HAL
open science

Numerical simulation of wind erosion : application to dune migration

Jianzhao Wu

To cite this version:

Jianzhao Wu. Numerical simulation of wind erosion : application to dune migration. Other. Université de Lyon, 2019. English. NNT : 2019LYSEC016 . tel-02166919

HAL Id: tel-02166919

https://theses.hal.science/tel-02166919v1

Submitted on 27 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre NNT: 2019LYSEC16

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de l'École centrale de Lyon

École Doctorale ED162
MEGA: Mécanique, Énergétique, Génie civil, Acoustique

Spécialité de doctorat: Mécanique des Fluides

Soutenue publiquement le 29/05/2019, par:
Jianzhao WU

Numerical simulation of wind erosion: application to dune migration

Devant le jury composé de:

Béatrice MARTICORENA	*-Examinatrice*
Catherine LE RIBAULT	*-Directrice de thèse*
Ivana VINKOVIC	*-Co-directrice de thèse*
Ning HUANG	*-Rapporteur*
Pascal FEDE	*-Rapporteur*
Serge SIMOËNS	*-Examinateur*
Sylvain DUPONT	*-Invité*

Acknowledgements

This thesis is supported by the China Scholarship Council (CSC). Numerical simulations are carried out on two scientific computing centers: PMCS2I of Ecole Centrale de Lyon and P2CHPD of Université Claude Bernard Lyon 1.

My first great gratitude goes to my supervisors: Catherine LE RIBAULT and Ivana VINKOVIC for their continuous support, encouragement, and particularly impatience during my PhD study. I would also like to offer my special thanks to Dr. Serge SIMOËNS, who provides lots of good ideas to improve my work. Besides, their great knowledge and rigorous research altitude shown in every discussions help me constantly improve myself.

I wish to acknowledge the two referees of my manuscripts: Mr. Ning HUANG and Mr. Pascal FEDE for their precious time and energy put in reviewing my manuscript and their interesting and valuable comments on my work. I also want to thank Mrs. Béatrice MARTICORENA and Mr. Sylvain DUPONT for accepting to attend my defense and giving their comments and discussions on my work.

I appreciate the help and scientific support provided by LMFA. I'm grateful for the information technology services and support provided by Mr. Bernard BARBIER, Mr. Dan-Gabriel CALUGARU and Mr. Laurent POUILLOUX.

I also want to thank my colleges in LMFA and my friends at ECL for the useful discussions and happy moments we enjoyed together. In particular, I was really happy to have weekly scientific exchanges and discussions with my friends in the lab (Haining LUO, Ying ZHU, Zecong Qin) on our own subjects.

Finally, I would like to give special thanks to my girl friend Si SHI. Her endless support, encouragement and love helped me keep moving forward and go through hard times. I thank also my parents for their selfless love and continuous support. This thesis is dedicated to them.

Abstract

Wind erosion is a complex dynamic process consisting in an atmospheric boundary layer, aeolian particle transport, sand dune deformation and their intricate interactions. This thesis undertakes this problems by conducting three-dimensional numerical simulations of solid particle transport over a fixed or deformable sand dune. Turbulent flow is calculated by a developed numerical solver (Large-eddy simulation (LES) coupled with immersed boundary method (IBM)). Solid particle trajectories are tracked by a Lagrangian approach. Particle entrainment, particle-surface interactions and particle deposition are taken into account by physical comprehensive wind erosion models.

Firstly, a new numerical solver has been developed to simulate turbulent flows over moving boundaries by introducing the IBM into LES. Two canonical simulation cases of a turbulent boundary layer flow over a Gaussian dune and over a sinusoidal dune are performed to examine the accuracy of the developed solver. Recirculation region characteristics, mean streamwise velocity profiles, Reynolds stress profiles as well as the friction velocity over the dune are presented. In the Gaussian case, a good agreement between experimental data and simulated results demonstrates the numerical ability of the improved solver. In the sinusoidal case, the developed solver with wall modeling over the immersed boundary shows a better performance than the pure one, when a relatively coarse grid is used.

Secondly, physical comprehensive modeling of wind erosion is described in detail, based on the forces acting an individual particle. An instantaneous entrainment model for both lifting and rolling-sliding modes is proposed to initialize particle incipient motions. Lagrangian governing equations of aeolian particle motion are presented and used to simulate the trajectories of solid particles. Particularly, Lagrangian governing equations of bed-load particle motion are originally deduced and applied to model the particle rolling-sliding movement on the bed surface. In addition, particle-surface interactions are taken into account by probabilistic rebound/splash models.

Thirdly, numerical simulations of particle transport over a fixed Gaussian dune and over a deformable sinusoidal dune are carried out. In the fixed Gaussian case, an overall good agreement on the particle concentration profiles over the dune between the simulated results and the experimental data of Simoëns et al. (2015) preliminarily validates the ability

and accuracy of the developed numerical solver coupled with physical comprehensive wind erosion models. In the deformable sinusoidal case, the simulated dune shapes are compared with the experimental ones of Ferreira and Fino (2012). A good agreement between them is observed at $t = 2.0$ min and an obvious underestimate of the dune shape is shown at $t = 4.0$ min and $t = 6.0$ min. By analyzing the simulated results, it is shown that the recirculation zone behind the dune is gradually reduced as the dune deforms and that windward erosion and lee side deposition is observed. It is also shown after testing that the splash entrainment is important for the lee side erosion. Moreover, a preliminary attempt is presented to apply an improved splash model with accounting for the bed slope effect to the simulation of sand dune deformation. A better performance on the simulated dune shape is achieved at $t = 4.0$ min in comparison with the experimental one.

Keywords: Wind erosion, Solid particle transport, Boundary layer, Large eddy simulation, Immersed boundary method, Sand dune deformation

Résumé

L'érosion èolienne est un phénomène complexe avec des interactions entre la couche limite atmosphérique, le transport des particules et la déformation des dunes. Dans cette thèse des simulations numériques de transport de particules solides sur des dunes fixes ou déformables sont effectuées. L'écoulement turbulent est calculé par des simulation des grandes échelles (LES) couplée avec une méthode de frontières immergées. Les particules solides sont tractées par une approche Lagrangienne. L'entraînement des particules, leur interaction avec la surface et leur dépôt sont pris en compte par des modèles physiques complets d'érosion.

D'un point de vue numérique, une méthode de frontières immergées a été introduite pour simuler les écoulements turbulents sur des frontières mouvantes. Le nouveau solveur a été validé en effectuant des comparaison avec les résultats expérimentaux de Simoëns et al. (2015) dans le cas d'une colline Gaussienne.

D'un point de vue physique, des modèles complets ont été développés pour l'érosion éolienne en se basant sur les forces agissant sur les particules. Des modèles instantanés pour l'envol, le roulement et le glissement des particules sont développés pour initier le mouvement des particules. Leur rebond et le splash sont également pris en compte. Des équations Lagrangiennes sont utilisées pour simuler la trajectoire des particules solides dans l'air. Une équation de transport d'un lit de particules a également été développée pour les cas de glissement et de roulement des particules sur la surface. La déformation de la dune est effectuée en faisant le bilan des particules qui s'envolent et se déposent. Ces modèles ont été validés en comparant les résultats de simulation avec les résultats expérimentaux de Simoëns et al. (2015) sur les profils de concentration autour d'une colline Gaussienne.

Enfin, des simulations numériques d'une dune sinusoïdale déformable sont effectuées. La forme de la dune simulée est comparée avec les résultats expérimentaux de Ferreira and Fino (2012). Un bon agrément est obtenu à $t = 2.0$ min, par contre la hauteur de la dune est sous-estimée entre 4.0 min et 6.0 min. Les résultats numériques montrent que la zone de recirculation diminue progressivement quand la dune se déforme. L'érosion, due à l'envol et au splash, est important à l'avant de la dune tandis que les particules se déposent à l'arrière de la dune. Le modèle de splash a été modifié pour prendre en compte l'effet de la pente, ce qui a permis une meilleure estimation de la hauteur de la dune à $t = 4.0$ min.

Mots-clefs: érosion èolienne, transport des particules solides, couche-limite atmosphérique, simulation des grandes échelles, méthode de frontières immergées, migration des dunes

Table of contents

Nomenclature xiii

List of figures xxiii

List of tables xxix

1 Introduction 1

1.1 Description of wind erosion . . . 1

1.1.1 Atmospheric boundary layers . . . 1

1.1.2 Aeolian particle transport . . . 2

1.1.3 Topographical evolution . . . 4

1.1.4 Intricate interactions . . . 4

1.2 Review of recent researches . . . 5

1.2.1 Research methods . . . 5

1.2.2 Research process . . . 6

1.2.3 Numerical approaches . . . 9

1.3 Objective, method and outline . . . 13

1.3.1 Objective . . . 13

1.3.2 Numerical method . . . 14

1.3.3 Outline . . . 14

2 Governing equations and numerical method 17

2.1 Governing equations . . . 17

2.1.1 Navier-Stokes equations . . . 18

2.1.2 Boussinesq approximation . . . 20

2.2 Large eddy simulation . . . 23

2.2.1 Filtered formulation . . . 24

2.2.2 Subgrid-scale modeling . . . 25

2.2.3 Near-wall treatment . . . 29

2.3 Numerical method . . . 30
2.3.1 Discretization . . . 30
2.3.2 Parallelization . . . 32
2.3.3 Terrain-following mesh and vertical grid stretching . . . 33
2.4 Initialization and boundary conditions . . . 35
2.4.1 Initialization . . . 35
2.4.2 Boundary conditions . . . 36
2.4.3 Wall modeling . . . 38
2.5 Conclusions . . . 40

3 Immersed boundary method 41
3.1 Basic description . . . 41
3.1.1 General consideration . . . 42
3.1.2 Formulation of the forcing term . . . 44
3.1.3 Advantages and disadvantages of the IBM . . . 46
3.2 Ghost-cell finite-difference method . . . 47
3.2.1 Representation of the immersed boundary . . . 47
3.2.2 Classification of computational nodes . . . 48
3.2.3 Flow reconstruction at the immersed boundary . . . 50
3.2.4 Wall modeling at the immersed boundary . . . 55
3.2.5 Instability analysis of moving boundaries . . . 59
3.3 Implementation of IBM in ARPS . . . 61
3.3.1 Initialization . . . 61
3.3.2 Time integration . . . 62
3.3.3 Parallelization . . . 63
3.3.4 Moving boundaries . . . 64
3.4 Validation of IBM-ARPS . . . 65
3.4.1 Turbulent boundary layer over a small Gaussian dune . . . 65
3.4.2 Results and comparisons . . . 68
3.4.3 Turbulent boundary layer over a large sinusoidal dune . . . 77
3.4.4 Results and comparisons . . . 79
3.5 Conclusions . . . 82

4 Physical modeling of wind erosion 83
4.1 Forces on an airborne particle . . . 83
4.1.1 Forces in undisturbed ambient flows . . . 84
4.1.2 Forces induced by the disturbance . . . 90

4.1.3 Gravity, cohesion and friction . . . 91
4.2 Particle incipient motion . . . 93
4.2.1 Threshold friction velocity . . . 93
4.2.2 Lifting, Sliding, Rolling mode . . . 95
4.2.3 Discussions on the threshold velocity . . . 99
4.2.4 Instantaneous entrainment model . . . 104
4.3 Particle transport motion . . . 108
4.3.1 Equation of airborne particle motion . . . 108
4.3.2 Equation of bed-load particle motion . . . 111
4.3.3 Lagrangian stochastic model . . . 114
4.3.4 Two-way coupling . . . 114
4.4 Particle surface interaction . . . 115
4.4.1 Rebound processes . . . 115
4.4.2 Splash entrainment . . . 117
4.4.3 Particle deposition . . . 118
4.5 Sand dune deformation . . . 119
4.5.1 Temporal evolution of dune height . . . 119
4.5.2 Avalanche model . . . 120
4.6 Conclusions . . . 120

5 Numerical simulation of wind erosion 123
5.1 Validation case: particle transport over a fixed Gaussian dune . . . 124
5.1.1 Experimental configuration . . . 124
5.1.2 Numerical configuration . . . 125
5.1.3 Results and analysis . . . 126
5.2 Wind erosion case: particle transport over a deformable sinusoidal dune . . . 129
5.2.1 Estimation of the duration of sand dune deformation . . . 129
5.2.2 Experimental set-up . . . 130
5.2.3 Numerical configuration . . . 131
5.2.4 Flow characteristics . . . 133
5.2.5 Sand dune deformation . . . 136
5.2.6 Improved splash model with accounting for bed slope effect . . . 144
5.3 Conclusions . . . 146

6 Conclusions and Perspectives 149
6.1 Conclusions . . . 149
6.2 Perspectives . . . 152

Appendix A Similarity theory in turbulent boundary layer flows **153**
A.1 Boundary layers on a smooth surface 153
A.2 Boundary layers on a rough surface 155

Appendix B Numerical details in ARPS **159**
B.1 Discretized formulation of governing equations 159
B.2 Detailed description of boundary conditions 163

References **167**

Nomenclature

Acronyms

ARPS Advanced Regional Prediction System

CAPS Center for Analysis and Prediction of Storms

CFD Computational Fluid Dynamics

DEM Discrete Element Method

DNS Direct Numerical Simulation

IBM Immersed Boundary Method

IBM-ARPS Immersed Boundary Method and Advanced Regional Prediction System

INRA Institut National de la Recherche Agronomique

LES Large Eddy Simulation

LES-NWM Large Eddy Simulation with Near-Wall Modeling

LES-NWR Large Eddy Simulation with Near-Wall Resolution

MPI Message Passing Interface

RANS Reynolds Average Numerical Simulation

SGS Subgrid-scale

TBL Turbulent boundary layer

Greek symbols

θ_{ejc} Ejection angle

θ_r Rebounding angle

(ζ, η, ξ) Curvilinear coordinate system after the vertical grid stretching

(ζ, η, ξ') Curvilinear coordinate system

α_0, β_0 Two coefficients in Equation (3.4)

$\boldsymbol{\delta}$ Dirac function

$\boldsymbol{\psi}$ Vector of physical variables

$\boldsymbol{\psi}^n$ Discrete vector of physical variables at the time step n

$\boldsymbol{\psi}_{ib}$ Desired immersed boundary condition

χ Distance from the wall normalized by the thickness of boundary layers δ

Δ Filtered scale

$\Delta\rho$ Fluid density variation to the base state

$\Delta\Theta$ Potential temperature variation to the base state

$\Delta\tilde{\Theta}$ Filtered potential temperature variation to the base state

$\Delta\xi$ Vertical grid spacing in the coordinate system (ζ, η, ξ)

$\Delta\xi'_{min}, \Delta\xi'_m$ Averaged grid spacing in the bottom and middle layer in Figure 2.3

δ Thickness of boundary layers

δ_ν Viscous lengthscale in the inner region of boundary layers

Δ_i, Δx, Δy,Δz Grid spacings

δ_{ij} Kronecker delta

ε_{ijk} Levi-Civita symbol

γ Specific heat ratio

γ_ρ Density ratio defined by $\gamma_\rho = (\rho_p - \rho)/\rho$

γ_g Gravity number

κ von Kàmàn constant

κ_s	Positive spring constant
κ_T	Fluid molecular conductivity
κ_{sgs}	Subgrid diffusivity
λ	First coefficient of fluid viscosity or shear viscosity, commonly, $\lambda = -\frac{2}{3}\mu$
λ_T	Thermal diffusivity
μ	Second coefficient of fluid viscosity or volume viscosity
μ_r	Rolling friction coefficient
μ_s	Sliding friction coefficient
μ_t	Turbulent viscosity
μ_e	Effective turbulent viscosity in Equation (2.49)
μ_{re}	Equivalent rolling friction coefficient defined by $\mu_{re} = \tan(\varphi + \varphi_r)$
μ_{se}	Equivalent sliding friction coefficient defined by $\mu_{se} = \tan(\varphi + \varphi_s)$
ν	Fluid kinematic viscosity defined by $\nu = \mu/\rho$
Ω_i	Angular velocity of the earth
Π	Wake strength parameter
ψ	A physical variable
ψ_0	Constant for the Dirichlet boundary condition in Equation (3.6)
ψ_G	Value of a physical variable ψ at the ghost point G
ψ_n	Constant for the Neumann boundary condition in Equation (3.7)
$\psi_{G'}$	Value of a physical variable ψ at the image point G'
ρ	Fluid density
ρ_r	Fluid density of the base state
σ_{ij}	Cauchy stress tensor

τ_p, τ'_p Particle characteristic timescale responding to the fluid

τ_w Mean wall shear stress

τ_w Wall shear stress

τ_{ij} Deviatoric stress tensor of σ_{ij}

Θ Potential temperature

θ Bed slope

Θ_r Potential temperature of the base state

$\tilde{\varepsilon}$ Dissipation rate of filtered motions

$\tilde{\sigma}_{ij}$ Subgrid-scale tensor

$\tilde{\tau}_{ij}$ Deviatoric subgrid-scale tensor of $\tilde{\sigma}_{ij}$

φ Angle depending on the particle geometrical arrangement as shown in Figures 4.12 and 4.13

φ_r Rolling friction angle

φ_s Sliding friction angle

$\vec{\omega}$ Fluid vorticity defined by $\vec{\omega} = \nabla \times \vec{u}$

K Medium permeability coefficient

Sh Shield number

Sh_c Critical Shield number

ν_{sgs} Subgrid viscosity

ρ_p Solid particle density

Roman symbols

v_{ejc} Ejection velocity

(x, y, z) Cartesian coordinate system

$-\langle u'w' \rangle$ Reynolds stress

$\boldsymbol{a}$	Vector composed by a_1, a_2, a_3, a_4
$\boldsymbol{P}$	Matrix given by Equation (3.13)
$\Delta\tilde{p}$	Filtered pressure variation to the base state
Δp	Fluid Pressure variation to the base state
δt	Time step
Δu	Shift velocity
Δu^+	Shift velocity normalized by the friction velocity u_*
$\Delta x^+, \Delta y^+, \Delta z^+$	Grid spacings normalized by the viscous lengthscale δ_ν
Δz_{min}	Refined grid increment in the bottom layer
Δz^+_{min}	Refined grid increment in the bottom layer normalized by the viscous lengthscale δ_ν
$\hat{k}$	Wavenumber in Fourier space
$\langle C \rangle (x,z)$	Mean concentration of solid particles
$\langle C_{max} \rangle$	Maximal mean concentration of solid particles
$\mathscr{B}_d$	Operator denoting the boundary conditions on the immersed dune surface ∂D_d
$\mathscr{B}_f$	Operator denoting the boundary conditions on the computational domain
$\mathscr{F}$	Operator denoting additional forcing terms when the immersed boundary method is introduced
$\mathscr{G}$	Operator denoting the governing equations
∂D_f	Boundaries of fluid computational domain D_f
∂D_d	Immersed dune surface
$\tilde{q}_{j\Theta}$	Subgrid-scale flux
$\tilde{S}$	Characteristic filtered rate of strain defined by $\tilde{S} = \sqrt{2\tilde{s}_{ij}\tilde{s}_{ji}}$
$\tilde{s}_{ij}$	Filtered strain tensor
$\tilde{u}_i$	Filtered fluid velocity

$\vec{u}(\vec{x}_p,t)$ Fluid velocity at the position of $\vec{x}_p$

$\vec{u}_p(\vec{x}_p(t),t)$ Particle velocity at the time t

$\vec{x}_p(t)$ Particle position at the time t

$\vec{F}_A, F_A$ Added mass force and its norm

$\vec{F}_B, F_B$ Basset force and its norm

$\vec{F}_D, F_D$ Aerodynamic drag force and its norm

$\vec{F}_L, F_L$ Aerodynamic lift force and its norm

$\vec{F}_P, F_P$ Acceleration by the fluid and its norm

$\vec{G}, G$ Gravity and its norm

a Dimensional constant in Equation (4.78)

a_1, a_2, a_3, a_4 Interpolation coefficients in Equation (3.8)

C_1, C_2 Coefficients in the transport equation of SGS kinetic energy

C_C Cohesion coefficient

C_D Drag coefficient normalized by the particle-fluid relative velocity u_r

C_D^w Wall drag coefficient normalized by the friction velocity u_*

C_f Skin friction coefficient

C_K Kolmogorov constant

C_L Lift coefficient normalized by the particle-fluid relative velocity u_r

C_L^{Saff} Saffman-lift coefficient

C_L^w Wall Lift coefficient normalized by the friction velocity u_*

C_M Coefficient in the SGS model based on the SGS kinetic energy transport equation

c_p Specific heat at constant pressure for an ideal gas

c_s Acoustic wave speed defined by $c_s = \sqrt{\gamma R T_r}$

d Depth of the entire domain in Figure 2.3

$D(x,t)$, $D(\vec{x},t)$ Deposition rate

d_1, d_2, d_3 Depth of the bottom, middle and top layer in Figure 2.3

D_f Fluid computational domain

d_p Particle diameter

d_p^+ Particle diameter normalized by the viscous lengthscale δ_v

d_p^- Particle diameter normalized by the viscous diameter d_v

d_v Viscous diameter

e Internal energy

$E(\hat{k},t)$ Energy spectrum

$E(x,t)$, $E(\vec{x},t)$ Emission rate

$F(d_p^+)$ Function in Equation (4.47) accounting for the aerodynamic effect

$f(Re_p)$ Function of particle Reynolds number given by Equation (4.4)

F_C Cohesion

F_d Velocity-defect law

F_f Sliding friction force

f_i Body force

f_w Law of the wall

$f_{\boldsymbol{\psi}}$ Boundary force for physical variable $\boldsymbol{\psi}$

g Gravitational acceleration

$G(d_p^-)$ Function in Equation (4.47) accounting for the inter-particle cohesion effect

H Dune height

h Effective height of roughness elements

h Fluid enthalpy

$H(\theta)$ Function in Equation (4.47) accounting for the bed slope effect

$h(x,y)$ Terrain height function

h^+ Effective height of roughness elements normalized by the viscous lengthscale δ_ν

I_p Solid particle inertial moment

k Vertical grid index

k_{sgs} Subgrid-scale kinetic energy

L_{AB} Maximum distance from the dune surface to the recirculation streamline in the vertical direction as shown in Figure 3.22

L_{CD} Longest length between two extremities of the recirculation zone as shown in Figure 3.22

L_x, L_y, L_z Domain size

M Mach number

m_p Solid particle mass

M_R Rolling friction torque

N Reaction force on the particle from the wall

n_1, n_2, n_3 Number of grid levels in the bottom, middle and top layer in Figure 2.3

$N_d(x,t)$, $N_d(\vec{x},t)$ Number of entrained particles per unit area and unit time

$N_e(x,t)$, $N_e(\vec{x},t)$ Number of entrained particles per unit area and unit time

N_{ejc} Number of ejected particle by the splash scheme

N_x, N_y, N_z Number of grid nodes

nz Total number of staggered grid points

p Fluid pressure

p_0 Reference level pressure

p_r Fluid pressure of the base state

Pr Prandtl number defined by $Pr = \nu/\kappa_T$

Pr_{sgs} Modeled SGS Prandtl number

q_i Heat flux

R Gas constant

R Particle radius

r Heat radiation

Re_p Particle Reynolds number normalized by particle-fluid relative velocity u_r

Re_δ Reynold number defined by $Re_\delta = U_e\delta/\nu$

s_{ij} Fluid strain tensor

St_L, St_η Stokes number

T Fluid temperature

T_r Fluid temperature of the base state

U Mean streamwise velocity

u^+ Mean streamwise velocity normalized by the friction velocity

u_* Friction velocity

U_e External velocity

u_i, u, v, w Fluid velocity

$u_{r,c}^{lifting}, u_{*,c}^{lifting}$ Critical (friction) velocity of lifting particle incipient motion

$u_{r,c}^{rolling}, u_{*,c}^{rolling}$ Critical (friction) velocity of rolling particle incipient motion

$u_{r,c}^{sliding}, u_{*,c}^{sliding}$ Critical (friction) velocity of sliding particle incipient motion

v_{imp} Impacting particle speed

v_{reb} Rebounding particle speed

$w(\chi)$ Wake function

w_p^t Particle terminal velocity

x_{reat} Maximum position of the recirculation zone as shown in Figure 3.22

x_{sep} Minimum position of the recirculation zone as shown in Figure 3.22

z^+ Distance form the wall normalized by the viscous lengthscale Δ_ν

z_0 Roughness height

z_d Shift displacement

z_{flat} Prescribed height above which the curvilinear vertical coordinate coincides with the Cartesian one, i.e., $\xi' = z$

z_{min} Height of the bottom boundary

$\vec{u}_r, u_r$ Particle-fluid relative velocity and its norm

C_S Smagorinsky coefficient

D_d Dune domain

F_L^+ Aerodynamic lift force normalized by ρv^2

l_S Smagorinsky length

u_i'' Fluctuated velocity of scales smaller than the filtered scale

List of figures

1.1 Schematic illustration of the complex dynamic process of wind erosion. . . 2

1.2 Different modes of airborne particle motion: suspension, saltation, creep. . 3

1.3 Research processes of wind erosion in view of fluid mechanism and solid particle transport. 7

2.1 Sketch of the Arakawa C-grid and of the variable position. 31

2.2 Sketch of the domain decomposition (left) and of the message passage between the sub-domains (right). Left figure taken from Xue et al. (1995). Right figure taken from http://www.idris.fr/formations (MPI). 32

2.3 Sketch of the vertical grid stretching (left) and of its corresponding terrain-following grid (right). The terrain-following mesh with the vertical stretching is generated by the ARPS code and the grid control parameters are given by $z_{flat} = 1.2$, $d_1 = 0.8$, $d_2 = 1.6$, $\Delta\xi'_{min} = 0.05$, $\Delta\xi = 0.1$ and $nz = 37$. 34

2.4 Different boundary conditions for the numerical simulation of boundary layers in ARPS. 37

2.5 Validation of the numerical model (ARPS). Blue square: experimental results [Vinçont et al. (2000) and Simoens et al. (2007)] with $Re_\tau = 500$. Black line: DNS results of Spalart (1988) with $Re_\tau = 660$. Red circle: ARPS results. Adapted from Huang (2015). 40

3.1 Sketch of a boundary layer over a single dune (left) and of the corresponding body-conformable grid (right). 42

3.2 Sketch of the computational domain (left) and of the corresponding Cartesian grid (right) of a boundary layer over a single dune resolved by the immersed boundary method. 43

3.3 Sketch of the location of the dune height variable $h(x,y)$ in an Arakawa C-grid cell. 48

3.4 An example of the representation of dune height at the Arakawa C-grid. . . 48

3.5 An example of the classification of computational nodes. 49
3.6 An example of reconstructing the physical variable ψ at the ghost node G. G' is the image point of G. P_1, P_2, P_3, P_4 are the four neighbors. $\vec{n}$ is the normal vector at P_1. 51
3.7 Sketch of case 1 for Dirichlet and Neumann boundary conditions. 51
3.8 Sketch of case 2 for Dirichlet and Neumann boundary conditions. 52
3.9 Sketch of case 3 for Dirichlet and Neumann boundary conditions. 52
3.10 Sketch of case 4 for Dirichlet and Neumann boundary conditions. 53
3.11 Sketch of case 5 for Dirichlet and Neumann boundary conditions. 54
3.12 An example of applying wall modeling at the wall node W. P is a fluid node, a grid interval above W. 56
3.13 An example of applying wall modeling at the wall node W. P is the interpolation point for W. P_1, P_2, P_3, P_4 are the four neighbors of P. 57
3.14 Situation 1 of the immersed boundary motion. 59
3.15 Situation 2 of the immersed boundary motion. 60
3.16 Situation 3 of the immersed boundary motion. 60
3.17 Initialization procedure in ARPS with the implementation of IBM. The red boxes denotes the additional initialization IBM modules 62
3.18 Time integration procedure in ARPS with the implementation of IBM. The red box denotes the flow reconstruction and wall modeling IBM modules. . 63
3.19 Time integration procedure for wind erosion problem in ARPS with the implementation of IBM. 64
3.20 Sketch of the wind tunnel set up of the experiment of Simoëns et al. (2015). Taken from Huang (2015). 65
3.21 Sketch of the computational domain of TBL over a small Gaussian dune. . . 66
3.22 Sketch of the key characteristics of the recirculation zone behind a 2D obstacle. Figure adapted from Huang et al. (2018). 68
3.23 Mean streamlines over a Gaussian dune and recirculation zones on the lee side. 69
3.24 Mean longitudinal velocity profiles over a small Gaussian dune. Points: experiments of Simoëns et al. (2015). Dashed lines: ARPS simulation (Test A). Solid lines: IBM-ARPS simulation (Test B). 69
3.25 Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a small Gaussian dune. Points: experiments of Simoëns et al. (2015). Dashed lines: ARPS simulation (Test A). Solid lines: IBM-ARPS simulation (Test B). 70
3.26 Comparison of the friction velocity between the ARPS simulation (Test A) and the IBM-ARPS simulation (Test B). 71

3.27 Mean streamlines over a Gaussian dune and recirculation zones on the lee side. 71

3.28 Comparison of flow fields simulated by the IBM-ARPS between with imposing zero velocity on the solid nodes (Test B) and without any interior treatments (Test C). 72

3.29 Mean streamlines over a small Gaussian dune and recirculation zones behind the dune. 73

3.30 Mean streamlines over a Gaussian dune and comparison of the recirculation zones behind the dune. 74

3.31 Comparison of flow fields simulated by the IBM-ARPS between without wall modeling (Test B) and with wall modeling (Test D). 75

3.32 Comparison of flow fields simulated by the IBM-ARPS between without pressure reconstruction (Test B) and with pressure reconstruction (Test E). . 76

3.33 Sketch of the computational domain of TBL over a large sinusoidal dune. . 77

3.34 Mean streamlines over a large sinusoidal dune and recirculation zones behind the dune. 79

3.35 Mean longitudinal velocity profiles over a large sinusoidal dune. Black points: ARPS simulation (Test A′). Blue dashed lines: IBM-ARPS simulation without wall modeling (Test B′). Red solid lines: IBM-ARPS simulation with wall modeling (Test C′). 80

3.36 Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a large sinusoidal dune. Black points: ARPS simulation (Test A′). Blue dashed lines: IBM-ARPS simulation without wall modeling (Test B′). Red solid lines: IBM-ARPS simulation with wall modeling (Test C′). 80

3.37 Comparison of the friction velocity between the ARPS simulation (Test A′), the ARPS simulation without wall modeling (Test B′) and the IBM-ARPS simulation with wall modeling (Test C′). 81

4.1 Sketch of the aerodynamic drag force $\vec{F}_D$ for a particle sitting on the wall. . 86

4.2 Sketch of the aerodynamic lift force $\vec{F}_L$ for a particle sitting on the wall. . . 86

4.3 Lift coefficient C_L^w by different formula. —: Equation (4.17); -- : Equation (4.14) and Equation (4.16); —: Saffman (1965) second order; —: Leighton and Acrivos (1985); —: Hall (1988); —: Mollinger and Nieuwstadt (1996); —: Zeng et al. (2009). 89

4.4 Ratio of lift to drag force for an immobile particle of diameter $d_p^+ = 1$ submerged in a boundary layer on the smooth and rough wall. F_D is calculated by Equation (4.1) and Equation (4.3). F_L is calculated by Equation (4.7) and Equation (4.11). The average streamwise velocity field is generated by the wall of the law. 89

4.5 Sketch of the submerged gravity $\vec{G}$. 91

4.6 Sketch of the cohesion force $\vec{F}_C$. 91

4.7 Sketch of the sliding friction F_f. 92

4.8 Sketch of the rolling friction M_R. 92

4.9 Probability of particle incipient motion versus the dimensional shear stress $\tau = u_*^2/(\gamma_\rho g d)$. Taken from Wu and Lin (2002). 93

4.10 Friction velocity u_* versus the flow velocity U_e without and with particle transport. Taken from Ferreira (2017). 94

4.11 Sketch of the lifting mode of particle incipient motion. Left: the surface is plate $\theta = 0$; Right: a general sketch with a slope θ. 95

4.12 Sketch of the sliding mode of particle incipient motion. Left: the surface is plate $\theta = 0$; Right: a general sketch with a slope θ. 96

4.13 Sketch of the rolling mode of particle incipient motion. Left: the surface is plate $\theta = 0$; Right: a general sketch with a slope θ. 98

4.14 Functions $F(d_p^+)$, $H(\theta)$ and $G(d_p^-)$. —: Lifting mode; —: Sliding mode with $\varphi_{se} = 48°$; —: Rolling mode with $\varphi_{re} = 33°$. 102

4.15 Critical Shield numbers Sh_c of different particle diameters. —: Lifting mode; —: Sliding mode with $\varphi_{se} = 48°$; —: Rolling mode with $\varphi_{re} = 33°$; : Equation (4.39); - - : Equation (4.40); - - : Equation (4.41); - - : Equation (4.44). ⊙: White (1982); ⊙: Williams (1986); ⊙: Fernandez Luque and Van Beek (1976); ⊙: Fletcher (1976); ⊙: Chepil (1945); ⊙: Greeley and Iversen (1985). 103

4.16 Bed slope effect on the critical Shield number Sh_c of lifting, sliding and rolling mode. —: Lifting mode; —: Sliding mode with $\varphi_{se} = 48°$; —: Rolling mode with $\varphi_{re} = 33°$. 104

4.17 Sketch of a complete particle incipient process by lifting. 105

4.18 Sketch of a complete particle incipient process by sliding. 106

4.19 Sketch of a complete particle incipient process by rolling. 107

4.20 Schematic illustration of the forces acting on an airborne particle. $\vec{u}$ is the fluid velocity, $\vec{\omega}$ is the vorticity and $\vec{u}_p$ is the particle velocity. The aerodynamic drag $\vec{F}_D$ is in the opposite direction of the particle-fluid relative velocity $\vec{u}_r = \vec{u}_p - \vec{u}_r$ according to Equation (4.1). The aerodynamic lift $\vec{F}_L$ is in the direction of $\vec{\omega} \times \vec{u}_r$ according to Equation (4.7). 109
4.21 Schematic illustration of the forces acting on a bed load particle and of its motion. $\vec{u}_p^c$ is the particle velocity and $\vec{\omega}_p^c$ is the particle angular velocity in the curvilinear coordinate system (ζ, η, ξ). F_D denotes the aerodynamic drag, F_L the aerodynamic lift, F_C the cohesion, G the gravity, N the reaction, F_f the friction, M_R the friction torque, respectively. 112
4.22 Sketch of the rebound process. $P-O-R$ describes the three-dimensional rebound process with the incident angle θ_i, the impacting speed v_{imp}, the rebound speed v_{reb}, the rebound angles θ_r and $\theta_{r,y}$. $P-O-R_y$ describes the two-dimensional rebound process projected on the incident plane Oxz with its rebound speed v_{reb}^{xz} and the rebound angle θ_r. 116

5.1 Sketch of sandbox-type particle-feeding device. Figure adapted from Taniere et al. (1997). 124
5.2 Sketch of numerical configuration of particle transport over a Gaussian dune. 125
5.3 Particle trajectories over a fixed Gaussian dune. Results obtained by the IBM-ARPS simulation coupled with integrated wind erosion models (Case B). 127
5.4 Comparison between particle longitudinal velocity (points) and fluid velocity (lines) over a Gaussian hill. Results obtained by the IBM-ARPS simulation coupled with integrated wind erosion models (Case B). 128
5.5 Particle concentration profiles over a Gaussian dune. Points: experiments of Simoëns et al. (2015). Dashed lines: ARPS simulation coupled with integrated wind erosion models (Case A). Solid lines: IBM-ARPS simulation coupled with integrated wind erosion models (Case B). 128
5.6 View of a single sand pile in the wind tunnel. Taken from Ferreira and Fino (2012) 131
5.7 Sketch of numerical configuration of wind erosion. 132
5.8 Mean streamlines over a deformable dune and recirculation zones on the lee side. Results obtained by Case A′. 134
5.9 Mean longitudinal velocity profiles over a deformable dune. Results obtained by Case A′. 135
5.10 Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a deformable dune. Results obtained by Case A′. 136

5.11 Friction velocity at $t = 0$ min, $t = 2$ min, $t = 4$ min, $t = 6$ min. Results obtained by Case A′. 137
5.12 Sand dune shape at $t = 2$ min. Red lines: experiment of Ferreira and Fino (2012). Blue lines and points: numerical simulation (Case A′). 138
5.13 Erosion and deposition. Erosion (violet lines and points): total area of particles removed from the sand dune between $t = 0$ min and $t = 2$ min. Deposition (red lines and points): total area of particles deposited on the sand dune between $t = 0$ min and $t = 2$ min. 139
5.14 Contributions to erosion between $t = 0$ min and $t = 2$min: lifting, rolling and splash. Results obtained by Case A′. 140
5.15 Sand dune deformation (Case A′) at $t = 4$ min. 141
5.16 Sand dune deformation (Case A′) at $t = 6$ min. 142
5.17 Comparison of the dune shape at $t = 2$ min between Case A′, Case B′ and Case C′. 143
5.18 Comparison of the dune shape at $t = 2$ min between Case A′, Case D′ and Case E′. 143
5.19 Dune shapes at $t = 2$ min obtained by the experiments of Ferreira and Fino (2012), Case A′ and Case F′. 145
5.20 Dune shapes at $t = 4$ min obtained by the experiments of Ferreira and Fino (2012), Case A′ and Case F′. 146

A.1 Experimental results of the shift velocity Δu^+ from Nikuradse (1933) and schematic illustration of different roughness regimes: hydrodynamically smooth regime, transitionally rough regime and fully rough regime. 156
A.2 Sketch of different flow regime depending on the roughness configuration: isolated flow regime, waked flow regime and skimming flow regime. Taken from Oke (1988). 157

List of tables

1.1 Numerical approaches of wind erosion at different scales. η is the dissipation scale and L is the integral length of atmospheric boundary layer. d_p is the particle diameter and L_s is the saturation length. 10

3.1 Physical parameters given for simulating TBL over a small Gaussian dune. The Reynolds numbers are defined by $Re_H = U_e H/\nu$, $Re_\delta = U_e \delta/\nu$ and $Re_\tau = u_* \delta/\nu$. 67

3.2 Numerical parameters given for simulating TBL over a small Gaussian dune. 67

3.3 Test cases with different treatments for a TBL over a small Gaussian dune. . 67

3.4 Characterization of the recirculation zone behind a small Gaussian dune. . . 69

3.5 Physical parameters given for simulating TBL over a small Gaussian dune. The Reynolds numbers are defined by $Re_H = U_e H/\nu$, $Re_\delta = U_e \delta/\nu$ and $Re_\tau = u_* \delta/\nu$. 78

3.6 Numerical parameters given for simulating TBL over a small Gaussian dune. 78

3.7 Test cases with different treatments for a TBL over a large sinusoidal dune. 78

3.8 Characterization of the recirculation zone behind a large sinusoidal dune. . 80

4.1 Some formulas of the non-dimensional lift force F_L^+ and the wall lift coefficient C_L^w. 88

4.2 Magnitude order of force ratio. Adapted from Vinkovic (2005). 91

4.3 Mathematical formula of $F(d_p^+)$, $G(d_p^-)$ and $H(\theta)$ for lifting, sliding and rolling mode. 102

5.1 Parameters of solid particles. 125

5.2 Numerical cases of particle transport in a TBL over a small Gaussian dune. 126

5.3 Parameters of solid particles. 131

5.4 Numerical cases of particle transport in a TBL over a small Gaussian dune. 133

5.5 Estimation of computational time for the numerical simulation of Case A′. Δt is the large time step. N_p denotes the number of processors. CPU-H represents CPU hours necessary for 6 min of Case A′. 133
5.6 Characterization of the recirculation zone behind a deformable dune. Results obtained by Case A′. 134
5.7 e_h and σ_h between the numerical results (Case A′) and the experimental data (Ferreira and Fino (2012)). 140

Chapter 1

Introduction

In this chapter, we give a brief description of the process of wind erosion, which consists in the atmospheric boundary layer, aeolian particle transport, topographical evolution and their intricate interactions. Then, we review the recent researches of wind erosion, specially the numerical modeling and approaches of wind erosion. Finally, the objective, method and organization of this thesis are given.

1.1 Description of wind erosion

Wind erosion is an environmental topics involving many serious issues such as desertification, land degradation, air pollution, *etc.*. Enormous desertification is slowly reducing our living space. Large area of land degradation decreases the agricultural production. Severe air pollution has a negative impact on our health and affects our daily life. Although wind erosion proceeds unnoticed, it destroys our living environment, threatens people's health and causes economic damages like a chronic disease. Therefore, understanding, modeling and controlling the physical mechanism of wind erosion becomes an increasingly important issue for researches.

Wind erosion is a complex dynamic process consisting in the atmospheric dynamics, sand particle motion, topography evolution, and their intricate interactions as shown in Figure 1.1 [Lancaster (2011)]. In the following, these main four parts are introduced in detail.

1.1.1 Atmospheric boundary layers

The understanding and modeling of atmospheric boundary layer is important for the study of wind erosion. In atmospheric boundary layers, the multi-scale turbulence is generated by both the wind shear and the buoyancy. In the layer close to the surface, the wind shear near

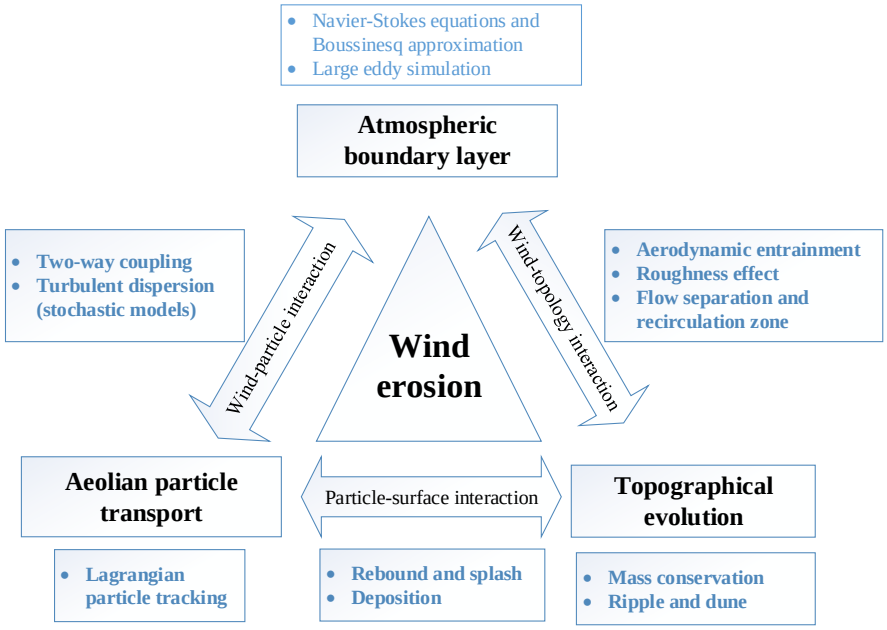


Figure 1.1 Schematic illustration of the complex dynamic process of wind erosion.

the surface is responsible for the particle entrainment, and turbulent coherent structures play a crucial role on particle transport and deposition. For the layer over a smooth surface, the wall shear stress is easily evaluated since the viscous dissipation is dominant and turbulence is weak in the viscous sublayer. When the surface consists of roughness elements, the presence of roughness strongly affects the transfer process of turbulent kinetic energy and the distribution of shear stress near the surface. Hence, it is of considerable importance to estimate the wall shear stress with accounting for roughness effects in the investigation of wind erosion. Commonly, the influence of wall roughness on the flow is taken into account in an average way. Some models to account for the roughness effect on the mean velocity profile is described in Appendix A. In computational fluid dynamics (CFD), the atmospheric boundary layer is modeled by the simplified Navier-Stokes equations using Boussinesq approximations [Stull (1988)], and the roughness effects are taken into account by imposing a rough wall model [Byun (1990); Mason and Callen (1986)].

1.1.2 Aeolian particle transport

Individual sand particles are entrained by the aerodynamic forces, then transported in the carried flow and lastly fall toward the surface due to the gravity effect. Owing to the presence of turbulent structures near the surface, the aeolian particle motion becomes complex.

Turbulent fluctuations bring certain uncertainty to the trajectory of individual particles and the preferential concentration appears in the regions where the longitudinal velocity is small [Vinkovic et al. (2011)]. Based on the experimental and field observations, Bagnold (1941) suggested that the aeolian particle motion can be classified into three categories as shown in Figure 1.2 (d_p is the particle diameter):

- Suspension ($d_p \leq 70\ \mu$m): such particles are often called dusts. They are easily suspended in the atmospheric boundary layer since their terminal velocities are small. Large quantities of such particles suspended in air result in the air pollution or dust storm.

- Saltation ($70\ \mu\text{m} < d_p \leq 500\ \mu$m): the diameter of sand particles is generally within this interval. The saltation is the major motion mode for the sand particle movement during the process of wind erosion. Massive particle transport by saltation results in the dune deformation, ripple formation, *etc.*.

