morphological time step size scale discussed above. The latter one is numerical acceleration but the scaling here is physical acceleration. In Figure 4.26, red lines represents morphological time step without scaling, and blue lines represents the morphological time step scaled by 10 times. This figure indicates that although scaled morphological cannot fully reproduce the original morphological time step, the difference is accepted considering the speed-up effects.
Figure 4.26: Time evolution of scour depth at the upstream edge and the downstream edge of cylinder. Scaling represents the scaling in physical acceleration.

The simulation results also reveals the difference flow direction with respect to the slope angle at different time. Figure 4.27 shows the contour of probability distribution for different combination of $\phi$ and $\beta$ around the cylinder. The domain for data sample is within three diameter from the cylinder. The darker region means higher probability of the corresponding combination of $\phi$ and $\beta$. Overall,
the majority of the combinations of angles are within the valid region, which shows that the slope effect algorithm works correctly. As the scour hole develops, the local area of high probability moves from bottom left to upper region. Since $\phi$ is the angle between the near-bed flow direction and the steepest slope direction, the shift indicates that within the scour hole, the area where flow works against gravity along a slope increases. After about 10 minutes, there exist two local areas of high probability. One is in the area with low bed slope and the flow is working against the gravity. The second is in the area with bed slope almost equal to repose angle and the flow works against the gravity. These distributions may be case specific. However, they provide some inside information on how the flow and bed co-evolve.

As discussed in Section 4.3.3, in the sand-slide algorithm, if the strict enforcement is applied, the bed slope shall be limited by red vertical line ($\beta_0$) instead of blue curve ($\beta'_0$). However, this strict enforcement induces high numerical stiffness to Equation (4.27). Considering both computational efficiency and physics of sand-slide, the iteration is limited in solving Equation (4.27), in which case, the bed slope may exceed $\beta_0$. The physical basis of the limited iteration is that the time period of sand-slide process shall be within one morphological time step instead of infinity time. Based on calculation, about 5% of the scour hole slope shown in Figure 4.27 exceed the red vertical line, while only 1.5% exceed the blue curve. This means the limited iteration in the sand-slide algorithm is reasonable despite of the very small amount of bed slopes located in invalid region ($\beta > \beta'_0$). According to
Section 4.3.3, $\beta'_0$ only depends on and can be determined by $\beta_0$, $\beta$ and $\phi$. Therefore, if $\beta'_0$ is also considered in the threshold, the sand-slide algorithm would have great improvement.

Figure 4.27: Probability distribution of different combination of $\phi$ and $\beta$ within the area three diameter away from cylinder at different moments. The vertical solid red line represents the location of $\beta_0$. The curved solid blue line represents the valid threshold of Equation (4.24).

4.5.3 A test case of complex bridge piers

One of the advantages of the proposed scour model is to simulate the burial and exposure of in-stream structure foundations, such as bridge piers. To test and demonstrate this capability, a set of complex bridge piers was chosen to be simulated. The shape of the bridge piers were based on the Chattahoochee River bridge model.
in Lee (2006). It consists of four rectangular concrete columns with large rectangular concrete footings, as shown in Figure 4.28. This is a hypothetical case because the flow and sediment conditions reported in Lee (2006) were not used. In their report, it took 25 hours to reach the equilibrium state, which is not feasible in numerical simulation. Also, the purpose of this case is to test the capability of this scour model to simulate the local exposure foundation. Thus, instead, the flow condition and sediment properties use similar values as in Roulund et al. (2005). The length scale of the structure is also adjusted by $1/250$ of field scale to ensure that the bed sediment is movable.

![Figure 4.28: A set of complex bridge piers. The pier length of field scale is 13.26 m.](image)

Figure 4.29 shows the time evolution of the scour development around these complex bridge piers. For each moment, right-hand-side is a top-view with bed only. The left figure is a 3D view with the structure. It should be noted that the
color bar of the bed elevation is different for each moment. The flow direction is parallel to the bridge bent, from left to right.