- Creep ($d_p > 500\ \mu$m): such particles are too heavy to be entrained by aerodynamic forces. However, they can slide or roll on the surface, pushed by wind or by impact of saltating particles. Hence, the creeping process plays a important role in the topography deformation.

In numerical simulations, a Lagrangian model is usually used for calculating the trajectory of aeolian particles [Vinkovic (2005)]. Moreover, continuum models are also developed and applied for the simulation of aeolian particle transport [Sauermann et al. (2001)].

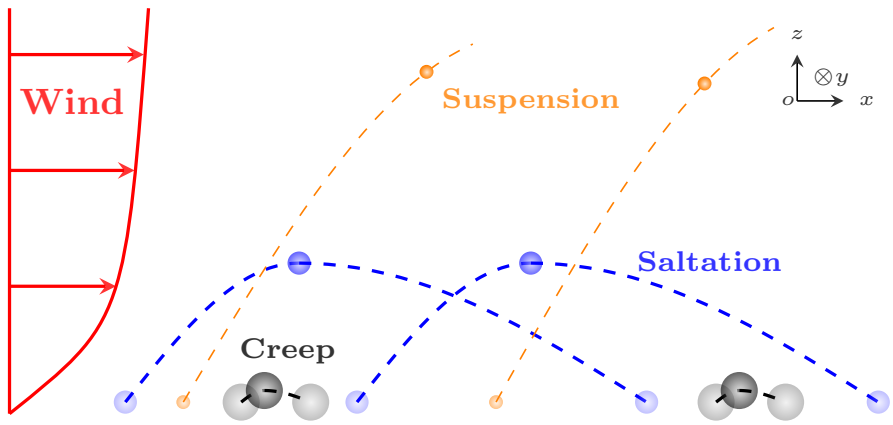


Figure 1.2 Different modes of airborne particle motion: suspension, saltation, creep.

1.1.3 Topographical evolution

In the geological view, wind erosion contributes greatly to the topographical evolution. Byun (1990) found that the long-range transport of large quantities of dusts from continent to ocean results in the formation of sand seas in the world. According to Raupach et al. (1994), wind erosion causes the huge loss of topsoil, which results in the loss of soil nutrients and thus in the agricultural land degradation. Moreover, desertification is a typical example of wind erosion in the arid area. Every year several million hectares global drylands are converted to deserts, and the rate is increasing [Greeley and Iversen (1985)]. Though desertification is caused by many factors such as global warming, human activity, the transport of solid particles is the direct cause of arable land transforming into deserts.

1.1.4 Intricate interactions

The interactions between the atmospheric boundary layer, the solid particle transport and the bed surface (topography) are complex. They are briefly described in the following:

Wind and particle: Solid particles are carried by the wind and dispersed in the atmosphere, whereas the motion of solid particle decelerates the carried flow. Technically, the interaction between the carried flow and solid particle motions should be taken into account at the same time, but it is obviously difficult to be achieved in numerical simulations. In practice, particle motions are governed by the forces acting on it, such as the aerodynamic forces (drag and lift) and gravity, *etc.*, and simulated by Lagrangian tracking method. When the effect of particles on the carried flow is accounted for, a strategy of two-way coupling is used by introducing the feedback forcing inside the governing equations of the atmospheric flow.

Wind and topography: Particle emission by the wind is one origin of the topographical evolution, whereas the topography complexity affects the near-wall flow characteristics. When the wall shear stress is larger than the threshold one, particles start their incipient motions, either to roll or slide on the surface, or to be lifted in the atmosphere. The loss of many particles modifies the local topography structure. In contrary, the induced topographical structure at micro-scale, also called roughness, breaks the near-wall flow characteristics and then decreases the mean velocity of wall flows. More details of roughness effect are introduced in Appendix A. In addition, for the topographical structures at the meso-scale, such as sand dunes, hills, Hoffmann et al. (1985) and Muck et al. (1985) found a convex curvature attenuates the pre-existing turbulence while a concave curvature leads

to a quasi-inviscid generation of longitudinal vortices. In the case of large bed slopes, flow separation takes place and the recirculation zones are created on the lee side of obstacles.

Particle and topography: Submitted to gravity, solid particles transported by wind in the saltation layer ultimately fall toward the ground and collide with particles on the surface [Beladjine et al. (2007)]. After the collision, there are two possibilities: firstly, if the particle impacting velocity is large enough, then it rebounds and continues its motion in the fluid with a new velocity (called rebound process). At the same time, some particles on the ground may be ejected due to the absorption of kinetic energy lost during the collision (called splash entrainment) [Kok and Renno (2009)]; secondly, if the impacting velocity is too small to achieve the rebound process, it may continue to slide or roll on the surface until deposition occurs due to the friction effect (called deposition). The processes of rebounding, splashing and deposition depend on both the characteristics of the impacting particle and the local properties of the surface topography, such as the sizes and shapes of the particles on the ground, and the arrangement structure of particles, *etc.*. Conversely, particle emission by splash entrainment and deposition contributes to the modification of local topographical structures.

1.2 Review of recent researches

In this section, we briefly review the recent researches on the wind erosion from three aspects: research methods, research process and numerical approaches. In particular, we list several numerical methods, which have been used for simulating wind erosion in recent years.

1.2.1 Research methods

The comprehensive understanding of the wind erosion physics has been progressing through three different approaches: experimental observations, theoretical studies and numerical studies.

Field measurements and wind-tunnel experiments: Field observations and wind-tunnel experiments are two effective ways of wind erosion researches. These studies focused on the the investigation of the fundamental physics, such as the estimation of the critical shear stress [Bagnold (1941); White (1982)], the influence of roughness on turbulent flows [Nikuradse (1933); Zingg (1953)], the statistics of particle entrainment or of particle-surface collision [Beladjine et al. (2007); Kok and Renno (2009)]. Recently, wind-tunnel experimental studies

of a saltation process were conducted to investigate the characteristics of particle transport in a boundary layer, such as the mean profile of particle concentration, the phenomena of particle preferential concentration [Creyssels et al. (2009); Ho et al. (2014); Simoëns et al. (2015)].

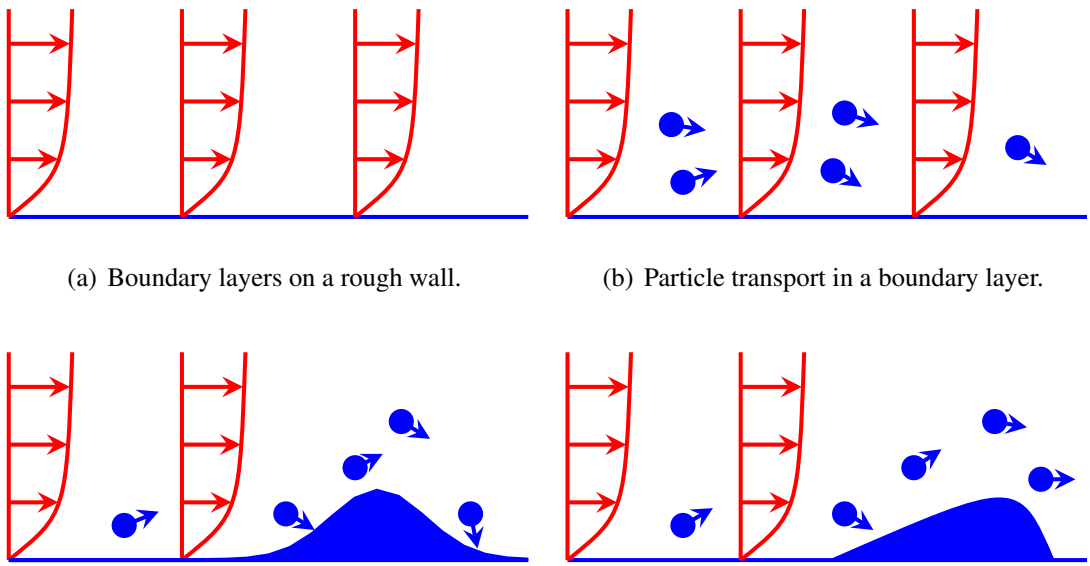
Theoretical studies: The dynamic model of wind erosion is complex. Theoretical investigations were carried out step by step. Firstly, these mathematical formulations of the forces acting on an individual particle were derived, and then the dynamic equation of wind-blown particles was obtained Maxey and Riley (1983). Based on the balance of these forces, the theoretical formulation of the threshold friction velocity was deduced and compared with experimental measurements [Greeley and Iversen (1985); Shao and Lu (2000)]. Secondly, using this mathematical modeling, the analytical trajectory of an individual particle in a boundary layer with mean velocity profiles was theoretically figured out [Sørensen (1991)]. This provides us a theoretical basis for the understanding and investigation of wind erosion. Thirdly, using the analytical saltation trajectory, the dynamics of saltation layers can be modeled from the uniform state to the non-uniform state [Shao (2008)]. Based on the momentum balance of the saltation layer, the relation of saturated saltation flux is obtained, namely, $q_s \sim (u_* - u_{*,c})^p$, where u_* and $u_{*,c}$ are the wind and critical friction velocities, and $p = 3$ is a scaling exponent [Bagnold (1941); Durán and Herrmann (2006); Owen (1964)].

Numerical simulations: Comprehensive numerical modeling of wind erosion attracts the attention of researchers in recent years [Durán et al. (2012); Huang et al. (2018); Kok and Renno (2009); Lopes et al. (2013)]. Compared with other approaches, numerical simulation of wind erosion provides more information on turbulent flows and particle motions, which are potentially useful in understanding the physics of wind erosion, even they are submitted to the validity and limited by the used modeling. In this thesis, the atmospheric boundary layer is resolved by large eddy simulation and the trajectories of solid particles are simulated by a Lagrangian tracking approach [Huang (2015); Wu et al. (2017)]. Different models have been introduced to take into account the interactions of the particles with the surface, especially the aerodynamic entrainment, the rebound and the splash. Other numerical approaches and modeling are described in detail in Subsection 1.2.3.

1.2.2 Research process

In the view of fluid dynamics, the research on wind erosion is gradually advanced through four different processes as shown in Figure 1.3: *i)*, studies of boundary layers on a rough wall estimates the roughness effect on turbulent flows; *ii)*, investigations of the particle transport

in a boundary layer figure out the characteristics of particle entrainment, particle transport and particle deposition; *iii)*, researches of the particle transport over a fixed dune focus on the influence of the presence of an obstacle (hill, building, tree . . .) on the flow and on the particle transport; *iv)*, studies on particle transport over a deformable dune contributes greatly to the physical understanding of wind erosion, thus of the moving morphology of the ground.



(a) Boundary layers on a rough wall.

(b) Particle transport in a boundary layer.

(c) Particle transport over a fixed dune.

(d) Particle transport over a deformable dune.

Figure 1.3 Research processes of wind erosion in view of fluid mechanism and solid particle transport.

Boundary layers on a rough flat wall: Roughness elements constitute the most frequently the ground from which occurs the atmospheric boundary layer. Raupach et al. (1980) firstly proposed the notion of roughness sublayer, in which roughness elements have an obvious influence on the flow structures. Based on many experimental and numerical results, Raupach et al. (1991) and Jiménez (2004) supported the point that the near-wall flow structures are strongly affected within the roughness sublayer whereas the outer flow structure is nearly unaffected, when the Reynolds number is high and the ratio of roughness height to the boundary layer thickness is small. Moreover, Oke (1988) defined three flow regimes: isolated flow, waked flow and skimming flow, according to the density of roughness elements in the wind tunnel experiments of Hussain and Lee (1980). Simoens et al. (2007) experimentally reproduced these three flow regimes using different spacings of squared obstacles and

analyzed the vortex structures created within the canyons. Recently, Huang et al. (2016) proposed an improved law of the wall with accounting for the roughness effect in the three flow regimes, based on a collected dataset. The roughness effects in a boundary layer are detailedly introduced in Appendix A.

Particle transport in a boundary layer on a flat plate: Particle entrainment, particle transport and particle deposition in a boundary layer are crucial in the fundamental studies of wind erosion. Numerous wind tunnel and numerical experiments of a saltation layer have been conducted to investigate the fundamental properties and physical mechanism of particle entrainment, transport and deposition. Creyssels et al. (2009) performed the experiment of saturated particle transport in a turbulent boundary layer, and analyzed the mean profiles of particle concentration and velocity. Ho et al. (2014) showed based on the wind-tunnel experiments that the distribution of saltation length and height obeys a log-normal law, which is independent on the flow strength. Based on numerical results, Rouson and Eaton (2001) and Picciotto et al. (2005) found that the spatial distribution of solid particles is related to the instantaneous flow topology. Solid particles are preferentially concentrated in the regions of low-speed streaks. Vinkovic et al. (2011) and Yu (2015) conducted the direct numerical simulation (DNS) cases of particle transport in channel flows and showed that the detachment of solid particles from the wall are usually surrounded by ejections. This result is confirmed more recently for non spheric particles [Ouchene et al. (2018)]. In the review of Soldati and Marchioli (2009), the physical mechanism of particle deposition is described. Particles fall through the accumulation region and deposit on the surface either by the direct impaction or by the indirect action of near-wall turbulent fluctuations.

Particle transport in a boundary layer over a fixed dune: The presence of obstacles in a boundary layer may cause a flow separation and create large recirculation zones, in which solid particles are trapped and deposited. Almeida et al. (1993) used Laser-Doppler measurements to investigate the recirculating flow behind two-dimensional, polynomial-shaped dunes. Cao and Tamura (2006) carried out the experiments of boundary layers over a rough sinusoidal dune and found that the roughness effect enhances the flow separation and extends the reattachment length, which agrees well with the experimental results of Simoëns et al. (2015). However, the investigation of Kanda et al. (2013) showed the opposite point that the upwind wall roughness can suppress the flow separation and lead to a smaller recirculation zone behind a trapezoidal dune. Thus the wall roughness effect on the control of flow separation behind an obstacle is unclear and still needs further investigations. In this thesis, we don't study the roughness effect on flow separation but focus on the behavior of

particle transport around the moving and deformable dune and the corresponding recirculation zone. In a priori study of Huang (2015), particles are entrained on the windward side of the dune due to the increase of upwind friction velocity and are deposited on the lee side due to the reversed flow velocity in the recirculation zone.

Particle transport in a boundary layer over a deformable dune: To our knowledge, only a few studies on the particle transport over a moving or deformable dune have been carried out. Most of researches focused on the understanding and modeling of dune dynamics without accounting for the turbulence effect on particle transport [Charru et al. (2013); Lancaster (2011); Wiggs (2001)]. As the wind speed increases up the windward slope of an individual dune and decreases on the lee side, sandy dunes migrate by erosion on the windward side and deposition on the lee side [Ernstsen et al. (2007)]. Durán et al. (2012) and Durán et al. (2014) developed numerical models of particle transport from bed load to saltation and carried out numerical simulations of the dynamic formation of sand ripples. Lopes et al. (2013) conducted the numerical simulation of sand dune deformation using classic wind erosion models and compared the simulated results with the wind-tunnel experimental data of Ferreira and Fino (2012). However, these two simulation cases are two-dimensional and use the Reynolds Average Numerical methods (RANS).

1.2.3 Numerical approaches

With the rapid development of computational fluid dynamics (CFD), numerical modeling and simulation of wind erosion attracts our attentions and becomes an effective research method to investigate the physical mechanics of particle transport in atmospheric boundary layers. Table 1.1 summarizes numerical approaches for simulation and modeling of the fluid phase and of solid particles at different scales. For the fluid phase, the atmospheric boundary layer is simulated by either DNS, Large eddy simulation (LES) or RANS, which is used either without any modeling, or with subgrid-scale modeling, or with turbulent modeling, respectively. For solid particles, the motion of dispersed particles in the carried flow is computed by Lagrangian particle tracking. Particle entrainment, particle collision and deposition are simulated either by Discrete element method (DEM) or by probabilistic models.

RANS + Flux-divergence model: This method has been usually applied for the simulation of the development and migration of large sand dunes, and of the emergence and evolution of sand ripples [Charru et al. (2013); Wippermann and Gross (1986)]. Generally, it is achieved by three steps:

Modeling	Grid size Δ	ABL numerical simulation	Grid size Δ	Numerical simulation of solid particle motions
Micro-scale modeling	$\Delta < \eta$	Direct numerical simulation (DNS)	$\Delta < d_p$	Direct Element method (DEM)
Large-scale modeling	$\Delta \gg \eta$ and $\Delta \ll L$	Large eddy simulation (LES) + subgrid-scale modeling	$\Delta \gg d_p$ and $\Delta \ll L_s$	Integrated wind erosion models (Particle entrainment, collision and deposition models) + Lagrangian particle tracking
Statistically Averaged modeling	$\Delta > L$	Reynold averaged numerical simulation (RANS) + turbulent modeling	$\Delta > L_s$	Flux-divergence model (saturated transport flux)

Table 1.1 Numerical approaches of wind erosion at different scales. η is the dissipation scale and L is the integral length of atmospheric boundary layer. d_p is the particle diameter and L_s is the saturation length.

- Step 1: computation of the atmospheric boundary layer using RANS in order to obtain the local friction velocity, which is responsible for the sand particle transport flux;
- Step 2: determination of the distribution of the particle transport flux q_s using the friction velocity u_* obtained from RANS simulation in Step 1, according to an empiric formula of saturated transport flux: $q_s \approx f(u_*)$;
- Step 3: determination of the erosion and deposition rate by evaluating the divergence of particle transport flux, and then evolution of sand dune form according to the conservation of mass.

Wippermann and Gross (1986) attempted to apply the flux-divergence model to simulate the migration of a barchan dune from a conical pile of sand using the simple saturated flux formula of Lettau (1978). The original flux-divergence model involved the local saturated assumption of particle transport, under which the local flux can be replaced by the saturated one. Andreotti et al. (2002); Charru et al. (2013); Durán et al. (2012) carried out the systematic analysis of particle transport in a boundary layer and concluded that the local transport flux is equal to the saturated one for the equilibrium transport state, and is suggested to be a function of the saturated length, the saturated time, the saturated flux for the non-equilibrium state. Charru and Hinch (2006) proposed an improved flux-divergence model by adding the erosion rate and deposition rate, which are related to the particle motions. Moreover, Sauermann

et al. (2001) developed a continuum saltation model coupled with the flux-divergence model for sand dunes migration with accounting for the aeolian transport of dispersed particles.

RANS + Integrated wind erosion models + Lagrangian particle tracking: Integrated wind erosion models consist in particle entrainment model, particle-surface collision model (rebound and splash models) and particle deposition model, which are established based on the forces acting on an individual particle. With this method, numerical simulation of wind erosion is conducted by three steps:

- Step 1: computation of the atmospheric boundary layer using RANS in order to supply the flow information for particle entrainment, transport and deposition;
- Step 2: determination of particle entrainment or deposition according to the corresponded criterion in integrated wind erosion models and calculation of aeolian particle motion by Lagrangian particle tracking. In particular, the particle-surface collision is simulated by a probabilistic rebound model and the particle ejection caused by the rebound is simulated by the splash model;
- Step 3: determination of the erosion and deposition rate by counting the number of entrained particles and deposited particles, and the evolution of sand dune form according to the conservation of mass.

Different from the flux-divergence model, the integrated wind erosion models coupled with Lagrangian particle tracking take into account the different behaviors of solid particles: particle entrainment, particle transport, particle deposition and particle-surface interaction. Hence, this modeling seems more physical, comprehensive and closer to the real situation. Kok and Renno (2009) applied the law of the wall of boundary layers on the rough wall for step 1 and proposed a comprehensive numerical model of steady state saltation. Lopes et al. (2013) used this approach for the numerical simulation of wind erosion with accounting for both saltation and creep, and investigated the development of a large sinusoidal sand dune. However, in the work of Lopes et al. (2013), the RANS simulation of boundary layers is two-dimensional and the integrated wind erosion models are derived from the saturated particle transport flux. Thus the local wind erosion is totally equivalent to an analytical function of the friction velocity, which is similar to that in the flux-divergence model. This severely reduces the accuracy of the wind erosion rate calculation for the unsaturated state of particle transport. Further, the saturated particle transport flux is unknown for complex terrains, *e.g.*, it is hard to reach a good approximation of the flux using this model for several successive dunes of different forms.

LES + Integrated wind erosion models + Lagrangian particle tracking: This method is almost the same as the previous one, except that LES is used for the calculation of the boundary layer for step 1. Compared with RANS, LES provides more informations of turbulent fluctuations, which are related to particle entrainment, transport and deposition. To account for the effect of solid particles on turbulent flows, two-way coupling is used by introducing an additional drag force in the fluid momentum equations. To account for the influence of turbulence at small scales on particle motion, a stochastic model is established by introducing the small-scale Lagrangian velocity fluctuation into the particle transport equation [Vinkovic (2005)]. Shao (2008) used this method to investigate the saltation in a boundary layer and found that the splash entrainment is important in the process of saltation. By using the immersed boundary method to handle the bed topography deformation, Nabi et al. (2012) successfully applied this method to simulate the dynamic process of underwater ripple and dune formation. Based on this wind erosion modeling, Dupont et al. (2013) conducted the numerical simulation of time-dependent saltation over a flat plate surface and focused on the investigation of saltation intermittency. Recently, Huang et al. (2018) carried out the numerical simulation of saltation over several successive fixed hills with different hill spacings covering three basic cases of skimming, wake or isolated flow regimes proposed by Oke (1988) and conducted the parametric studies of the presence of obstacles on the characteristics of particle transport.

RANS + DEM: DEM is used for the direct numerical simulation of particle motions, including particle entrainment, transport, deposition and particle-surface collision. A two-way coupling is applied to take into account the influence of solid particles on the turbulent flow. Durán et al. (2012) and Durán et al. (2014) used this two-phase numerical approach based on DEM for particles coupled with RANS for boundary layers and conducted the numerical investigation of saltation and of the aeolian ripple dynamics. However, these simulations are two-dimensional. The computational cost of a three-dimensional simulation of particle motions using DEM is extremely expensive.

DNS + Lagrangian particle tracking: The turbulent flow is simulated by DNS and solid particles are tracked by a Lagrangian approach. This method is usually used for the investigation of particle distribution in a turbulent flow, for example, the preferential concentration of particles in isotropic turbulence [Squires and Eaton (1991)] or in channel flows [Fessler et al. (1994)]. Moreover, Vinkovic et al. (2011) and Yu (2015) used this method to study particle transport in channel flows and showed that the detachment of solid particles from the wall are usually surrounded by ejections. However, the employment of DNS in the simulation

of boundary layers is extremely computationally expensive, especially with high Reynolds number and with moving boundaries.

1.3 Objective, method and outline

As described in Section 1.1, wind erosion is a complex dynamic process and the physics behind it has not yet been in-depth studied. As reviewed in Section 1.2, numerical models have been developed for comprehensively simulating the dynamics of wind erosion. However, most of numerical simulations focus on investigation of saltation dynamics over a rough wall without obstacles or with immobile obstacles, and these simulations of sand ripple formation or dune migration are either inaccurate using traditional saturated flux-divergence models, or two-dimensional or underwater using more comprehensive wind erosion models. Hence, it inspires us to conduct a three-dimensional numerical simulation of wind erosion using an integrated wind erosion model coupled with LES.

1.3.1 Objective

The main goal of this thesis is to understand the physics of wind erosion by conducting numerical simulation of a turbulent boundary layer over a deformable sand dune. Concretely, the objectives are to:

- Develop a numerical method for wind simulation with moving boundaries: large eddy simulation coupled with immersed boundary method (LES-IBM) is developed and validated in this thesis. This numerical solver has the ability of simulating boundary layers at high Reynold number and over moving boundaries.

- Develop physical models of wind erosion: a comprehensive wind erosion model is proposed to simulate the processes of particle entrainment, transport, deposition and particle/surface interaction (rebound and splash);

- Conduct numerical simulations of wind erosion and understand the physics of wind erosion by analyzing the simulated results.

This work is an extension of the thesis of Huang (2015) for a fixed dune (as shown in Figure 1.3(c)) to a deformable dune (as shown in Figure 1.3(d)). In this thesis, both the implementation of the IBM into the ARPS code and physical modeling of wind erosion with accounting for the interaction between solid particles and the moving surface are original.

1.3.2 Numerical method

For the fluid phase, LES is used for the simulation of boundary layers and the immersed boundary method is applied to treat moving boundaries. Compared with RANS, LES allows the computation of the instantaneous evolution of large-scale turbulent structures able to produce sweeping events responsible for aerodynamic entrainment of solid particles. Compared with DNS, LES takes less computational cost.

For solid particles, the Lagrangian particle tracking approach coupled with integrated wind erosion models is adopted for the simulation of particle motions. Concretely, the Lagrangian particle tracking method is used to calculate the particle trajectory. The integrated wind erosion models are applied to simulate the processes of particle entrainment, deposition and particle-surface interaction (rebound and splash). This approach saves computing resource compared with DEM, and accounts for particle motions and does not depend on the saturated assumption compared with the flux-divergence model.

In this thesis, numerical simulation of wind erosion is achieved using the Advanced Regional Prediction System (ARPS) code, which is originally developed by the Center for Analysis and Prediction of Storms (CAPS) at the university of Oklahoma. Through the efforts of our teams for many years, ARPS has been developed into a numerical solver that can simulate the transport of solid particles in boundary layers. Vinkovic et al. (2006) developed a Lagrangian stochastic model and implemented it in ARPS for the study of the dispersion of solid particles in a turbulent boundary layer. Dupont et al. (2013) used ARPS for the investigation of saltation dynamics over a flat surface. Huang (2015) implemented the particle entrainment and rebound model in ARPS for the study of particle transport over one or several Gaussian hills. Thanks to the work of S. Dupont at INRA (Institut National de la Recherche Agronomique), Bordeaux, the MPI (Message Passing Interface) parallelization of ARPS code has been completely accomplished, notably for the module of the calculation of solid particle motion. The work of this thesis is also part of the continuous development of this numerical tool.

1.3.3 Outline

This manuscript is organized as follows:

- In Chapter 2, the theoretical and numerical formulations of the LES code (ARPS with the additional development of Vinkovic (2005) and Huang (2015)) are described.

- In Chapter 3, the implementation of immersed boundary method in ARPS is introduced and the validation of this new numerical solver is presented.

- In Chapter 4, physical modeling of wind erosion related to particle entrainment, transport, deposition and particle-surface interaction is described.
- In Chapter 5, numerical simulation of sand dune deformation is conducted and the analysis of results is given.
- In Chapter 6, the final conclusion is given.

Chapter 2

Governing equations and numerical method

The objective of this thesis is to investigate the natural phenomena of wind erosion through numerical approaches. The turbulent boundary layer flow is simulated by large eddy simulation (LES) using the Advanced Regional Prediction System (ARPS) code, primarily developed by the Center for Analysis and Prediction of Storms (CAPS) at the university of Oklahoma. This numerical model resolves the compressible Navier-Stokes equations in a generalized terrain-following coordinate system, which describes a non-hydrostatic atmospheric turbulent flow.

In the context of this thesis, we only focus in this chapter on the theoretical and numerical description of the dynamic equations used in ARPS. In Section 2.1, the governing equations of atmospheric flows are deduced from the general compressible Navier-Stokes equations using the Boussinesq approximation; In Section 2.2, LES is described, including the filtered governing equations, subgrid-scale modeling and near-wall treatment; In Section 2.3, numerical details such as the discretization, the parallelization and the vertical grid stretching, are introduced; In Section 2.4, the initialization, different boundary conditions and wall modeling, are briefly presented.

2.1 Governing equations

The mechanical problem can be described in different scales: the quantum mechanics is used to study the microscopic scale (molecular scale), the motion of a rigid structure or body is figured out in macroscopic scale, and the fluid mechanics is a description in a scale, which is larger than the molecular scale and smaller than the macroscopic scale. The basic theory

of the fluid mechanics can be classified into the category of continuum mechanics, which regards the material as a continuum phase and then deals with the mechanic behaviors and the kinetic motion.

According to the basic concepts of continuum mechanics, the governing equations of the continuum phase are derived from the conservation laws: conservation of mass resulting in the continuity equation, conservation of linear momentum in the momentum equation, conservation of angular momentum in the symmetry property of Cauchy stress tensor, and the first thermodynamic law in the energy equation.

In this section, the general Navier-Stokes equations are introduced and an abridgment is derived after using the Boussinesq approximation. These simplified governing equations enable us to only focus on the temporal evolution of density, pressure, temperature variations, due to the fact that the real density, pressure and temperature are close to the hydrostatic state.

2.1.1 Navier-Stokes equations

Considering a fluid particle in a Cartesian coordinate system, according to the conservation of mass, momentum and energy from the basic theory of continuum mechanics and the thermodynamic properties of the fluid phase, the general Navier-Stokes equations are given by:

$$\text{Conservation of mass: } \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = 0, \tag{2.1a}$$

$$\text{Conservation of linear momentum: } \rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = \frac{\partial \sigma_{ji}}{\partial x_j} + \rho f_i, \tag{2.1b}$$

$$\text{Conservation of angular momentum: } \sigma_{ij} = \sigma_{ji}, \tag{2.1c}$$

$$\text{Conservation of energy: } \rho \frac{\partial e}{\partial t} + \rho u_j \frac{\partial e}{\partial x_j} = s_{ij}\sigma_{ji} + \frac{\partial q_j}{\partial x_j} + \rho r, \tag{2.1d}$$

$$\text{Equation of state for a perfect gas: } p = \rho R T, \tag{2.1e}$$

where ρ is the fluid density, x_i the fluid position, u_i the fluid velocity in $i-$th direction, σ_{ij} the Cauchy stress tensor, f_i the body force, e the internal energy, $s_{ij} = (\partial u_i/\partial x_j + \partial u_j/\partial x_i)/2$ the strain tensor, q_i the heat flux, r the heat radiation, R the gas constant and T the temperature, respectively. For an atmospheric flow, the body force contains the gravity due to the attraction of the earth, and the Coriolis force due to the rotation of the earth, *i.e.*, $f_i = -g\delta_{i3} - 2\varepsilon_{ijk}\Omega_j u_k$, where g is the gravitational acceleration and Ω_k is the angular velocity. Here, δ_{ij} denotes the

Kronecker delta tensor and ε_{ijk} is the Levi-Civita symbol. Equations (2.1) are not closed and σ_{ij}, q_i and r have to be modeled.

Generally, the Cauchy stress tensor is split up into two terms:

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij}, \tag{2.2}$$

where p is the hydrostatic pressure, and τ_{ij} the deviatoric stress tensor, respectively. Stokes proposed a linear constructive law for the deviatoric stress tensor τ_{ij}, based on three assumptions: i), τ_{ij} is a linear function of the strain rate tensor s_{ij} or equivalently the velocity gradient; ii), the fluid is isotropic; iii), for a fluid at rest, $\partial\tau_{ji}/\partial x_j$ must be zero. The formula proposed for a Newtonian fluid is given by:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \lambda\frac{\partial u_k}{\partial x_k}\delta_{ij} = 2\mu s_{ij} + \lambda s_{kk}\delta_{ij}, \tag{2.3}$$

where μ is the first coefficient of viscosity or shear viscosity, and λ is second coefficient of viscosity or volume viscosity. Commonly, we take $\lambda = -\frac{2}{3}\mu$. As a consequence, the momentum conservation equation (Equation (2.1b)) is rewritten as:

$$\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial}{\partial x_i}\left(p + \frac{2}{3}\mu s_{kk}\right) + \frac{1}{\rho}\frac{\partial}{\partial x_j}\left(2\mu s_{ji}\right) - g\delta_{i3} - 2\varepsilon_{ijk}\Omega_j u_k. \tag{2.4}$$

Since $e = h - p/\rho$ where h is the enthalpy of the fluid, an alternative form of the energy equation (Equation (2.1d)) is obtained as:

$$\rho\frac{\partial h}{\partial t} + \rho u_j\frac{\partial h}{\partial x_j} = \rho\frac{D}{Dt}\left(\frac{p}{\rho}\right) + s_{ij}\sigma_{ji} + \frac{\partial q_j}{\partial x_j} + \rho r, \tag{2.5}$$

where $D\cdot/Dt$ denotes the material derivative, *i.e.*, $D\cdot/Dt = \partial\cdot/\partial t + u_j\partial\cdot/\partial x_j$, and r is the thermal radiation. Note that the material derivative is a Lagrangian derivative following the motion of the fluid particle.

For the ideal gas, the enthalpy and the heat flux can be expressed as a function of the temperature, *i.e.*, $h = c_pT$, $q_i = \lambda_T\partial T/\partial x_i$, where c_p is the specific heat at constant pressure, λ_T the thermal diffusivity, and T the temperature, respectively. Substituting the law of Cauchy strain tensor (Equations (2.2) and (2.3)) into Equation (2.5), we have in the adiabatic case ($r = 0$) by introducing the continuity equation (Equation (2.1a)):

$$\frac{\partial T}{\partial t} + u_j\frac{\partial T}{\partial x_j} = \frac{1}{c_p\rho}\left(\frac{Dp}{Dt} + 2\mu\left(s_{ij}s_{ji} - \frac{1}{3}s_{ii}s_{kk}\right) + \frac{\partial}{\partial x_j}\left(\lambda_T\frac{\partial T}{\partial x_j}\right)\right). \tag{2.6}$$

Defining the potential temperature $\Theta = T\,(p_0/p)^{(\gamma-1)/\gamma}$ with the coefficient $\gamma = c_p/(c_p - R)$ and p_0 the reference level pressure, we obtain $\frac{1}{\Theta}\frac{D\Theta}{Dt} = \frac{1}{T}\frac{DT}{Dt} - \frac{\gamma-1}{\gamma}\frac{1}{p}\frac{Dp}{Dt}$ by taking the material derivative. Due to the state equation (Equation (2.1e)), we have $\frac{\gamma-1}{\gamma}\frac{1}{p} = \frac{1}{c_p\rho T}$. Therefore, Equation (2.6) can be rewritten as:

$$\frac{\partial\Theta}{\partial t} + u_j\frac{\partial\Theta}{\partial x_j} = \frac{\Theta}{c_p\rho T}\left(2\mu(s_{ij}s_{ji} - \frac{1}{3}s_{ii}s_{kk}) + \frac{\partial}{\partial x_j}\left(\lambda_T\frac{\partial T}{\partial x_j}\right)\right). \tag{2.7}$$

For the subsonic or non-strongly heated flow, we can neglect the viscous dissipation, Equation (2.7) can be approximated by defining the molecular conductivity $\kappa_T = \lambda_T/(\rho c_p)$:

$$\frac{\partial\Theta}{\partial t} + u_j\frac{\partial\Theta}{\partial x_j} = \kappa_T\frac{\Theta}{T}\frac{\partial^2 T}{\partial x_j\partial x_j}. \tag{2.8}$$

Note that the Prandtl number is defined as the ratio of momentum diffusivity to thermal diffusivity, *i.e.*, $Pr = \nu/\kappa_T$ where the molecular viscosity is defined by $\nu = \mu/\rho$. In fact, Pr is a dimensionless number, which indicates the relative significance of the thermal conduction to the fluid convection. The hydrodynamic convection dominates if $Pr \gg 1$ whereas the thermal conduction is dominant if $Pr \ll 1$. Generally, the Prandtl number takes a constant value for a fluid at a certain state. For example, $Pr \approx 0.7$ in the air at the ambient temperature, which means that the heat diffusion speed is nearly equal to the fluid momentum diffusion's.

2.1.2 Boussinesq approximation

Assuming that the thermodynamic physical variables are considered as a combination of a time-invariant base state and a deviation, thus a decomposition can be written as:

$$p(x_i,t) = p_r(x_i) + \Delta p(x_i,t), \tag{2.9a}$$

$$\rho(x_i,t) = \rho_r(x_i) + \Delta\rho(x_i,t), \tag{2.9b}$$

$$\Theta(x_i,t) = \Theta_r(x_i) + \Delta\Theta(x_i,t), \tag{2.9c}$$

where p_r, ρ_r, Θ_r denote the fluid pressure, the density, the potential temperature of the base state, and Δp, $\Delta\rho$, $\Delta\Theta$ the corresponding variations, respectively. The Boussinesq approximation assumes that the ratio of variations to base states is small, *i.e.*, $\Delta p \ll p_r$, $\Delta\rho \ll \rho_r$ and $\Delta\Theta \ll \Theta_r$.

The base state is governed by three equations: *i*), the hydrostatic equation; *ii*), the state equation; *iii*), an equation specifying the atmosphere condition. Firstly, the base state should

satisfy a hydrostatic equilibrium equation:

$$\frac{1}{\rho_r}\frac{\partial p_r}{\partial x_i} = -g\delta_{i3}. \tag{2.10}$$

It indicates that the base state is horizontally homogeneous. Secondly, it should also satisfy the equation of state for the base state (Equation (2.1e) for the perfect gas), *i.e.*, $p_r = \rho_r R T_r$ where the base state temperature is given by $T_r = \Theta_r \left(p_0/p_r\right)^{(1-\gamma)/\gamma}$. Thirdly, the atmospheric condition should be specified, such as, an isentropic atmosphere, an isothermal atmosphere, an atmosphere with constant static stability, *etc.*.

When the base state is specified, the Navier-Stokes equations are transformed and simplified. In the momentum equation (Equation (2.1b)), the term of pressure gradient force becomes by taking the linear approximation:

$$\begin{aligned}\frac{1}{\rho}\frac{\partial p}{\partial x_i} &= \frac{1}{\rho_r+\Delta\rho}\left(\frac{\partial p_r}{\partial x_i}+\frac{\partial(\Delta p)}{\partial x_i}\right) = \left(\frac{1}{\rho_r}-\frac{\Delta\rho}{\rho_r^2}+O((\Delta\rho)^2)\right)\left(\frac{\partial p_r}{\partial x_i}+\frac{\partial(\Delta p)}{\partial x_i}\right)\\ &= \left(1-\frac{\Delta\rho}{\rho_r}\right)\frac{1}{\rho_r}\frac{\partial p_r}{\partial x_i}+\frac{1}{\rho_r}\frac{\partial(\Delta p)}{\partial x_i}+O(\Delta\rho).\end{aligned} \tag{2.11}$$

Then following the definition of the potential temperature $\Theta = T\left(p_0/p\right)^{(\gamma-1)/\gamma}$, Equation (2.1e) can be rewritten as:

$$\rho = p_0^{(\gamma-1)/\gamma}\frac{p^{1/\gamma}}{R\Theta} = p_0^{R/c_p}\frac{p^{(1-R/c_p)}}{R\Theta}, \tag{2.12}$$

which yields an approximation of the ratio of the density variation to the base state after the linearization:

$$\begin{aligned}\frac{\Delta\rho}{\rho_r} &= \frac{\rho}{\rho_r}-1 = \frac{(1+\Delta p/\rho_r)^{(1-R/c_p)}}{1+\Delta\Theta/\Theta_r}-1 = (1-R/C_p)\frac{\Delta p}{p_r}-\frac{\Delta\Theta}{\Theta_r}+O(\Delta p,\Delta\rho)\\ &= \frac{\Delta p}{\rho_r\gamma R T_r}-\frac{\Delta\Theta}{\Theta_r}+O(\Delta p,\Delta\rho) = \frac{\Delta p}{\rho_r c_s^2}-\frac{\Delta\Theta}{\Theta_r}+O(\Delta p,\Delta\rho),\end{aligned} \tag{2.13}$$

with the acoustic wave speed defined as $c_s = \sqrt{\gamma R T_r}$. Applying Equation (2.9), substituting Equation (2.11) into Equation (2.4) and then using Equation (2.13), a simplified form of momentum equation is obtained after the linear approximation:

$$\begin{aligned}\frac{\partial u_i}{\partial t}+u_j\frac{\partial u_i}{\partial x_j} &= -\frac{1}{\rho_r}\frac{\partial}{\partial x_i}\left(\Delta p+\frac{2}{3}\mu s_{kk}\right)+\frac{1}{\rho_r}\frac{\partial}{\partial x_j}\left(2\mu s_{ji}\right)\\ &\quad -g\delta_{i3}\left(\frac{\Delta p}{\rho_r c_s^2}-\frac{\Delta\Theta}{\Theta_r}\right)-2\varepsilon_{ijk}\Omega_j u_k.\end{aligned} \tag{2.14}$$

Taking the material derivative of Equation (2.12) and then replacing the material derivative of density $\rho^{-1}D\rho/Dt$ by the divergence of fluid velocity $\partial u_j/\partial x_j$ using the continuity equation (Equation (2.1a)), it becomes:

$$\frac{1}{\gamma p}\frac{Dp}{Dt} = \frac{1}{\Theta}\frac{D\Theta}{Dt} - \frac{\partial u_j}{\partial x_j}. \tag{2.15}$$

Using the decomposition of a variable into a base state and a derivation (Equation (2.9)), the material derivative of pressure can be simplified as:

$$\begin{aligned}\frac{1}{\gamma p}\frac{Dp}{Dt} &= \frac{1}{\gamma(p_r+\Delta p)}\left(\frac{Dp_r}{Dt}+\frac{D(\Delta p)}{Dt}\right) = \frac{1}{\gamma p_r}\left(\frac{Dp_r}{Dt}+\frac{D(\Delta p)}{Dt}\right)+O(\Delta p)\\ &= \frac{1}{\rho_r c_s^2}\left(\frac{\partial(\Delta p)}{\partial t}+u_j\frac{\partial(\Delta p)}{\partial x_j}+u_j\frac{\partial p_r}{\partial x_j}\right)+O(\Delta p).\end{aligned} \tag{2.16}$$

Substituting Equation (2.16) into Equation (2.15) and then using Equation(2.10), the dynamic equation of pressure variation is obtained:

$$\frac{\partial(\Delta p)}{\partial t}+u_j\frac{\partial(\Delta p)}{\partial x_j} = \rho_r g u_j\delta_{j3} - \rho_r c_s^2\frac{\partial u_j}{\partial x_j} + \rho_r c_s^2\frac{1}{\Theta}\frac{D\Theta}{Dt}. \tag{2.17}$$

The divergence terms are usually the dominant terms for most meteological applications. The diabatic term is usually small, and is therefore neglected, then Equation (2.17) becomes:

$$\frac{\partial(\Delta p)}{\partial t}+u_j\frac{\partial(\Delta p)}{\partial x_j} = \rho_r g u_j\delta_{j3} - \rho_r c_s^2\frac{\partial u_j}{\partial x_j}. \tag{2.18}$$

To our knowledge, the pressure variations are of the order of M^2 (M the Mach number), that is $\Delta p/p_r \sim M^2$. At low Mach number ($M < 0.3$), after the Boussinesq approximation, the hydrostatic pressure is dominant, and the relative pressures from the base state are of lower order of $\Delta\rho/\rho_r$, *i.e.*, $\Delta p/p_r \ll \Delta\rho/\rho_r$ [Lesieur (1987)]. In this case, the approximation states:

$$\frac{\Delta\Theta}{\Theta_r} \approx \frac{\Delta T}{T_r} \approx -\frac{\Delta\rho}{\rho_r}. \tag{2.19}$$

Since $\Theta_0 = T_0$ from the definition $\Theta = T\,(p_0/p)^{(\gamma-1)/\gamma}$, we can take the approximation $\Delta T = \Delta\Theta$ at the lowest order. Neglecting the diffusion term of the base state temperature $\frac{\partial^2 T_r}{\partial x_k \partial x_k}$ and taking the lowest approximation $\frac{\Theta}{T}\frac{\partial^2(\Delta T)}{\partial x_k \partial x_k} \approx \frac{\partial^2(\Delta\Theta)}{\partial x_k \partial x_k}$, the deviation of potential

temperature satisfies from Equation (2.8):

$$\frac{\partial(\Delta\Theta)}{\partial t}+u_j\frac{\partial(\Delta\Theta)}{\partial x_j}=-u_j\frac{\partial\Theta_r}{\partial x_j}+\kappa_T\frac{\partial^2(\Delta\Theta)}{\partial x_j\partial x_j}. \tag{2.20}$$

In conclusion, combining Equations (2.14), (2.18), and (2.20), the governing equations after using the Boussinesq approximation are obtained:

$$\frac{\partial(\Delta p)}{\partial t}+u_j\frac{\partial(\Delta p)}{\partial x_j}=\rho_r g u_j\delta_{j3}-\rho_r c_s^2\frac{\partial u_j}{\partial x_j}, \tag{2.21a}$$

$$\begin{aligned}\frac{\partial u_i}{\partial t}+u_j\frac{\partial u_i}{\partial x_j}=&-\frac{1}{\rho_r}\frac{\partial}{\partial x_i}\left(\Delta p+\frac{2}{3}\mu s_{kk}\right)+\frac{1}{\rho}\frac{\partial}{\partial x_j}\left(2\mu s_{ji}\right)\\&-g\delta_{i3}\left(\frac{\Delta p}{\rho_r c_s^2}-\frac{\Delta\Theta}{\Theta_r}\right)-2\varepsilon_{ijk}\Omega_j u_k,\end{aligned} \tag{2.21b}$$

$$\frac{\partial(\Delta\Theta)}{\partial t}+u_j\frac{\partial(\Delta\Theta)}{\partial x_j}=-u_j\frac{\partial\Theta_r}{\partial x_j}+\kappa_T\frac{\partial^2(\Delta\Theta)}{\partial x_j\partial x_j}. \tag{2.21c}$$

We remark that Boussinesq approximation replaces the real pressure, density, temperature by their corresponding variations through using the hydrostatic equation, and then reduces the number of governing equations from six to five. This is helpful to save the computational resources and to improve the computational efficiency.

2.2 Large eddy simulation

Large eddy simulation (LES) is becoming an attractive practical numerical approach in the community of computational fluid dynamics (CFD). The basic concept of LES is that the large scales of the fluid velocity are directly resolved by the numerical method, while the small scales are modeled by subgrid-scale (SGS) models. Hence, LES takes less computing cost than the direct simulation (DNS) and captures more information on the large scale structures than the Reynolds average numerical simulation (RANS).

Similar to the RANS closure for the Reynolds stress, SGS modeling is required for the closure of the subgrid stress in LES. In isotropic turbulence, it is acceptable for most of the researchers that the small scales probably obey to the universal law, *e.g.*, $-5/3$ scaling law of energy spectrum in the inertial range in isotropic turbulence, whereas the large scales are dependent on the flow characteristics. Therefore, SGS models seem universal and independent on the flow complexity.

When LES is used for the simulation of turbulent boundary flows, near-wall resolution is important for the accuracy of simulated results. Commonly, LES methods are categorized

into two approaches by whether taking into account the near-wall resolution or not: LES with near-wall resolution (LES-NWR) and LES with near-wall modeling (LES-NWM). In practice, LES-NWM is more useful in the simulation of complex flows, since it does not require the high resolution in the near-wall region. When LES-NWM is applied in numerical simulations, a wall model is usually used to specify the near-wall stresses.

In this section, we firstly deduce the filtered governing equations of LES, and then introduce the subgrid-scale modeling and the near-wall treatment for LES-NWM. In particular, we detail the SGS models, i.e., Smagorinsky's model and the model based on the subgrid kinetic energy, which are implanted in ARPS.

2.2.1 Filtered formulation

Since we have $\Delta\rho/\rho_r \ll 1$ from the Boussinesq approximation, taking the approximation at the lowest order, the continuity equation (Equation (2.1a)) reduces to:

$$\frac{\partial u_i}{\partial x_i} = s_{ii} = 0. \tag{2.22}$$

It implies that the incompressibility assumption can be obtained when the Boussinesq approximation is applied.

Considering a filter $\tilde{G}(x_i,\Delta)$ with the filtered scale Δ, the filtered part of an arbitrary function $f(x_i,t)$ is defined as:

$$\tilde{f}(x_i,t) = \int_{\mathscr{R}} \tilde{G}(x_i-\xi_i,\Delta) f(\xi_i,t) d\xi_i, \tag{2.23}$$

with $\mathscr{R}$ denoting the control volume. Thus, a physical variable can be decomposed into the large-scale part and the small-scale one. For example, the decomposition of the fluid velocity is given by:

$$u_i = \tilde{u}_i + u_i'', \tag{2.24}$$

where $\tilde{u}_i$ is the filtered velocity, and u_i'' is the fluctuated velocity of scales smaller than the filtered scale Δ.

Using the incompressibility assumption (Equation (2.22)) and applying the filter on the simplified governing equations (Equations (2.21)), the filtered formulation are obtained as:

$$\frac{\partial(\Delta\tilde{p})}{\partial t} + \tilde{u}_j \frac{\partial(\Delta\tilde{p})}{\partial x_j} = \rho_r g \tilde{u}_j \delta_{j3} - \rho_r c_s^2 \frac{\partial \tilde{u}_j}{\partial x_j}, \tag{2.25a}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j} = -\frac{1}{\rho_r}\frac{\partial(\Delta\tilde{p})}{\partial x_i} + \frac{1}{\rho_r}\frac{\partial}{\partial x_j}\left(2\mu\tilde{s}_{ji}\right) + \frac{\partial \tilde{\sigma}_{ji}}{\partial x_j} - g\delta_{i3}\left(\frac{\Delta\tilde{p}}{\rho_r c_s^2} - \frac{\Delta\tilde{\Theta}}{\Theta_r}\right) - 2\varepsilon_{ijk}\Omega_j\tilde{u}_k, \tag{2.25b}$$

$$\frac{\partial(\Delta\tilde{\Theta})}{\partial t} + \tilde{u}_j \frac{\partial(\Delta\tilde{\Theta})}{\partial x_j} = -\tilde{u}_j \frac{\partial \Theta_r}{\partial x_j} + \kappa_T \frac{\partial^2(\Delta\tilde{\Theta})}{\partial x_j \partial x_j} + \frac{\partial \tilde{q}_{j\Theta}}{\partial x_j}, \tag{2.25c}$$

in which $\tilde{\sigma}_{ij}$ is the subgrid tensor and $\tilde{q}_{j\Theta}$ the subgrid flux. They are given by:

$$\tilde{\sigma}_{ij} = \tilde{u}_i\tilde{u}_j - \widetilde{u_i u_j}, \tag{2.26a}$$

$$\tilde{q}_{j\Theta} = (\Delta\tilde{\Theta})\tilde{u}_j - \widetilde{(\Delta\Theta)u_j}. \tag{2.26b}$$

It should be noted that Equations (2.25) are not closed since the subgrid terms $\tilde{\sigma}_{ij}$ and $\tilde{q}_{j\Theta}$ are unknown. Commonly, $\tilde{\sigma}_{ij}$ and $\tilde{q}_{j\Theta}$ are modeled by a SGS model, which are introduced in detail in Subsection 2.2.2.

2.2.2 Subgrid-scale modeling

According to the energy cascade theory in turbulence, the energetic action of the subgrid scales on the resolved scales is assumed and the mechanism of energy transfer from the resolved scales to subgrid scales is supposed to be similar to the mechanism of energy dissipation from the subgrid scales to the molecular scales, Hence, the eddy-viscosity assumption reads:

$$\tilde{\tau}_{ij} = \tilde{\sigma}_{ij} - \frac{1}{3}\tilde{\sigma}_{kk}\delta_{ij} = 2\nu_{sgs}\left(\tilde{s}_{ij} - \frac{1}{3}\tilde{s}_{kk}\delta_{ij}\right), \tag{2.27}$$

where ν_{sgs} is the subgrid viscosity, $\tilde{\tau}_{ij}$ is the deviatoric part of $\tilde{\sigma}_{ij}$ and $\tilde{s}_{ij}$ the filtered strain tensor. Equation (2.27) takes the linear relation between the subgrid tensor $\tilde{\sigma}_{ij}$ and the filtered strain tensor $\tilde{s}_{ij}$ with a modeled coefficient ν_{sgs}. According to the local equilibrium hypothesis, an equilibrium state of the energy transfer from larger scales and the dissipation to smaller scales is achieved. In the spectral view, this means that there is no energy accumulation in the inertial range, and that the energy spectrum in high wave-numbers keeps time invariant. This enables us to express ν_{sgs} by the local resolved scales or subgrid scales, rather than by the statistically average physical properties.

In the following, we introduce in detail Smagorinsky's model and the model based on the subgrid-scale kinetic energy, which are implanted in ARPS.

Smagorinsky's model: This model is based on the mixed-length hypothesis [Smagorinsky (1963)]. It is widely used in numerical simulations of research and industrial domain, due to its simple form and simplicity to be implanted. Based on the mixed-length hypothesis, an eddy viscosity is given by:

$$\nu_{sgs} = l_S^2 \widetilde{S} = (C_S \Delta)^2 \widetilde{S}, \tag{2.28}$$

where $\widetilde{S} = \sqrt{2\tilde{s}_{ij}\tilde{s}_{ji}}$ is the characteristic filtered rate of strain, l_S the Smagorinsky length and C_S the smagorinsky coefficient, respectively. In incompressible isotropic turbulence, it is assumed that the size of filtered scale Δ is located in the equilibrium inertial range, where the energy spectrum satisfies the $-5/3$ scaling law, *i.e.*, $\hat{E}(\hat{k},t) \sim \tilde{\varepsilon}^{2/3}\hat{k}^{-5/3}$ where $\hat{k}$ is the wavenumber and $\tilde{\varepsilon}$ denotes the dissipation rate of filtered motions. In the limit of infinite Reynold number, the inertial range is theoretically supposed to be infinite and the energy spectrum reads $\hat{E}(\hat{k},t) = C_K \tilde{\varepsilon}^{2/3}\hat{k}^{-5/3}$ where C_K is the Kolmogorov constant, it is then obtained:

$$\langle 2\tilde{s}_{ij}\tilde{s}_{ji}\rangle = \int_0^{\pi/\Delta} 2\hat{k}^2 \hat{E}(\hat{k},t) d\hat{k} = \int_0^{\pi/\Delta} 2C_K \tilde{\varepsilon}^{2/3}\hat{k}^{1/3} d\hat{k} = \frac{3}{2}\pi^{4/3} C_K \tilde{\varepsilon}^{2/3}\Delta^{-4/3}. \tag{2.29}$$

Setting the ensemble averaged subgrid-scale kinetic energy dissipation identical to $\tilde{\varepsilon}$, that is, $\tilde{\varepsilon} = \langle \tilde{\tau}_{ij}\tilde{s}_{ji}\rangle$, then eliminating $\tilde{\tau}_{ij}$ using Equations (2.27) and (2.28), an approximated formula of $\tilde{\varepsilon}$ is obtained:

$$\tilde{\varepsilon} \approx \langle (C_S\Delta)^2 \sqrt{2\tilde{s}_{ij}\tilde{s}_{ji}}\rangle \langle 2\tilde{s}_{ij}\tilde{s}_{ji}\rangle \approx (C_S\Delta)^2 \langle 2\tilde{s}_{ij}\tilde{s}_{ji}\rangle^{3/2}. \tag{2.30}$$

Substituting Equation (2.29) into Equation (2.30) results in:

$$C_S \approx \frac{1}{\pi}\left(\frac{2}{3C_K}\right)^{3/4}. \tag{2.31}$$

Theoretically, $C_K = 1.4$ yields $C_S = 0.18$. In practice, C_S is usually between 0.10 and 0.20: 0.10 [Deardorff (1970)], 0.15 [Pope and Pope (2000)], 0.17 [Berselli et al. (2005)], 0.20 [Clark et al. (1979)]. Most researchers actually prefer $C_S = 0.10$, which is consistent with the experimental value $C_S \in [0.10, 0.12]$ [Meneveau (1994); O'Neil and Meneveau (1997)]. In fact, the Smagorinsky model gives a good performance in the simulation of isotropic turbulence [Clark et al. (1979)]. But it fails to simulate the quasi two-dimensional flow or to capture the coherent structures near the wall in the simulations of boundary layers due its too dissipative property.

Some drawbacks of Smagorinsky's model are summarized as follows:

- It is a time-irreversible model, which may result in the failure of capturing the energy backscatter phenomena for complex turbulence simulation;

- Against the limiting consistency, ν_{sgs} does not vanish when the flow is transited into a laminar regime;

- It is not a universal model. The Smagorinsky coefficient C_S is not a single constant for different turbulent flows. For example. $C_S \sim (z^+)^3$ in the near-wall region rather than a limited value, where z^+ is the normal distance from the wall normalized by the viscous lengthscale, hence, the Smagorinsky model is too dissipative for the resolution of near-wall dynamics .

Actually, in the simulations of complex flows, many corrections have been proposed to improve the Smagorinsky's model, *i.e.*, Moin and Kim (1982) proposed a damping function to correct the Smagorinsky coefficient C_S, Germano et al. (1991) developed an approach to dynamically adjust C_S.

Model based on the subgrid-scale kinetic energy transport equation: By introducing the subgrid-scale information, Deardorff (1980) proposed a SGS model where the eddy viscosity is related to the subgrid-scale kinetic energy:

$$\nu_{sgs} = C_M \Delta \sqrt{k_{sgs}}, \tag{2.32}$$

where C_M is the modeled coefficient, and $k_{sgs} = \widetilde{u_i'' u_i''}/2$ is the subgrid-scale kinetic energy. Here, u_i'' denotes the subgrid-scale fluctuation, *i.e.*, $u_i'' = u_i - \tilde{u}_i$ from Equation (2.24). In general, k_{sgs} is governed by the modeled subgrid-scale kinetic energy transport equation:

$$\frac{\partial k_{sgs}}{\partial t} + \tilde{u}_j \frac{\partial k_{sgs}}{\partial x_j} = -\tilde{\tau}_{ij}\tilde{s}_{ji} - C_1 \frac{k_{sgs}\sqrt{k_{sgs}}}{\Delta} + C_2 \frac{\partial}{\partial x_j}\left(\Delta\sqrt{k_{sgs}})\frac{\partial k_{sgs}}{\partial x_j}\right) + \nu \frac{\partial^2 k_{sgs}}{\partial x_j \partial x_j} \tag{2.33}$$

with $C_1 = 1.0$ and $C_2 = 0.1$ proposed by Yoshizawa (1982), Yoshizawa and Horiuti (1985) and Horiuti (1985). In practical simulation, to take into account the near-wall effect, Deardorff (1980) and Moeng (1984) proposed $C_1 = 3.9$ at the first grid near the wall, and $C_1 = 0.93$ at the others. In Equation (2.33), the different terms are respectively the temporal evolution term, the convection by the resolved modes, production by the resolved modes, turbulent dissipation, and viscous dissipation.

In isotropic turbulence of high Reynold numbers, assuming the equilibrium energy spectrum $\hat{E}(\hat{k},t) = C_K \tilde{\varepsilon}^{2/3}\hat{k}^{-5/3}$ and Δ in the inertial range, the subgrid-scale kinetic energy

is given by:

$$k_{sgs} = \int_{\pi/\Delta}^{\infty} \hat{E}(\hat{k},t)d\hat{k} = \int_{\pi/\Delta}^{\infty} C_K \tilde{\varepsilon}^{2/3}\hat{k}^{-5/3}d\hat{k} = \frac{3}{2}\pi^{-2/3}C_K\varepsilon^{2/3}\Delta^{2/3}. \tag{2.34}$$

Substituting Equation (2.34) into Equation (2.32), and then compared with the basic SGS model $\nu_{sgs} = \frac{A}{C_K\pi^{4/3}}\varepsilon^{1/3}\Delta^{4/3}$ deduced from dimensional analysis, the coefficient C_M is resolved as:

$$C_M = \frac{A}{\pi}\left(\frac{2}{3C_K^3}\right)^{1/2}. \tag{2.35}$$

Aupoix and Cousteix (1982) found $A = 0.438$ using the two-fluid model (TFM) and $A = 0.441$ using the eddy-damping quasi-normal model (EDQNM). Theoretically, $C_K = 1.4$ leads to $C_M \approx 0.069$. In practice, Moeng and Wyngaard (1989) proposed $C_M = 0.1$ for the closure.

Similarly, the subgrid flux $\tilde{q}_{j\Theta}$ in Equation (2.25c) can be modeled based on the eddy-diffusivity assumption:

$$\tilde{q}_{j\Theta} = \kappa_{sgs}\frac{\partial\tilde{\Theta}}{\partial x_j} \tag{2.36}$$

where κ_{sgs} is the subgrid diffusivity. Equation (2.36) uses a linear relation between the subgrid flux and the filtered heat flux with a modeled coefficient κ_{sgs}. Generally, the subgrid diffusivity κ_{sgs} is related to the subgrid viscosity by:

$$\kappa_{sgs} = \frac{\nu_{sgs}}{Pr_{sgs}}, \tag{2.37}$$

where Pr_{sgs} is the modeled turbulent Prandtl number. In ARPS, Pr_{sgs} is evaluated by [Deardorff (1980)]:

$$Pr_{sgs} = \left(1 + 1.52\frac{\sqrt{k_{sgs}}}{\Delta}\left|\frac{g}{\Theta_r}\frac{\partial\Theta_r}{\partial z}\right|^{1/2}\right)^{-1}, \tag{2.38}$$

where k_{sgs} is the subgrid-scale kinetic energy evaluated by Equation (2.33), g is the gravity and Θ_r the potential temperature of base state.

When an anisotropic filter is used, Deardorff (1970) proposed in the Cartesian case:

$$\Delta = (\Delta x_1 \Delta x_2 \Delta x_3)^{1/3} \tag{2.39}$$

where Δx_i is the filter cutoff length in the $i-$th direction. In ARPS, the filtered size is different

in horizontal and vertical direction, which yields:

$$\Delta = \begin{cases} \sqrt{\Delta x \Delta y}, & \text{in the horizontal directions;} \\ \Delta z, & \text{in the vertical direction,} \end{cases} \tag{2.40}$$

where Δx and Δy are the filter cutoff length in horizontal directions, and Δz is the filter cutoff length in the vertical direction.

2.2.3 Near-wall treatment

In recent years, many researches found coherent structures in boundary layer flows, specially, organized large scale structures – streaks formed by the fast fluid action (ejections) and the slow fluid effect (sweeps). In the view of kinetic energy dynamics, it is found that sweeps are related to the backward energy cascade) and ejections to the forward energy cascade. Therefore, the near-wall resolution is important in the simulation of turbulent boundary layer flows.

According to whether the flow field in the viscous wall layer is resolved or not, LES is classified into two categories: LES-NWR called large-eddy simulation with near-wall resolution, and LES-NWM called large-eddy simulation with near-wall modeling. When LES-NWM is used, special treatments are required in order to capture the structures in the near-wall region. Supposing Δz_{min} the grid spacing near the wall and u_* the friction velocity, the dimensionless value $\Delta z^+_{min} = u_* \Delta z_{min}/\nu$ is used as the quantitative indicator for this classification.

Near-wall dynamics resolving ($\Delta z^+_{min} < 2$): If we want to investigate the kinetic energy mechanism, the fluid field in the near wall region should be resolved. In numerical simulation, a sufficiently fine grid should be used, *i.e.*, the first grid point should be imposed deeply in the viscous sublayer $\Delta z^+_{min} < 1$ and the non-slip boundary condition is imposed on the solid wall. Numerical tests indicates that $\Delta z^+_{min} < 2$ can get the ability of estimating the near wall dynamics and capturing the coherent structures. However, the restriction of $\Delta z^+_{min} < 2$ takes a lot of computing cost.

In addition, the near-wall dynamics resolving for rough wall is very difficult, because the exact boundary condition on a rough wall is hard to be specified theoretically.

Near-wall dynamics modeling ($20 < \Delta z^+_{min} < 200$): To reduce the computing costs, a coarse mesh in the near-wall region is usually chosen. Commonly, the first mesh is out of the viscous layer and is usually near or located in the logarithmic inertial layer, *i.e.*,

$20 < \Delta z^{+}_{min} < 200$. Hence, the near-wall dynamics should be modeled rather than resolved. This special model used for representing the near wall dynamics is called the wall model. In practice, the simulated physical quantities are averaged at the first grid, which then results in the inaccurate value of wall stresses. Therefore, the wall model should specify the wall stresses ($\tau_{13}|_w$ and $\tau_{23}|_w$) and the wall-normal velocity component.

In the numerical simulations of wall flows with high Reynold numbers, a very fine grid near wall is required to achieve the resolution of the viscous sublayer. Therefore, to reduce the computational cost, LES-NWM is usually used in the simulation of atmospheric boundary layer. A free-slip condition with imposing the wall models is usually used to specify the velocity field on the wall.

2.3 Numerical method

In this thesis, the LES code ARPS is used to simulate the turbulent boundary layer flow. The standard version is described in detail in the ARPS user's guide Xue et al. (1995) and in Xue et al. (2000) and Xue et al. (2001). In the recent yeas, this code has been developed to study the characteristics of particle or scalar transport in a turbulent boundary layer. Vinkovic (2005) developed the Lagrangian stochastic model for the dispersion of passive scalars or solid particles, and implemented it in ARPS code. Dupont et al. (2013) parallelized the module of the computation of solid particle motion, and simulated the sand saltation over a flat surface. Huang (2015) improved the near-wall treatment for the rough surface and the technique for generating the inlet boundary condition. After the validation of these improvements, Huang et al. (2018) carried out the numerical simulation of particle transport over Gaussian hills to study the behavior of aeolian erosion and deposition.

In this section, the numerical method used in ARPS is briefly presented, including the numerical discretization, the parallelization, the vertical grid stretching.

2.3.1 Discretization

In the numerical simulation, the finite difference method is applied to discretize the computational time-space domain.

Time discretization: The mode-splitting time integration approach proposed by Klemp and Wilhelmson (1978) is used to eliminate the acoustic waves effect. According to this approach, the time domain is firstly discretized into a set of sub-domains with a large time step, which is then divided into a number of computationally inexpensive small time steps.

Only the acoustically active terms are updated in a small time integration while all the other terms are advanced every large time step. As a consequence, the small time step is limited by the acoustic wave effect.

For the large time integration, a centered three-level (leapfrog) time differencing scheme is used and the time step size is limited by a stability condition based on convective and (optionally) on gravity wave speeds. For the small time step integration, a forward scheme is used in horizontal direction and the Crank-Nicolson scheme in vertical direction. This results in an explicit discrete form of u and v equations and an implicit form of w and p equations. The algorithm implicitly solving the vertical equations is absolutely stable regardless of the acoustic waves. Thus, the small time step size only depends on the horizontal grid spacing.

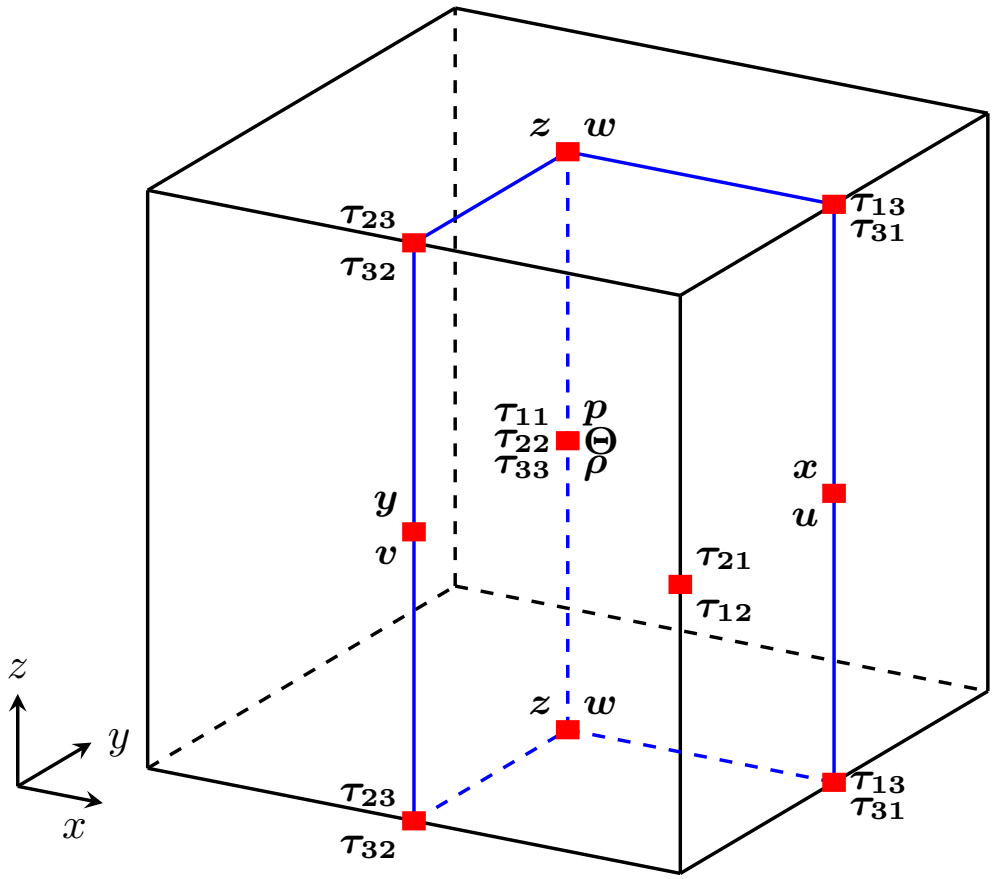


Figure 2.1 Sketch of the Arakawa C-grid and of the variable position.

Spatial discretization: In ARPS, the governing equations are spatially discretized on a staggered grid using finite difference. In a staggered grid, the different physical variables sharing the same grid index (i, j, k) are not located at the same position. Figure 2.1 shows

the location of the physical variables staggered on the Arakawa C-grid [Arakawa (1966)]. The scalars, such as the pressure p, the density ρ, the potential temperature Θ, are defined at the center of the grid cell. The three components of vectors, such as the velocity, are located at the center of the corresponding cell faces, whose normal directions are same with the components. For the shear stress tensor, the diagonal components are defined at the center of the cell and the others are located at the center of cell sides, *e.g.*, $\tau_{13} = \partial w/\partial x$ is evaluated on the Oxz plane, a half a grid interval below the w-point.

Except for the convection terms, a second-order accurate finite difference scheme is used for the spatial discretization of governing equations. The convection terms can be discretized by a second-order or fourth-order accurate finite difference. The discrete formulation of governing equations are given in Appendix B.1.

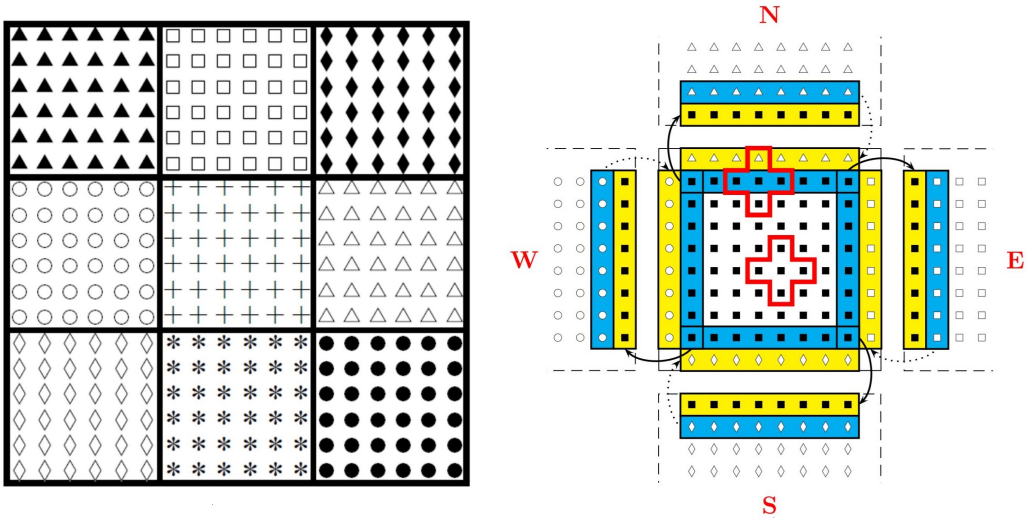


Figure 2.2 Sketch of the domain decomposition (left) and of the message passage between the sub-domains (right). Left figure taken from Xue et al. (1995). Right figure taken from http://www.idris.fr/formations (MPI).

2.3.2 Parallelization

A parallelization strategy based on message passing interface is implemented in ARPS. As shown in Figure 2.2, a large computing domain is divided into several small sub-domains, which are assigned to different processors. The boundary information between the sub-domains is supplied by using a message passage strategy. Each sub-domain sends the boundary information to its four neighbors in the east, south, west, north direction and also receives the information from them for a two-dimensional case. The MPI parallelization improves the computing ability through using many processors. The efficiency depends on

the balance of the computing resources required by each sub-domain and the ratio of the message passing load to the computing load.

In ARPS, the finite difference method used for the discretization facilitates the process of domain decomposition. The domain decomposition is applied only in horizontal directions, due to the implicit scheme employed in the vertical direction. Thanks to the work of S. Dupont at INRA, Bordeaux, the MPI parallelization for the module of the calculation of solid particle motion in the APRS code has been accomplished.

2.3.3 Terrain-following mesh and vertical grid stretching

In ARPS, a vertical coordinate transformation is applied to convert the computational domain with an irregular geometry to a regular, rectangular one. A general vertical coordinate transformation is given by:

$$\begin{aligned} \zeta &= x, \\ \eta &= y, \\ \xi &= \xi(x,y,z), \end{aligned} \tag{2.41}$$

where (x,y,z) are the Cartesian coordinates, and (ζ,η,ξ) are the transformed curvilinear coordinate (or the computational coordinate).

Two steps are required for the vertical coordinate transformation: *i)* the terrain-following coordinate transformation $\xi' = \xi'(x,y,z)$; ii), the vertical grid stretching $\xi = \xi(\xi')$. Firstly, the terrain-following coordinate transformation is given by:

$$\xi' = \begin{cases} (z_{flat} - z_{min})\dfrac{z - h(x,y)}{z_{flat} - h(x,y)} + z_{min}, & \text{if } z_{min} \le z \le z_{flat}, \\ z, & \text{if } z > z_{flat}, \end{cases} \tag{2.42}$$

where ξ' is the transformed vertical coordinate, z the Cartesian vertical coordinate, $h(x,y)$ the terrain height in the Cartesian coordinate system, z_{min} the height of the bottom boundary in the transformed coordinate system, z_{flat} the prescribed height above which the transformed coordinate coincides with the Cartesian coordinate.

Secondly, we generate a vertically non-uniform grid in the coordinate system (ζ,η,ξ'), where the grid spacing $\Delta\xi'$ varies in the ξ'- direction, and then apply the vertical grid stretching $\xi = \xi(\xi')$ to obtain a vertically uniform one with a grid spacing $\Delta\xi$ in the coordinate system (ζ,η,ξ). Note that the grid spacings in the ζ- and η- directions are uniform. Particularly, if the vertical grid stretching is not used, we have $\xi = \xi'$ and $\Delta\xi = \Delta\xi'$,

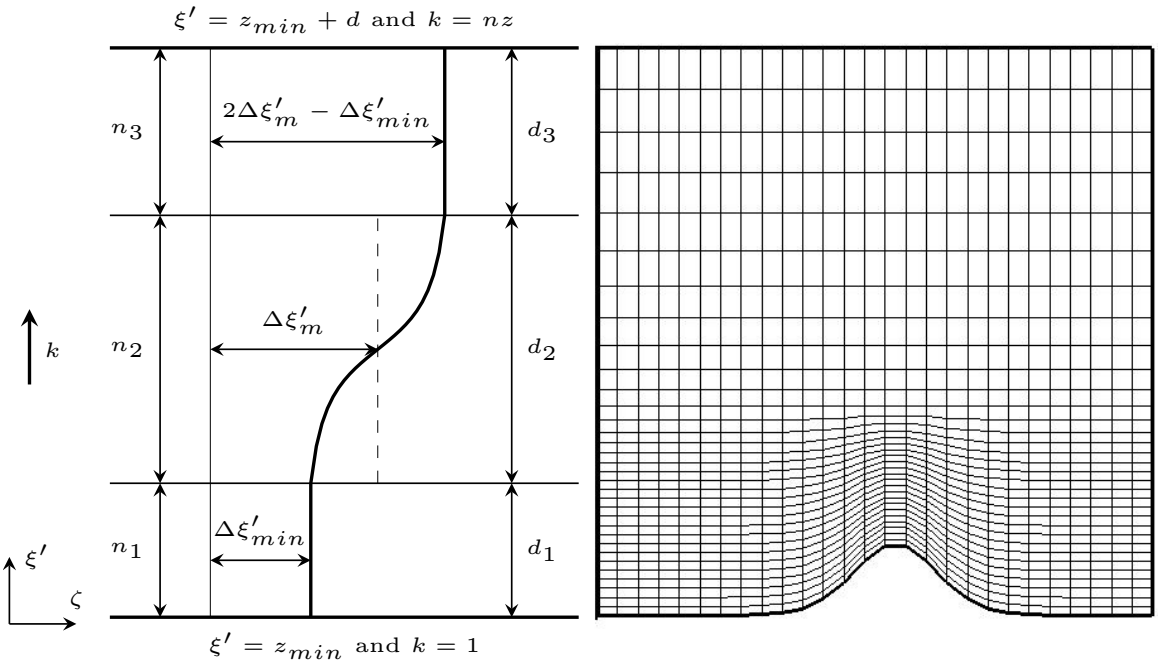


Figure 2.3 Sketch of the vertical grid stretching (left) and of its corresponding terrain-following grid (right). The terrain-following mesh with the vertical stretching is generated by the ARPS code and the grid control parameters are given by $z_{flat} = 1.2$, $d_1 = 0.8$, $d_2 = 1.6$, $\Delta\xi'_{min} = 0.05$, $\Delta\xi = 0.1$ and $nz = 37$.

which means that the grid is also uniform in the vertical direction. In numerical simulations, when the vertical grid stretching is applied, the discrete non-uniform grid spacing $\Delta\xi'_k$ in the coordinate system (ζ, η, ξ') is evaluated by:

$$\Delta\xi'_k = \begin{cases} \Delta\xi'_{min}, & \text{if } 1 \le k \le n_1, \\ \Delta\xi'_m + \dfrac{\Delta\xi'_m - \Delta\xi'_{min}}{\tanh(2\alpha_c)} \tanh\left(2\alpha_c \dfrac{k - n_m}{1 - n_m}\right), & \text{if } n_1 + 1 \le k \le n_1 + n_2, \\ 2\Delta\xi'_m - \Delta\xi'_{min}, & \text{if } n_1 + n_2 + 1 \le k \le nz, \end{cases} \tag{2.43}$$

where k is the vertical grid index, α_c is a control parameter which takes a value in the range of 0.2 - 5.0, $n_m = (1 + n_2)/2$, $\Delta\xi'_{min}$, $\Delta\xi'_m$, $2\Delta\xi'_m - \Delta\xi'_{min}$ and n_1, n_2, n_3 are the average grid spacing and the number of grid levels in the bottom, middle and top layer in the terrain-following coordinate system as shown in Figure 2.3, respectively. Concretely, nz is the total number of staggered grid points and d is the depth of the entire domain, and d_1, d_2, $d - d_1 - d_2$ are the depth of the bottom, middle and top layer respectively. $\xi' = z_{min}$ denotes the position of bottom wall and $\xi' = z_{min} + d$ is the location of upper boundary. $\Delta\xi$ is the vertical uniform grid increment in the coordinate system (ζ, η, ξ). In ARPS, $\Delta\xi'_{min}$, $\Delta\xi$, d_1,

d_2 are the control parameters given in the input file and d can be evaluated by $d = (nz-3)\Delta\xi$ for a staggered grid. The only unknown parameter $\Delta\xi'_m$ can be resolved by:

$$\frac{d_1}{\Delta\xi'_{min}} + \frac{d_2}{\Delta\xi'_m} + \frac{d-d_1-d_2}{2\Delta\xi'_m - \Delta\xi'_{min}} = \frac{d}{\Delta\xi}. \tag{2.44}$$

The vertical stretching proposes a technique strategy for the grid refinement in the vertical direction. In practice, a grid spacing less than $\Delta\xi$ is used in the bottom layer, in order to refine the mesh near wall. This is helpful to capture the coherent structures located in the near-wall region in the simulation of turbulent boundary layer flows.

2.4 Initialization and boundary conditions

To numerically solve the governing equations (Equations (2.25)), the flow field has to be initialized for the time integration and the boundary conditions should be imposed to specify the solution. Especially, the wall shear stresses must be specified as LES-NWM is used in ARPS.

2.4.1 Initialization

Before beginning the time integration in a numerical simulation, the variables must be initialized, including the time-independent base state, and the time-dependent variable variations from base state.

The hydrostatic state can be resolved by using three equations: the hydrostatic equation (Equation (2.10)), the state equation $p_r = \rho_r R T_r$ and an equation specifying the atmosphere condition, which have been introduced in Subsection 2.1.2.

For the velocity field, an average information is assigned. In ARPS, an empirical mean velocity profile for a fully developed turbulent boundary layer is given by:

$$U = \begin{cases} u_* f_w\left(\dfrac{z}{\delta_v}\right) + u_* \dfrac{\Pi}{\kappa} w\left(\dfrac{z}{\delta}\right), & \text{for } z/\delta \leq 1; \\ U_e, & \text{for } z/\delta > 1; \end{cases} \tag{2.45}$$

where U denotes the mean streamwise velocity, u_* the friction velocity, $\delta_v = \nu/u_*$ the viscous lengthscale in the inner layer, δ the thickness of boundary layers, z the vertical distance from the wall, U_e the external velocity, respectively. The function f_w represents the law of the wall on a smooth or rough wall. $w(\chi) = 2\sin^2(\chi\pi/2)$ is the wake function and Π is called

the flow-dependent wake strength parameter. More detailed information on f_w and Π is presented in Appendix A.

For a boundary layer, since $U = U_e$ at $z = \delta$, we obtain from Equation (2.45):

$$\sqrt{\frac{2}{C_f}} = \frac{U_e}{u^*} = f_w\left(\frac{\delta}{\delta_v}\right) + \frac{2\Pi}{\kappa} = f_w\left(Re_\delta \frac{u_*}{U_e}\right) + \frac{2\Pi}{\kappa} = f_w\left(Re_\delta \sqrt{\frac{C_f}{2}}\right) + \frac{2\Pi}{\kappa}, \quad (2.46)$$

where $Re_\delta = U_e\delta/\nu$ and $C_f = 2(u_*/U_e)^2$ are the Reynolds number and the skin-friction coefficient of a boundary layer. Equation (2.46) is called the friction law of boundary layers. It indicates that the skin-friction coefficient C_f is a function of the flow Reynold number Re_δ. Therefore, for a boundary layer with a given Reynold number Re_δ, solving Equation (2.46) yields a solution of the skin-friction coefficient C_f.

In numerical simulations, δ and U_e are given, we then calculate C_f from Equation (2.46), namely, get the value of u_*. Therefore, the average velocity field is evaluated using Equation (2.45) and the initialization of velocity field is achieved.

2.4.2 Boundary conditions

In the numerical simulation of a three-dimensional boundary layer, different boundary conditions are developed and imposed on the boundaries of the computational domain. Some of them are physical and some are non-physical or artificial. As shown in Figure 2.4, the rigid wall condition, upper radiative boundary condition, periodic boundary condition, recycling inlet boundary conditions, and radiative outlet boundary condition are used for the simulation of boundary layers. Moreover, a zero-gradient boundary condition is set for the base state. These boundary conditions are briefly described in the following. More details are found in Appendix B.2.

Rigid wall condition: In ARPS, a non-penetrative free-slip boundary condition (or a mirror-type boundary condition) is imposed on the bottom wall. In addition, for the simulation of boundary layers, an additional wall model is required to evaluate the wall shear stress. The detailed description of wall modeling is presented in Subsection 2.4.3.

Upper boundary condition: When the explicit scheme in the vertical direction is used, the zero-gradient boundary condition can be considered as the top boundary conditions. When the implicit scheme is used, the zero-normal gradient boundary condition is imposed for the horizontal velocity and other physique quantities, except for the w and Δp, which are governed by the upper boundary condition proposed by Klemp and Durran (1983).

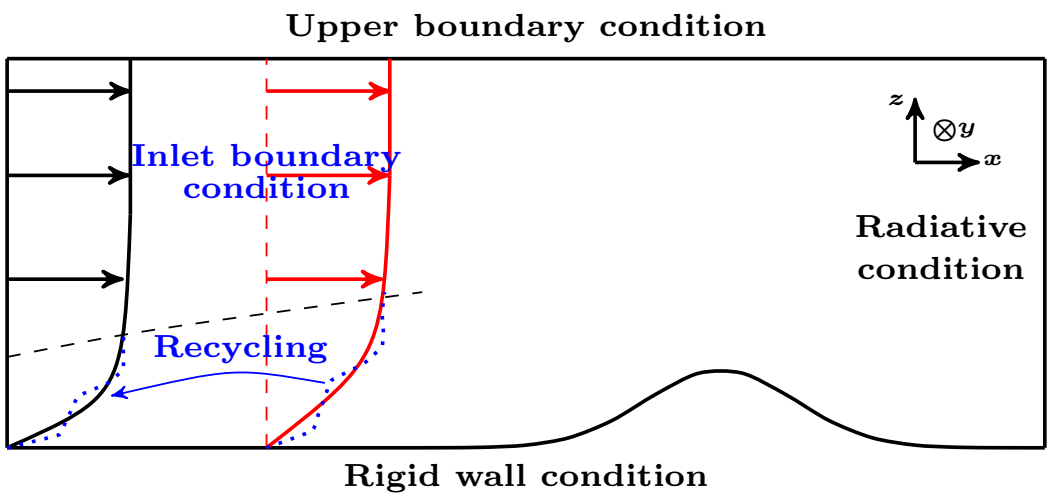


Figure 2.4 Different boundary conditions for the numerical simulation of boundary layers in ARPS.

Periodic lateral boundary condition: The periodic boundary condition is usually used in the simulation of homogeneous isotropic turbulence. It assumes that the flow field can repeat itself indefinitely outside the computational domain. If the periodic boundary condition is set on the left and right side, then the variable value at the left boundary copies that at the right boundary. Under periodic boundary conditions, the conservation of mass and of linear momentum is achieved but the conservation of angular momentum is failed [Kuzkin (2015)]. As boundary layers are usually statistically homogeneous in the spanwise direction, the periodic boundary condition is applied on the two lateral sides.

Inlet boundary condition: The inlet boundary condition is important for boundary layers simulation and the generation method is complicated. It should not only supply the average velocity fields but also the random fluctuations, which contains the information on coherent structures. In the works of Vinkovic (2005) and Huang (2015), a recycling method for generating inflow turbulence has been implemented in ARPS. Figure 2.4 gives the schematic illustration of the recycling procedure. This method, firstly proposed by Lund et al. (1998), generates the velocity fluctuations from that at a downstream station, which is obtained by resolving the discretized governing equations and thus seems more physical and realistic. For the mean flow, the streamwise mean velocity profile is given by the similar law of boundary layers (Equation (2.45)), and the spanwise and normal velocities are assumed to be null. The inflow data generated by this method provides information of coherent structures and reduces the flow transition length.

Outlet boundary condition: An open radiative boundary condition is used at the exit of the computational domain. This condition allows internally generated waves to propagate from the outlet boundary with few reflections Durran and Klemp (1982). In radiative boundary conditions, a simplified wave propagation equations is employed to evaluate the flux of the predicted model variables at the outlet boundary. In ARPS, both the formulation of Orlanski (1976) and of Klemp and Wilhelmson (1978) are implemented for the radiative outlet condition.

Base state boundary condition: The base state boundary condition is applied at the end of initialization. At the lateral boundaries, including the inlet and outlet boundaries, the fluid base density ρ_r, pressure p_r and potential temperature Θ_r are generated by the same method as the interior domain. At the bottom and top boundaries, a zero-gradient condition is set for Θ_r, and then ρ_r and p_r are resolved using the hydrostatic relation.

2.4.3 Wall modeling

As discussed in Subsection 2.2.3, wall modeling is required to specify the near-wall flow when LES-NWM is used. In the following, we firstly give a theoretical analysis on wall modeling and then introduce the wall-stress model, which is widely used in the simulation of boundary layers with high Reynold number.

As the average total stress is almost a constant in the inner layer, we have:

$$\langle \tau_{13} \rangle|_w = (\mu_t(z) + \mu)\frac{\partial U(z)}{\partial z}, \tag{2.47}$$

where μ_t is the turbulent viscosity and $U(z)$ is the average stream-wise velocity. Integrating Equation (2.47) from 0 to Δz_{min} in the wall-normal direction, it is obtained:

$$\langle \tau_{13} \rangle|_w \int_0^{\Delta z_{min}} \frac{1}{\mu_t(z) + \mu} dz = U(\Delta z_{min}). \tag{2.48}$$

which yields:

$$\langle \tau_{13} \rangle|_w = \left(\frac{1}{\Delta z_{min}} \int_0^{\Delta z_{min}} \frac{1}{\mu_t(z) + \mu} dz \right) \frac{U(\Delta z_{min})}{\Delta z_{min}} = \mu_e \frac{U(\Delta z_{min})}{\Delta z_{min}}, \tag{2.49}$$

where μ_e is called the effective viscosity, Δz_{min} is the grid increment near the wall. Equation (2.49) shows that μ_e can not be equal to the local viscosity, when Δz is large. Hence, $\langle \tau_{13} \rangle|_w$ can not be calculated accurately in a coarse grid and needs to be specified by wall modeling.

τ_{23} is in the same situation. This explains why it is difficult to simulate correctly the dynamics of the inner layer when Δz^+_{min} is large and implies that an effective way of wall modeling is to directly modify the wall shear stress.