Initially, all the footings were buried in the sand. Erosion started from the corners of the left (upstream) column and sediment was transported and deposited between the middle two columns. Although there was slight erosion on the side edge of the right (downstream) column at \( t = 1 \) min, the small scour hole was quickly filled by the sand transported from the big scour hole to the left. With time, the scour around left column grew deeper. This is because the bed is in clear water scour regime. In this case, all the sand would be transported away and the bed elevation would continue to decrease. Eventually, the two footings in the middle were fully exposed, which can be observed from the top view figures. It can also be observed that once the middle two footings were exposed, the sediment deposition in between them also gradually moved away.
Figure 4.29: Time evolution of the scour development around complex bridge piers. The flow direction is from left to right.

Figure 4.30 shows the probability distribution of different combination of $\phi$ and $\beta$ near the complex bridge piers at different moments. It can be observed in Figure 4.30 that the dark area moves to the right as the scour hole develops from initially flat bed. In addition, the focal area of the distribution slightly moves up because the area where the flow works against gravity increases. These trends continue until about $t = 40$ min before the two middle footings were exposed. After that, the scour hole shape changed such that the overall bed slope in the scour hole decreases. This is probably due to the fact that the exposure of the two middle
footing removed the shielding and support for sediment. Thus, the bed lost some
capability to sustain high bed slope. After \( t = 40 \) min, i.e., the exposure of the two
middle columns, the scour holes started to reconfigure and merged into one larger
hole. As a result, the overall bed slope decreases as seen the figure that the dark
area moves to the left. The distribution of \( \phi \) also shows some changes after the
two middle columns are exposed. It decreases slightly and becomes more uniform
until \( t = 65 \) min. Afterward, it shifts toward higher values. These changes are
again case specific. They depend on how the structure is configured and the scour
process. Nevertheless, the time evolution of the probability distribution provides
some insights on the dynamics.

Also shown in Figure 4.30 are the percentage of areas where \( \beta > \beta_0 \) and \( \beta > \beta'_0 \),
which are areas where the bed slope violates the angle of repose (both the original
definition and the one considering the flow direction effect). It is found that these
percentage values are all very small. It shows that the numerical algorithms for the
sand slide and angle of repose correction are correct and accurate enough.
Figure 4.30: Probability distribution of different combination of $\phi$ and $\beta$ near the complex bridge piers at different moments. The vertical red line represents the location of $\beta_0$. The curved blue line represents the valid threshold of Equation (4.24), noted as $\beta'_0$.

4.6 Conclusions

Based on an immersed boundary method for unstructured mesh, a three-dimensional local scour model is presented in this chapter. This scour model is capable of simulating the scour process around complex geometries. This model was tested and validated against experimental measurement for a simple pier scour problem. A more complex in-stream structure, a bridge with multiple piers, was also simulated and analyzed. The main conclusions and contributions of this chapter are summarized as follows:
• Three wall models with corresponding roughness consideration were analyzed and proper adaptation was made such that they can be used with the immersed boundary method.

• In the scour model, the effect of flow direction on the critical Shield number is discussed. This work adopts a modified formulation for the calculation of critical Shields number.

• A diffusion-based sand-slide algorithm is introduced. The diffusive flux is limited based on local angle of the bed.

• The coupling between hydrodynamics and morphodynamics is discussed. A robust and parallelized interpolation scheme between 3D hydro mesh and 2D morpho mesh is proposed.

• The problematic effect of mesh non-orthogonality on the divergence calculation in the Exner equation is addressed. An edge-center storage method is proposed to alleviate this problem.

• The computational cost of the coupled model is evaluated to find the proper mesh resolution criterion and interpolation stencil size.

• In order to speed up the scour modeling for practical use, acceleration schemes were discussed and analyzed.