Schumann (1975) proposed a model to specify the instantaneous wall shear stress $\tau_{13}|_w$ and $\tau_{23}|_w$:

$$\tau_{13}|_w = \left(\frac{\tilde{u}(\Delta z_{min})}{U(\Delta z_{min})}\right)\tau_w, \tag{2.50a}$$

$$\tau_{23}|_w = \left(\frac{\tilde{v}(\Delta z_{min})}{U(\Delta z_{min})}\right)\tau_w, \tag{2.50b}$$

where $\tilde{u}, \tilde{v}$ denote the resolved stream-wise, spanwise velocity and τ_w the mean wall shear. In this model, $U(\Delta z_{min})$ and τ_w are statistical physics quantities and are given a priori. Grötzbach (1987) made an extension by using the logarithmic law to evaluate τ_w. Firstly, the statistic average is approximated by the spatial average in the span-wise direction, that is, $U(\Delta z_{min}) = \langle \tilde{u}(\Delta z_{min})\rangle_y$; Secondly, applying the logarithmic law to resolve the friction velocity u_*, then the mean wall shear stress is obtained by $\tau_w = \rho u_*^2$. For example, a simple logarithmic law $U/u_* = \kappa^{-1}\ln(z/z_0)$ is used for a rough case in ARPS, the mean wall shear is then calculated by:

$$\tau_w = \rho\langle \tilde{u}(\Delta z_{min})\rangle_y^2 \left(\frac{1}{\kappa}\ln\left(\frac{\Delta z_{min}}{z_0}\right)\right)^{-2}, \tag{2.51}$$

where z_0 is the roughness length and κ denotes the von Kámán constant.

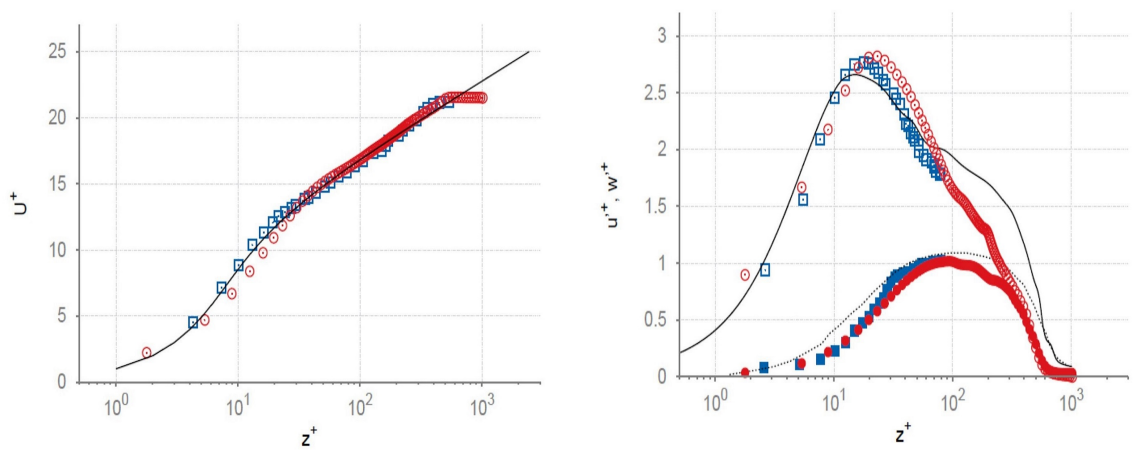
Another modification proposed by Mason and Callen (1986) is to replace the mean stream-wise velocity $U(\Delta z_{min})$ by the local instantaneous surface velocity $u_s(\Delta z_{min}) = \sqrt{\tilde{u}^2+\tilde{v}^2}\Big|_{z=\Delta z_{min}}$. By assuming that $u_s(\Delta z_{min})$ also obeys the logarithmic law, then Equation (2.51) becomes:

$$\tau_w = \rho\left(\tilde{u}^2+\tilde{v}^2\right)\big|_{z=\Delta z_{min}} \left(\frac{1}{\kappa}\ln\left(\frac{\Delta z_{min}}{z_0}\right)\right)^{-2}. \tag{2.52}$$

Note that this first modification is based on the assumption that the logarithmic law is satisfied for the spatial average velocity in span-wise direction, which is easily acceptable since the flow is homogeneous in span-wise direction, and that the second modification requires that the logarithmic law is obeyed for the local and instantaneous velocity, which may be true as the resolved large scale velocity can be approximated by the mean velocity for a very coarse grid.

2.5 Conclusions

In this chapter, we introduce not only the theoretical formulation of governing equations used in ARPS, but also the numerical details, such as discretization, parallelization, initialization and boundary conditions. The governing equations of atmospheric boundary layers are resolved by LES with a SGS model. After the development of Vinkovic (2005), the recycling method for generating inflow turbulence was implemented and the wall modeling was improved by introducing a new law of the wall of boundary layers on a rough surface [Huang et al. (2016)]. The performance of this numerical model in the simulation of boundary layers was validated in the thesis of Huang (2015). As shown in Figure 2.5, a good agreement is achieved between the mean velocity and velocity RMS profile of the numerical results and of experimental data.



(a) Mean streamwise velocity profile (b) Streamwise and normal velocity RMS profile

Figure 2.5 Validation of the numerical model (ARPS). Blue square: experimental results [Vinçont et al. (2000) and Simoens et al. (2007)] with $Re_\tau = 500$. Black line: DNS results of Spalart (1988) with $Re_\tau = 660$. Red circle: ARPS results. Adapted from Huang (2015).

Chapter 3

Immersed boundary method

In ARPS, the numerical simulation of turbulent boundary layer flows over complex geometries is achieved through using terrain-following grids. However, it's difficult to generate good quality grids for geometries with extreme slopes. Moreover, the grid generation should be repeatedly done at each time step for moving boundaries. An alternative choice overcoming these difficulties is the immersed boundary method (IBM). This approach introduces an additional forcing term into the governing equations in the vicinity of the boundary. Thus irregular boundaries can be taken into account within a Cartesian grid and the flow fields can be simulated by numerically resolving the modified governing equations. The immersed boundary method facilitates the grid generation and the treatment of moving boundaries. Developing a new numerical solver by introducing IBM into ARPS is of first importance in this thesis, as our aim is to simulate boundary layers over moving dunes. The implementation of IBM is one original point of this work.

In this chapter, we detail the implementation of the immersed boundary method into the ARPS code. In Section 3.1, the development of the IBM is briefly introduced; In Section 3.2, the direct forcing method (ghost-cell finite-difference method) is detailedly described; In Section 3.3, the details of the implementation of the IBM into ARPS are presented; In Section 3.4, the developed numerical solver (IBM-ARPS) is validated through two numerical simulation cases: a turbulent boundary layer (TBL) over a small dune and a TBL over a large dune.

3.1 Basic description

The immersed boundary method was primarily proposed by Peskin (1972) in the simulation of the blood flow in the heart [Peskin (1977)]. In the original method, an elastic boundary condition was represented by introducing a boundary force, which satisfies the Hooke's

law. After that, the rigid boundary for IBM was developed by Briscolini and Santangelo (1989) and Goldstein et al. (1993). In both approaches, the immersed rigid body was regarded as the virtual fluid, whose velocity was enforced to zero by a feedback mechanism. Soon after, Mohd-Yusof (1997) and Fadlun et al. (2000) proposed a discrete immersed boundary method by directly imposing the immersed forcing on the computational nodes near the boundary, rather than on the immersed interface. With this improvement, the desired boundary conditions at the immersed boundary are achieved by reconstructing the flow field in the vicinity of the boundary. For instance, the no-slip boundary condition was imposed by reconstructing the velocity field through the interpolation or extrapolation between the resolved velocities at the neighbor nodes and the zero velocity at the boundary [Mittal et al. (2008); Tseng and Ferziger (2003)].

In this section, we introduce the basic conception of the IBM, give a brief description of the development of the IBM and summarize its advantages and disadvantages.

3.1.1 General consideration

The basic idea of the immersed boundary method is that the irregular boundary geometries are converted into the regular ones by enlarging the fluid domain and submerging the complex boundaries, and the Navier-Stokes equations are modified by introducing the additional forcing in the immersed zones. This modification facilitates the resolution of the governing equations by using simple boundaries and computational grids.

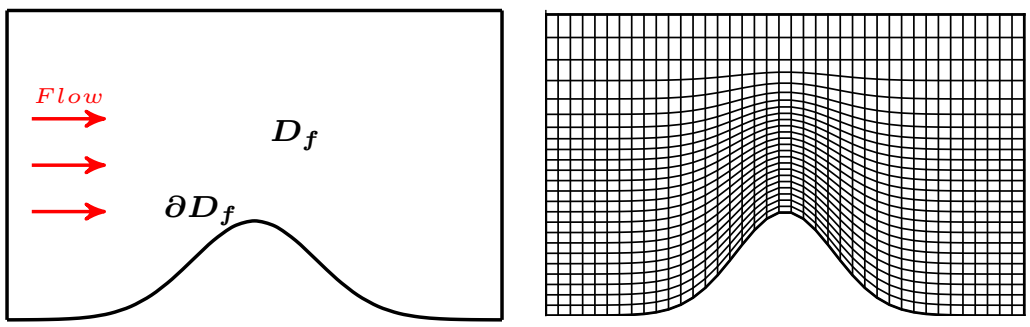


Figure 3.1 Sketch of a boundary layer over a single dune (left) and of the corresponding body-conformable grid (right).

For example, considering a turbulent boundary layer flow over a single dune, the governing equations of fluid fields are simply given by:

$$\partial_t \boldsymbol{\psi} + \mathscr{G}\boldsymbol{\psi} = \mathbf{0} \text{ in } D_f, \tag{3.1a}$$

$$\mathscr{B}_f \boldsymbol{\psi} = \mathbf{0} \text{ on } \partial D_f, \tag{3.1b}$$

where $\boldsymbol{\psi}$ means the vector of physical variables, D_f and ∂D_f are the computational domain and the boundaries as shown in Figure 3.1, the operator $\mathscr{G}$ denotes the governing equations without the temporal evolution term, and $\mathscr{B}$ represents the operator of boundary conditions on ∂D_f. To simulate this case using finite-difference method, ARPS generates the body-conformable grid as shown in Figure 3.1 and the flow field is obtained by resolving the governing equations in this terrain-following coordinate system.

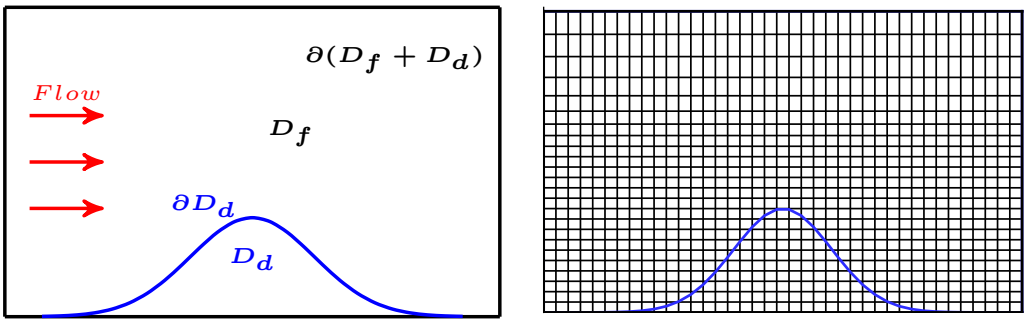


Figure 3.2 Sketch of the computational domain (left) and of the corresponding Cartesian grid (right) of a boundary layer over a single dune resolved by the immersed boundary method.

According to the immersed boundary method conception, the complex dune surface ∂D_d is immersed into an enlarged and regular domain $D_f + D_d$, and then the previous boundary conditions $\mathscr{B}_f \boldsymbol{\psi} = \mathbf{0}$ on ∂D_f (Equation (3.1b)) is replaced by $\mathscr{B}_f \boldsymbol{\psi} = \mathbf{0}$ on $\partial(D_f + D_d)$ and an additional boundary condition on the immersed dune $\mathscr{B}_d \boldsymbol{\psi} = \mathbf{0}$ on ∂D_d. This immersed condition $\mathscr{B}_d \boldsymbol{\psi} = \mathbf{0}$ on ∂D_d is eliminated by introducing forcing terms $\mathscr{F} \boldsymbol{\psi}$ into the general governing equations (Equation (3.1a)) in the vicinity of the immersed boundary ∂D_d. Thus Equation (3.1) is rewritten as:

$$\partial_t \boldsymbol{\psi} + \mathscr{G} \boldsymbol{\psi} = \mathscr{F} \boldsymbol{\psi} \text{ in } D_f + D_d, \tag{3.2a}$$

$$\mathscr{B}_f \boldsymbol{\psi} = \mathbf{0} \text{ on } \partial(D_f + D_d), \tag{3.2b}$$

where $D_f + D_d$ denotes the new regular computational domain and $\partial(D_f + D_d)$ represents its corresponding boundaries. Figure 3.2 gives a sketch of domains D_f, D_d, $D_f + D_d$ and of boundaries ∂D_f, ∂D_d, $\partial(D_f + D_d)$. In general, Equation (3.2) is resolved using a simple Cartesian grid as shown in Figure 3.2. Obviously, the task of grid generation is greatly simplified by introducing the immersed boundary approach and the grid regeneration at each time step is avoided for the moving boundary.

3.1.2 Formulation of the forcing term

According to the way of imposing the boundary conditions, the immersed boundary method can be classified into two groups [Mittal and Iaccarino (2005)]: *i)*, the continuous forcing method, in which the forcing function is introduced into the continuous governing equations in the vicinity of immersed boundaries before the discretization; *ii)*, the discrete forcing method, in which the forcing term is incorporated into the discrete governing equations. In the following, the mathematical formulations of the forcing term for these two methods are introduced.

Continuous forcing method: The forcing function, also called the force density function $\mathscr{F}\boldsymbol{\psi}$ in Equation (3.2), is introduced into the governing equations to implicitly impose the boundary conditions on the immersed surface. The formulation of continuous forcing function $\mathscr{F}\boldsymbol{\psi}$ is usually given by:

$$\mathscr{F}\boldsymbol{\psi}(\boldsymbol{x}_s,t) = \int_{D_f+D_d} \boldsymbol{f}_{\boldsymbol{\psi}}(\boldsymbol{x},t)\boldsymbol{\delta}(\boldsymbol{x}-\boldsymbol{x}_s)d\boldsymbol{x}, \tag{3.3}$$

where $\boldsymbol{f}_{\boldsymbol{\psi}}$ is the boundary force for physical variable $\boldsymbol{\psi}$, $\boldsymbol{x}_s$ denotes the position of immersed boundary, *i.e.*, $\boldsymbol{x}_s \in \partial D_d$, and $\boldsymbol{\delta}$ is the Dirac function, respectively. In numerical simulations, the location of immersed surface does not usually coincide with the grid points, and the forcing is then distributed over these surrounding grid points through replacing the $\boldsymbol{\delta}$ function by a smooth distribution function.

For rigid boundaries, a common formula of $\boldsymbol{f}_{\boldsymbol{\psi}}$ for imposing no-slip condition is given by [Goldstein et al. (1993)]:

$$\boldsymbol{f}_{\boldsymbol{\psi}}(\boldsymbol{x},t) = \alpha_0 \int_0^t \boldsymbol{\psi}(\boldsymbol{x},\tau)d\tau + \beta_0\boldsymbol{\psi}(\boldsymbol{x},t), \tag{3.4}$$

where α_0, β_0 are two coefficients and here ψ denotes the fluid velocity. Equation (3.4) proposes a feedback control for the velocity field in the vicinity of the immersed boundaries. When $\beta_0 = 0$ and $\alpha_0 = -\kappa_s$ where κ_s is a positive spring constant, Equation (3.4) reduces to the approaches of Beyer and LeVeque (1992) and Lai and Peskin (2000). When $\alpha_0 = 0$ and $\beta_0 = -\mu/K$ where K is the medium permeability coefficient, the penalization approach proposed by Khadra et al. (2000) can be deduced from Equation (3.4).

The continuous forcing method proposes a mathematical formula of the immersed forcing and thus is easy to implement into the existed codes. The drawback of continuous IBM with rigid boundaries is that the coefficients in Equation (3.4) need to be specified by users and

that the forcing usually induces spurious oscillations near the boundary, which are associated with the numerical instability.

Discrete forcing method: The discrete method gives a good performance for the simulation of turbulent flows with high Reynolds number, due to the imposition of the immersed boundary conditions without calculating the forcing term. In discrete form, the forcing term is regarded as a direct corrector on the resolved fluid flow in the vicinity of the immersed boundary. This term can be determined at each time step by using the immersed boundary conditions. For example, for the simple time-stepping scheme, to yield the desired boundary conditions at the next time step, *i.e.*, $\boldsymbol{\psi}^{n+1} = \boldsymbol{\psi}_{ib}$, the prescribed force at the time step n can be expressed as:

$$\mathscr{F}\boldsymbol{\psi}^n = \frac{1}{\Delta t}(\boldsymbol{\psi}_{ib} - \boldsymbol{\psi}^n) + \mathscr{G}\boldsymbol{\psi}^n, \tag{3.5}$$

where Δt is the time step and the supscript $(\cdot)^n$ denotes the n-th time step. When the grid coincides with the immersed boundary, $\boldsymbol{\psi}_{ib}$ can be easily specified according to the boundary conditions. In general, the immersed boundary cut off the grids, thus some questions are raised: where are the immersed forcing located? how to reconstruct $\boldsymbol{\psi}_{ib}$? The answers are summarized as follows:

- **Location of the forcing:** there are two possible locations of the discrete immersed forcing $\mathscr{F}\boldsymbol{\psi}^n$: *i)*, Mohd-Yusof (1997) firstly suggested that the discrete forcing was imposed inside the solid field. This treatment was extended in the ghost-cell finite difference method; *ii)*, Fadlun et al. (2000) proposed the external forcing in the simulation of three-dimensional complex flows. With this method, the forcing is applied on the fluid nodes closest to the immersed boundary. Both the internal forcing and the external forcing can effectively reconstruct the sharp representation of the immersed boundary condition and give good simulated results.

- **Reconstruction scheme for $\boldsymbol{\psi}_{ib}$:** many reconstruction methods are available for reconstructing the flow field at the immersed boundary. The standard reconstruction method consists of a one-dimensional interpolation scheme and a reconstruction direction. Fadlun et al. (2000) used the linear interpolation in the grid-line direction; Gilmanov et al. (2003) and Balaras (2004) proposed the linear interpolation along the well-defined line normal to the immersed boundary, in order to avoid the ambiguities of choosing the grid lines; Peller et al. (2006) used the high-order Lagrange and least squares interpolation along the grid-line direction for the flow reconstruction. The reconstruction method is improved by applying the multi-dimensional interpolation scheme. Iaccarino and Verzicco (2003) reconstructed the velocity at the solid nodes

near the boundary using the linear multi-dimensional and quadratic multi-dimensional interpolation; in the ghost-cell approach developed by Tseng and Ferziger (2003), the linear and quadratic multi-dimensional interpolation is commonly used to evaluated the velocity at the ghost cell, and the interpolation combined with the mirroring technique is proposed for the special treatment in case of large negative value appearing when the ghost cell is very near the immersed boundary; Gao et al. (2007) developed an inverse distance weighted interpolation scheme based on the Taylor expansion analysis about the boundary point for the improvement of the ghost approach.

The discrete (or direct) forcing method enforces the immersed boundary condition through the reconstruction approach. The velocity gradient near wall can be predicted more accurately through applying a high-order interpolation, especially for boundary layers and for high Reynolds number flows. The advantage is that we implement this method without calculating the forcing term and that no unknown parameters should be specified by users.

It is clearly concluded that the immersed boundary method is very suitable for the simulation of complex or moving boundary problems. Commonly, the discrete forcing method performs better in the simulation of turbulent flows with high Reynolds number than the continuous method. Therefore, the ghost-cell finite-difference method, a direct forcing approach proposed by Tseng and Ferziger (2003), has been chosen to be implemented in the ARPS code, in order to simulate boundary layers over complex terrains, and thus to investigate the wind erosion with moving dunes.

3.1.3 Advantages and disadvantages of the IBM

The first advantage of applying immersed boundary method in the simulation of turbulent flows with complicated geometrical boundaries is that the process of grid generation is much easier to be achieved. Commonly, it is difficult to generate an acceptable body-conformable structured or unstructured mesh for complex boundaries, since the geometrical complexity has a great influence on the quality of the generated grid, thus on the numerical accuracy and stability. Moreover, as the geometry becomes more complicated, the process of generating a good-quality body-conformable grid becomes increasingly difficult and the computational operation count per grid point greatly increases. In contrary, the geometrical complexity can not significantly affect the task of grid generation in immersed boundary method. Secondly, the immersed boundary method can handle the moving boundary problem. Compared with the usual methods, neither the grid-regeneration nor the grid deformation are required in the numerical simulation. Hence, using immersed boundary method has a considerable decrease of the numerical difficulties and of the computational time-consuming. In addition, it is

obvious that the implementation of immersed boundary method in an existed code is much simpler compared with the other methods.

One drawback is that the imposition of immersed boundary conditions is not straightforward as the traditional approaches. Another one is that the problem of accuracy on boundary treatment and of conservation properties of numerical schemes in vicinity of immersed boundaries is not trivial. Moreover, when simulating turbulent flows with high Reynolds number, the grid size greatly increases and this will lead to a substantial fraction of grid points inside the body, where the resolved flow is not physical and useless.

3.2 Ghost-cell finite-difference method

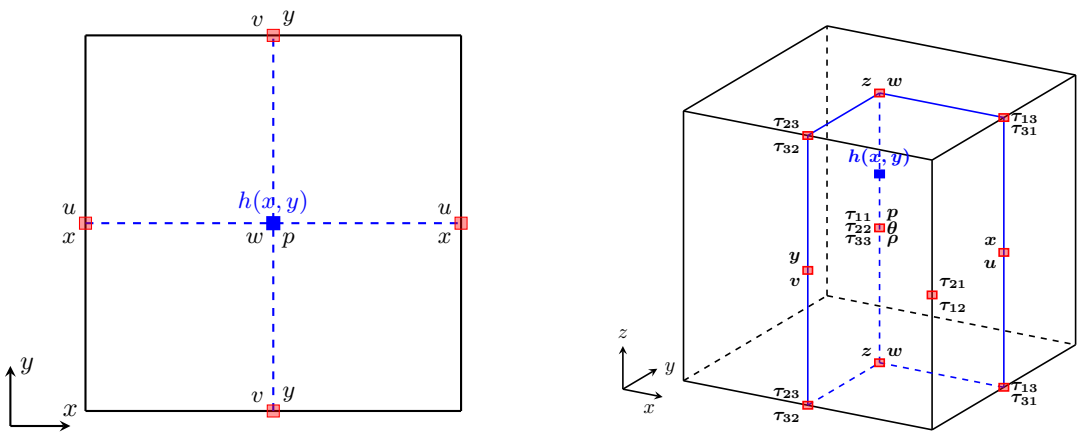
The ghost-cell finite-difference method of Tseng and Ferziger (2003) is an extension of the discrete immersed method of Mohd-Yusof (1997) and Fadlun et al. (2000). The basic idea of this approach is to construct a sharp representation of the immersed boundary using the ghost cells, which are defined as cells in the solid that have at least one neighbor in the fluid. Then, the immersed boundary conditions can be achieved through reconstructing the velocity fields on these ghost cells. Suggested by Lundquist et al. (2010), the bilinear interpolation scheme combined with the mirroring technique is used to calculate the ghost cell velocity. This method is suitable for the simulation of high Reynolds number flows.

In this section, we will describe in detail the ghost-cell finite-difference method, including the representation of the immersed boundary, the classification of computational nodes, the treatment of Dirichlet and Neumann Boundary condition, and near-wall models at the immersed boundary.

3.2.1 Representation of the immersed boundary

To implement IBM into ARPS, the first step is to construct the discrete representation of the immersed dune surface. The dune height $h(x,y)$, a function of horizontal coordinates x and y, is considered as a scalar variable on the horizontal plane defined on the staggered grid. As shown in Figure 3.3, $h(x,y)$ is defined at the center of a grid cell on xy-plane, and takes the value of dune height in the z-direction.

Figure 3.4 shows an example of the numerical representation of dune elevation in ARPS, which has the same resolution as the horizontal grid. This allows the treatment of the dune height as a scalar variable located at the bottom wall. One advantage of this representation is that the message passing between the dune height arrays at different processors can be easily achieved by calling the parallelization modules in ARPS. Another advantage is that

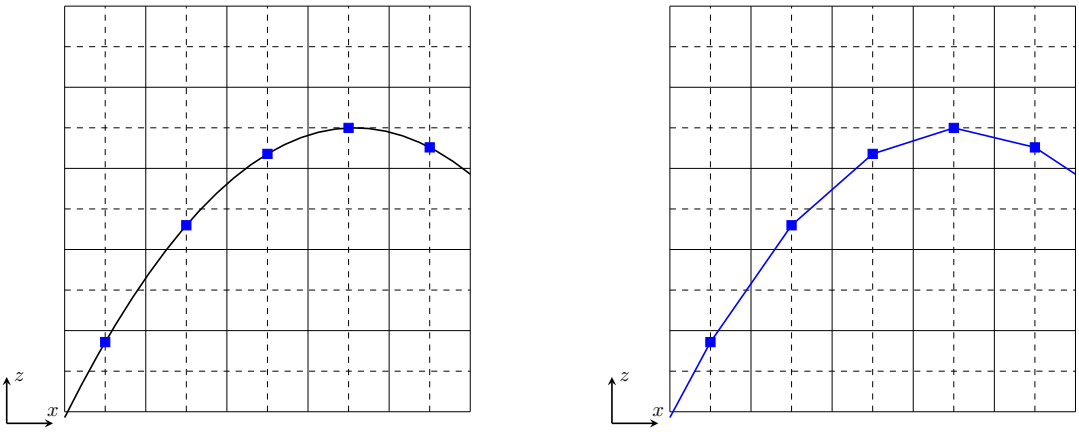


(a) Two-dimensional top view.

(b) Three-dimensional view.

Figure 3.3 Sketch of the location of the dune height variable $h(x,y)$ in an Arakawa C-grid cell.

this representation will be compatible with the regional wind erosion model proposed in Chapter 4, which determines the dune deformation.



(a) Continuous dune surface.

(b) Discrete dune surface.

Figure 3.4 An example of the representation of dune height at the Arakawa C-grid.

3.2.2 Classification of computational nodes

The computational nodes can be classified into four groups: fluid nodes, wall nodes, ghost nodes and solid (dune) nodes. They are described in detail as follows:

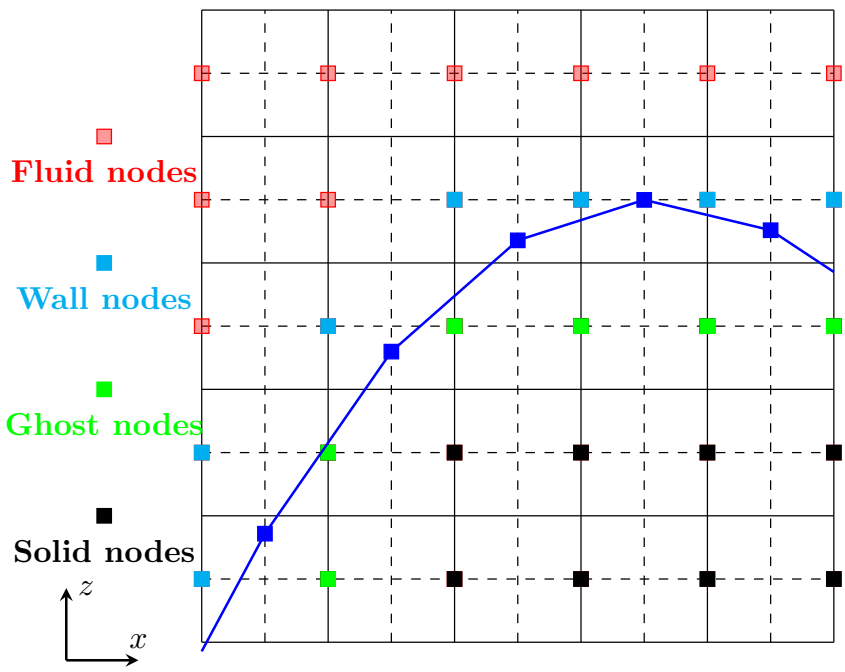


Figure 3.5 An example of the classification of computational nodes.

1. Fluid nodes: located at the cells in the fluid field, which have no neighbors in the solid. The flow field at the fluid nodes is obtained by resolving the normal Navier-Stokes equations without the additional forcing term.

2. Wall nodes: located at the cells in the fluid field, which have at least one neighbor in the solid. When the grid is fine enough, the wall nodes are considered as the fluid nodes and the resolution at these nodes is credible. When the grid is coarse, the wall models should be applied to correct the resolved fields at wall nodes. The details are introduced in Subsection 3.2.4.

3. Ghost nodes: located at the cells in the solid (dune) field, which have at least one neighbor in the fluid. Within the immersed boundary method, the forcing term is directly imposed to reconstruct the flow field in the ghost nodes and thus to enforce the immersed boundary conditions. The flow reconstruction is detailedly described in Subsection 3.2.3.

4. Solid nodes: located at the cells in the solid (dune) field, which have no neighbor in the fluid. Fadlun et al. (2000) and Iaccarino and Verzicco (2003) found that the interior treatment has no influence on the simulated results of extremal flows for the stationary surface, whereas the investigation of Liao et al. (2010) shows the importance

of applying the internal forcing in the simulations of moving boundary problems. Therefore, the treatment at solid nodes should be tested in this work.

For the staggered grid, the classification of the computational nodes should be done for each physical variable. Figure 3.5 gives a schematic illustration of the locations of fluid nodes, wall nodes, ghost nodes and solid nodes.

3.2.3 Flow reconstruction at the immersed boundary

The immersed forcing is implicitly imposed on the ghost cells by reconstructing the flow field. The velocity on the ghost cells are reconstructed by a bilinear interpolation scheme combined with the the mirroring technique proposed by Lundquist et al. (2010). The basic two boundary conditions are Dirichlet and Neumann conditions, which are expressed as:

$$\psi|_{\partial D} = \psi_0, \tag{3.6}$$

and

$$\left.\frac{\partial \psi}{\partial n}\right|_{\partial D} = \psi_n, \tag{3.7}$$

where ψ_0 and ψ_n are two constants and ∂D represents the boundary.

The flow reconstruction at each ghost node can be achieved through the following three steps:

- Step 1: Finding the image point using the mirroring technique. Given an example as shown in Figure 3.6, we find G' the image point of the ghost node G through the immersed boundary (the dune surface in this thesis).

- Step 2: Determining the value of the physical variable at the image point through the interpolation. As shown in Figure 3.6, the physical variable $\psi_{G'}$ at G' is calculated through the bilinear interpolation between these four neighbors P_1, P_2, P_3,P_4, in which P_1 is on the immersed boundary surface, and the others are fluid points or wall points.

- Step 3: Reconstruction at the ghost cells. In Figure 3.6, the value at the ghost cell G equals to $\psi_G = 2\psi_0 - \psi_{G'}$ for the Dirichlet boundary condition and $\psi_G = \psi_{G'} - \overline{GG'}\psi_n$ for the Neumann boundary condition.

Next, we introduce the details of the identification of these neighbors and the interpolation scheme in the second step. Since the immersed boundary cuts the computational cell in an arbitrary manner, the identification of the four neighbors depends on the geometric

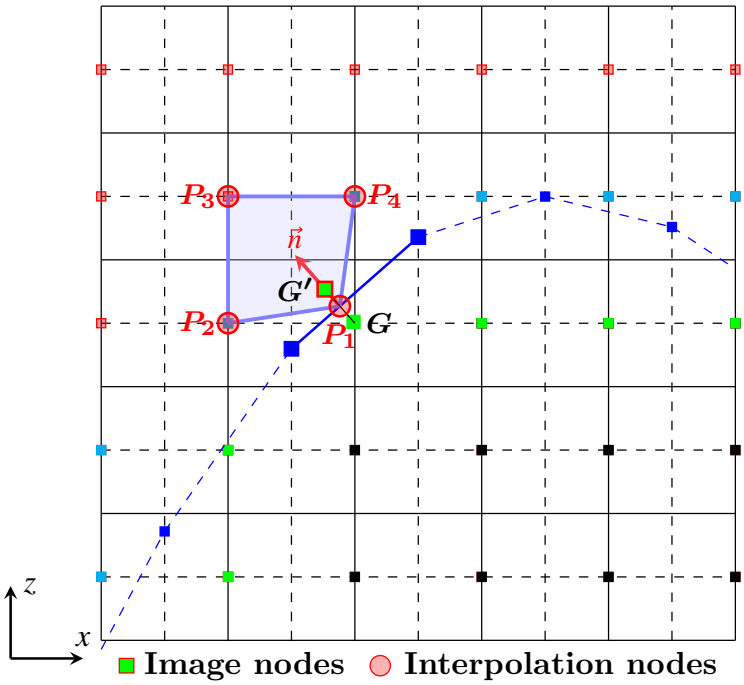
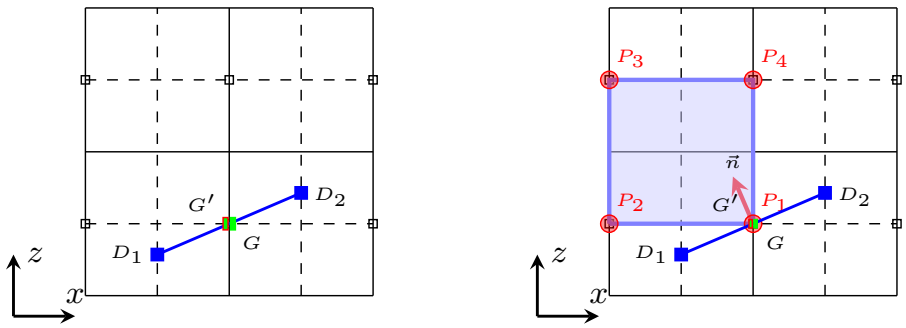


Figure 3.6 An example of reconstructing the physical variable ψ at the ghost node G. G' is the image point of G. P_1, P_2, P_3, P_4 are the four neighbors. $\vec{n}$ is the normal vector at P_1.



(a) Dirichlet boundary condition.

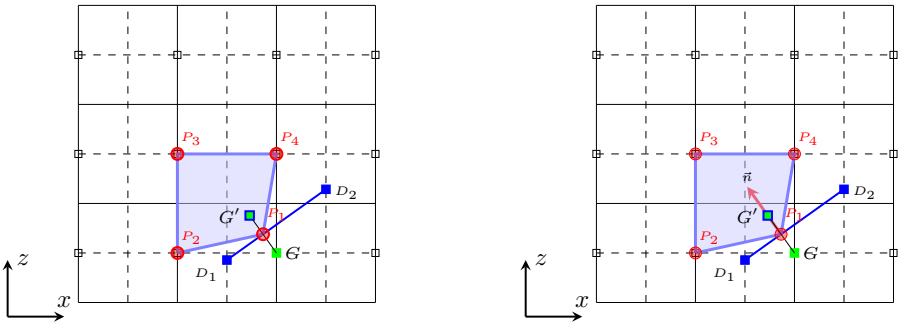
(b) Neumann boundary condition.

Figure 3.7 Sketch of case 1 for Dirichlet and Neumann boundary conditions.

relationship between the image point and the immersed boundary. We summarize the different choices of these interpolation points as follows:

- Case 1: the immersed boundary is coincident with the computational nodes, thus the image point is itself as shown in Figure 3.7. In this simple case, the treatment for the Dirichlet boundary is different from that for the Neumann boundary. For the Dirichlet boundary, the model variable is directly modified by $\psi_G = \psi_0$ according to Equation

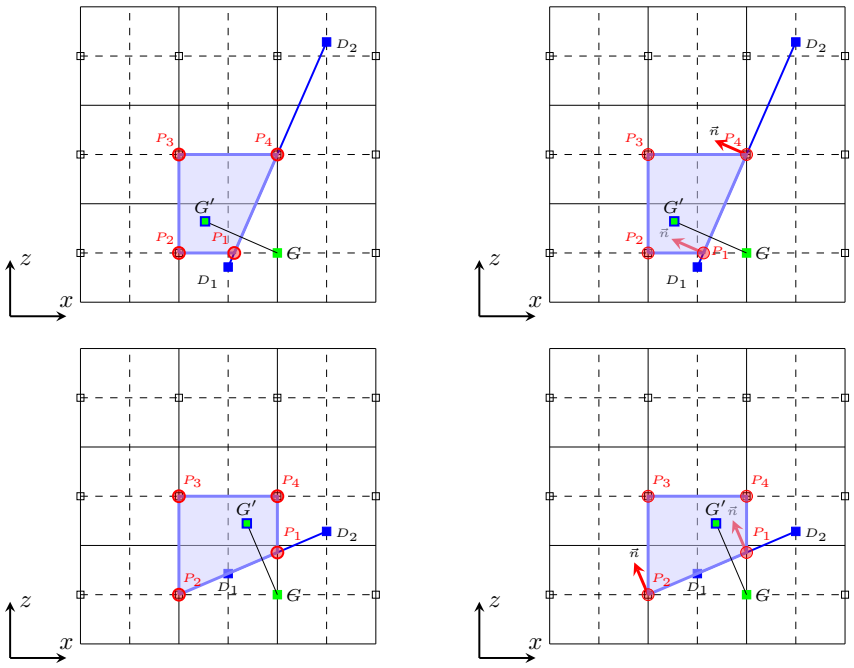
(3.6). For the Neumann boundary, ψ_G can not be assigned by Equation (3.7), thus should be specified through the interpolation between the neighbors P_1, P_2, P_3, P_4.



(a) Dirichlet boundary condition. (b) Neumann boundary condition.

Figure 3.8 Sketch of case 2 for Dirichlet and Neumann boundary conditions.

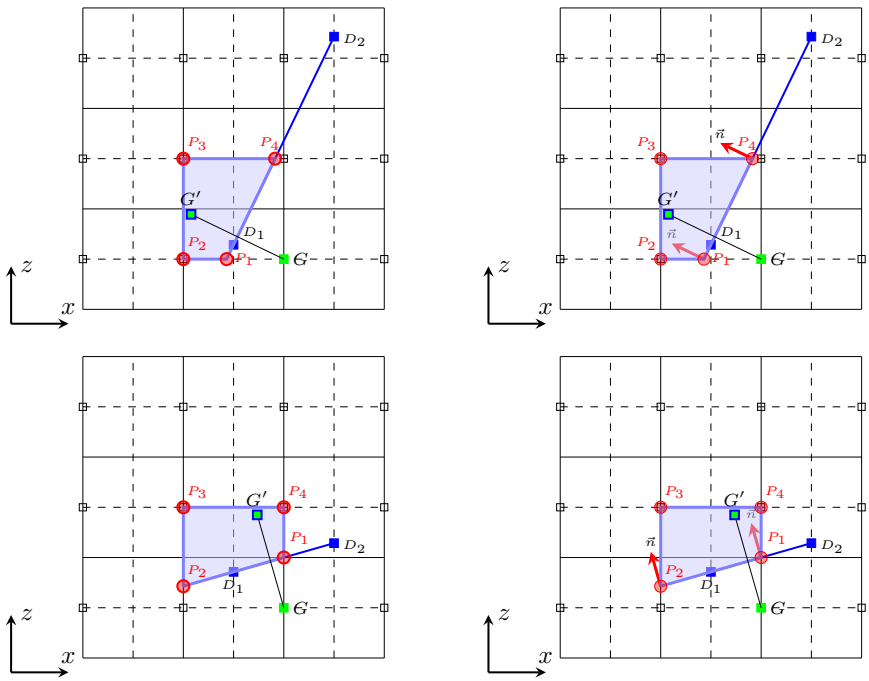
- Case 2: the image point is located in the cut computational cell and the immersed boundary cuts two adjacent sides as shown in Figure 3.8. This is a typical case shown in Figure 3.6. One of the four neighbors P_1 is located at the interface. The treatment for the Dirichlet and Neumann boundary is almost same.



(a) Dirichlet boundary condition. (b) Neumann boundary condition.

Figure 3.9 Sketch of case 3 for Dirichlet and Neumann boundary conditions.

- Case 3: it is a special case of Case 2 or of Case 4, in which one computational node is coincident with the immersed boundary. As shown in Figure 3.9, except for two computational nodes, two points of intersection between the line $\overline{D_1D_2}$ and the cut computational cell are chosen as the interpolation points. Therefore, Equation (3.6) or Equation (3.7) will be applied twice during the interpolation.



(a) Dirichlet boundary condition. (b) Neumann boundary condition.

Figure 3.10 Sketch of case 4 for Dirichlet and Neumann boundary conditions.

- Case 4: the immersed boundary cut two opposite sides of a computational cell and the image point is located in the cut grid cell. In this case as shown in Figure 3.10, in order not to lose generality, two intersection points between $\overline{D_1D_2}$ and the cut grid cell are chosen as two neighbors for the interpolation and the others are two fluid nodes identified in the same way as Case 3. Hence, Equation (3.6) or Equation (3.7) will be used twice during the interpolation.

- Case 5: the image point is out of the cut grid cell as shown in Figure 3.11. In this case, the neighbors are four fluid points. Equation (3.6) and Equation (3.7) are no longer used and the same interpolation treatment is applied for the Dirichlet and Neumann boundary.

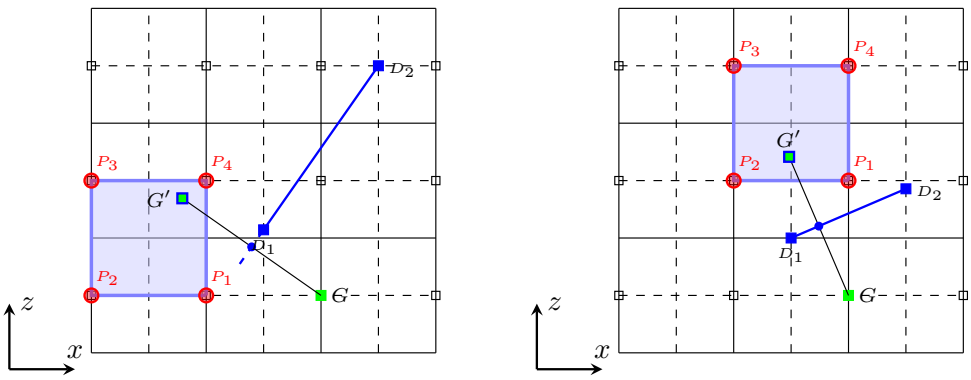


Figure 3.11 Sketch of case 5 for Dirichlet and Neumann boundary conditions.

In this thesis, the immersed sand dune is homogeneous in the spanwise ($y-$) direction. Hence, the bilinear interpolation scheme appreciate for two-dimensional dune is used for the reconstruction:

$$\psi(x,z) = a_1 + a_2 x + a_3 z + a_4 xz, \tag{3.8}$$

where a_1, a_2, a_3, a_4 are the four unknown coefficients. Note that the trilinear reconstruction scheme can be used for the extension to three-dimensional dune. The coefficients are resolved using the flow field on the neighbors P_1, P_2, P_3 and P_4. The details are introduced as follows.

Dirichlet boundary condition: Using the flow field on the four neighbors (P_i with $i \in \{1,2,3,4\}$), we have from Equation (3.8):

$$\boldsymbol{P}\boldsymbol{a} = \boldsymbol{\psi}, \tag{3.9}$$

where the matrix $\boldsymbol{P}$, $\boldsymbol{a}$ and $\boldsymbol{\psi}$ are given by:

$$\boldsymbol{P} = \begin{bmatrix} 1 & x_1 & z_1 & x_1 z_1 \\ 1 & x_2 & z_2 & x_2 z_2 \\ 1 & x_3 & z_3 & x_3 z_3 \\ 1 & x_4 & z_4 & x_4 z_4 \end{bmatrix}, \boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}, \boldsymbol{\psi} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix}. \tag{3.10}$$

Here (x_i, z_i) denotes the position of point P_i and ψ_i takes the value of the physical variable at the position P_i. When one of them is located at the immersed boundary, the physical variable should take the value ψ_0 according to Equation (3.6). By resolving Equation (3.9), the four coefficients are obtained:

$$\boldsymbol{a} = \boldsymbol{P}^{-1}\boldsymbol{\psi}, \tag{3.11}$$

where $\boldsymbol{P}^{-1}$ is the inverse matrix of $\boldsymbol{P}$. In this thesis, the matrix inversion is achieved using the algorithm of Gaussian elimination.

Neumann boundary condition: For the case 5 as shown in Figure 3.11, the interpolation for the Neumann boundary condition is accomplished in the same manner as for the Dirichlet boundary condition, since the four interpolation points are computational nodes. The unknown interpolation coefficients in Equation (3.8) are resolved by Equation (3.11), where the matrix $\boldsymbol{P}$ and $\boldsymbol{\psi}$ are constructed by Equation (3.10).

For the other cases where one or more identified neighbors coincide with the interface, Equation (3.7) should be used for the interpolation to specify the immersed boundary condition. To satisfy the Neumann boundary condition for the coincident nodes on the immersed surface, substituting Equation (3.8) into Equation (3.7) yields:

$$\frac{\partial \psi}{\partial n} = \vec{n} \cdot \vec{\nabla} \psi = a_2 n_x + a_3 n_z + a_4 (n_x z + x n_z) = \psi_n, \tag{3.12}$$

where the normal vector is denoted by $\vec{n} = (n_x, n_y, n_z)$. In the present case that a two-dimensional dune is used, we always have $n_y = 0$. For instance, if only one of the neighbors is on the immersed surface, the matrix $\boldsymbol{P}$ and $\boldsymbol{\psi}$ are rewritten as:

$$\boldsymbol{P} = \begin{bmatrix} 0 & n_x & n_z & n_x z_1 + n_z x_1 \\ 1 & x_2 & z_2 & x_2 z_2 \\ 1 & x_3 & z_3 & x_3 z_3 \\ 1 & x_4 & z_4 & x_4 z_4 \end{bmatrix}, \boldsymbol{\psi} = \begin{bmatrix} \psi_n \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix}. \tag{3.13}$$

After the coefficients a_i are resolved using Equation (3.11), the variable value $\psi_{G'}$ can be obtained according to Equation (3.8). Therefore, ψ_G at the ghost point is reconstructed following step 3 noted at the beginning of Subsection 3.2.3 and the immersed boundary condition is then achieved.

In practice, no-slip boundary condition is imposed on the immersed boundary, that is, $\tilde{u}_i = 0$ treated as a Dirichlet boundary condition. If the pressure condition is considered, a zero-gradient type condition is used, *i.e.*, $\partial \tilde{p}/\partial n = 0$, which can be rewritten as $\partial(\Delta \tilde{p})/\partial n = -\partial \tilde{p}_r/\partial n$. This can be treated as a Neumann boundary condition.

3.2.4 Wall modeling at the immersed boundary

As discussed in Section 2.2.3 , when LES-NWM is used, the wall models should be used to evaluate the near-wall flows. The numerical strategy of wall modeling has been implemented

in IBM-ARPS for the simulation of high Reynolds number turbulent flows using a coarse grid.

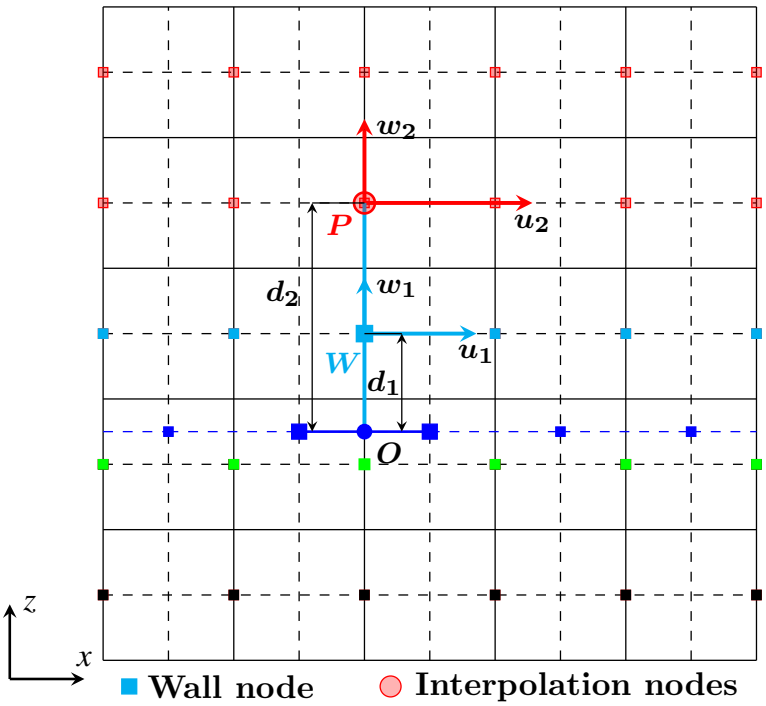


Figure 3.12 An example of applying wall modeling at the wall node W. P is a fluid node, a grid interval above W.

In our works, the strategy of wall modeling proposed by Mason and Callen (1986) is used. Within this wall model, the fluid velocity at the first computational point is reconstructed using the logarithmic law including the roughness effects. When the immersed boundary method is applied, the first computational point is changed to the wall node, which is defined as the first grid node near the interface in the fluid phase, thus applying the wall modeling results in reconstructing the velocity field on the wall nodes.

Firstly, considering a plate immersed surface as shown in Figure 3.12, assuming that the law of the wall f_w is verified locally and instantaneously for the streamwise velocity, the relation between (u_1, w_1) and (u_2, w_2) is given by:

$$u_1 = \frac{f_w(d_1^+)}{f_w(d_2^+)} u_2 \quad \text{and} \quad w_1 = \frac{d_1}{d_2} w_2, \tag{3.14}$$

where d_1 and d_2 are the displacements from the points W and P to the immersed boundary in the vertical direction, and d_1^+ and d_2^+ are the corresponding non-dimensional ones by the viscous lengthscale. The friction velocity u_* can be calculated by solving the equation

$u_2 = u_* f_w(d_2 u_*/\nu)$. Equation (3.14) proposes that the streamwise velocity is reconstructed by the law of the wall while the normal velocity is evaluated by the linear interpolation. In this simple case, the velocity reconstruction in the streamwise direction is completely independent of that in the normal direction, which greatly simplifies the numerical implementation.

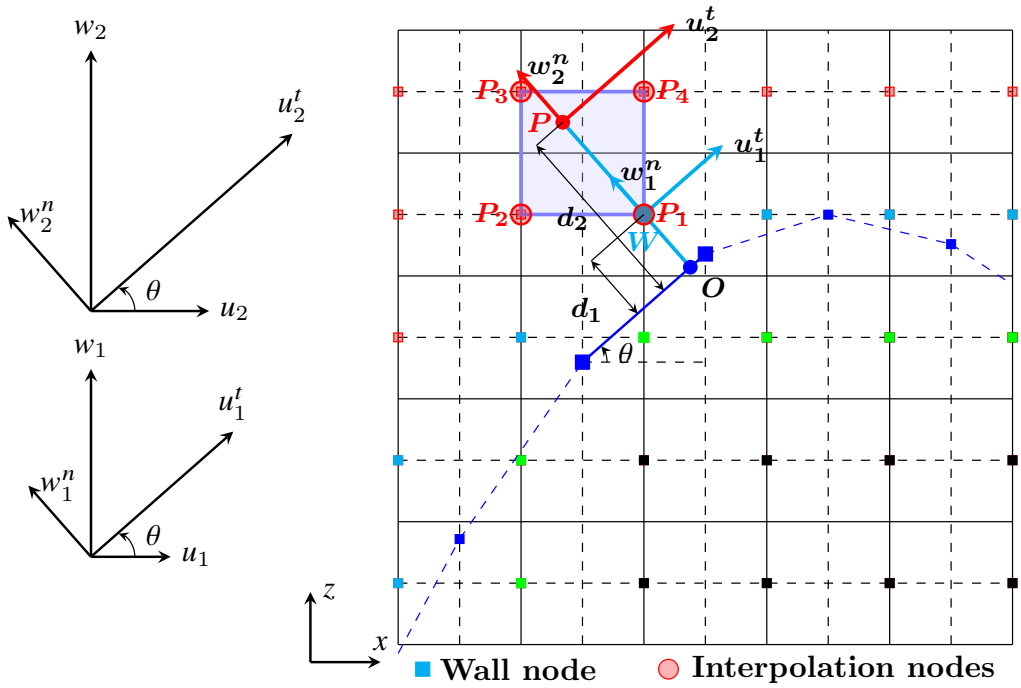


Figure 3.13 An example of applying wall modeling at the wall node W. P is the interpolation point for W. P_1, P_2, P_3, P_4 are the four neighbors of P.

For a two-dimensional immersed boundary of any shape, taking into account the bed slope effect, we decompose the velocity into the normal one and the tangential one [DeLeon et al. (2018)]. As shown in Figure 3.13, the tangential and normal velocities (u^t, w^n) are given by:

$$\begin{bmatrix} u^t \\ w^n \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix}, \tag{3.15}$$

where θ is the local bed slope and (u, w) are the fluid velocities in x- and z- directions. Assuming that the law of the wall f_w is verified locally and instantaneously for the tangential velocity, similar to Equation (3.14), the relation between (u_1^t, w_1^n) and (u_2^t, w_2^n) is given by

$$\begin{bmatrix} u_1^t \\ w_1^n \end{bmatrix} = \begin{bmatrix} f_w(d_1^+)/f_w(d_2^+) & 0 \\ 0 & d_1/d_2 \end{bmatrix} \begin{bmatrix} u_2^t \\ w_2^n \end{bmatrix}, \tag{3.16}$$

where d_1 and d_2 are the displacement from the points W and P to the immersed boundary in the normal direction, and d_1^+ and d_2^+ are the corresponding non-dimensional ones by the viscous lengthscale. The friction velocity u_* can be calculated by solving the equation $u_2^t = u_* f_w(d_2 u_*/\nu)$. Equation (3.16) proposes that the tangential velocity is reconstructed by the law of the wall and the normal velocity is evaluated by the linear interpolation. Combining Equation (3.15) and Equation (3.16), the reconstructed velocities on the wall node W are obtained:

$$\begin{bmatrix} u_1 \\ w_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}^{-1} \begin{bmatrix} f_w(d_1^+)/f_w(d_2^+) & 0 \\ 0 & d_1/d_2 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u_2 \\ w_2 \end{bmatrix}. \tag{3.17}$$

Expanding Equation (3.17), we obtain the wall model formulation:

$$u_1 = \left(\frac{f_w(d_1^+)}{f_w(d_2^+)} \cos^2\theta + \frac{d_1}{d_2} \sin^2\theta \right) u_2 + \left(\frac{f_w(d_1^+)}{f_w(d_2^+)} - \frac{d_1}{d_2} \right) \cos\theta \sin\theta w_2, \tag{3.18a}$$

$$w_1 = \left(\frac{f_w(d_1^+)}{f_w(d_2^+)} - \frac{d_1}{d_2} \right) \cos\theta \sin\theta u_2 + \left(\frac{f_w(d_1^+)}{f_w(d_2^+)} \sin^2\theta + \frac{d_1}{d_2} \cos^2\theta \right) w_2. \tag{3.18b}$$

Equation (3.18) with $\theta = 0$ reduces to the original rough wall model of Mason and Callen (1986). It should be noted that Equation (3.18) shows a coupling between the velocity u and w when the local bed slope is not zero, *i.e.*, $\theta \neq 0$. This is different from the velocity reconstruction at the ghost nodes, which can be separately done at the staggered grid.

In ARPS, the implementation of wall modeling is divided into three steps:

1. Finding the point P, whose normal distance from the immersed boundary is d_2 as shown in Figure 3.13. We set the difference of d_1 and d_2 equal to the grid size, *i.e.*, $d_2 - d_1 = \sqrt{(\Delta x)^2 + (\Delta z)^2}$.

2. Evaluating the velocity (u_2, w_2) at the point P. In practice, the velocities u_2 and w_2 are obtain through the bilinear interpolation (Equation (3.8)) using its four neighbors P_1, P_2, P_3 and P_4. The neighbors as interpolation points are identified according to five cases shown in Figure 3.7, Figure 3.8, Figure 3.9, Figure 3.10 and Figure 3.11, respectively.

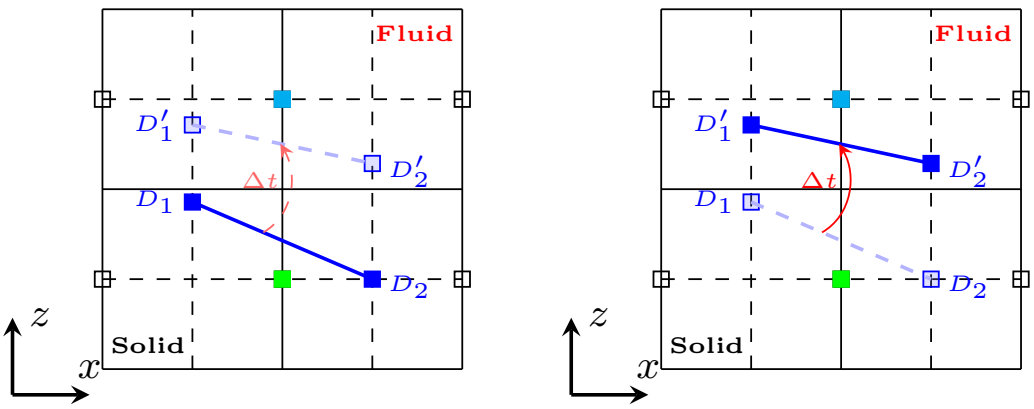
3. Reconstructing the velocity (u_1, w_2) at the wall point W. Firstly, we calculate the tangential velocity u_2^t at the point P using Equation (3.15). Then supposing that u_2^t satisfies the law of the wall f_w, the friction velocity u_* can be obtained by solving $u_2^t = u_* f_w(d_2 u_*/\nu)$. Finally, we calculate d_1^+ and d_2^+ and reconstruct u_1 and w_1 using Equation (3.18).

Note that the law of the wall f_w of boundary layers is introduced in detail in Appendix A. In numerical simulations, we use the formulation (Equation (A.8) and Equation (A.9)) with accounting for the different roughness regimes. The wall modeling is applied after the flow reconstruction at every small time step. For the further development, a shear stress near-wall model will introduced in IBM-ARPS. Like the velocity model, the shear stress tensor at the wall nodes should be reformulated in terms of tangential and normal components with accounting the bed slope and then be reconstructed according to the common wall models for the plate walls. This will be complicated since the shear stress tensor has six independent components for a general case.

3.2.5 Instability analysis of moving boundaries

As the flow field at the current time step is used to get the solution at the next time step, non-physical value near the interface may be obtained if the role of ghost nodes changes. In the simulation of wind erosion, the time step of flow simulation is usually much smaller than the characteristic time scale of particle transport, that is, $\Delta t \ll \tau_p$. Thus, the immersed boundaries can not pass through more than one grid cell in a large time step. In this case, there are three possible situations of the immersed boundary motion:

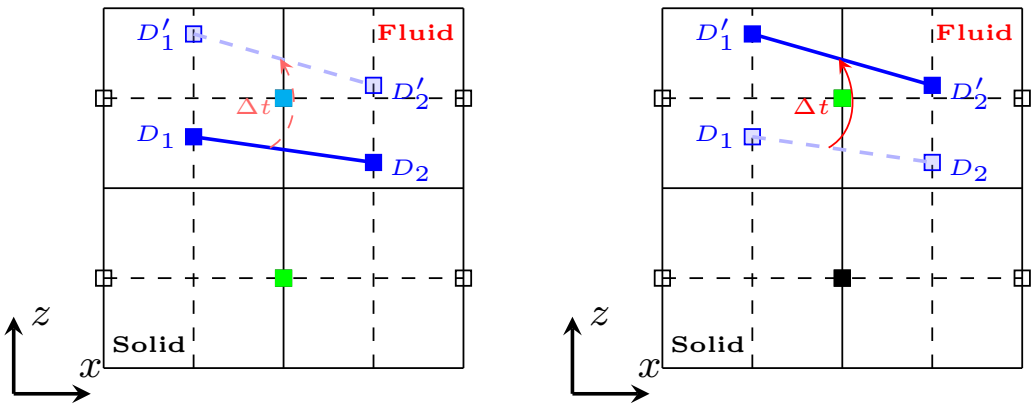
- Situation 1: Moving between two grid nodes. As shown in Figure 3.14, the immersed boundary does not pass any computational grid nodes. Thus, the computational grid nodes do not change their own role and no numerical stability problems arise.



(a) Current time step $t = t_0$ (b) Next time step $t = t_0 + \Delta t$

Figure 3.14 Situation 1 of the immersed boundary motion.

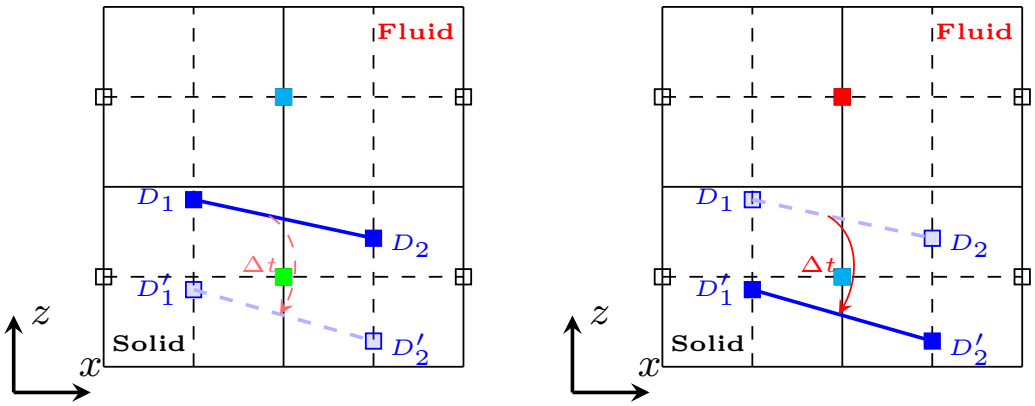
- Situation 2: Moving toward the fluid and passing a wall node. As shown in Figure 3.15, the ghost node changes to a solid node and the wall node becomes a ghost node. In this case, it does not cause the problems of numerical instability, as the flow field at the new ghost point has a history from the previous time step.



(a) Current time step $t = t_0$ (b) Next time step $t = t_0 + \Delta t$

Figure 3.15 Situation 2 of the immersed boundary motion.

- Situation 3: Moving toward the solid and passing a ghost node. As shown in Figure 3.16, the ghost node becomes a wall nodes and the nearest solid node changes to a ghost node. In this case, numerical instability may happen as the flow field at the new wall node is not physical and the forcing at the new ghost node loses the history effect.



(a) Current time step $t = t_0$ (b) Next time step $t = t_0 + \Delta t$

Figure 3.16 Situation 3 of the immersed boundary motion.

Nabi et al. (2012) found that the velocity derivatives near the immersed boundary are not accurately calculated, since nonphysical velocities are introduced in the situation 3. In this work, the procedure of wall modeling is applied to construct the velocity at the wall nodes, and just help to correct the nonphysical value of the velocity in the situation 3. This treatment is similar to the field-extension procedure proposed by Yang and Balaras (2006), which is introduced to treat the grid points changing from solid phase to fluid phase. Therefore, the reconstruction method coupled with wall modeling can be directly applicable to moving boundary problems. In the simulation of wind erosion, a test of this immersed boundary method found that the numerical instability sometimes takes place if the wall modeling procedure is removed.

3.3 Implementation of IBM in ARPS

If the immersed boundary method is implemented in ARPS, some modifications will be made in the modules of initialization, time integration and parallelization. In this section, the detailed implementation of IBM in the ARPS code is given and the special treatment for moving boundaries is introduced.

3.3.1 Initialization

In ARPS, the standard initialization module is used to specify the model control parameters, to generate the terrain-following grid, and to create the initial flow field. Firstly, the physical and numerical parameters of numerical cases are assigned by reading the input file. According to the MPI technique, the computational domain is then divided into several sub-domains. In each sub-domain, the computational grid is set up and the corresponding Jacobian matrix is numerically calculated. At last, the base-state and time-dependent variation of physical variables are initialized using the internal functions.

Figure 3.17 shows the initialization procedure after the implementation of IBM into ARPS. An additional IBM module is implemented after the procedure of model variables configuration. In this IBM module, the immersed boundary is firstly configured and then the computational nodes are classified into the fluid nodes, wall nodes, ghost nodes and solid nodes. At the end of initialization, the value of model variables at the ghost nodes are re-assigned by the reconstruction method, in order to impose the immersed boundary condition.

Since the scalar transport is not the research topic in this work, there is no treatment for scalars at the immersed boundary. The pressure and density are evaluated by the state

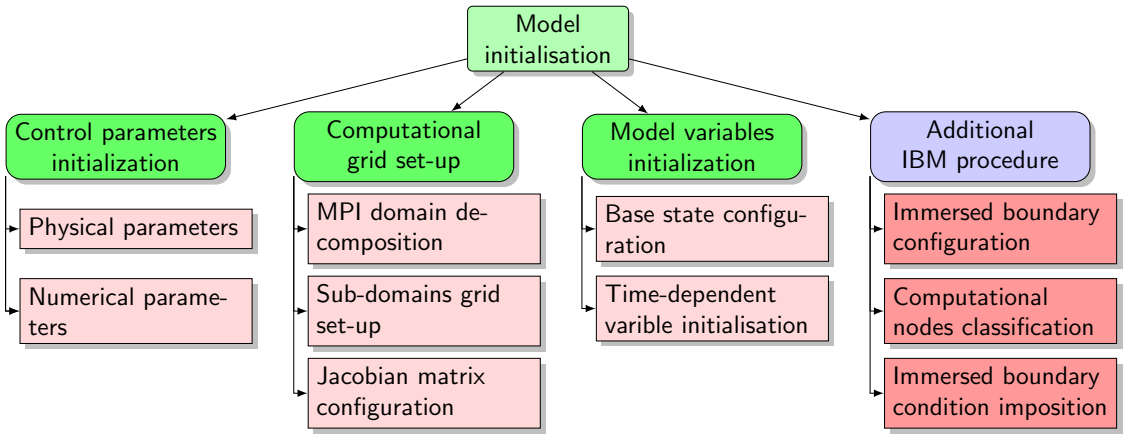


Figure 3.17 Initialization procedure in ARPS with the implementation of IBM. The red boxes denotes the additional initialization IBM modules

equation, and thus do not need a forced boundary condition. The potential temperature is allowed to be freely developed without imposing the forcing.

3.3.2 Time integration

As introduced in Section 2.3, a mode-splitting time integration is used for the numerical time discretization in ARPS. The terms of acoustic and gravity wave modes are advanced every small time step, while the other forcing terms are updated on the large time integration. In particular, the subgrid kinetic energy k_{sgs} used for evaluating the subgrid viscosity ν_{sgs} is advanced on the large time step. In addition, Lagrangian type equation of tracking the particle trajectories has been implemented in the subroutine for solving the TKE equation.

As illustrated in Figure 3.18, an IBM module is added in the procedure of time integration. This module is implemented in the subroutine of small time integration. Within this module, we reconstruct the field of velocity and pressure at the ghost nodes at the end of each small time advancement. If a coarse grid is used, the wall modeling is used to correct the fluid velocity at the wall nodes. In addition, the interior treatment at the solid nodes is also implemented in the reconstruction modules. Obviously, the modification at the interior nodes can affect the accuracy of velocity or scalar derivatives near the immersed boundary. There are two possible options available for the artificial treatment of the interior nodes. One is to impose the null velocity at the solid nodes. This is equivalent to applying the direct forcing at each interior node as suggested by Saiki and Biringen (1996). The other is to leave the interior domain free to develop a nonphysical flow without any treatments. From the studies of Fadlun et al. (2000), Iaccarino and Verzicco (2003) and Lundquist et al. (2010), it was

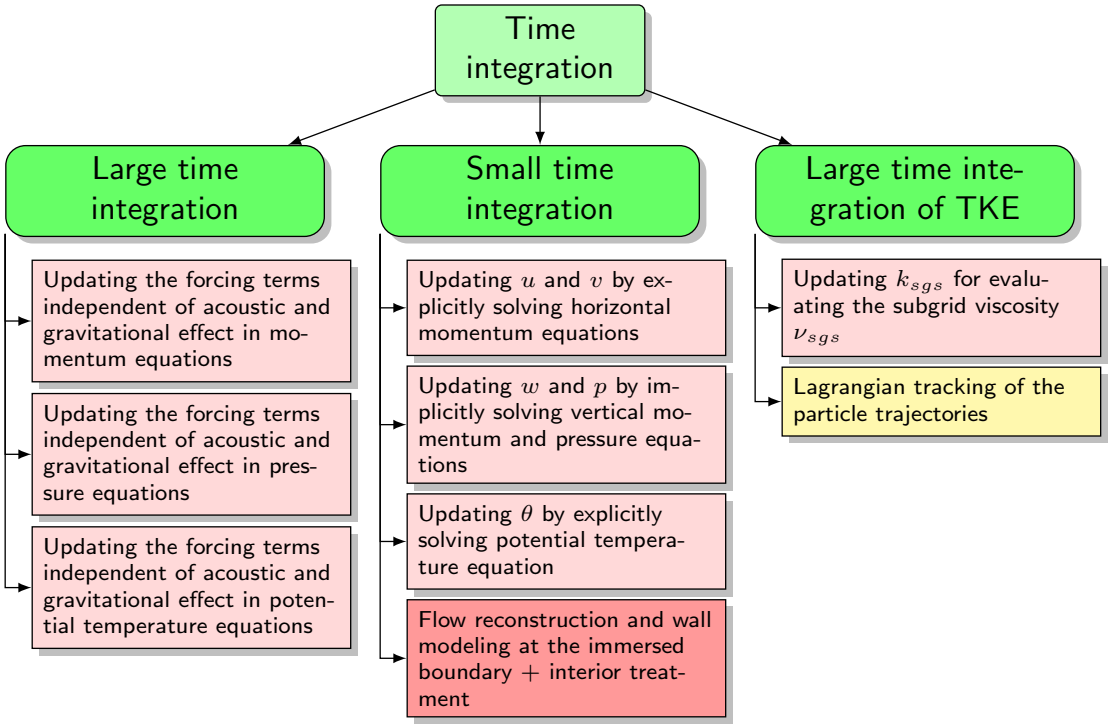


Figure 3.18 Time integration procedure in ARPS with the implementation of IBM. The red box denotes the flow reconstruction and wall modeling IBM modules.

found that no influence on the flow fields at the fluid nodes appeared in test cases of different interior treatments. In this thesis, we also investigate the effect of interior treatment on the implementation of IBM in ARPS by carrying out numerical test cases

3.3.3 Parallelization

The IBM modules implemented in ARPS are parallelized in the horizontal directions. In each sub-domain, the physical variables at the boundaries should be assigned according to the boundary conditions imposed on the real boundaries, or through the message passing from the neighbors, which are not located at the real boundaries. At the end of IBM modules, the message passing is used again to exchange the information at the boundaries with its neighbors since the value of some physical variables near the immersed boundary is modified by the flow reconstruction procedure. In particular, if some nodes, used for the interpolation, are located out of the sub-domain, then an additional message passing will be applied to obtain the model variable value at these nodes.

3.3.4 Moving boundaries

The ghost-cell finite-difference method introduced in Section 3.2 is directly used in the simulation of moving boundary problems. As shown in Figure 3.19, the computational nodes must be re-classified and the flow field at the ghost and wall nodes have to be reconstructed again every time the immersed surface is updated according to the physical models of wind erosion. In addition, when the immersed boundary is moving in a fixed grid, the role of computational node near the interface may change during time integration, for example, a ghost node becomes a wall node, or conversely.

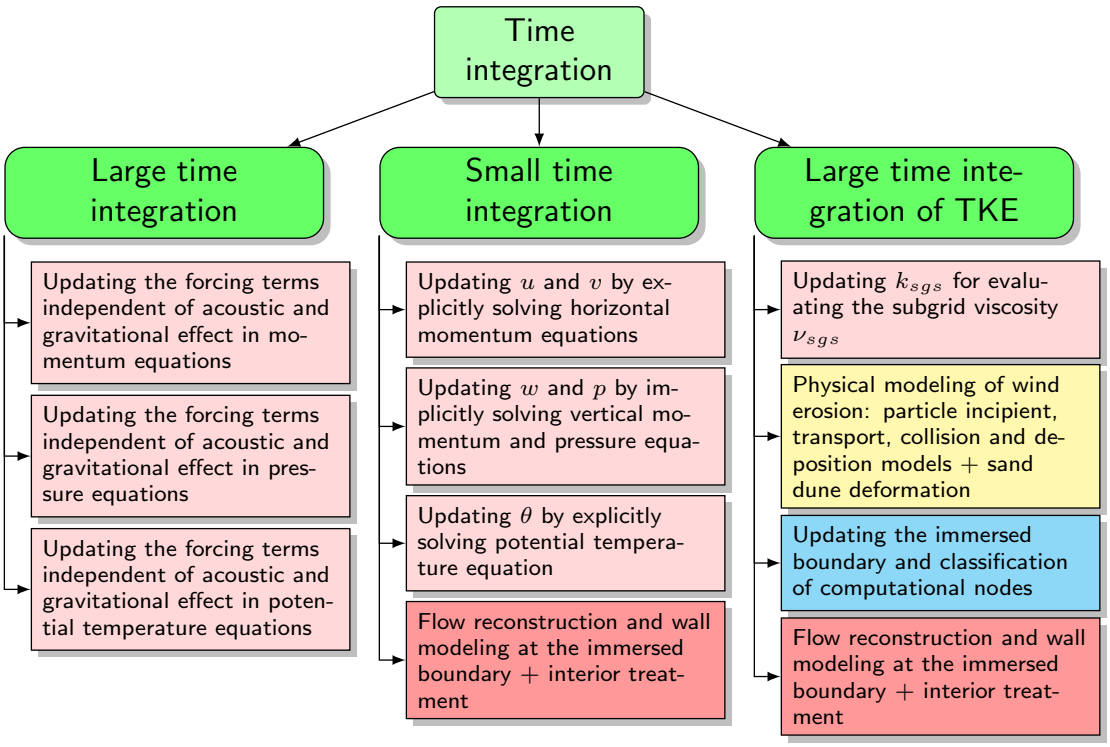


Figure 3.19 Time integration procedure for wind erosion problem in ARPS with the implementation of IBM.

In Figure 3.19, integrated physical models of wind erosion, including the particle entrainment, transport, deposition and particle/surface interaction models, are implemented to simulate the wind erosion in an atmospheric boundary layer. In the case of sand dune deformation, one particle is removed from a grid cell once the criterion of particle incipient motion is satisfied. Commonly, the immersed interface is not able to completely cross one or more grid cells every time step integration, as the time step is usually assigned by a small value. For this case, the problems of numerical stability are discussed in Subsection 3.2.5.

3.4 Validation of IBM-ARPS

In this section, two canonical simulation cases of a turbulent boundary layer (TBL) over a fixed obstacle are performed to verify the accuracy of the immersed boundary method implemented in ARPS. The first numerical case is a TBL over a small Gaussian dune, in which the computational grid near the wall is very fine. The second is a TBL over a large sinusoidal dune, in which the computational grid near the wall is relatively coarse. In both cases, recirculation region characteristics, fluid velocity profiles and Reynolds stress profiles at different stream-wise positions as well as the friction velocity are presented.

3.4.1 Turbulent boundary layer over a small Gaussian dune

The first canonical simulation case of a turbulent boundary layer flow over a small Gaussian dune is performed to verify the accuracy of the immersed boundary method coupled with ARPS. In this case, the dune height is much smaller than the thickness of the boundary layer, *i.e.*, the blockage ratio is $H/\delta = 1/7$. In practice, a fine grid can be used for this simulation, especially the vertical grid spacing in the near-wall region is able to take the value of order of the characteristic length in the viscous sublayer using the vertical grid stretching approach, *i.e.*, $\Delta z^{+}_{min} = 15$ in the buffer layer. This is helpful to obtain an accurate numerical solution of near-wall flow fields. Using the simulated results, recirculation region characteristics, longitudinal velocity profiles as well as Reynolds stress profiles at different streamwise positions are presented and compared with the experimental results of Simoëns et al. (2015).

Experimental configuration: The experimental set-up is sketched in Figure 3.20. For the

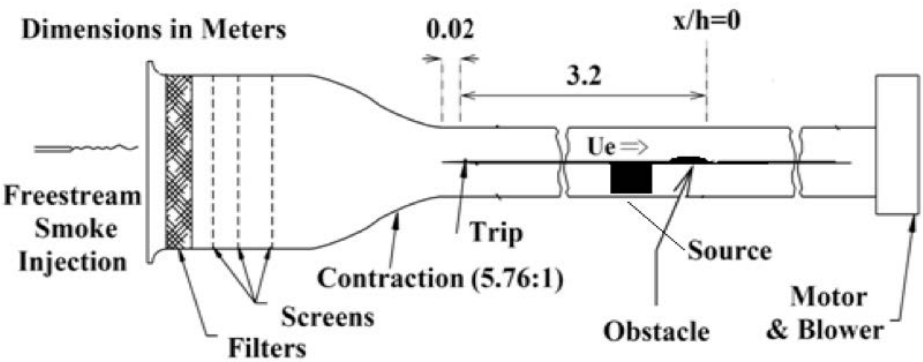


Figure 3.20 Sketch of the wind tunnel set up of the experiment of Simoëns et al. (2015). Taken from Huang (2015).

wind tunnel flow, the external velocity is set to $U_e = 11.2$ m/s, the thickness of the boundary

layer is $\delta = 0.07$ m and the friction velocity is $u_* = 0.46$ m/s. Small particles of an average diameter $d_p = 200$ μm are uniformly and closely spread and glued on the ground in order to obtain the rough wall condition. The obstacle shape is a Gaussian dune given by

$$h(x,y) = H \exp\left(-\left(0.7\frac{x}{H}\right)^2\right), \tag{3.19}$$

where $H = 0.01$ m is the dune height. Equation (3.19) shows that the dune height is independent on y, and thus is homogeneous in the spanwise direction. The particle image velocimetry (PIV) is used for the measurement of the fluid velocity.

Numerical configuration: The computational domain is illustrated in Figure 3.21. The

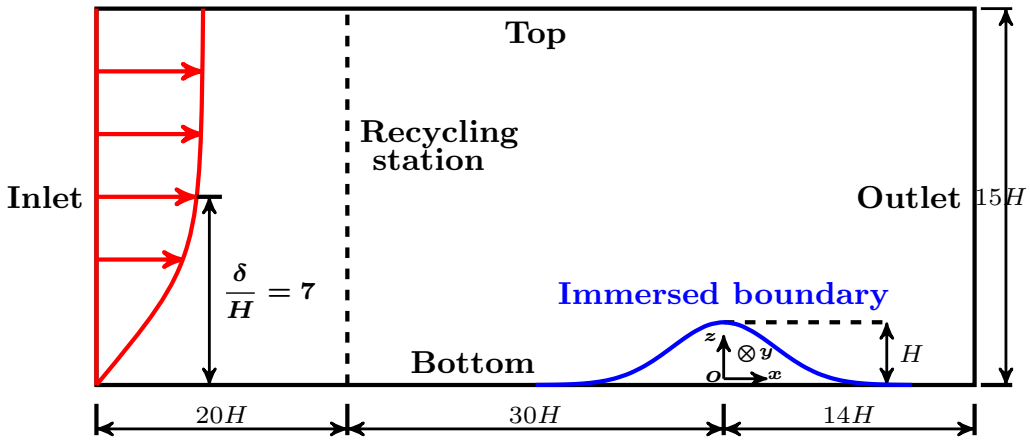


Figure 3.21 Sketch of the computational domain of TBL over a small Gaussian dune.

Gaussian dune is located at the origin, and is considered as the immersed boundary, on which no-slip boundary condition is imposed using the immersed boundary method. At the beginning of the domain, there is a flow development region from $x = -50H$ to $x = -30H$ for generating the inlet flow using the recycling method [Lund et al. (1998)]. The physical parameters are shown in detail in Table 3.1 and the numerical parameters in Table 3.2. The external velocity U_e, the boundary layer thickness δ and the friction velocity u_* are given to generate the inlet streamwise velocity profile and the initial base state using an improved log-law formula [Huang et al. (2016)]. The size of the domain is determined by L_x, L_y and L_z and N_x, N_y and N_z are the number of nodes in the three direction. The grid is uniform in the streamwise and spanwise direction with the assigned values Δx and Δy. It is slightly stretched in the wall-normal direction with an average grid increment Δz and a refined increment near the wall Δz_{min}. Note that the refined region should cover the Gaussian dune to ensure the accuracy of the interpolation used in the immersed boundary method. The grid testing has

been done in the thesis of Huang (2015). Here, we chose the grid size shown in Table 3.2, which has been examined and given a good performance for the numerical simulation.

H(m)	δ(m)	U_e(m·s^{-1})	u_*(m·s^{-1})	Re_H	Re_δ	Re_τ
0.01	0.07	11.2	0.46	7467	52267	2147

Table 3.1 Physical parameters given for simulating TBL over a small Gaussian dune. The Reynolds numbers are defined by $Re_H = U_e H/\nu$, $Re_\delta = U_e \delta/\nu$ and $Re_\tau = u_* \delta/\nu$.

$N_x \times N_y \times N_z$	L_x/H	L_y/H	L_z/H	$\Delta x/H$	$\Delta y/H$	$\Delta z/H$	$\Delta z_{min}/H$	Δx^+	Δy^+	Δz^+_{min}
$643 \times 63 \times 100$	64	6	15	0.1	0.1	0.15	0.05	30	30	15

Table 3.2 Numerical parameters given for simulating TBL over a small Gaussian dune.

Test cases: As presented in Table 3.3, five numerical test cases with different treatments are carried out using the same physical and numerical parameters shown in Table 3.2. Test A is performed using ARPS without the implementation of immersed boundary method as in Huang (2015). The simulated results of Test A are considered as the reference for the comparisons with the results by IBM-ARPS in the investigation of the accuracy of the immersed boundary method. Test B is conducted by IBM-ARPS, in which only the no-slip boundary condition is imposed on the immersed surface and the fluid velocity inside the dune is enforced to zero. Test C is performed using IBM-ARPS without any treatments inside the dune. Test D is simulated by IBM-ARPS with near-wall treatments on the immersed boundary. Test E is conducted by IBM-ARPS with imposing the zero-gradient condition of the pressure fields on the immersed boundary. The latter three numerical cases are used to investigate the influence of different treatments on the performance of the IBM-ARPS simulation.

Test	ARPS or IBM-ARPS	Interior treatment	Wall modeling	Pressure reconstruction
Test A	ARPS	-	-	-
Test B	IBM-ARPS	Zero velocity	Non	Non
Test C	IBM-ARPS	Non	Non	Non
Test D	IBM-ARPS	Zero velocity	Yes	Non
Test E	IBM-ARPS	Zero velocity	Non	Yes

Table 3.3 Test cases with different treatments for a TBL over a small Gaussian dune.

3.4.2 Results and comparisons

According to the works of Huang (2015), recirculation zones are characterized not only by the separation position x_{sep} and the reattachment position x_{reat}, but also by a maximal height L_{AB} and a major axis L_{CD} as shown in Figure 3.22. x_{sep} is defined as the minimum streamwise position of the recirculation zone and x_{reat} the maximum one. L_{AB} is defined as the maximum distance from the dune surface to the recirculation streamline in the vertical direction and L_{CD} the longest one between the two extremities of the recirculation bubble. The ratio L_{AB}/L_{CD} is then a useful parameter to characterize the shape of the bubble.

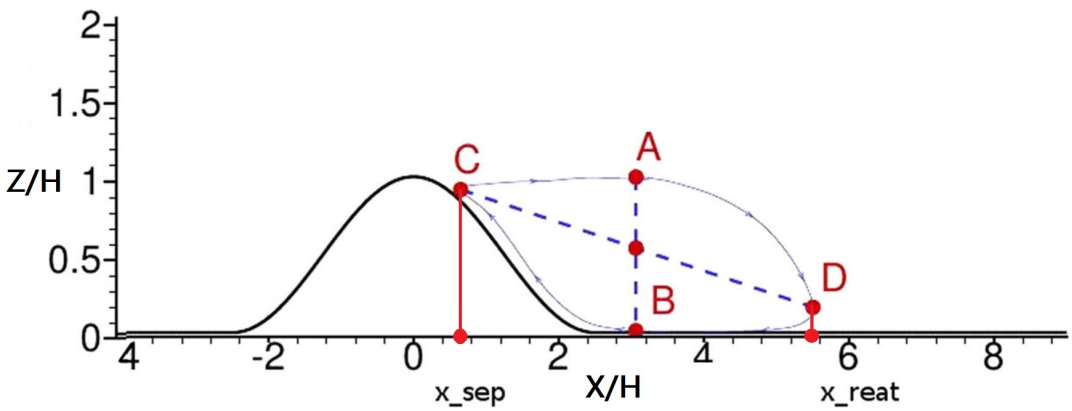
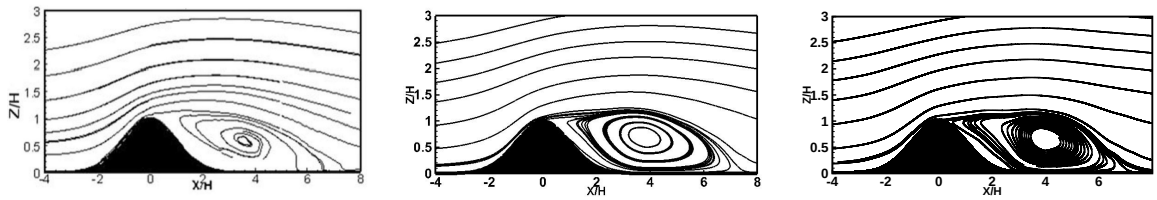


Figure 3.22 Sketch of the key characteristics of the recirculation zone behind a 2D obstacle. Figure adapted from Huang et al. (2018).

Figure 3.23 shows the recirculation zones obtained by the experiments, by the ARPS simulation and by the IBM-ARPS simulation. In Figure 3.23, the coordinates are scaled by the dune height H and the origin is set at the dune center. Table 3.4 gives the values of key parameters characterizing the form of recirculation zone, i.e., x_{sep}, x_{reat}, L_{AB}, L_{CD} and L_{AB}/L_{CD}. The separation position x_{sep} is the same between the two simulations but slightly larger than that of the experiment, while x_{reat} given by the ARPS simulation is a little bigger than the experimental value and it becomes a little smaller when the immersed boundary method is introduced. The length L_{AB} is completely the same between the three cases, whereas L_{CD} obtained by the ARPS is slightly greater than that of the experiment and L_{CD} obtained by the IBM-ARPS is a little smaller. The ratio of L_{AB} and L_{CD} is nearly the same and is around 0.18. This means that the recirculation zone simulated by the ARPS becomes slightly smaller when the immersed boundary method is used, and that both the simulated recirculation zone sizes are consistent with the experimental results of Simoëns et al. (2015).



(a) Experiment of Simoëns et al. (2015).

(b) Numerical simulation using the ARPS (Test A).

(c) Numerical simulation using IBM-ARPS (Test B).

Figure 3.23 Mean streamlines over a Gaussian dune and recirculation zones on the lee side.

	x_{sep}/H	x_{reat}/H	L_{AB}/H	L_{CD}/H	L_{AB}/L_{CD}
Experiment	0.2	6.9	1.2	6.7	0.179
Test A	0.3	7.0	1.2	6.8	0.176
Test B	0.3	6.8	1.2	6.6	0.182

Table 3.4 Characterization of the recirculation zone behind a small Gaussian dune.

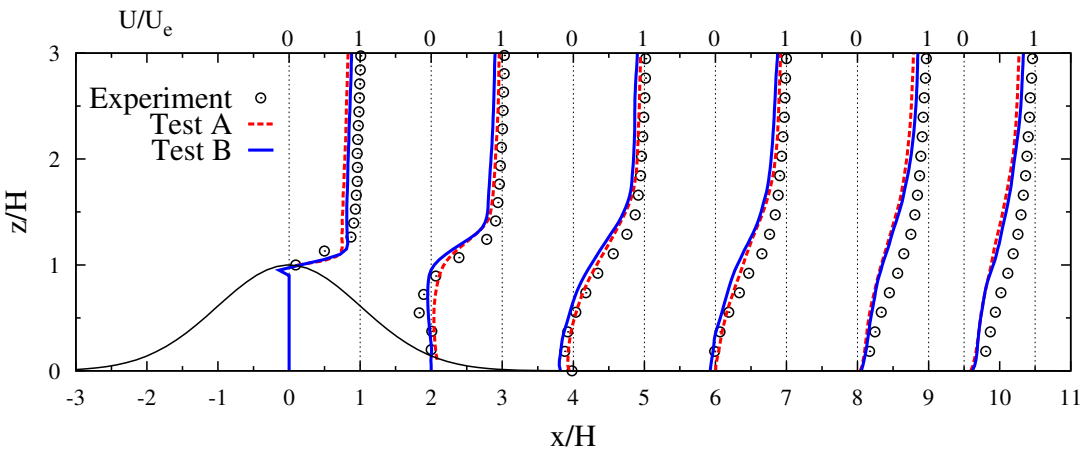


Figure 3.24 Mean longitudinal velocity profiles over a small Gaussian dune. Points: experiments of Simoëns et al. (2015). Dashed lines: ARPS simulation (Test A). Solid lines: IBM-ARPS simulation (Test B).