The validation against the pier scour case in Roulund et al. (2005) shows the
capability of this scour model to capture the horseshoe vortex system around the vertical pier. The results also indicate that the present scour model can predict scour development and the time evolution of the scour depth. The analysis on the probability distribution of bed slope and flow direction can gain some insights into the dynamics within the scour hole. The demonstration case of a complex bridge with multiple piers shows the ability of this scour model to simulate the local exposure of bridge pier foundation.

References


180


Escauriaza, C. and Sotiropoulos, F. (2011). Reynolds number effects on the coherent dynamics of the turbulent horseshoe vortex system, volume 86.

ESI-OpenCFD (2018). OpenFOAM.


Chapter 5  

Summary and future work

5.1 Summary

Hydrodynamics and morphodynamics around in-stream structures are of great significance to many aspects associated with these structures, e.g., ecological benefits and structural integrity. Most previous experimental and numerical studies focused on in-stream structures with simple geometries. For example cylinders were used to model large woody debris (LWD) and simple vertical piers were used for real bridge pier scour problems. In the practices of either stream restoration or bridge engineering, the in-stream structures usually have great complexity geometrically, which creates important micro- and meso-scale flow structures which are critical for mixing and transport. Measurement and quantification of these small scale flow patterns are difficult in the lab and field. On the other hand, high-resolution CFD models are attractive tool for the investigation of local flow structures and...
morphological changes around these structures.

This thesis achieved following objectives: (1) evaluation of the hydraulic impact of in-stream structures with different geometrical simplifications in CFD models; (2) a new $y^+$-adaptation scheme based immersed boundary wall function to improve the wall shear stress result; (3) a new local scour model based on the immersed boundary method; (4) analysis and improvement on the computational efficiency of the local scour model.

In Chapter 2, the effects of different modeling representation of engineered log jam (ELJ), a popular in-stream structure for stream restoration and erosion control, were evaluated using three-dimensional computational fluid dynamics models. Modeling results were validated with experimental measurement. Three different geometrical representations were studied, including a fully-resolved case, a porosity model case, and a solid barrier case. The analyses of the simulation results show that the porosity model and solid barrier representation significantly reduced the computational expense at the cost of missing some important physical information, which might be important for the intended purpose of in-stream structures. The porosity model, after careful calibration, appeared to have better performance than the solid barrier representation in terms of the bleeding flow through the complex structure. However, this work also identified substantial loss of accuracy in terms of velocity, turbulent shear, sediment transport capacity and coherent structure in the vicinity and the wake zone of ELJ. Thus, care needs to be
taken to interpret and use the porosity model results in practice. If the emphasis is in the far field, then the simulation results using both the porosity model and the solid barrier representation did not show dramatic difference from the fully-resolved case. Therefore, the use of such simplified models may be justified.

In preparation for the hydrodynamic part of the local scour model, Chapter 3 introduced a new immersed boundary method and a \( y^+ \)-adaptation strategy for turbulent flow simulations with RANS models. One distinctive feature of the new model is that the resulted local wall shear stress is smoother, which is important for the sediment transport process on immersed boundary. The model was developed with unstructured mesh and equipped with efficient geometric operation algorithms, which helps resolve the complex geometries.

With the new immersed boundary method, Chapter 4 further introduced a three-dimensional local scour model for complex in-stream structures. The main components of the scour model were described in detail. A diffusion-based sand-slide algorithm was introduced and a physically based determination of diffusivity was discussed. The parallelized coupling scheme between hydrodynamics and morphodynamics modules was proposed. The problematic effect of mesh non-orthogonality on divergence calculation in the Exner equation was addressed with an edge-center storage method. A numerical acceleration method associated with morphological time step, and a physical acceleration method of morphological scaling were introduced to achieve computationally feasible scour modeling. The
validation of pier scour case showed the present scour model can capture the horseshoe vortex system around the pier. It also showed that the present scour model can well predict scour development and the time evolution of the scour depth. The testing case against complex bridge piers showed the capability of this scour model to simulate the dynamic exposure of bridge pier foundations. The probability distribution of bed slope and flow direction reveals the dynamic process within the scour hole.

5.2 Future work

Although the basic framework of the coupling system between hydrodynamics and morphodynamics has been established for the local scour model in this thesis, future work is needed for additional functionalities and improved performance.

(1) Addition of suspended sediment transport. The current local scour model only considers bedload transport, which is not suitable for cases with significant entrainment and deposition. In order to model suspended load within the current immersed boundary method framework, special consideration should be given to how to incorporate the entrainment and deposition fluxes through an immersed boundary.

(2) Six-degrees-of-freedom (6DoF) movement of in-stream structures and the interaction with the evolving bed. In the current scour model, all the in-stream
structures are fixed. However, one of the concerns in the engineering practice of stream restoration is the stability of the engineered structure, for example rock vane and weirs. To model the movement of structure or its components, the structure can also be modeled with the immersed boundary method. The difficulty is then how to distinguish the structure and the bed which are both modeled with the immersed boundary method. The movement of objects also need to consider the contact and response with solid boundaries and other objects in the domain.

(3) Further improvement of implicit interpolation in immersed boundary method. The current immersed boundary method uses explicit interpolation to enforce Dirichlet or Neumann boundary conditions, which may cause stability problem in the pressure correction step. This can be improved by using implicit interpolation. In the literature, implicit interpolation is usually used with structured meshes. The challenges of implementing implicit scheme in immersed boundary with unstructured mesh are how to deal with matrix coefficients and how to parallelize.
The finite-volume discretizations of temporal and spatial derivative terms are listed in the following:

- **Time derivative**, e.g. \( \frac{\partial u}{\partial t} \). For first-order Euler scheme,

\[
\frac{\partial u}{\partial t} = \frac{u^n - u^{n-1}}{\Delta t}
\]  

(A.1)

where \( u^n \) denotes the unknown velocity at current timestep, \( u^{n-1} \) denotes the known velocity at last timestep.

- **Convection term**, e.g. \( \nabla \cdot (uu) \), \( \nabla \cdot (ku) \), \( \nabla \cdot (\epsilon u) \). According to Gauss’s theorem in Eq. (3.8),

\[
\int_V \nabla \cdot (uu) \, dV = \sum_f (S_f \cdot u_f) \, u_f, \quad \int_V \nabla \cdot (ku) \, dV = \sum_f (S_f \cdot u_f) \, k_f, \quad \int_V \nabla \cdot (\epsilon u) \, dV = \sum_f (S_f \cdot u_f) \, \epsilon_f
\]  

(A.2)
where $u_f$, $k_f$ and $\epsilon_f$ are interpolated values through the face $f$. Physically, $S_f \cdot u_f$ means the flow flux on the face $f$, which can be also expressed as $\Phi_f$.

Substitute $\Phi_f$ into Eq. (A.2)

\[
\begin{align*}
\int_V \nabla \cdot (uu) dV &= \sum_f \Phi_f u_f, \\
\int_V \nabla \cdot (ku) dV &= \sum_f \Phi_f k_f, \\
\int_V \nabla \cdot (\epsilon u) dV &= \sum_f \Phi_f \epsilon_f
\end{align*}
\] (A.3)

- Laplacian term, e.g. $\nabla \cdot (\nu + \nu_T)\nabla u$.

\[
\int_V \nabla \cdot (\nu + \nu_T)\nabla u dV = \sum_f (\nu + \nu_T)_f S_f \cdot (\nabla u)_f
\] (A.4)

where $(\nu + \nu_T)_f$ and $(\nabla u)_f$ are interpolated values of eddy viscosity and velocity gradient on face $f$, respectively.