The mean longitudinal velocities over a Gaussian dune at several different streamwise stations are presented in Figure 3.24. The mean streamwise velocities are normalized by the external one and located at the corresponding stations, *i.e.*, $x = 0.0$, $x = 2.0H$, $x = 4.0H$, $x = 6.0H$, $x = 8.0H$, $x = 9.5H$. The coordinates are scaled by the dune height H and the origin is set at the dune center. It is shown that the flow near the boundary accelerates on the windward side of the dune and then decreases on its lee side. The velocity profiles located in the recirculation zone are reversed. A good agreement is achieved between the two simulation results, even which is slightly smaller than experimental data.

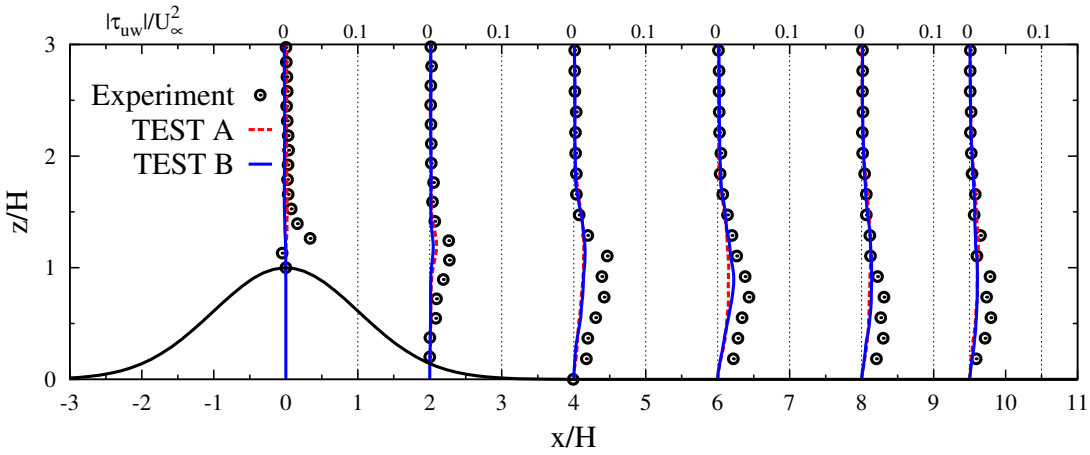


Figure 3.25 Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a small Gaussian dune. Points: experiments of Simoëns et al. (2015). Dashed lines: ARPS simulation (Test A). Solid lines: IBM-ARPS simulation (Test B).

In Figure 3.25, we illustrate profiles of mean Reynolds stress $\langle u'w' \rangle$ over the small Gaussian dune at six different stremwise stations. Here, u' denotes the fluctuation velocity in the streamwise direction and w' the fluctuation in the normal direction. The mean Reynolds stress is scaled by the square of the external velocity. As shown in Figure 3.25, because of the separation of shear flow, the Reynold stress increases obviously in the recirculation zone. A good agreement between the ARPS simulation and the IBM-ARPS simulation is observed, whereas the Reynolds stress obtained by both simulations is smaller than that of experiments. Globally, the agreement on Reynolds stress, particularly near the wall, ensures us the validity of the numerical methods, even though the numerical simulations underestimate the value of Reynolds stress.

Figure 3.26 shows the friction velocity u_* over a small Gaussian dune, which is responsible for the particle incipient motion. The friction velocity is defined by $u_* = \sqrt{\tau_w/\rho}$ as in Huang (2015), where τ_w is wall shear stress and ρ is the fluid velocity, and it is normalized by the initial value. The friction velocity slightly decreases as the boundary layer flow develops, increases on the windward side of the dune and until near the dune crest, suddenly drops sharply due to the presence of flow separation and takes very small values in the recirculation zone, then gradually grows after the reattachment point. Obviously, the friction velocity vanishes at the separation and reattachment points. This is helpful to determine the value of x_{sep} and x_{reat}. In Figure 3.26, the friction velocity obtained by the IBM-ARPS agrees well with that by the ARPS.

In short, the recirculation zone characteristics, mean longitudinal velocity profiles and Reynolds stress profiles obtained by the IBM-ARPS are in a good agreement with the

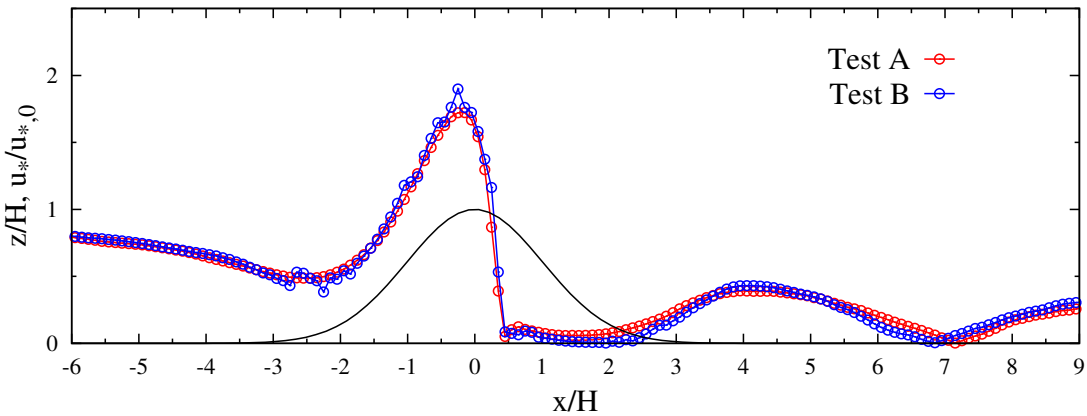
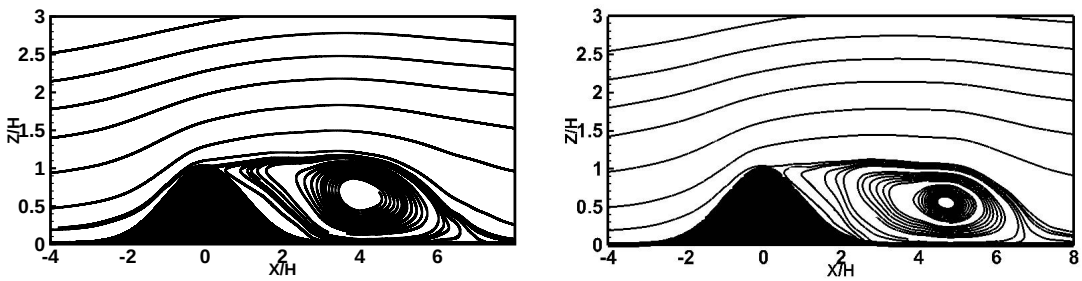


Figure 3.26 Comparison of the friction velocity between the ARPS simulation (Test A) and the IBM-ARPS simulation (Test B).

simulated results by the ARPS and the experimental data of Simoëns et al. (2015). The friction velocity obtained by the IBM-ARPS is totally in agreement with the simulated one by the ARPS without the immersed boundary method. This basically validates the accuracy of the new numerical solver (called the standard IBM-ARPS), in which only no-slip boundary condition is imposed on the immersed boundary.

In the following, we investigate the effect of interior treatment, wall modeling, pressure reconstruction on the standard IBM-ARPS code through the comparisons of recirculation zone characteristics, mean longitudinal velocity profiles, Reynolds stress profiles and the friction velocity between simulated results.

Effect of interior treatment: When the immersed boundary method is used, there are

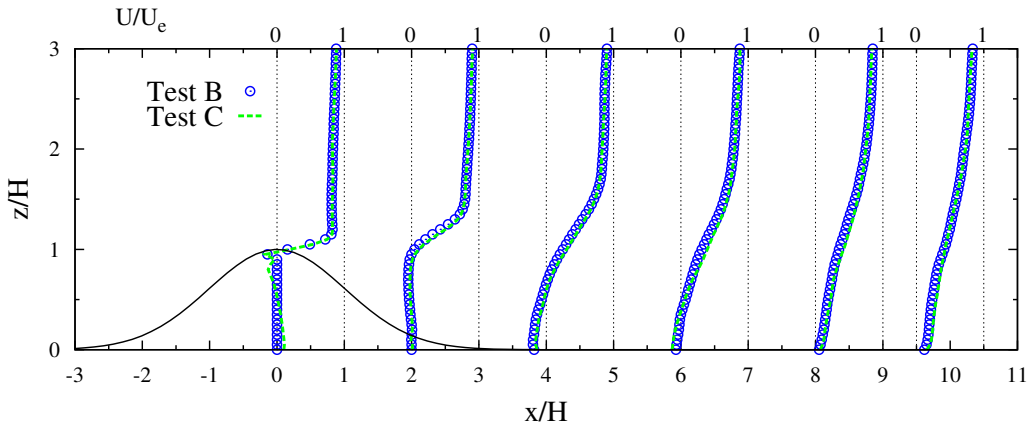


(a) IBM-ARPS simulation with imposing zero velocity on the solid nodes (Test B).

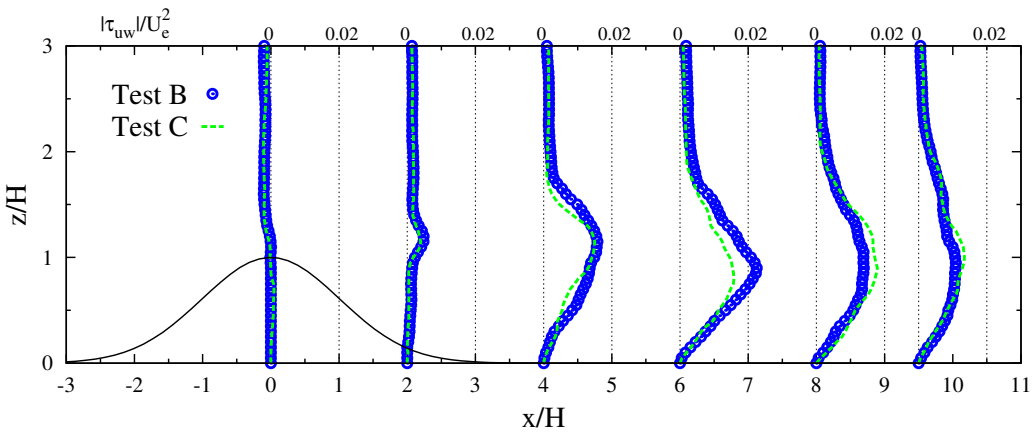
(b) IBM-ARPS simulation without any treatments on solid nodes. (Test C).

Figure 3.27 Mean streamlines over a Gaussian dune and recirculation zones on the lee side.

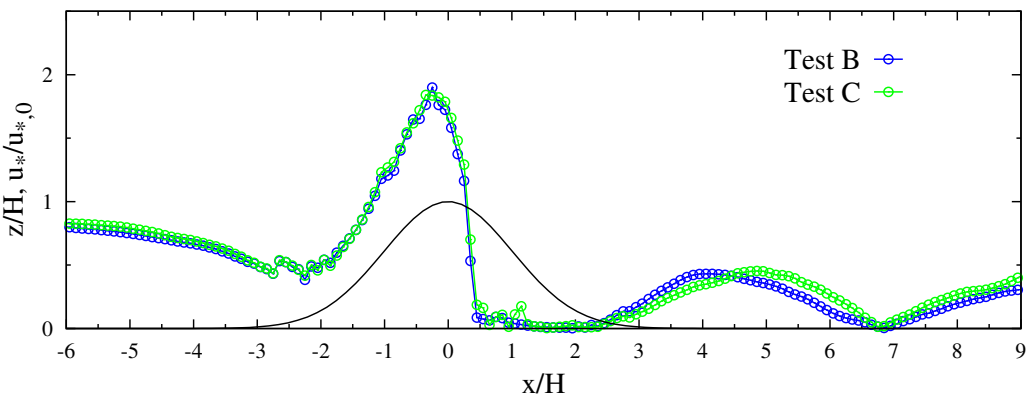
two special treatments for the interior computational nodes (identified as solid nodes in



(a) Mean longitudinal velocity profiles over a small Gaussian dune.



(b) Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a small Gaussian dune.

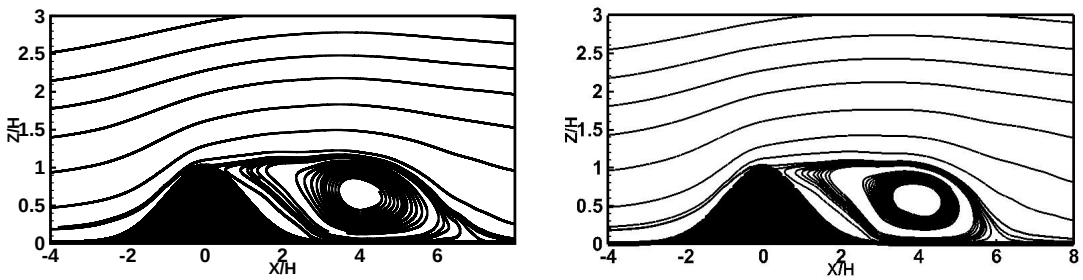


(c) Normalized friction velocity over a small Gaussian dune.

Figure 3.28 Comparison of flow fields simulated by the IBM-ARPS between with imposing zero velocity on the solid nodes (Test B) and without any interior treatments (Test C).

this thesis): imposing zero velocity at the solid nodes or leaving solid nodes free without any modifications. Test B is conducted by the standard IBM-ARPS with directly enforcing null velocity at all solid nodes and Test C without any special treatments in the interior domain. Figure 3.27 shows the recirculation zones behind the small Gaussian dune of the two numerical simulations. The size of these two bubbles is found nearly the same. For instance, it is observed in Figure 3.28(c) that the starting and ending points of null friction velocity are located at the same position for both Test B and Test C , which indicates that x_{sep} and x_{reat} take almost the same values. Figure 3.28 presents mean longitudinal velocity profiles, Reynold stress profiles and friction velocity of Test B and Test C. A good agreement between them is observed, though a slight difference in Reynolds stress and friction velocity is found within the recirculation zone. Briefly, the good agreement between the flow fields of Test B and Test C shows no significant influence of interior treatments on the simulated results. This point is supported by the studies of Fadlun et al. (2000), Iaccarino and Verzicco (2003) and Lundquist et al. (2010), in which the influence of the internal treatments has been checked and no essential influence was found.

Effect of wall modeling: The wall modeling is detailedly described in Subsection 3.2.4.



(a) IBM-ARPS simulation without wall modeling (Test B).

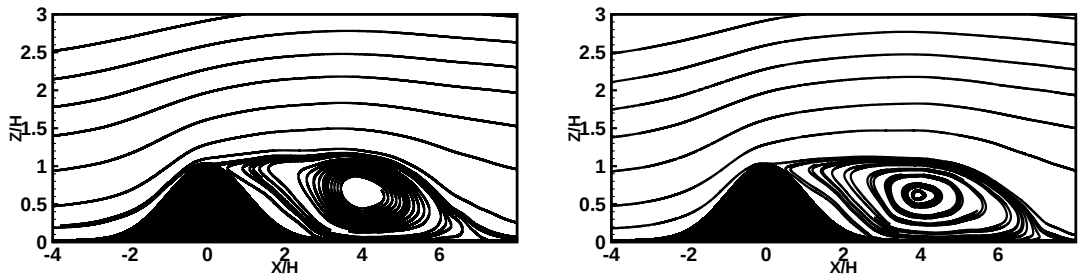
(b) IBM-ARPS simulation with wall modeling (Test D).

Figure 3.29 Mean streamlines over a small Gaussian dune and recirculation zones behind the dune.

It is interesting to investigate the effect of wall modeling on the numerical results when a relatively fine grid is used. If the wall modeling is employed, the fluid velocity on the wall nodes in the fluid domain is required to be reconstructed using a rough law of the wall. Test B is conducted by the standard IBM-ARPS without wall modeling and Test D with wall modeling. Figure 3.29 illustrates the recirculation zones of these two cases. It is found that applying the wall modeling weakly reduces the size of recirculation zone. In Figure 3.31, mean longitudinal velocity profiles, Reynolds stress profiles and friction velocity are presented and compared between Test B and Test D. A good agreement is observed on the

longitudinal velocity as well as the friction velocity, and a slight difference exists for the Reynolds stress. Moreover, as shown in Figure 3.31(c), the starting point of zero friction velocity of Test D is slightly bigger than that of Test B, whereas the ending point of Test D is a little smaller than that of Test B. This indirectly indicates that a smaller recirculation zone is obtained using wall modeling coupled with immersed boundary method. In brief, a very small effect on the simulated fields is observed when the wall modeling is applied in the standard IBM-ARPS simulation with a fine grid.

Effect of pressure reconstruction: Technically, zero-gradient condition for fluid pressure



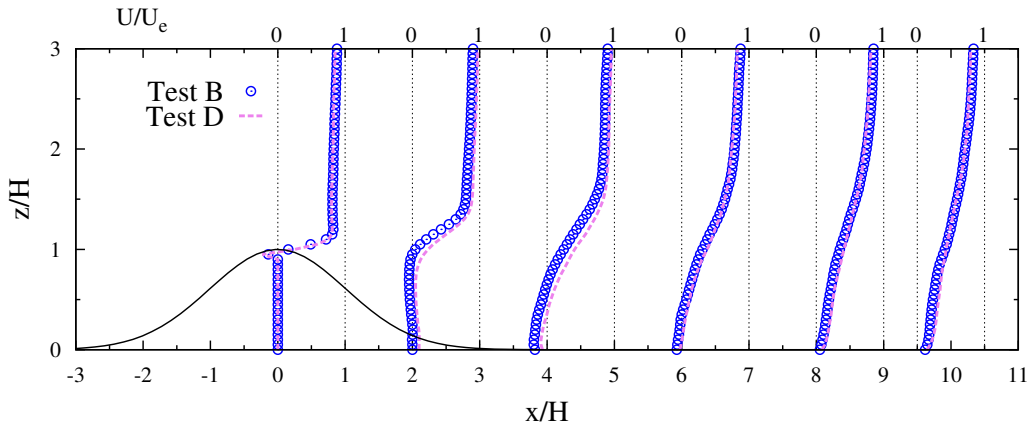
(a) IBM-ARPS simulation without pressure reconstruction (Test B).

(b) IBM-ARPS simulation with pressure reconstruction (Test E).

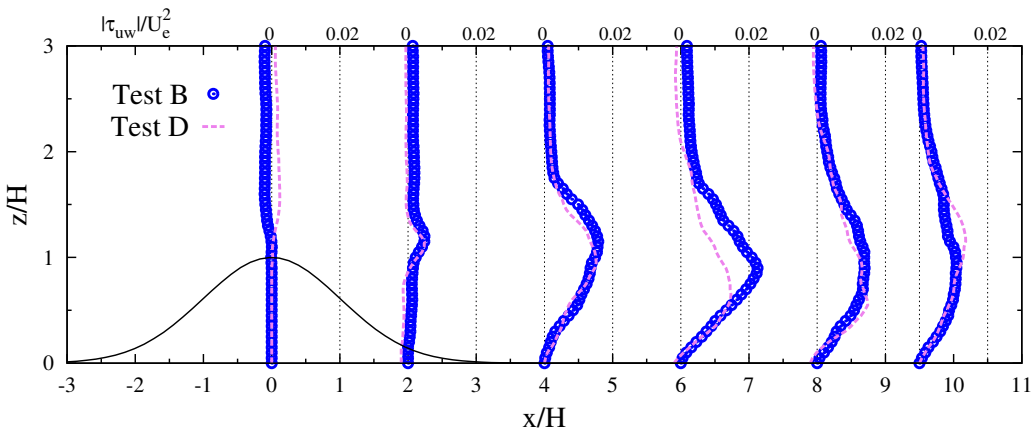
Figure 3.30 Mean streamlines over a Gaussian dune and comparison of the recirculation zones behind the dune.

should be imposed on the immersed boundary while no-slip condition for fluid velocity is imposed on it. Zero-gradient condition can be regarded as the Neumann boundary condition, *i.e.*, $\partial(\Delta p)/\partial n = -\partial p_r/\partial n$. In Test E, zero-gradient condition for pressure is achieved by directly reconstructing the pressure value on the ghost nodes through the interpolation introduced in Subsection 3.2.3. Figure 3.30 shows the recirculation zones behind the dune of Test B and Test E, and that the bubble size of these two cases is almost the same. In Figure 3.32, the mean longitudinal velocity profiles, Reynolds stress profiles and the friction velocity are shown and a perfect agreement between Test B and Test E is observed. Hence, it is concluded that the pressure reconstruction has no effect on the flow fields simulated by the standard IBM-ARPS.

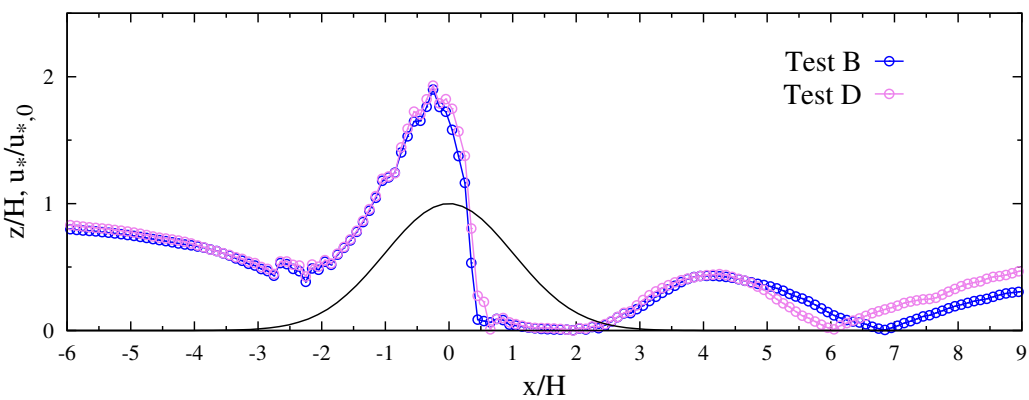
In a word, the good agreement between the (ARPS and IBM-ARPS) simulation results and experimental data demonstrates the accuracy of the new numerical solver with the implementation of immersed boundary method. It is also shown that neither the interior treatment nor the pressure reconstruction has effect on the simulated results, whereas the wall modeling has a slight influence, e.g., weakly reducing the recirculation zone.



(a) Mean longitudinal velocity profiles over a small Gaussian dune.

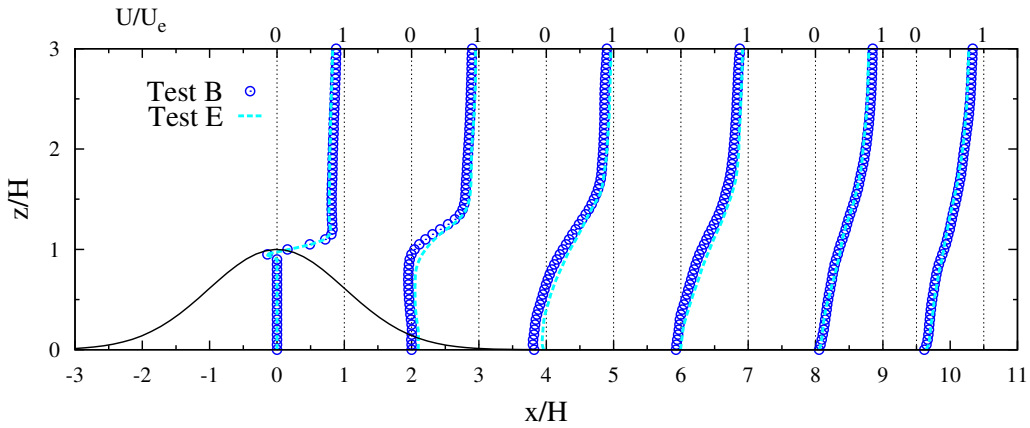


(b) Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a small Gaussian dune.

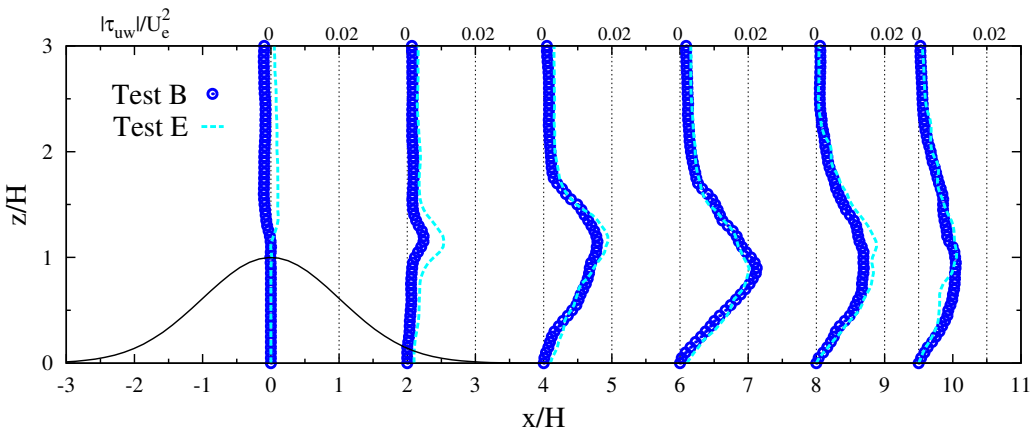


(c) Normalized friction velocity over a small Gaussian dune.

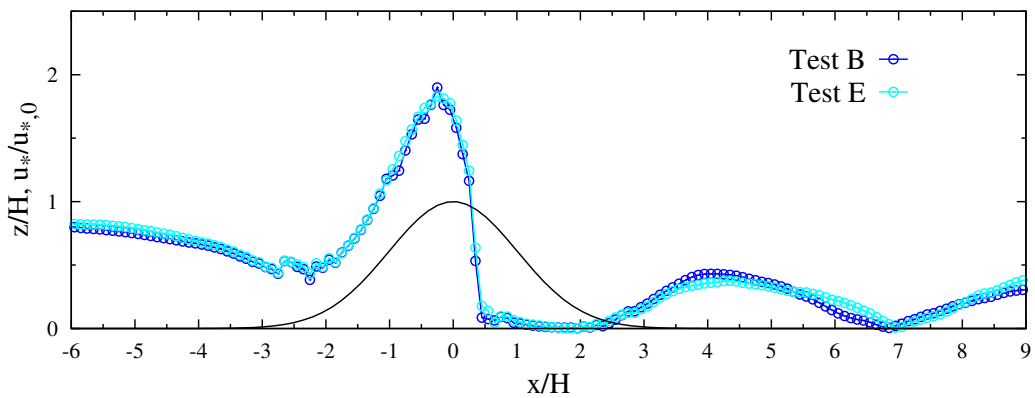
Figure 3.31 Comparison of flow fields simulated by the IBM-ARPS between without wall modeling (Test B) and with wall modeling (Test D).



(a) Mean longitudinal velocity profiles over a small Gaussian dune.



(b) Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a small Gaussian dune.



(c) Normalized friction velocity over a small Gaussian dune.

Figure 3.32 Comparison of flow fields simulated by the IBM-ARPS between without pressure reconstruction (Test B) and with pressure reconstruction (Test E).

3.4.3 Turbulent boundary layer over a large sinusoidal dune

The second canonical simulation case of a turbulent boundary layer flow over a large sinusoidal fixed dune is performed. This case comes from one of the few existing wind-tunnel experiments on wind erosion over a deformable sand dune [Ferreira and Fino (2012)]. The main aim of this thesis is to reproduce the results of sand dune deformation in this experiment by conducting a three-dimensional LES simulation with integrated wind erosion models, even though a two-dimensional RANS simulation has been carried out by Lopes et al. (2013). To achieve a complete wind erosion simulation with sand dune deformation, we firstly focus on the simulation of a boundary layer over a fixed dune. Concretely, in this case, the dune height is close to the thickness of the boundary layer, i.e., the blockage ratio is $H/\delta = 3/5$. In practice, a relatively coarse grid is used in this simulation, since the refined vertical grid spacing is greatly larger than the characteristic length in the viscous sublayer, *i.e.*, $\Delta z^{+}_{min} = 42$ in the log layer. Hence, the presence of this large dune increases the difficulty of near-wall resolution, specially, the estimation of wall shear stress over the dune, the prediction of flow separation and the formation of recirculation zone on the lee side. Based on the simulated results, the flow characteristics over this large dune, including recirculation zones, mean longitudinal velocity, Reynolds stress and friction velocity, are studied, in order to validate the ability of the new numerical solver (IBM-ARPS).

Numerical configuration: In this simulation, the dune shape takes form of:

$$h(x,y) = \frac{H}{2}\left\{1 + \sin\left[\pi \times \left(\frac{x}{3H} + \frac{1}{2}\right)\right]\right\}, \tag{3.20}$$

where H is the dune height. Equation (3.20) shows that h is independent on y, and thus is homogeneous in the y- direction. The computational domain is illustrated in Figure 3.33.

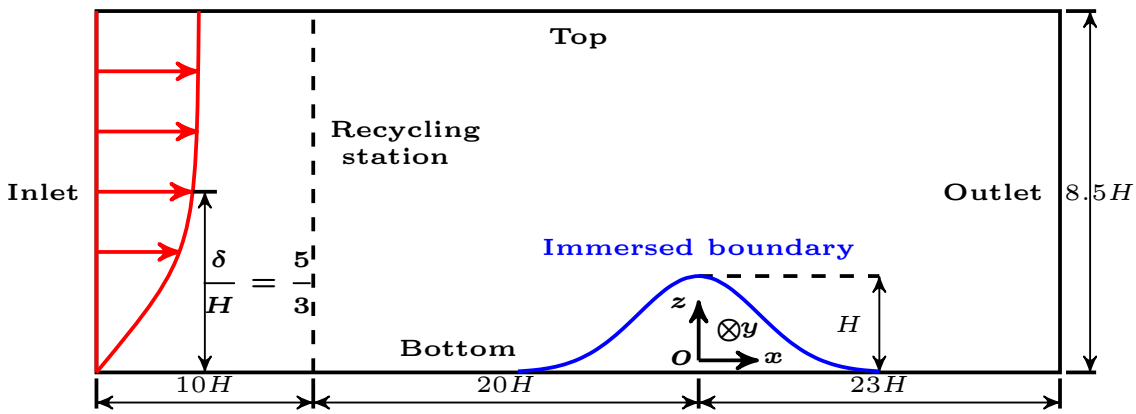


Figure 3.33 Sketch of the computational domain of TBL over a large sinusoidal dune.

The large sinusoidal dune is located at the origin, and is regarded as the immersed boundary. At the beginning of the domain, there is a flow development region from $x = -30H$ to $x = -20H$ for generating the inlet flow using the recycling method. The physical parameters are shown in detail in Table 3.5 and the numerical parameters in Table 3.6. The initial base state of flow field is generated using an improved log-law formula [Huang et al. (2016)], where the external velocity U_e, the boundary layer thickness δ and the friction velocity u_* are given in an input file. L_x, L_y and L_z mean the domain size in the three dimensions and N_x, N_y and N_z are the number of nodes. The grid is uniform in the streamwise and spanwise direction with the assigned values Δx and Δy. It is slightly stretched in the wall-normal direction with an average grid increment Δz and a refined increment near the wall Δz_{min}.

H(m)	δ(m)	U_e(m·s^{-1})	u_*(m·s^{-1})	Re_H	Re_δ	Re_τ
0.06	0.10	9.1	0.42	36400	60667	2800

Table 3.5 Physical parameters given for simulating TBL over a small Gaussian dune. The Reynolds numbers are defined by $Re_H = U_e H/\nu$, $Re_\delta = U_e \delta/\nu$ and $Re_\tau = u_* \delta/\nu$.

$N_x \times N_y \times N_z$	L_x/H	L_y/H	L_z/H	$\Delta x/H$	$\Delta y/H$	$\Delta z/H$	$\Delta z_{min}/H$	Δx^+	Δy^+	Δz^+_{min}
$1063 \times 63 \times 100$	53	6	8.5	0.05	0.10	0.10	0.025	84	168	42

Table 3.6 Numerical parameters given for simulating TBL over a small Gaussian dune.

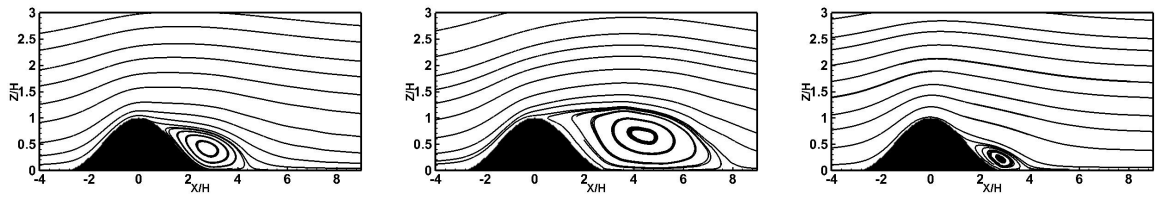
Test cases: Three numerical test cases are conducted for the simulation of a TBL flow over a large sinusoidal dune. Test A′ is carried out using the ARPS with a terrain-following grid. Test B′ is performed by the standard IBM-ARPS. Test C′ is conducted using the standard IBM-ARPS with the additional wall modeling. On the one hand, comparisons between these simulations can show the ability of the standard IBM-ARPS solver. On the other hand, the effect of wall modeling on the IBM-ARPS simulation with a coarse grid can be figured out.

Test	ARPS or IBM-ARPS	Interior treatment	Wall modeling	Pressure reconstruction
Test A′	ARPS	-	-	-
Test B′	IBM-ARPS	Zero velocity	Non	Non
Test C′	IBM-ARPS	Zero velocity	Yes	Non

Table 3.7 Test cases with different treatments for a TBL over a large sinusoidal dune.

3.4.4 Results and comparisons

The recirculation zones obtained by Test A′, Test B′ and Test C′ are shown in Figure 3.34 and the key parameters (x_{sep}, x_{reat}, L_{AB}, L_{CD} and L_{AB}/L_{CD}) are presented in Table 3.8. Compared with the bubble obtained by the ARPS, the recirculation zone simulated by the standard IBM-ARPS is greatly enlarged when a relatively coarse grid is employed. On the contrary, the bubble is severely reduced when the wall modeling is coupled with the standard IBM-ARPS. This indicates that the recirculation zone behind the dune is not able to be accurately predicted by the standard IBM-ARPS with a coarse grid, and that the wall modeling has a great influence on the characteristics of recirculation zone. Unfortunately, there is no experimental data for the recirculation zone size in this case. In both the sinusoidal case and the Gaussian dune case, the maximal dune slope is nearly the same: 32° for Gaussian dune and 33° for the sinusoidal dune. According to the study of Huang (2015), it is found that the blockage ratio has weak influence on the lee side flow separation and thus on the bubble size, whereas an increase of Reynold number Re_H delays the flow separation and then reduce the recirculation zone. Therefore, the recirculation zone behind a large sinusoidal dune is supposed to be much smaller than that behind a small Gaussian dune previous one, since the Reynolds number $Re_H = 36400$ in the former case is much greater than $Re_H = 7467$ in the latter case. This proves that the small recirculation obtained by IBM-ARPS simulation with the wall modeling seems more reasonable and physical. This point is also supported by the RANS simulation of Lopes et al. (2013), in which a very small recirculation zone behind the dune is observed.



(a) Numerical simulation using the ARPS (Test A′).

(b) Numerical simulation using IBM-ARPS (Test B′).

(c) IBM-ARPS simulation with wall modeling (Test C′).

Figure 3.34 Mean streamlines over a large sinusoidal dune and recirculation zones behind the dune.

Figure 3.35 shows the mean longitudinal velocity profiles over a sinusoidal dune at seven different streamwise stations, *i.e.*, $x = -1.5H$, $x = 0.0H$, $x = 1.5H$, $x = 3.0H$, $x = 4.5H$, $x = 6.0H$ and $x = 7.5H$. The velocity is scaled by the external one U_e. It is found that the flow accelerates on the windward side and separates near the crest on the lee side. Due to the flow separation, the recirculation zone is created behind the dune. Within this bubble,

	x_{sep}/H	x_{reat}/H	L_{AB}/H	L_{CD}/H	L_{AB}/L_{CD}
Test A′	0.97	4.48	0.80	3.60	0.222
Test B′	0.36	8.22	1.20	7.91	0.152
Test C′	1.67	3.83	0.44	2.21	0.199

Table 3.8 Characterization of the recirculation zone behind a large sinusoidal dune.

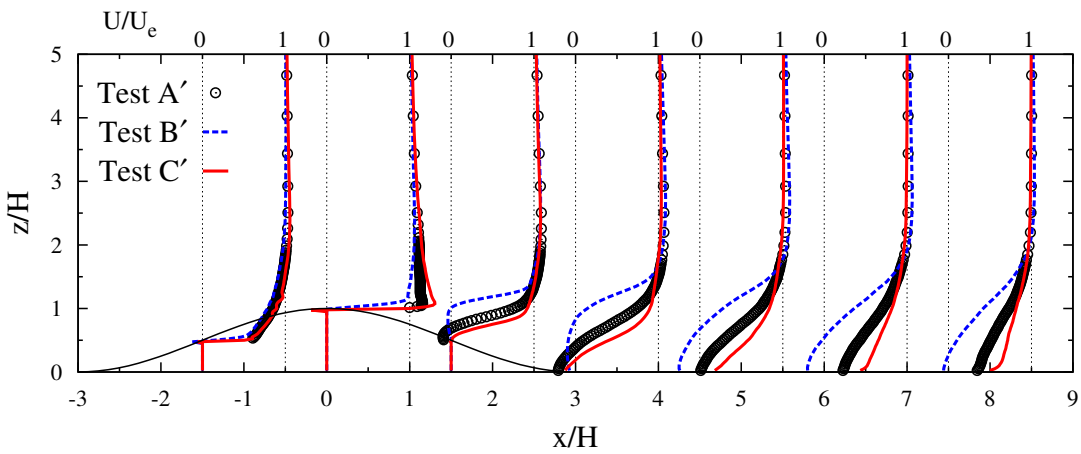


Figure 3.35 Mean longitudinal velocity profiles over a large sinusoidal dune. Black points: ARPS simulation (Test A′). Blue dashed lines: IBM-ARPS simulation without wall modeling (Test B′). Red solid lines: IBM-ARPS simulation with wall modeling (Test C′).

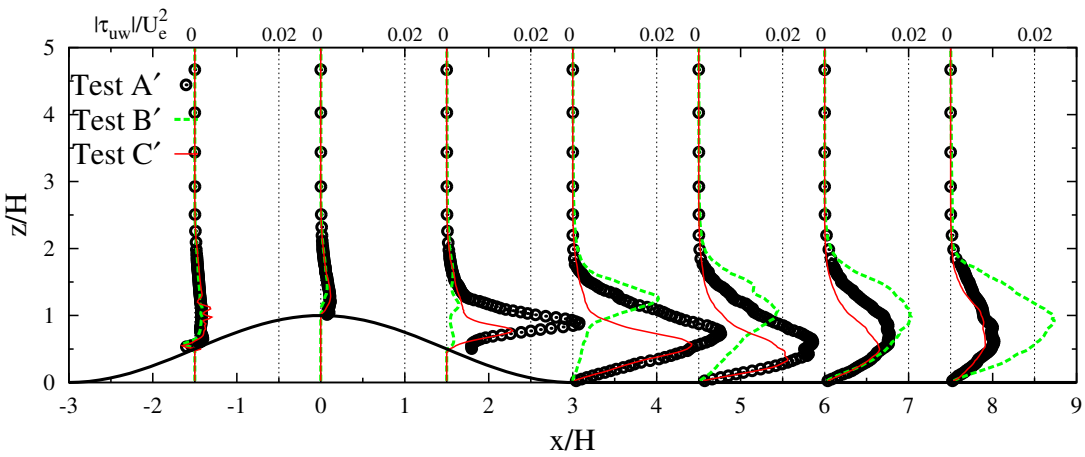


Figure 3.36 Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a large sinusoidal dune. Black points: ARPS simulation (Test A′). Blue dashed lines: IBM-ARPS simulation without wall modeling (Test B′). Red solid lines: IBM-ARPS simulation with wall modeling (Test C′).

the longitudinal velocity near the wall is reversed. Figure 3.36 presents the Reynolds stress profiles over the sinusoidal dune. The Reynolds stress is normalized by U_e^2. It is shown that

the Reynolds stress is enhanced after the flow separation, especially in the recirculation zone. As shown in Figures 3.35 and 3.36, a good agreement on mean longitudinal velocity and Reynold stress is observed on the windward side, but an obvious difference is shown on the lee side. Globally, the results simulated by the IBM-ARPS with the wall modeling is much closer to that of the ARPS simulation than that of the standard IBM-ARPS simulation.

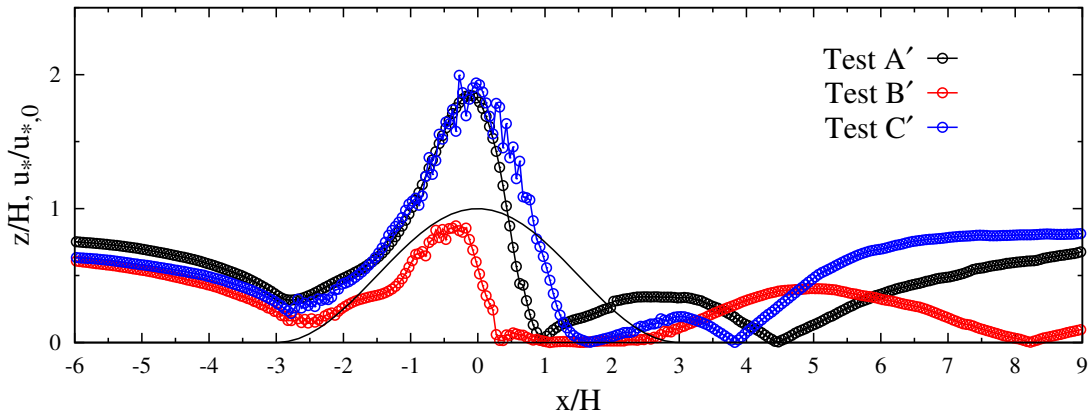


Figure 3.37 Comparison of the friction velocity between the ARPS simulation (Test A′), the ARPS simulation without wall modeling (Test B′) and the IBM-ARPS simulation with wall modeling (Test C′).

In Figure 3.37, the friction velocities obtained by the ARPS simulation, the IBM-ARPS simulation and the IBM-ARPS simulation with wall modeling are presented. The friction velocity is scaled by the initial value. It is shown that the friction velocity over the dune obtained by the standard IBM-ARPS simulation is obviously much smaller than that by the other two simulations. Although the friction velocity of IBM-ARPS simulation with wall modeling slightly differs from that of ARPS simulations on the lee side, a good agreement is observed on the windward side and at the dune crest, especially on that of large values, which is important for the particle entrainment. Briefly, Applying wall modeling in the standard IBM-ARPS simulation is helpful to obtained a relatively accurate friction velocity over a large dune when a coarse grid is used.

In conclusion, through the comparisons with the simulated results by the ARPS, the accuracy of the IBM-ARPS simulation with wall modeling is more acceptable than the standard IBM-ARPS simulation for the TBL flow over a large sinusoidal dune. It is also concluded that the wall modeling is an effective way to improve the simulated flow fields when the IBM-ARPS is used with a coarse grid.

3.5 Conclusions

In this chapter, a new numerical solver of large-eddy simulation coupled with immersed boundary method (IBM-APRS) is developed to simulate turbulent boundary layer flows over complex or deformable obstacles of different sizes. The accuracy of this solver is validated through the numerical simulations of a turbulent boundary layer over a small Gaussian and a large sinusoidal dune.

In the first canonical validation case, we conduct several numerical simulations a TBL over a small Gaussian dune using a relatively fine grid. A good agreement on the recirculation zone characteristics, mean longitudinal velocity profiles, Reynolds stress profiles between the (ARPS and IBM-ARPS) simulation results and experimental data demonstrates the ability of this developed solver. Moreover, it is shown that neither the interior treatment nor the pressure reconstruction has influence on the standard IBM-ARPS simulation results, whereas the wall modeling has a slight effect, *e.g.*, weakly reducing the recirculation zone.

In the second canonical validation case, we preforms numerical simulations a TBL over a large sinusoidal dune using a relatively coarse grid. It is shown that the IBM-ARPS with wall modeling gives a better performance than the standard IBM-ARPS by comparing the results with simulated data by the ARPS with terrain-following grid. It is concluded that the wall modeling plays a more important role in the IBM-ARPS simulation with a coarse grid than that with a very fine grid. In our following works, this numerical approach will be used to simulate the turbulent boundary layer flow with Lagrangian tracking of solid particles and with a deformable dune. The dune deformation will be modeled by sand particle movement with the physical wind erosion models proposed in Chapter 4.

Chapter 4

Physical modeling of wind erosion

Wind erosion consists in four complex processes: particle entrainment, particle transport, particle deposition and particle/surface interaction. In order to study the wind erosion through numerical simulations, physical modeling of these processes is required. In the previous models of wind erosion, the erosion and deposition rates are usually modeled as functions of the Shields number. However, these empiric functions are obtained from the saturated (equilibrium) transport state, and hence, lack universality. Therefore, we propose a more comprehensive and physical model of wind erosion, in which particle motions are determined by the forces acting on it and particle/surface interaction are taken into account using a probabilistic rebound/splash model.

In this chapter, original physical modeling of wind erosion is presented and integrated wind erosion models are proposed. In Section 4.1, the forces acting on an individual airborne particle are introduced and discussed. In Section 4.2, particle incipient motions are theoretically analyzed and an instantaneous particle entrainment model is proposed. In Section 4.3, Lagrangian governing equations of airborne and bed-load particle motions are given. Moreover, a Lagrangian stochastic model and a two-way coupling strategy are briefly described. In Section 4.4, particle-surface collision (rebound and splash) models are introduced and the criterion of particle deposition is briefly given. In Section 4.5, an avalanche model is presented and the governing equation of sand deformation is introduced.

4.1 Forces on an airborne particle

The forces acting on an airborne particle can be classified into two categories: *i)* forces produced by the undisturbed ambient flow, *ii)* forces induced by the disturbance. The undisturbed ambient flow is defined as the steady uniform or non-uniform flow in the absence

of the particle, and the disturbance is produced by either the unsteady flow or the particle and its unsteady motion with respect to the ambient flow.

Since the 19-th century, Stokes (1851) began the study on drag force acting on a sphere in a steady uniform flow. Then Boussinesq (1885), Basset (1888a) and Oseen (1927) provided the force formula of a moving particle in the fluid at rest, in order to study the dynamic motion of a particle submerged in fluid. Faxén (1922) extended Stokes' drag expression to the non-uniform flow though replacing the uniform velocity by the undisturbed non-uniform ambient velocity averaged over the particle surface. To study the particle motion in a moving fluid, Tchen (1947) made two extensions on the forces: firstly, to an unsteady and uniform flow; secondly, to an unsteady and non-uniform flow. Unfortunately, Corrsin and Lumley (1956) lately stated several inconsistencies in Tchen's second extension, specially the effect of pressure gradient on the contribution to the forces on the particle. Lately, Buevich (1966) made his contribution to improve this term induced by the undisturbed effect. Maxey and Riley (1983) proposed the dynamic motion of a particle in unsteady and non-uniform flow with complete and accurate force expressions. However, this equation is only applied to particles of size smaller than the flow dissipation scale. Gatignol (1983) corrected this limitation by several dissipation scales by using the Faxén correction [Faxén (1922)].

In this section, considering a singular particle flying in an unsteady turbulent boundary flow or moving on the bed surface, we will briefly introduce analytical expressions of these forces acting on it. We divide these forces into three groups: *1)* the forces in undisturbed ambient flows, including the aerodynamic drag and lift; *2)* forces induced by the disturbance, including the added-mass force, the Basset force, the effect by the pressure gradient; and *3)* the other forces independent on the flow, consisting in the submerged gravity, the adhesion force and the friction due to the contact with the ground.

4.1.1 Forces in undisturbed ambient flows

The undisturbed ambient flows is defined as the steady uniform or non-uniform flow without the effect of particles and particle motions. The main forces acting on a spheric particle induced by the flow are the aerodynamic drag and the aerodynamic lift. Specially, the aerodynamic lift of an immobile particle vanishes in an uniform flow, and it is generally induced by the vortex (or shear) and particle rotation.

Drag force: The aerodynamic drag force is aligned with the relative velocity of particle to fluid and slows down or accelerates the particle motion. This force for a spherical particle

can be written as:

$$\vec{F}_D = -\frac{\pi}{8} C_D \rho d_p^2 u_r^2 \hat{\vec{u}}_r, \tag{4.1}$$

where C_D is the non-dimensional drag coefficient, ρ the fluid density, d_p the particle diameter, u_r the norm of particle-fluid relative velocity $\vec{u}_r(\vec{x}_p,t) = \vec{u}_p - \vec{u}(\vec{x}_p,t)$ with $\vec{u}_p$ the particle velocity and $\vec{u}(\vec{x}_p,t)$ the fluid velocity at the particle center position $\vec{x}_p$, and $\hat{\vec{u}}_r$ the unit vector of $\vec{u}_r$, respectively. Hence, from Equation (4.1), the drag coefficient C_D is evaluated by:

$$C_D = \frac{F_D}{\frac{\pi}{8}\rho d_p^2 u_r^2}, \tag{4.2}$$

where F_D is the norm of the aerodynamic drag force $\vec{F}_D$.

From experimental and simulated results, the drag coefficient can be written as a function of particle Reynolds number Re_p [Shields (1936)]:

$$C_D(Re_p) = \frac{24}{Re_p} f(Re_p), \tag{4.3}$$

where Re_p is defined as $Re_p = u_r d_p/\nu$, and the function f is given by:

$$f(Re_p) = \begin{cases} 1 + 0.15 Re_p^{0.687} & , \quad \text{if } Re_p < 1000; \\ 0.0183 Re_p & , \quad \text{otherwise.} \end{cases} \tag{4.4}$$

In this thesis, special attention is paid to the aerodynamic drag force for a particle sitting on the wall as shown in Figure 4.1. We define a new non-dimensional drag coefficient C_D^w as follows:

$$C_D^w = \frac{F_D}{\frac{\pi}{8}\rho d_p^2 u_*^2}, \tag{4.5}$$

with u_r in Equation (4.2) replaced by the friction velocity u_*. For a particle lying on a rough wall, we have $u_r = u(h + d_p/2)$ where h is the effective height of rough elements. Using the law of the rough wall $u^+ = f_w(z^+, h^+)$, we have $u_r = u_* f_w(h^+ + d_p^+/2, h^+)$ and $Re_p = d_p^+ f_w(h^+ + d_p^+/2, h^+)$ with $d_p^+ = d_p u_*/\nu$. Combining Equation (4.2) and Equation (4.5), the relation between C_D^w and C_D is obtained:

$$C_D^w(d_p^+) = C_D \left(d_p^+ f_w \left(h^+ + \frac{d_p^+}{2}, h^+ \right) \right) f_w^2 \left(h^+ + \frac{d_p^+}{2}, h^+ \right). \tag{4.6}$$

Note that the law of the wall f_w has been detailedly described in Appendix A for both the smooth and rough cases. Specially, we have $h^+ = 0$ in the case of a smooth wall.

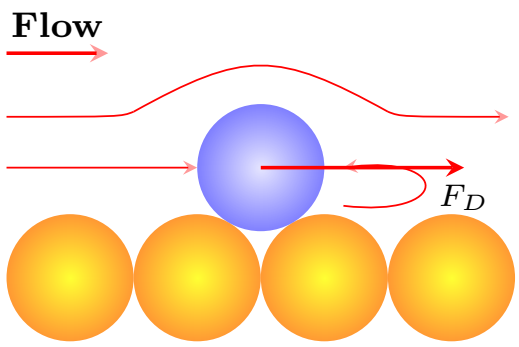


Figure 4.1 Sketch of the aerodynamic drag force $\vec{F}_D$ for a particle sitting on the wall.

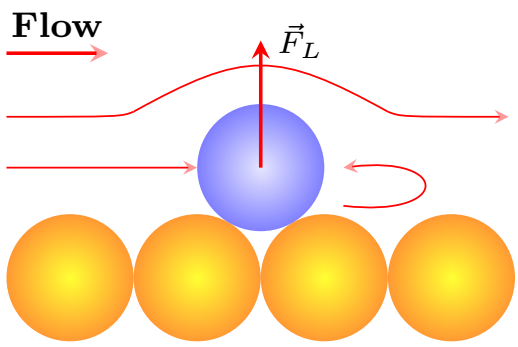


Figure 4.2 Sketch of the aerodynamic lift force $\vec{F}_L$ for a particle sitting on the wall.

Lift force: The aerodynamic lift is the component of the aerodynamic force orthogonal to the direction of the fluid-particle relative velocity. The aerodynamic lift acting on a particle can be divided into two parts: the fluid vortex induced part, and the particle rotation induced part [Loth (2008)]. There is not an empirical and useful formula to describe the total lift force, due to the nonlinear relation between these two parts. In our works, we study the motion of spherical particles in boundary flows without accounting for the particle rotation, therefore, the rotation-induced lift, also called the Magnus force, is neglected, and the vortex-induced lift is then dominant. Generally, the vortex-induced lift takes form of [Legendre and Magnaudet (1998)]:

$$\vec{F}_L = \frac{\pi}{8} C_L \rho d^2 u_r^2 \hat{\vec{\omega}} \times \hat{\vec{u}}_r, \tag{4.7}$$

with C_L the non-dimensional lift coefficient, $\hat{\vec{\omega}} \times \hat{\vec{u}}_r$ the direction of lift force. $\vec{\omega} = \nabla \times \vec{u}$ is the fluid vorticity at the center of the particle, and $\hat{\vec{\omega}}$ is its unit vector. Hence, the lift coefficient C_L is defined in the same manner as C_D (Equation (4.2)):

$$C_L = \frac{F_L}{\frac{\pi}{8} \rho d_p^2 u_r^2}, \tag{4.8}$$

where F_L is the norm of the aerodynamic lift force $\vec{F}_L$.

In a simple case of linear-shear vorticity, Saffman (1965) found an analytical formula of F_L for $Re_p \ll 1$:

$$F_L = 1.615 \rho \nu u_r d_p^2 \sqrt{\frac{\omega_{shear}}{\nu}}, \tag{4.9}$$

where ω_{shear} denotes the vorticity in a linear shear flow. In boundary flows, we approximately take $\omega_{shear} = (u(z_p + d_p/2) - u(z_p - d_p/2))/d_p$ with u the streamwise velocity and z_p the

particle position in the wall-normal direction. Substituting Equation (4.9) into Equation (4.8), the Saffman-lift coefficient is expressed by a function of particle Reynolds number Re_p and of the non-dimensional vorticity number Re_ω:

$$C_L^{\text{Saff}}(Re_p, Re_\omega) = \frac{12.92}{\pi}\sqrt{\frac{Re_\omega}{Re_p}}, \tag{4.10}$$

where Re_ω is defined as $Re_\omega = d_p\omega_{shear}/u_r$. To eliminate the strong restriction $Re_p \ll 1$, Mei (1992) used the numerical results of Dandy and Dwyer (1990) and obtained an approximate formula of C_L:

$$C_L(Re_p, Re_\omega) = \begin{cases} C_L^{\text{Saff}}\left((1-0.2343Re_\omega^{1/2})\exp(-Re_p/10) + 0.2343Re_\omega^{1/2}\right), & \text{if } Re_p \le 40, \\ 0.0441C_L^{\text{Saff}}\left(Re_\omega^{1/2}Re_p\right)^{1/2}, & \text{if } Re_p > 40, \end{cases} \tag{4.11}$$

where C_L^{Saff} is the Saffman lift coefficient given by Equation (4.10).

In particular, for a particle resting on the wall, Mollinger and Nieuwstadt (1996) proposed an empirical formula of the lift force by fitting with their experimental data:

$$F_L = 15.57\rho\nu^2\left(\frac{u_* d_p}{\nu}\right)^{1.87}, \tag{4.12}$$

where u_* is the friction velocity. By introducing non-dimensional lift force $F_L^+ = F_L/(\rho\nu^2)$ and particle diameter $d_p^+ = u_* d_p/\nu$, Equation (4.12) can be rewritten as:

$$F_L^+ = 15.57(d_p^+)^{1.87}. \tag{4.13}$$

Some other formula of F_L^+ are given in Table 4.1. It should be noted that the formula of Hall (1988) and Mollinger and Nieuwstadt (1996) are obtained from the experiments of a particle on a wall in a turbulent boundary layer. Zeng et al. (2009) found that the measured lift force in a boundary layer is greater than those in a linear shear flow. In numerical simulations, Descamps (2004) and Huang (2015) chosen the formula of Mollinger and Nieuwstadt (1996) to simulate particle entrainment in an atmospheric boundary layers. Moreover, Zeng et al. (2009) propose a simple expression of the shear-induced lift coefficient for a particle sitting on the wall:

$$C_L(Re_p) = \frac{3.63}{(Re_p^2 + 0.1173)^{0.22}}. \tag{4.14}$$

Note that C_L is a pure function of the particle Reynolds number Re_p in Equation (4.14),

which is the same as the drag coefficient. This may facilitate the calculation of the lift force of a particle contacted on the wall.

Formula of F_L^+	**Formula of C_L^w**	**Literature**
$F_L^+ = 0.81(d_p^+)^3$	$C_L^w = \frac{6.46}{\pi} d_p^+$	Saffman (1965) first order
$F_L^+ = 0.81(d_p^+)^3 - 0.27(d_p^+)^4$	$C_L^w = \frac{6.46}{\pi} d_p^+ - \frac{4.32}{\pi}(d_p^+)^2$	Saffman (1965) second order
$F_L^+ = 0.58(d_p^+)^4$	$C_L^w = \frac{9.22}{\pi}(d_p^+)^2$	Leighton and Acrivos (1985)
$F_L^+ = 4.21(d_p^+)^{2.31}$	$C_L^w = \frac{20.90}{\pi}(d_p^+)^{0.31}$	Hall (1988)
$F_L^+ = 15.57(d_p^+)^{1.87}$	$C_L^w = \frac{56.90}{\pi}(d_p^+)^{-0.13}$	Mollinger and Nieuwstadt (1996)
$F_L^+ = 0.49(d_p^+)^{3.12}$	$C_L^w = \frac{4.24}{\pi}(d_p^+)^{1.12}$	Zeng et al. (2009)

Table 4.1 Some formulas of the non-dimensional lift force F_L^+ and the wall lift coefficient C_L^w.

In this thesis, we define a new non-dimensional lift coefficient C_L^w for a particle sitting on a wall as shown in Figure 4.2, as follows:

$$C_L^w = \frac{F_L}{\frac{\pi}{8}\rho d_p^2 u_*^2}, \tag{4.15}$$

with u_r in Equation (4.8) replaced by the friction velocity u_*. For a particle lying on a rough wall, we have $u_r = u(h + d_p/2)$ where h means the effective height of rough elements. Using the log law of the wall $u^+ = f_w(z^+, h^+)$, we have $u_r = u_* f_w(h^+ + d_p^+/2, h^+)$, $Re_p = d_p^+ f_w(h^+ + d_p^+/2, h^+)$ and $Re_\omega = (f_w(h^+ + d_p^+, h^+) - f_w(h^+, h^+))/f_w(h^+ + d_p^+/2, h^+)$. Combining Equation (4.8) and Equation (4.15), the relation between C_L^w and C_L is obtained:

$$C_L^w(d_p^+) = f_w^2\left(h^+ + \frac{d_p^+}{2}, h^+\right) C_L\left(d_p^+ f_w\left(h^+ + \frac{d_p^+}{2}, h^+\right), \frac{f_w(h^+ + d_p^+, h^+) - f_w(h^+, h^+)}{f_w(h^+ + d_p^+/2, h^+)}\right). \tag{4.16}$$

Specially, we take $h^+ = 0$ in the smooth case. For example, it is known that the linear relation lies in the viscous sublayer for a smooth wall, *i.e.*, $f_w(z^+) = z^+$, when C_L takes the Saffman's formula (Equation (4.10)), applying Equation (4.16) yields:

$$C_L^w = \frac{6.46}{\pi} d_p^+. \tag{4.17}$$

It indicates that C_L^w is theoretically linear to d_p^+ when $d_p^+ \ll 1$. Other formulas of C_L^w are given in Table 4.1. C_L^w of different empirical formula are presented in Figure 4.3. Unfortunately, these formula can not reach a common curve of C_L^w. This implies indirectly that the intrinsic behavior of the aerodynamic lift is complicated. In Figure 4.3, we find that

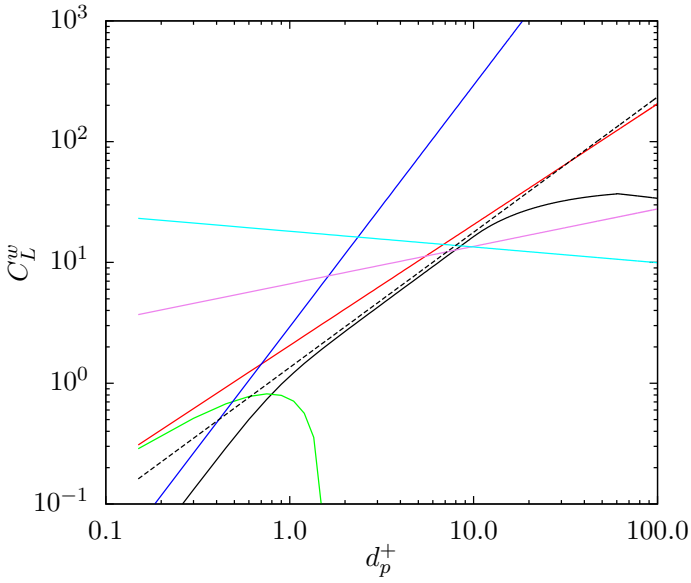


Figure 4.3 Lift coefficient C_L^w by different formula. —: Equation (4.17); - - : Equation (4.14) and Equation (4.16); —: Saffman (1965) second order; —: Leighton and Acrivos (1985); —: Hall (1988); —: Mollinger and Nieuwstadt (1996); —: Zeng et al. (2009).

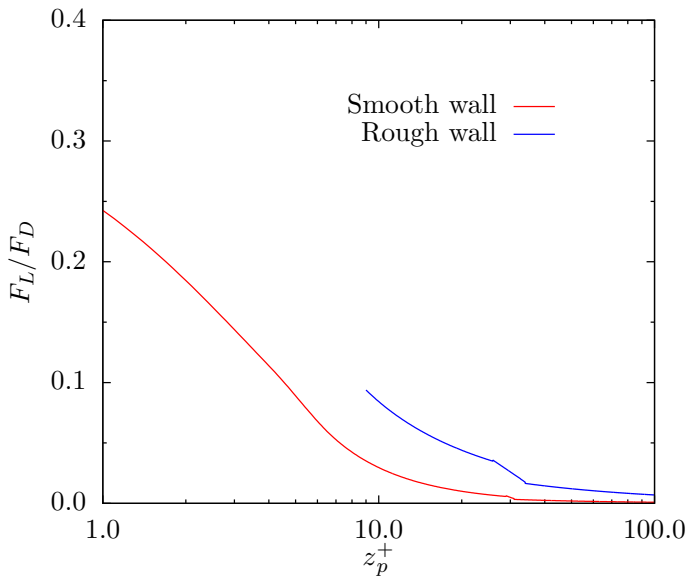


Figure 4.4 Ratio of lift to drag force for an immobile particle of diameter $d_p^+ = 1$ submerged in a boundary layer on the smooth and rough wall. F_D is calculated by Equation (4.1) and Equation (4.3). F_L is calculated by Equation (4.7) and Equation (4.11). The average streamwise velocity field is generated by the wall of the law.

a good agreement is achieved between Equation (4.14), Equation (4.17) and the formula of Zeng et al. (2009). Hence, we prefer to chose the simple formula (Equation (4.14)) and use it in the analysis of the threshold friction velocity of particle incipient motions.

Figure 4.4 shows the ratio of lift to drag force of an immobile particle in a boundary layer with $d_p^+ = 1.$. It is found that F_L/F_D decreases rapidly with the distance between the particle and wall z_p^+. If $z_p^+ > 10$, we have $C_L/C_D < 0.1$ for both smooth and rough wall. This indicates that the lift force is important for particle motions in the near-wall region, particularly for the particle incipient motion, but can be negligible for particles detached far away from the wall.

4.1.2 Forces induced by the disturbance

The disturbance is defined as the effect produced by either the unsteady flows or the particle and its motion. The forces induced by the disturbance consist in the acceleration by the fluid $\vec{F}_P$, the added mass $\vec{F}_A$ and the Basset force $\vec{F}_B$.

The acceleration by the fluid: It is produced by the effects of the undisturbed stresses from the pressure and viscosity. Buevich (1966) proposed that $\vec{F}_P$ equals to the product of fluid local acceleration and fluid mass, that is, for a spherical particle:

$$\vec{F}_P = \frac{1}{6}\pi\rho d_p^3 \frac{D\vec{u}}{Dt}, \tag{4.18}$$

where $\pi\rho d_p^3/6$ is the fluid mass displaced by the moving particle, and $D(\cdot)/Dt$ denotes the time derivative following the fluid particle motion.

The added mass force: It is the added effect acting on the submerged particle, due to the fact that the moving submerged particle has to displace the same volume of the surrounding fluid. The added mass force takes an opposite direction of particle-fluid relative motion and equals:

$$\vec{F}_A = -\frac{1}{12}\pi\rho d_p^3 \left(\frac{d\vec{u}_p}{dt} - \frac{d\vec{u}}{dt}\right), \tag{4.19}$$

where $d(\cdot)/dt$ denotes the time derivative following the solid particle motion.

The Basset force: It is a history force, which accounts for viscous effects and describes the time-delay of particle-fluid relative velocity in boundary layers. Basset (1888b) gave its

mathematical form of an accelerating sphere in a viscous flow:

$$\vec{F}_B = -\frac{3}{2}\pi\rho\mu d_p^2 \int_0^t \frac{1}{\sqrt{\pi\rho\mu(t-\tau)}} \left(\frac{d\vec{u}_p}{d\tau} - \frac{d\vec{u}}{d\tau}\right) d\tau. \tag{4.20}$$

For a spherical particle of diameter d_p in a boundary layer with the friction velocity u_* and the thickness δ, the ratio of these forces induced by the disturbance to the drag is estimated in Table 4.2. It is found that these ratios are related to $(d_p^+)^2/\delta^+$ with $\delta^+ = \delta u_*/\nu$. In this thesis, d_p, u_* and δ are of the order of 10^{-4}m, $10^{-1}\mathrm{m}\cdot\mathrm{s}^{-1}$ and 10^{-1}m, then we have $d_p^+ \sim 1$ and $\delta^+ \sim 10^3$, which yields $(d_p^+)^2/\delta^+ \sim 10^{-3}$. This indicates that F_P, F_A and F_B can be negligible when comparing to F_D. Therefore, the forces induced by the disturbance are not taken into account for the particle motion.

Force ratio	F_P/F_D	F_A/F_D	F_B/F_D
Order of magnitude	$\sim (d_p^+)^2/\delta^+$	$\sim (d_p^+)^2/\delta^+$	$\sim \left((d_p^+)^2/\delta^+\right)^{1/2}$

Table 4.2 Magnitude order of force ratio. Adapted from Vinkovic (2005).

4.1.3 Gravity, cohesion and friction

Gravity: As illustrated in Figure 4.5, the gravity of a particle submerged in a viscous flow is given by

$$\vec{G} = \frac{1}{6}\pi(\rho_p - \rho) g d_p^3 \hat{\vec{g}}, \tag{4.21}$$

where ρ_p is the particle density, ρ denotes the fluid density, and $\vec{g} = (0, 0, -g)$ is the gravitational acceleration with its norm g and its unit vector $\hat{\vec{g}}$.

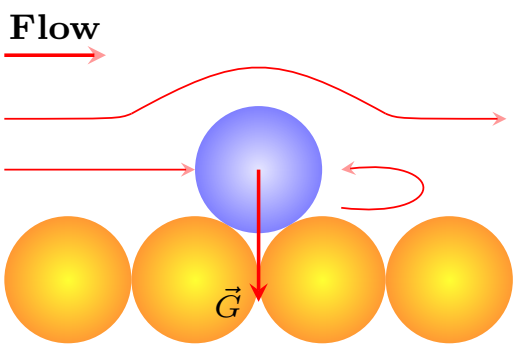


Figure 4.5 Sketch of the submerged gravity $\vec{G}$.

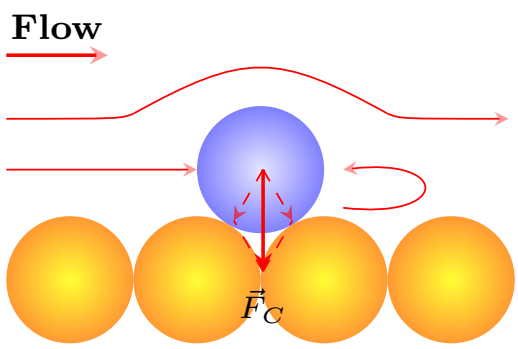


Figure 4.6 Sketch of the cohesion force $\vec{F}_C$.

Cohesion: When the particle is small, the cohesion force is usually important and should be taken into account. In the microscopic view, the cohesion force comes from these contacted particles and equals the ensemble of the forces exerted by the neighbors. Following Zimon et al. (1969), F_C is modeled as a linear function of the particle diameter d_p:

$$F_C = C_C d_p, \tag{4.22}$$

where the coefficient of cohesion is given by $C_C = 1.43 \times 10^{-5} \mathrm{N} \cdot \mathrm{m}^{-1}$. The cohesion is opposed to the particle motion and its direction is perpendicular to the contacted surface. In this thesis, solid particles are regularly arranged on the surface, the cohesions from the left neighbor and the right one are equal, hence, their resultant can be regarded as the adhesion from the bed surface, which is always perpendicular to the surface as shown in Figure 4.6. In this case, the cohesion coefficient will be modified by $\sqrt{3}C_C = 2.48 \times 10^{-5} \mathrm{N} \cdot \mathrm{m}^{-1}$.

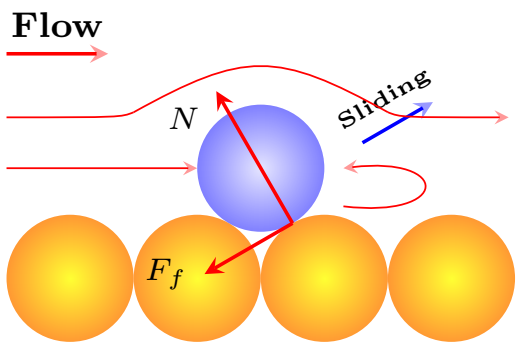


Figure 4.7 Sketch of the sliding friction F_f.

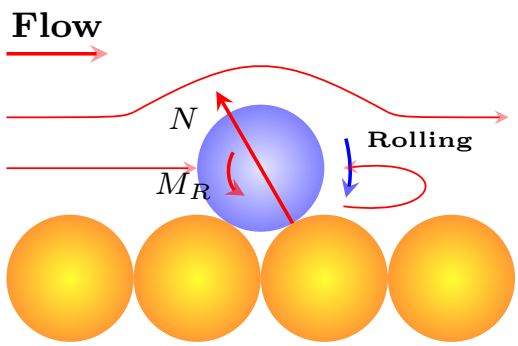


Figure 4.8 Sketch of the rolling friction M_R.

Friction: When a particle moves on a surface, there exists the friction resisting its motion. According to Coulomb's law, the sliding friction force F_f is a linear function of the reaction N:

$$F_f = \mu_s N, \tag{4.23}$$

where μ_s is called the sliding friction coefficient. F_f and N are sketched out in Figure 4.7. The sliding friction angle φ_s is defined as $\varphi_s = \arctan \mu_s$. Similarly, the reaction rolling friction torque M_R is also suggested to be a linear function of the reaction:

$$M_R = \mu_r R N, \tag{4.24}$$

where μ_r denotes the rolling friction coefficient and $R = d_p/2$ is the radius of particle. M_R

and N are sketched out in Figure 4.8. The rolling friction angle φ_r is defined as $\varphi_r = \arctan \mu_r$. The friction force (or torque) is always in the opposite direction of the particle sliding (or rolling) motion.

4.2 Particle incipient motion

In this section, particle incipient motions are theoretically studied, including lifting mode, sliding mode and rolling mode. Firstly, the definition of the threshold friction velocity $u_{*,c}$ is presented and discussed. Based on the balance of forces or moments, we then obtain the mathematical formulas of $u_{*,c}$ and compare them with the experimental results. At last, we propose an instantaneous criterion of particle incipient motions, which will be used in the wind erosion simulation.

4.2.1 Threshold friction velocity

Particle incipient motion is the first subject to the wind erosion. In these classic models, one defined a critical friction velocity $u_{*,c}$, above which a particle begins to move. The threshold friction velocity firstly proposed by DuBoys (1879), is usually modeled by the Shield diagram and is widely used for the quantitative description of the saltation (stream-wise) flux [Bagnold (1941); Shields (1936)].

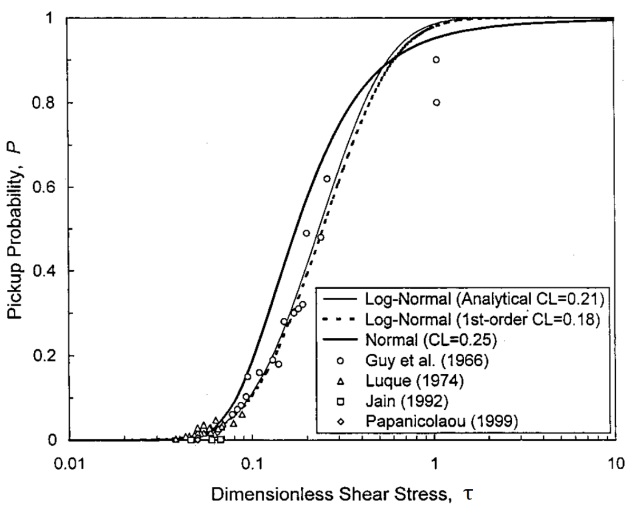


Figure 4.9 Probability of particle incipient motion versus the dimensional shear stress $\tau = u_*^2/(\gamma_\rho g d)$. Taken from Wu and Lin (2002).

In the macroscopic view, the threshold velocity is defined as the transition point from no particle transport to particle transport regime. However, in the microscopic view at the scale of particle size, how to define the event that a particle begins to move? it starts to slide, to roll, or to be lifted? It is obviously difficult to make the decision according to the visual observation. Moreover, in the stochastic view, the random instantaneous forces is not able to result in a deterministic transition point from non-motion to motion of a particle. This is why Lavelle and Mofjeld (1987) doubted the existence of critical threshold velocity. The probability of particle entrainment has been evaluated by some stochastic models proposed by Einstein et al. (1950), Cheng and Chiew (1998), Wu and Lin (2002) and Elhakeem et al. (2017). For example, Figure 4.9 shows the particle pickup probability in a turbulent boundary layer, it is found that the band of dimensional shear stress is wide from the null probability to the unit. This convinces us that it is impossible to find an unique value of the friction velocity to describe the transition band.

To most of researchers, it is acceptable that the conception of threshold is considered as a time-space average notion. For example, Ibrahim et al. (2008) measured the threshold friction velocities by defined the transition point as detaching 50% of particles on the surface. Recently, Ho (2012) distinguished the static threshold and dynamic threshold using the discontinuous transition from no transport to the transport regimes as shown in Figure 4.10. The static threshold $u^s_{*,c}$, which is totally determined by the turbulent flow, is defined as the maximum friction velocity in the no transport regime. The dynamic threshold $u^d_{*,c}$, also called the rebound threshold, is defined as the minimum friction velocity in the transport regime. Its value can be obtained from the intersection point between the two measured curves with and without transport.

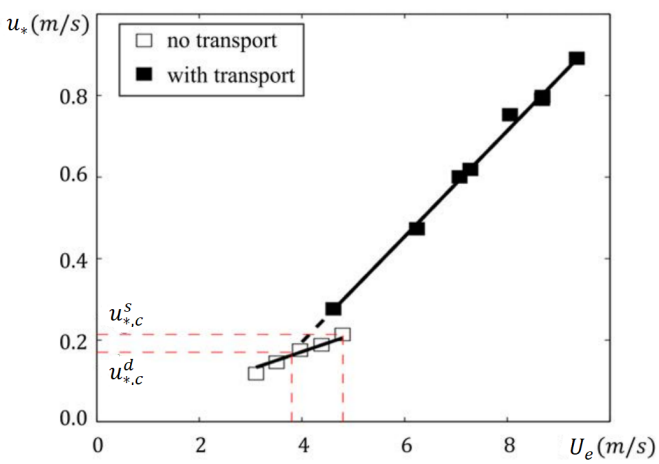


Figure 4.10 Friction velocity u_* versus the flow velocity U_e without and with particle transport. Taken from Ferreira (2017).

In this section, we focus on the (static) threshold velocity in the view of dynamic mechanism by distinguishing different modes of particle incipient motions: lifting mode, sliding mode and rolling mode. The particle incipient motions caused by the rebound events are simulated by a probabilistic splash model.

4.2.2 Lifting, Sliding, Rolling mode

Considering a particle on a rough surface, it begins to move either by lifting, by sliding or by rolling. Based on the balance of forces or torques, the critical condition of lifting, sliding, and rolling can be expressed by

- Lifting mode: the resultant in the wall-normal direction is positive, *i.e.*, $\sum F_N > 0$, with F_N denoting the force (or force component) in the wall-normal direction;
- Sliding mode: the resultant in the stream-wise (tangential) direction is positive, *i.e.*, $\sum F_T > 0$, with F_T denoting the force (or force component) in the stream-wise direction,
- Rolling mode: the total torque in the span-wise direction is positive, *i.e.*, $\sum M_S > 0$, with M_S denoting the torques in the span-wise direction.

Next, we will introduce these three modes in detail, by using the forces acting on a particle presented in Section 4.1.

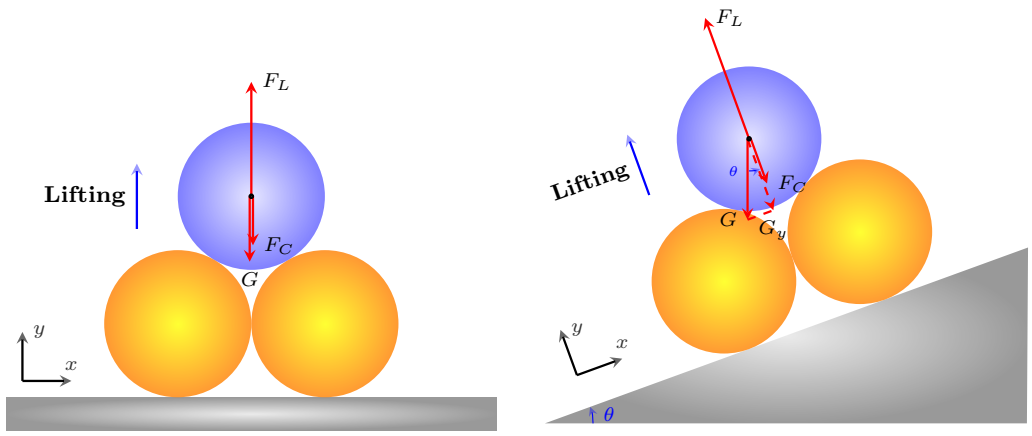


Figure 4.11 Sketch of the lifting mode of particle incipient motion. Left: the surface is plate $\theta = 0$; Right: a general sketch with a slope θ.

Lifting mode: The forces in the wall-normal direction are sketched in Figure 4.11. According to the balance of forces, *i.e.*, $\sum F_N > 0$, the general criterion of particle incipient motion by lifting on a rough sloping surface is:

$$F_L - G\cos\theta - F_C > 0, \tag{4.25}$$

where F_L is the aerodynamic lift force, G the particle gravity, θ the bed slope, F_C the cohesion force, respectively. Assuming that a spherical particle is studied, these forces can be written as $F_L = C_L \frac{\pi}{8}\rho u_r^2 d_p^2$, $G = \pi(\rho_p - \rho)g d_p^3/6$, $F_C = C_C d_p$, then the critical velocity is obtained as:

$$u_{r,c}^{lifting} = \sqrt{\frac{4}{3}\frac{1}{C_L}\gamma_\rho g d_p \cos\theta + \frac{C_C}{C_L}\frac{8}{\pi\rho d_p}}, \tag{4.26}$$

with $\gamma_\rho = (\rho_p - \rho)/\rho$. Supposing that the aerodynamic lift is related to the friction velocity, *i.e.*, $F_L = C_L^w \frac{\pi}{8}\rho u_*^2 d_p^2$, Equation (4.26) becomes:

$$u_{*,c}^{lifting} = \sqrt{\frac{4}{3}\frac{1}{C_L^w}\gamma_\rho g d_p \cos\theta + \frac{C_C}{C_L^w}\frac{8}{\pi\rho d_p}}, \tag{4.27}$$

which is called the threshold friction velocity of the particle lifting mode. When the fluid velocity (or the friction velocity) is greater than the critical value, the particle is then able to be lifted by the carried flow.

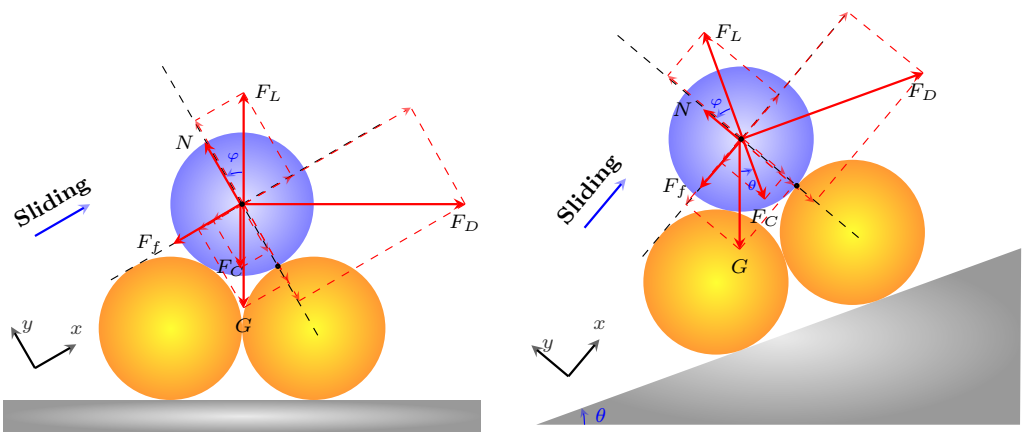


Figure 4.12 Sketch of the sliding mode of particle incipient motion. Left: the surface is plate $\theta = 0$; Right: a general sketch with a slope θ.

Sliding mode: When the particle is not able to be lifted, it will possibly slide or roll. The

exerted forces for sliding mode are showed in Figure 4.12. Note that the support (reaction) force exerted by the left neighbor is null, if the particle has the ability to slide over its right neighbor one. According the force balance in the sliding direction (x-direction shown in Figure 4.12), *i.e.*, $\sum F_T > 0$, the criterion of sliding mode can be expressed as:

$$\begin{aligned} N + F_L \cos\varphi &= F_D \sin\varphi + G\cos(\theta+\varphi) + F_C \cos\varphi, \\ F_D \cos\varphi + F_L \sin\varphi &> G\sin(\theta+\varphi) + F_C \sin\varphi + F_f, \end{aligned} \tag{4.28}$$

where F_D is the aerodynamic drag, N the reaction by the right neighbor particle, F_f the sliding friction force, and φ the angle shown in Figure 4.12, respectively. Note that the angle φ depends on the particle geometrical arrangement, ideally, $\varphi = \pi/6$. According to Coulomb law, we have $F_f = \mu_s N$ with μ_s the sliding friction coefficient. It is found that the normal force N is null when the lifting criterion is satisfied. This means that the sliding mode is easier to take place than the lifting mode.

Eliminating N and F_f in Equation (4.28) by the use of $F_f = \mu_s N$, we obtain after the simplification:

$$F_D - \mu_{se}(G\cos\theta + F_C - F_L) - G\sin\theta > 0, \tag{4.29}$$

where the equivalent sliding friction coefficient is $\mu_{se} = (\mu_s + \tan\varphi)/(1-\mu_s\tan\varphi)$. Assuming $\mu_s = \tan\varphi_s$, we have $\mu_{se} = \tan(\varphi_s + \varphi)$. When an ideal case is considered, *i.e.*, $F_D = 0$, $F_L = 0$ and $F_C = 0$, a particle slides automatically at the opposite direction when the local bed slope $\theta > \varphi_s + \varphi$. This is consistent with the result of Equation (4.29), namely, $\tan(-\theta) + \mu_{se} < 0$. (since the direction of particle motion changes, the sign of bed slope will also change $\theta \to -\theta$.)

If these forces take the form of $F_D = C_D \frac{\pi}{8}\rho u_r^2 d_p^2$, $F_L = C_L \frac{\pi}{8}\rho u_r^2 d_p^2$, $G = \pi(\rho_p - \rho) g d_p^3/6$, $F_C = C_C d_p$, the critical velocity is obtained from Equation (4.29) as:

$$u_{r,c}^{sliding} = \sqrt{\frac{4}{3}\frac{\mu_{se}}{C_D + \mu_{se}C_L}\gamma_\rho g d_p \left(\cos\theta + \frac{\sin\theta}{\tan\varphi_{se}}\right) + \frac{\mu_{se}C_C}{C_D + \mu_{se}C_L}\frac{8}{\pi\rho d_p}}, \tag{4.30}$$

with $\varphi_{se} = \varphi_s + \varphi$. Supposing the aerodynamic forces are related to the friction velocity, *i.e.*, $F_D = C_D^w \frac{\pi}{8}\rho u_*^2 d_p^2$ and $F_L = C_L^w \frac{\pi}{8}\rho u_*^2 d_p^2$, Equation (4.30) becomes:

$$u_{*,c}^{sliding} = \sqrt{\frac{4}{3}\frac{\mu_{se}}{C_D^w + \mu_{se}C_L^w}\gamma_\rho g d_p \left(\cos\theta + \frac{\sin\theta}{\tan\varphi_{se}}\right) + \frac{\mu_{se}C_C}{C_D^w + \mu_{se}C_L^w}\frac{8}{\pi\rho d_p}}, \tag{4.31}$$

which is called the threshold friction velocity of the particle sliding mode. When the fluid

velocity (or the friction velocity) is larger than the critical value of sliding mode and smaller than the critical value of lifting mode, the particle has the ability to slide on the surface.

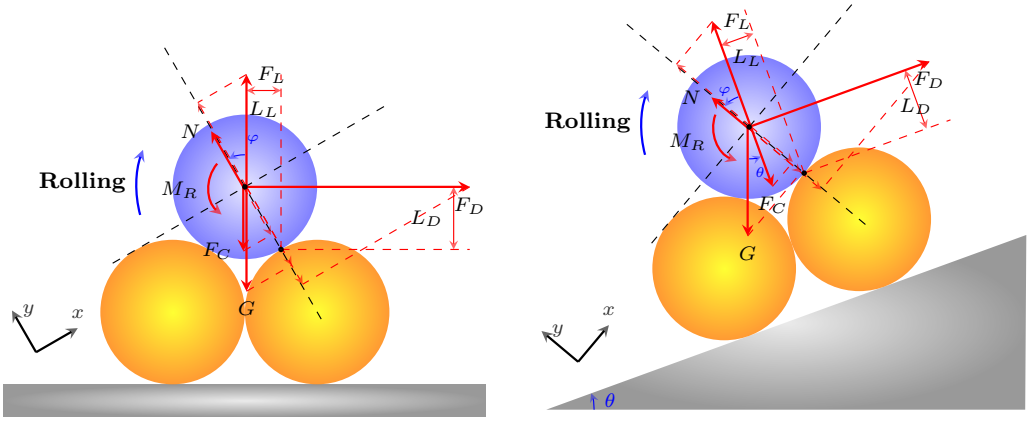


Figure 4.13 Sketch of the rolling mode of particle incipient motion. Left: the surface is plate $\theta = 0$; Right: a general sketch with a slope θ.

Rolling mode: When both lifting and sliding mode are not able to be achieved, the rolling mode probably takes place. The forces and moments are illustrated in Figure 4.13. Since the particle can roll over the right neighbor one, the support (reaction) force exerted by the left neighbor one should be null. According to the force and moment balance, *i.e.*, $\sum M_S > 0$, the criterion of particle rolling mode is given by:

$$\begin{aligned} N + F_L \cos\varphi &= F_D \sin\varphi + G\cos(\theta+\varphi) + F_C \cos\varphi, \\ L_D F_D + L_L F_L &> GR\sin(\theta+\varphi) + L_L F_C + M_R, \end{aligned} \tag{4.32}$$

where the rolling friction moment is determined by the rolling friction law $M_R = \mu_r RN$ with μ_r the rolling friction coefficient and R the particle radius. Eliminating N and M_R using $M_R = \mu_r RN$, Equation (4.32) is simplified to:

$$F_D - \frac{R\sin\varphi + \mu_r R\cos\varphi}{L_D - \mu_r R\sin\varphi} G\cos\theta - \frac{L_L + \mu_r R\cos\varphi}{L_D - \mu_r R\sin\varphi}(F_C - F_L) - \frac{R\cos\varphi - \mu_r R\sin\varphi}{L_D - \mu_r R\sin\varphi} G\sin\theta > 0. \tag{4.33}$$

When particles are regularly arranged as shown in Figure 4.13, the moment arms take the value of $L_D = R\cos\varphi$ and $L_L = R\sin\varphi$, then we get a similar formula of Equation (4.29):

$$F_D - \mu_{re}(G\cos\theta + F_C - F_L) - G\sin\theta > 0, \tag{4.34}$$

where the equivalent rolling friction coefficient is $\mu_{re} = (\mu_r + \tan\varphi)/(1 - \mu_r \tan\varphi)$. Assum-

ing $\mu_r = \tan\varphi_r$, we have $\mu_{re} = \tan(\varphi_r + \varphi)$. Note that the angle φ depends on the particle geometrical arrangement as shown in Figure 4.13, ideally, $\varphi = \pi/6$. Commonly, the rolling friction angle $\varphi_r = \arctan\mu_r$ is smaller that the sliding friction angle $\varphi_s = \arctan\mu_s$, since $\mu_r < \mu_s$.