- Gradient term, e.g. $\nabla p$ and $(\nabla u)_f$ in Eq. (A.4). Eq. (3.8) can be directly applied here to discretize $\nabla p$. For $(\nabla u)_f$, it can be calculated based on $u_P$ and $u_N$. 

190
Appendix B

Estimation of shear velocity in fully developed open channel flow

It is assumed that in fully developed open channel flow, velocity profile follows log-law in the whole water column. A class two-layer wall function is adopted as following

\[
  u^+ = \begin{cases} 
  \frac{1}{\kappa} \log(Ey^+) & \text{if } y^+ > y_{Laminar}^+ \\
  y^+ & \text{if } y^+ \leq y_{Laminar}^+ 
  \end{cases} \tag{B.1}
\]

For rough bed case, coefficient \( E \) can be estimated as a function of \( k_s^+ \), as shown in Eq. (3.41). In a general form, \( u^+ = u/u_\tau \) can be represented as a function of \( y^+ \) and \( k_s^+ \), which equals to \( yu_\tau/\nu \) and \( k_s u_\tau/\nu \) respectively. Then, Eq. (B.1) can be written as

\[
  \frac{u}{u_\tau} = f(y, k_s, u_\tau) \text{ or } g(u, y, k_s, u_\tau) = 0 \tag{B.2}
\]
For any given mean streamwise velocity $U$, water depth $H$ and bed roughness height $\delta_k$ in a fully developed open channel flow, shear velocity can be estimated by following steps

Step 1: Give an initial $u^0_\tau$, which could be any positive number within reasonable range. Tests show that the initial value won’t change the final result or slow down the iteration.

Step 2: Assume that edge velocity (very close to free surface) $u^n_e$ equals to the given mean velocity $U$, and $u^n_\tau = u^0_\tau$.

Step 3: Use Newton’s method or any other numerical method to solve Eq. (B.2), with $u = u^n_e$, $y = H$ and $k_s = \delta_k$. Specifically, in Newton’s method, the iterative process can be written as

$$u^{n+1}_\tau = u^n_\tau - g(u^n_e, H, \delta_k, u^n_\tau) \left[ \frac{\partial g(u^n_e, H, \delta_k, u_\tau)}{\partial u_\tau} \bigg|_{u_\tau = u^n_\tau} \right]^{-1}$$ (B.3)

Step 4: With an estimated $u^{n+1}_\tau$, the whole velocity profile from $y = 0$ to $y = H$ can be estimated based on Eq. (B.1). An estimated mean velocity can be computed as

$$U^{n+1} = \frac{1}{H} \int_{y=0}^{y=H} f(y, \delta_k, u^{n+1}_\tau) dy$$ (B.4)
Step 5: Adjust the edge velocity by

$$u_e^{n+1} = u_e^n \frac{U^{n+1}}{U}$$  \hspace{1cm} (B.5)

Step 6: Repeat Step 3 - 5 until the estimated mean velocity $U^{n+1}$ is very close to the given mean velocity $U$, in which case, the iteration converges. Therefore, an estimate value of $u_e^{n+1}$ is obtained.
Vita

Yuncheng Xu

Yuncheng Xu was born in Jiangyin, China, in 1990. The name of this city means the South Side of the Yangtze River, which is one of the largest rivers in the world. His childhood house was only 10 min drive from the Yangtze River. The rise and fall of the Yangtze River accompanied his childhood life. From 2005 to 2008, Yuncheng studied at Jiangsu Nanjing Senior High School in Jiangyin. After that, he went to China Agricultural University (CAU). From 2008 to 2014, at CAU, he obtained a bachelor degree in engineering mechanics, a double degree in English and a master degree in agricultural engineering. Since 2014, Yuncheng has been a Ph.D. student at Penn State University in the Department of Civil and Environmental Engineering. He was advised by Dr. Xiaofeng Liu in the program of water resources engineering. His Ph.D. work focused on developing computational fluid dynamics (CFD) models to study hydrodynamics and morphodynamics around complex in-stream structures.