If these forces take the form of $F_D = C_D\frac{\pi}{8}\rho u_r^2 d_p^2$, $F_L = C_L\frac{\pi}{8}\rho u_r^2 d_p^2$, $G = \pi(\rho_p - \rho)gd_p^3/6$, $F_C = C_C d_p$, the critical velocity is obtained from Equation (4.34):

$$u_{r,c}^{rolling} = \sqrt{\frac{4}{3}\frac{\mu_{re}}{C_D + \mu_{re}C_L}\gamma_\rho g d_p\left(\cos\theta + \frac{\sin\theta}{\tan\varphi_{re}}\right) + \frac{\mu_{re}C_C}{C_D + \mu_{re}C_L}\frac{8}{\pi\rho d_p}}, \tag{4.35}$$

with $\varphi_{re} = \varphi_r + \varphi$. Supposing the aerodynamic forces are related to the friction velocity, *i.e.*, $F_D = C_D^w\frac{\pi}{8}\rho u_*^2 d_p^2$ and $F_L = C_L^w\frac{\pi}{8}\rho u_*^2 d_p^2$, Equation (4.35) becomes:

$$u_{*,c}^{rolling} = \sqrt{\frac{4}{3}\frac{\mu_{re}}{C_D^w + \mu_{re}C_L^w}\gamma_\rho g d\left(\cos\theta + \frac{\sin\theta}{\tan\varphi_{re}}\right) + \frac{\mu_{re}C_C}{C_D^w + \mu_{re}C_L^w}\frac{8}{\pi\rho d}}, \tag{4.36}$$

which is called the threshold friction velocity for the rolling mode.

Note that the particle incipient motion can also be caused by a complex process – the bouncing mode. According to this mode, particles begin to move due to the collision with the rebounding particle carrying great momentums. In our simulations, the splash function is introduced to model the bouncing incipient motion. The splash entrainment model is further discussed in Subsection 4.4.2.

4.2.3 Discussions on the threshold velocity

To facilitate the following discussion, we introduce the non-dimensional Shield number Sh, which is defined by:

$$Sh = \frac{u_*^2}{\gamma_\rho g d_p}. \tag{4.37}$$

Then the critical Shield number is given by $Sh_c = u_{*,c}^2/(\gamma_\rho g d_p)$ if u_* is replaced by the threshold friction velocity $u_{*,c}$ in Equation (4.37). According to dimension analysis, we proposed that Sh_c is a function of dimensionless particle numbers and bed slope:

$$Sh_c = T\left(\frac{d_p}{\delta_\nu}, \frac{d_p}{d_\nu}, \theta\right) = T(d_p^+, d_p^-, \theta), \tag{4.38}$$

where $\delta_\nu = \nu/u_*$ means the flow characteristic length in the near-wall region, $d_\nu = (\nu^2/(\gamma_\rho g))^{1/3}$ represents the viscous diameter and θ is the bed slope. Note that d_p^+ represents the ratio

of the inertial effect to viscous effect of a fluid particle of a same diameter with the solid particle, and that d_p^- represents the ratio of particle inertial and gravitational effect to the viscous effect.

According to previous results, several formulas of Sh_c with $\theta = 0$ have been proposed and are summarized as follows:

- **Foucaut & Stanislas's curve:** By fitting with the measured result in wind tunnel experiments, Foucaut and Stanislas (1996) proposed an empirical formula of the critical shield number:

$$Sh_c = \frac{1}{d_p^-}\left(22.71(d_p^-)^{0.043} + 10.23(d_p^-)^{-0.118} - 32.5\right)^2. \tag{4.39}$$

  It is found that Sh_c is a pure function of d_p^-, and is independent on d_p^+.

- **Bagnold's model:** From the balance of force, Bagnold (1941) obtained:

$$Sh_c = F(d_p^+). \tag{4.40}$$

  Equation (4.40) indicates that the critical Shield number Sh_c is a pure function of d_p^+. Bagnold (1941) suggested $F(d_p^+) = 0.01$ for larger grains from the examination with experimental data. However, $Sh_c = 0.01$ fails for small particles, *i.e.*, $d_p < 100\ \mu$m, since observations show that Sh_c increases rapidly with decreasing the particle diameter d_p.

- **Greeley-Iversen's model:** Based on the idea that the inter-particle cohesion more likely causes the rapid increase of Sh_c with decreasing the particle diameter d_p, Greeley and Iversen (1985) proposed a correction of Equation (4.40):

$$Sh_c = F(d_p^+)(1 + G(d_p^-)), \tag{4.41}$$

  where $F(d_p^+)$ depends on the aerodynamic effect, and $G(d_p^-)$ depends on the effects of inter-particle cohesion. $F(d_p^+)$ and $G(d_p^-)$ were obtained by fitting with experimental data:

$$F(d_p^+) = \begin{cases} 0.04(1 + 2.5d_p^+)^{-1}, & \text{if } 0.03 \le d_p^+ < 0.3, \\ 0.0169(1.928(d_p^+)^{0.092} - 1)^{-1}, & \text{if } 0.3 \le d_p^+ < 10, \\ 0.0144(1 - 0.0858\exp(-0.0617)(d_p^+ - 10)))^2, & \text{if } d_p^+ \ge 10, \end{cases} \tag{4.42}$$

and

$$G(d_p^-) = \frac{A_{GI}}{(d_p^-)^{2.5}}, \tag{4.43}$$

with $A_{GI} = 6 \times 10^{-10.5}/(\rho g d_v^{2.5})$. For example, if we take $\rho = 1.25\text{kg}\cdot\text{m}^{-3}$, $g = 9.8\text{m}\cdot\text{s}^{-2}$, $\nu = 1.5 \times 10^{-5}\text{m}^2\cdot\text{s}^{-1}$ and $\rho_p = 2650\text{kg}\cdot\text{m}^{-3}$, we obtain $A_{GI} = 6.72$.

- **Shao-Lu's model:** Inspired by the finding that a good agreement with experimental results was achieved when $F(d_p^+)$ could be simply given by a constant, *i.e*, $F(d_p^+) = C_{SL}$, Shao and Lu (2000) proposed a simple expression with accounting for the inter-particle cohesion effect:

$$Sh_c = C_{SL}(1 + G(d_p^-)), \tag{4.44}$$

where G is a function accounting for the cohesion effect given by:

$$G(d_p^-) = \frac{A_{SL}}{(d_p^-)^2}, \tag{4.45}$$

with $C_{SL} = 0.0123$, $A_{SL} = \alpha/(\rho\gamma_\rho g d_v^2)$. Shao and Lu (2000) suggested that $\alpha = 3 \times 10^{-4}\text{kg}\cdot\text{s}^{-2}$, which yields $A_{SL} = 23.60$.

From Equations (4.27), (4.31) and (4.36), the critical Shield number of lifting mode, sliding mode and rolling mode can be rewritten as:

$$Sh_c^{lifting} = \frac{4}{3}\frac{1}{C_L^w(d_p^+)}\left(\cos\theta + \frac{A_C}{(d_p^-)^2}\right), \tag{4.46a}$$

$$Sh_c^{sliding} = \frac{4}{3}\frac{\mu_{se}}{C_D^w(d_p^+) + \mu_{se}C_L^w(d_p^+)}\left(\cos\theta + \frac{\sin\theta}{\tan\varphi_{se}} + \frac{A_C}{(d_p^-)^2}\right), \tag{4.46b}$$

$$Sh_c^{rolling} = \frac{4}{3}\frac{\mu_{re}}{C_D^w(d_p^+) + \mu_{re}C_L^w(d_p^+)}\left(\cos\theta + \frac{\sin\theta}{\tan\varphi_{re}} + \frac{A_C}{(d_p^-)^2}\right), \tag{4.46c}$$

where $A_C = 6C_C/(\pi\mu u_\nu)$ with $u_\nu = (\gamma_\rho g \nu)^{1/3}$. According to Equation (4.46), we propose a common formula of Sh_c as follows:

$$Sh_c = F(d_p^+)(H(\theta) + G(d_p^-)), \tag{4.47}$$

where $F(d_p^+)$ accounts for the aerodynamic effect, $H(\theta)$ accounts for the bed slope effect, and $G(d_p^-)$ accounts for the inter-particle cohesion effect. The expressions of $F(d_p^+)$, $H(\theta)$ and $G(d_p^-)$ are presented in Table 4.3 and are shown in Figure 4.14. We find that $G(d_p^-)$ takes the same expression for lifting, sliding and rolling mode, and that the formulas of $F(d_p^+)$ and $H(\theta)$ of sliding and rolling mode are similar, but both of them are different from the

lifting mode. Obviously, when the bed slope is null, i.e., $H(0) = 1$, Equation (4.47) reduces to $F(d_p^+)(1+G(d_p^-))$, which is consistent with Equation (4.41) and Equation (4.44).

	Lifting mode	**Sliding mode**	**Rolling mode**
$F(d_p^+)$	$\frac{4}{3}\frac{1}{C_L^w(d_p^+)}$	$\frac{4}{3}\frac{\mu_{se}}{C_D^w(d_p^+)+\mu_{se}C_L^w(d_p^+)}$	$\frac{4}{3}\frac{\mu_{re}}{C_D^w(d_p^+)+\mu_{re}C_L^w(d_p^+)}$
$H(\theta)$	$\cos\theta$	$\cos\theta+\frac{\sin\theta}{\tan\varphi_{se}}$	$\cos\theta+\frac{\sin\theta}{\tan\varphi_{re}}$
$G(d_p^-)$	$\frac{A_C}{(d_p^-)^2}$		

Table 4.3 Mathematical formula of $F(d_p^+)$, $G(d_p^-)$ and $H(\theta)$ for lifting, sliding and rolling mode.

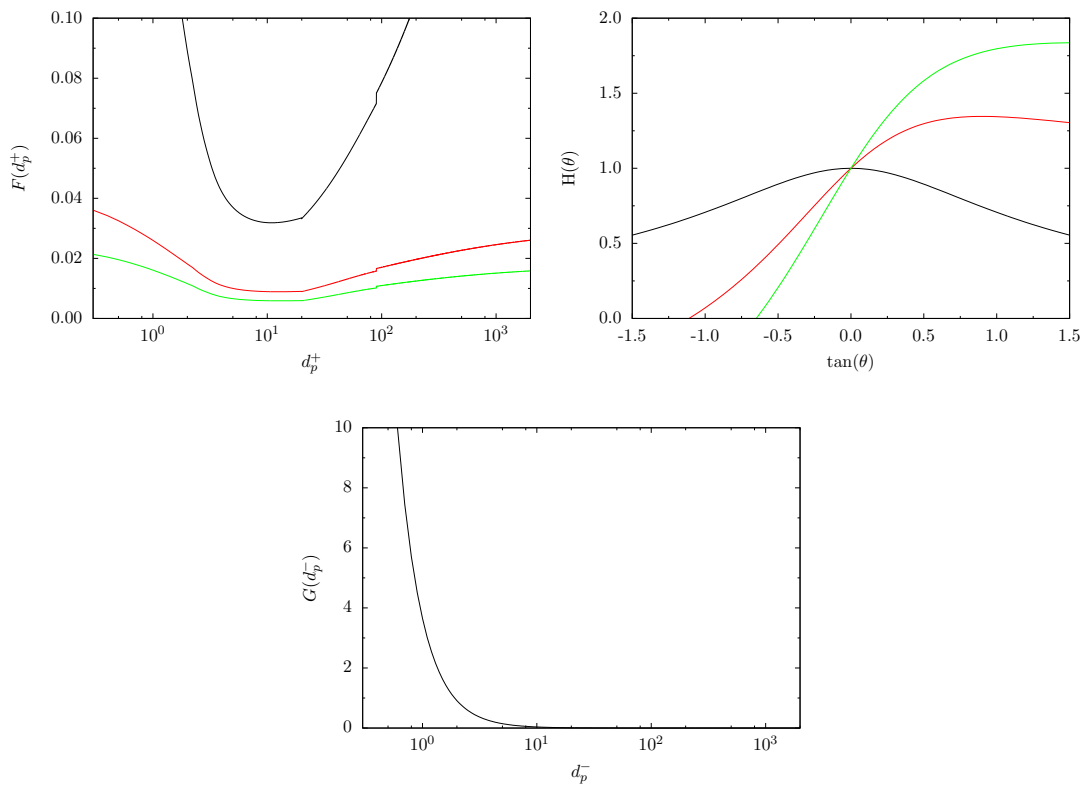


Figure 4.14 Functions $F(d_p^+)$, $H(\theta)$ and $G(d_p^-)$. —: Lifting mode; —: Sliding mode with $\varphi_{se} = 48°$; —: Rolling mode with $\varphi_{re} = 33°$.

From Equation (4.46), since $\mu_{se} > \mu_{re}$, it is easy to find:

$$Sh_c^{rolling} < Sh_c^{sliding} < Sh_c^{lifting}, \tag{4.48}$$

which is also examined in Figure 4.15. It indicates that rolling is the easiest mode for particle incipient motion. In practice, the critical Shield number Sh_c is measured using the transition from no transport to particle transport in wind tunnel experiments. The measured value is definitely a non-linear combination of $Sh_c^{lifting}$, $Sh_c^{sliding}$ and $Sh_c^{rolling}$. As shown in Figure 4.15, $Sh_c^{sliding}$ and $Sh_c^{rolling}$ are consist with the experimental data, whereas $Sh_c^{lifting}$ is greater than them, specially for a large d_p^-.

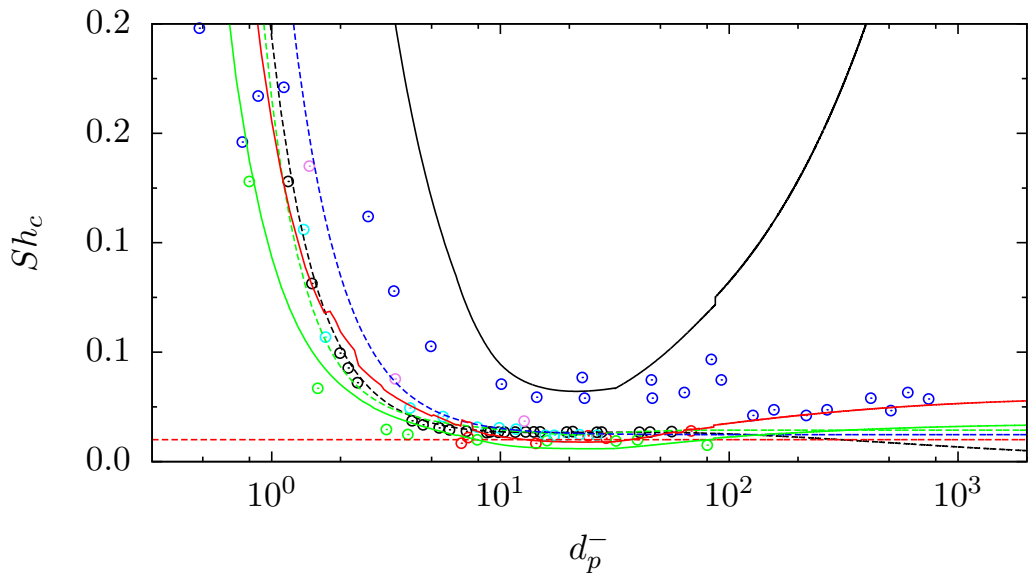


Figure 4.15 Critical Shield numbers Sh_c of different particle diameters. —: Lifting mode; —: Sliding mode with $\varphi_{se} = 48°$; —: Rolling mode with $\varphi_{re} = 33°$; - - : Equation (4.39); - - - : Equation (4.40); - - : Equation (4.41); - - : Equation (4.44). ⊙: White (1982); ⊙: Williams (1986); ⊙: Fernandez Luque and Van Beek (1976); ⊙: Fletcher (1976); ⊙: Chepil (1945); ⊙: Greeley and Iversen (1985).

Figure 4.16 shows the effect of bed slope on the critical Shield number Sh_c of lifting, sliding and rolling mode. It is found that Sh_c of lifting mode is always larger than that of sliding or rolling mode for an arbitrary bed slope θ smaller than 60°. For the lifting mode, Sh_c is an even function of θ and decreases with increasing θ if $\theta > 0$. For the sliding or rolling mode, Sh_c is an increasing function of θ and tends to a constant about 0.1 for large slopes.

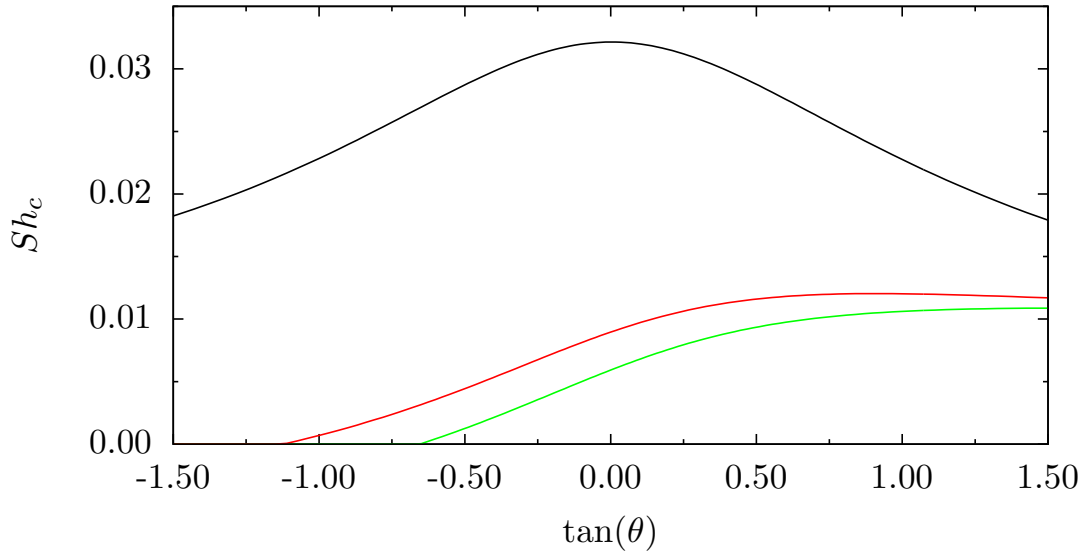


Figure 4.16 Bed slope effect on the critical Shield number Sh_c of lifting, sliding and rolling mode. —: Lifting mode; —: Sliding mode with $\varphi_{se} = 48°$; —: Rolling mode with $\varphi_{re} = 33°$.

4.2.4 Instantaneous entrainment model

Inspired by the energy model of particle entrainment proposed by Valyrakis et al. (2013) and Diplas et al. (2008), we propose a momentum criterion of particle incipient motions:

$$\gamma_{eff}(s_\Delta, s_p) \int_{t_0}^{t_0+T} P_i(t)dt > M_i, \text{ with } P_i(t) > 0, t_0 < t < t_0 + T, \tag{4.49}$$

where $P_i(t)$ denotes the summation of all the instantaneous forces (or torques) at the initial state, M_i the required minimum linear momentum (or angular momentum) for a particle incipient motion, and $\gamma_{eff}(s_\Delta, s_p)$ the effective coefficient of the momentum passing from the fluid in a grid cell on the particle during the incipient processes, respectively. It is assumed that $\gamma_{eff}(s_\Delta, s_p)$ is a function of the cell surface $s_\Delta = \Delta x \Delta y$ and the particle section area $s_p = \pi d_p^2/4$. Ideally, the coefficient $\gamma_{eff}(s_\Delta, s_p)$ is linear to the ratio of s_Δ and s_p, *i.e.*, $\gamma_{eff} \sim s_\Delta/s_p$. This means that if the grid becomes twice coarser, for example, $s'_\Delta = 2s_\Delta$, to keep the same number of entrained particles during the same duration under the same flow condition, the momentum passing from the fluids in a grid cell to a particle will become twice more, that is, the coefficient should satisfy $\gamma'_{eff}(s'_\Delta, s_p) = 2\gamma_{eff}(s_\Delta, s_p)$. Note that this entrainment criterion considers not only the magnitude of the forces, but also the spatial and

temporal effect in the numerical simulations. For the simplification, the effective coefficient is assumed to take the same value for different modes.

In detail, an instantaneous entrainment model for lifting mode is given by:

$$\gamma_{eff}(s_\Delta, s_p)\int_{t_0}^{t_0+T_i^{lifting}} F_i^{lifting}dt > m_p u_i^{lifting}, \text{ with } F_i^{lifting}(t) > 0, t_0 < t < t_0 + T_i^{lifting}, \tag{4.50}$$

where $F_i^{lifting}(t) = F_L' - G\cos\theta - F_C$ means the summation of forces acting the particle at the initial state with F_L' the instantaneous lift, $T_i^{lifting}$ the characteristic time for an incipient processes, m_p the particle mass, $u_i^{lifting}$ the required minimum velocity for an incipient process, respectively. Valyrakis et al. (2013) and Diplas et al. (2008) defined an incipient motion of lifting as displacing a height equal to the particle diameter in the direction perpendicular to the surface as shown in Figure 4.17. To accomplish a complete incipient lifting process only under the gravity, the minimum velocity should be assigned by $u_i^{lifting} = \sqrt{2gd_pH^{lifting}(\theta)}$, where the bed slope function takes form of $H^{lifting}(\theta) = \cos\theta$.

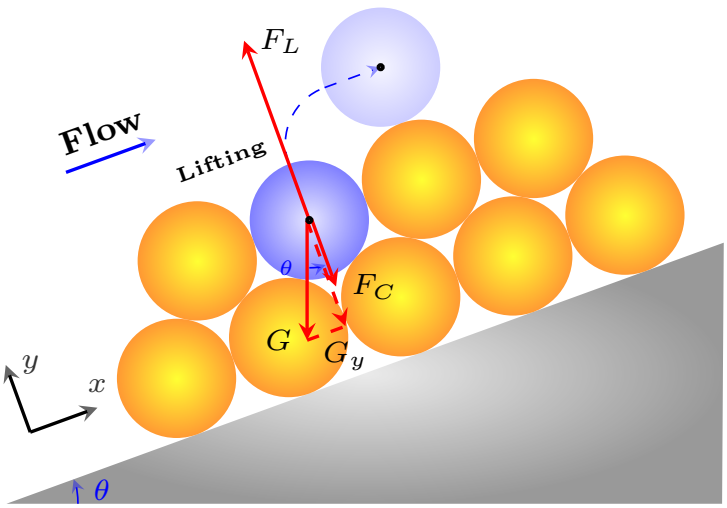


Figure 4.17 Sketch of a complete particle incipient process by lifting.

Similarly, an instantaneous entrainment model for sliding mode is given by:

$$\gamma_{eff}(s_\Delta, s_p)\int_{t_0}^{t_0+T_i^{sliding}} F_i^{sliding}dt > m_p u_i^{rolling}, \text{ with } F_i^{sliding}(t) > 0, t_0 < t < t_0 + T_i^{sliding}, \tag{4.51}$$

where $F_i^{sliding}(t) = F_D' - \mu_{si}(G\cos\theta + F_C - F_L') - G\sin\theta$ denotes the summation of the instantaneous torques acting the particle at the initial state, $T_i^{sliding}$ the characteristic time for an incipient processes, m_p the particle mass, $u_i^{sliding}$ the required minimum velocity for an

incipient process, respectively. We defined in the same way an incipient motion of sliding as displacing a distance equal to the particle diameter in the direction parallel to the surface as shown in Figure 4.18. To accomplish a complete incipient lifting process, the minimum velocity is approximated to $u_i^{sliding} = \sqrt{2\mu_{ri} g d_p H^{sliding}(\theta)}$, where the bed slope function takes form of $H^{sliding}(\theta) = \left(\cos\theta + \dfrac{\sin\theta}{\cos\varphi_{se}}\right)$.

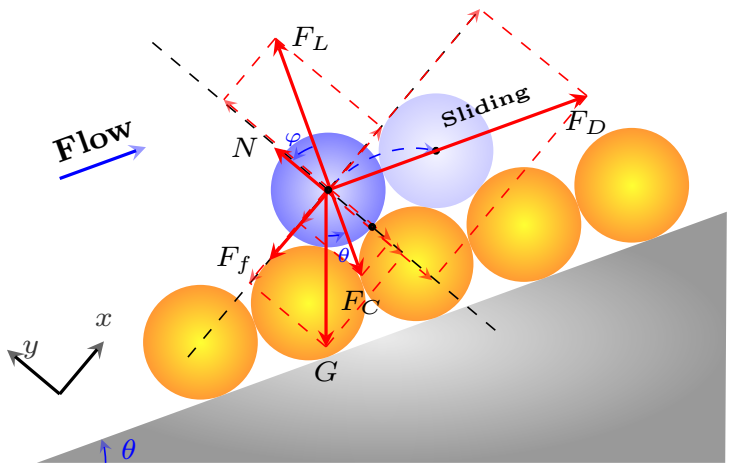


Figure 4.18 Sketch of a complete particle incipient process by sliding.

In the same manner, an instantaneous entrainment model for rolling mode is given by:

$$\gamma_{eff}(s_\Delta, s_p) \int_{t_0}^{t_0+T_i^{rolling}} M_i^{rolling} dt > I_p \omega_i^{rolling}, \text{ with } M_i(t) > 0, t_0 < t < t_0 + T_i^{rolling}, \quad (4.52)$$

where $M_i^{rolling}(t) = (F_D' - \mu_{ri}(G\cos\theta + F_C - F_L') - G\sin\theta)d_p/2$ represents the summation of the instantaneous torques acting the particle at the initial state, $T_i^{rolling}$ the characteristic time for an incipient processes, I_p the particle inertial moment, $\omega_i^{rolling}$ the required minimum velocity for an incipient process, respectively. We defined in the same way an incipient motion of rolling as rolling a distance equal to the particle diameter in the direction parallel to the surface as shown in Figure 4.19. To accomplish a complete incipient rolling process, the minimum rotation velocity is approximated to $\omega_i^{rolling} = \sqrt{20\mu_{ri} g H^{rolling}(\theta)/(7d_p)}$, where the bed slope function takes form of $H^{rolling}(\theta) = \left(\cos\theta + \dfrac{\sin\theta}{\cos\varphi_{re}}\right)$. Since we have

$I_p = 7m_p d_p^2/20$ for spherical particles, Equation (4.52) can be rewritten as:

$$\gamma_{eff}(s_\Delta, s_p)\int_{t_0}^{t_0+T_i^{rolling}} F_i^{rolling}dt > m_p u_i^{rolling}, \text{ with } F_i^{rolling}(t) > 0, t_0 < t < t_0 + T_i^{rolling}, \tag{4.53}$$

where the summation of the forces is given by $F_i^{rolling}(t) = F_D' - \mu_{ri}(G\cos\theta + F_C - F_L') - G\sin\theta)$, the required minimum velocity equals $u_i^{rolling} = 7\omega_i^{rolling} d_p/10$.

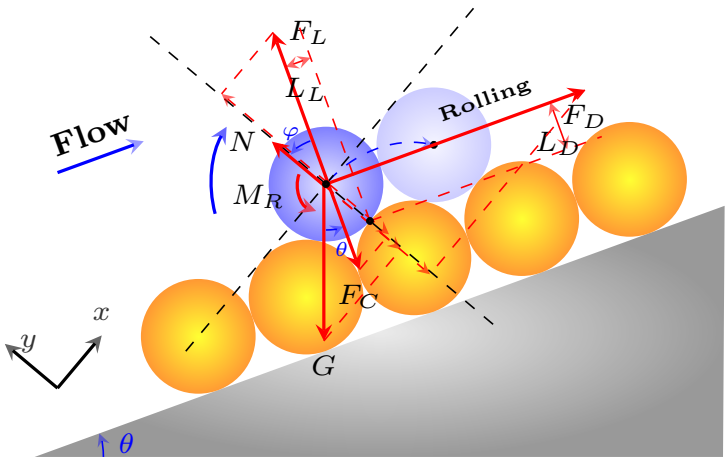


Figure 4.19 Sketch of a complete particle incipient process by rolling.

It is easy to find that the model of rolling mode is nearly same with that of sliding mode except the desired minimum velocity, if μ_{ri} in Equation (4.53) is replaced by μ_{si}. Moreover, Figure 4.15 shows that the threshold friction velocity of sliding mode is approximately equal to that of rolling mode. Inspired by the similarity between these two modes, we propose a mixed mode — rolling-sliding mode, to describe the particle incipient motion on the bed surface. The corresponding instantaneous entrainment model is then given by:

$$\gamma_{eff}(s_\Delta, s_p)\int_{t_0}^{t_0+T_i^{mixed}} F_i^{mixed}dt > m_p u_i^{mixed}, \text{ with } F_i^{mixed}(t) > 0, t_0 < t < t_0 + T_i^{mixed}, \tag{4.54}$$

where the resultant force is given by $F_i^{mixed}(t) = F_D' - \mu_{mixed}(G\cos\theta + F_C - F_L') - G\sin\theta)$, the required minimum velocity equals $u_i^{mixed} = \sqrt{2\mu_{mixed} g d_p H^{mixed}(\theta)}$ with the bed slope function of the mixed rolling-sliding mode given by $H^{mixed}(\theta) = \left(\cos\theta + \dfrac{\sin\theta}{\cos\varphi_{mixed}}\right)$. In practice, the mixed friction angle φ_{mixed} is assigned by the angle of repose, *i.e.*, $\varphi_{mixed} = 33°$ for sand particles, and the static friction coefficient of the mixed rolling-sliding mode is then given by $\mu_{mixed} = \tan\varphi_{mixed}$.

In a turbulent boundary layer, the mechanism of particle incipient motion is complex due to the randomness of the aerodynamic force and the impact of near-wall turbulent structures. White (1940) claimed that the particle entrainment events mainly takes place in the region of high speed fluids. By using quadrant analysis on the experimental wind velocity, Wiggs and Weaver (2012) then found that the particle incipient motion is highly correlated with the the sweep events of a turbulent bursting process. Inspired by these findings, Huang (2015) proposed a relation between the instantaneous aerodynamic forces and the average ones:

$$F_L' = F_L \frac{u_{\oplus}' w_{\ominus}'}{\langle u_{\oplus}' w_{\ominus}' \rangle}, \tag{4.55a}$$

$$F_D' = F_D \frac{u_{\oplus}' w_{\ominus}'}{\langle u_{\oplus}' w_{\ominus}' \rangle}, \tag{4.55b}$$

where $u_{\oplus}' w_{\ominus}'$ means Reynolds stress of the sweep events related to the motion of high speed fluid, and $\langle \cdot \rangle$ denote the ensemble average. Equation (4.55) builds a strong relationship between the sweep action and the particle entrainment. According to quadrant analysis, the sweep event is given by $u_{\oplus}' w_{\ominus}' = \max(u',0) \times \min(w',0)$. In a boundary layer, the ensemble average can be replaced by the spatial average in the homogeneous span-wise direction.

4.3 Particle transport motion

After the particle incipient motion, the particle moves into the nearby flow or slides on the bed surface. When the particle detaches far away from the surface and flies within the flow, the particle motion is mainly governed by the aerodynamic forces and the gravity. When the aerodynamic lift force is not large enough to entrain the particle or the falling particle fails to rebound, it will probably slide or roll on the bed surface due to the drag or gravity, and finally slows down by the wall friction.

4.3.1 Equation of airborne particle motion

Contrary to the turbulent flow described in an Eulerian description, the particle motion is studied in a Lagrangian description. The governing equations of particle motion are given by:

$$\frac{d\vec{x}_p(t)}{dt} = \vec{u}_p(\vec{x}_p(t),t), \tag{4.56a}$$

$$\frac{d\vec{u}_p(\vec{x}_p(t),t)}{dt} = \frac{1}{m_p} \sum \vec{F}, \tag{4.56b}$$

$$\frac{d\vec{\omega}_p(\vec{x}_p(t),t)}{dt} = \frac{1}{I_p}\sum \vec{M}, \tag{4.56c}$$

where $\vec{x}_p(t)$ is the particle position at the time t, $\vec{u}_p(\vec{x}_p(t),t)$ (or $\vec{\omega}_p(\vec{x}_p(t),t)$) the particle velocity (or angular velocity) at the position $\vec{x}_p(t)$ and at the time t, m_p the particle mass, I_p the inertial moment, $\vec{F}$ the forces acting on the particle, $\vec{M}$ the torques acting on the particle, respectively.

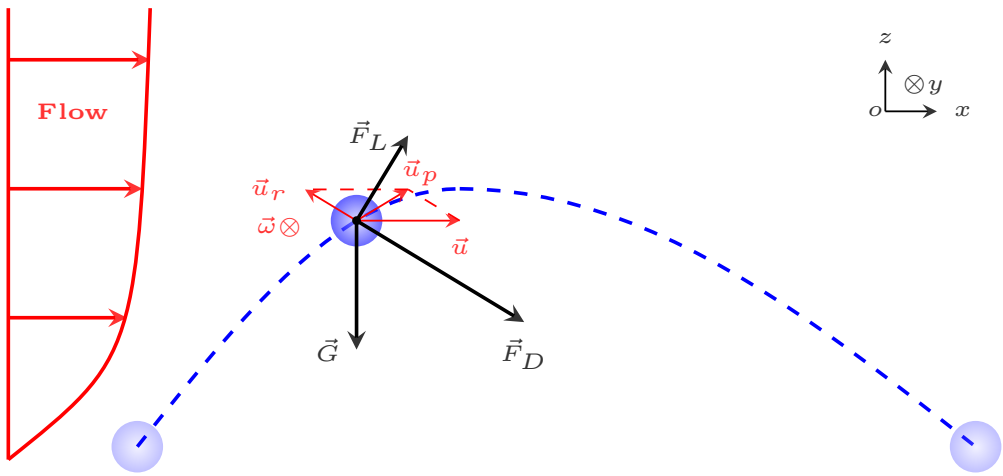


Figure 4.20 Schematic illustration of the forces acting on an airborne particle. $\vec{u}$ is the fluid velocity, $\vec{\omega}$ is the vorticity and $\vec{u}_p$ is the particle velocity. The aerodynamic drag $\vec{F}_D$ is in the opposite direction of the particle-fluid relative velocity $\vec{u}_r = \vec{u}_p - \vec{u}_r$ according to Equation (4.1). The aerodynamic lift $\vec{F}_L$ is in the direction of $\vec{\omega} \times \vec{u}_r$ according to Equation (4.7).

Considering a heavy airborne particle whose density is much larger than the fluid's, *i.e.*, $\rho_p \gg \rho$, it is known that the aerodynamic forces and the gravity are dominant, and that these forces induced by the disturbance are negligible. Moreover, the particle rotation is usually neglected. Hence, as shown in Figure 4.20, the motion of an airborne particle is determined only by the forces of drag, lift and gravity. Therefore, Equation (4.56) is rewritten as:

$$\frac{d\vec{x}_p(t)}{dt} = \vec{u}_p(\vec{x}_p(t),t), \tag{4.57a}$$

$$\frac{d\vec{u}_p(\vec{x}_p(t),t)}{dt} = \frac{1}{m_p}(\vec{F}_D + \vec{F}_L + \vec{G}). \tag{4.57b}$$

As discussed in Section 4.1, the aerodynamic lift is important in the near-wall region, and can be neglected when the particle stays far away from the surface. For spherical heavy particles, we have $(\rho_p - \rho)/\rho_p \approx 1$ and $m_p = \pi\rho_p d_p^3/6$. Then, inserting Equations (4.1) and

(4.21) into Equation (4.57b) yields by neglecting $\vec{F}_L$:

$$\frac{d\vec{u}_p(\vec{x}_p(t),t)}{dt} = \frac{\vec{u}(\vec{x}=\vec{x}_p(t),t) - \vec{u}_p(\vec{x}_p(t),t)}{\tau'_p} - \vec{g}, \tag{4.58}$$

where τ'_p is the particle characteristic time scale responding to the fluid, which is given by:

$$\tau'_p = \frac{\rho_p d_p^2}{18\rho\nu}\frac{1}{f(Re_p)} = \frac{\tau_p}{f(Re_p)}, \tag{4.59}$$

where $\tau_p = \rho_p d_p^2/(18\rho\nu)$ denotes the particle response time for $Re_p \ll 1$. Since τ'_p depends on the instantaneous particle Reynolds number Re_p, it is impossible to be evaluated a priori. Therefore, τ_p is widely used to estimate the characteristic respond time scale of solid particles in numerical simulations [Dupont et al. (2013); Vinkovic et al. (2006)].

When the force balance in the vertical direction is reached, the particle vertical acceleration should be null, which yields the terminal velocity w_p^t from Equation (4.58):

$$0 = \frac{w_p^t}{\tau'_p} - g. \tag{4.60}$$

Substituting Equation (4.59) into Equation (4.60), we obtain the formula of the particle terminal velocity:

$$w_p^t = \frac{g\tau_p}{f(Re_p)} = \frac{\rho_p g d_p^2}{18\rho\nu}\frac{1}{f(Re_p)}. \tag{4.61}$$

According to Equation (4.4), we have:

$$w_p^t = \begin{cases} \frac{\rho_p g d_p^2}{18\rho\nu}, & \text{if } Re_p \ll 1; \\ 1.66\sqrt{\gamma_\rho g d}, & \text{if } Re_p > 1000. \end{cases} \tag{4.62}$$

To characterize the airborne particle motion modes, some non-dimensional parameters are introduced. Firstly, the Stokes number is defined as the ratio between the particle and fluid characteristic time, *i.e.*, $St = \tau_p/\tau_f$ which yields:

$$\begin{aligned} &St_L = \tau_p/\tau_L, \text{ if using the time scale of turbulent large structures } \tau_L; \\ &St_\eta = \tau_p/\tau_\eta, \text{ if using the time scale of turbulent small structures } \tau_\eta. \end{aligned} \tag{4.63}$$

Given a boundary layer with the thickness δ and the friction velocity u_*, the time scales are defined by $\tau_L = \delta/u_*$ and $\tau_\eta = \sqrt{\nu\delta/u_*^3}$. The Stokes number reflects the effect of the particle inertia, which can characterize the particle motion modes as:

- $St_\eta \ll 1$, particles are carried by the turbulent small eddies. Thus, particles are in pure suspension;
- $St_L \gg 1$, even large scale fluctuations is not able to affect the particle motions. The trajectories of solid particles are mainly determined by the gravity and the interaction with the ground. Thus, particles are in pure saltation.

Another useful non-dimensional parameter is the gravity parameter, which is defined as the ratio of particle terminal velocity and the flow friction velocity, that is:

$$\gamma_g = \frac{w_p^t}{u_*}. \tag{4.64}$$

When $Re \ll 1$, according to Equation (4.62), Equation (4.64) reduces to an simple formula $\gamma_g = g\tau_p/u_*$, which is widely used to evaluate the value of the gravity parameter in a priori study. When τ_p is of the order of τ_η or of τ_L, *i.e.*, $St_\eta \approx 1$ or $St_L \approx 1$, the motion of solid particles is determined by the gravity and the inertia. In this case, Taniere et al. (1997) used the gravity parameter to distinguish two new regimes of particle motions:

- $\gamma_g < 1$, modified suspension;
- $\gamma_g > 1$, modified saltation.

4.3.2 Equation of bed-load particle motion

In this thesis, we firstly focus on the motion of bed-load particles, which are defined as moving particles on the wall. For instance, some particles after the collision with the surface fail to rebound and then slide or roll on the wall if their tangential velocities are large enough. The dynamic equation of bed-load particle motion in the Lagrangian description is given by:

$$\frac{d\vec{x}_p(t)}{dt} = \vec{u}_p(\vec{x}_p(t),t) + \vec{\omega}_p \times \vec{r}_p, \tag{4.65a}$$

$$\frac{d\vec{u}_p(\vec{x}_p(t),t)}{dt} = \frac{1}{m_p}\sum \vec{F}, \tag{4.65b}$$

$$\frac{d\vec{\omega}_p(\vec{x}_p(t),t)}{dt} = \frac{1}{I_p}\sum \vec{M}, \tag{4.65c}$$

where $\vec{x}_p(t)$ is the particle position at the time t, $\vec{u}_p(\vec{x}_p(t),t)$ (or $\vec{\omega}_p(\vec{x}_p(t),t)$) the particle velocity (or angular velocity), $\vec{r}_p$ the vector from the particle center to the contact point with the bed, m_p the particle mass, I_p the inertial moment, $\vec{F}$ the forces acting on the particle, $\vec{M}$ the torques acting on the particle, respectively.

Considering a spherical particle submerged in a turbulent boundary layer rolling and sliding on the wall surface, the aerodynamic lift, the friction and the cohesion forces must obviously be taken into account, and the rotation is important for the displacement of rolling particles. Hence, Equation (4.65) is rewritten as:

$$\frac{d\vec{x}_p(t)}{dt} = \vec{u}_p(\vec{x}_p(t),t) + \vec{\omega}_p \times \vec{r}_p, \tag{4.66a}$$

$$\frac{d\vec{u}_p(\vec{x}_p(t),t)}{dt} = \frac{1}{m_p}(\vec{F}_D + \vec{F}_L + \vec{G} + \vec{F}_f + \vec{N}), \tag{4.66b}$$

$$\frac{d\vec{\omega}_p(\vec{x}_p(t),t)}{dt} = \frac{1}{I_p}(\vec{M}_{F_D} + \vec{M}_{F_L} + \vec{M}_G + \vec{M}_R), \tag{4.66c}$$

where $\vec{F}_D$, $\vec{F}_L$, $\vec{G}$, $\vec{F}_f$, $\vec{N}$ respectively denote the aerodynamic drag, the aerodynamic lift, the gravity, the friction and the reaction from the wall, and $\vec{M}_{F_D}$, $\vec{M}_{F_L}$, $\vec{M}_G$ respectively represent the moment induced by the aerodynamic drag, the aerodynamic lift, the gravity, and I_p is the inertial moment of a sphere to the pivot point.

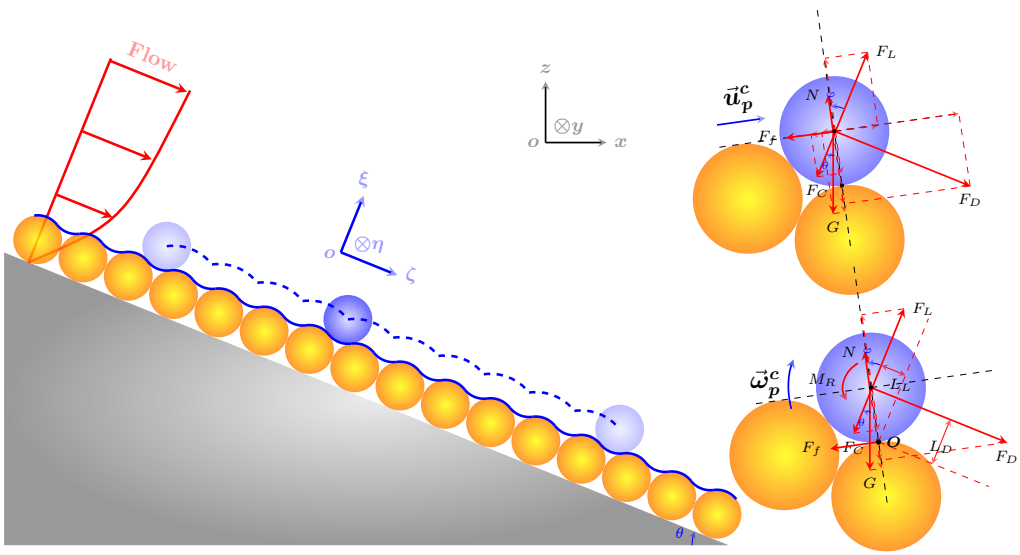


Figure 4.21 Schematic illustration of the forces acting on a bed load particle and of its motion. $\vec{u}_p^c$ is the particle velocity and $\vec{\omega}_p^c$ is the particle angular velocity in the curvilinear coordinate system (ζ, η, ξ). F_D denotes the aerodynamic drag, F_L the aerodynamic lift, F_C the cohesion, G the gravity, N the reaction, F_f the friction, M_R the friction torque, respectively.

As shown in Figure 4.21, supposing the sliding (and rolling) friction coefficient μ_s (and μ_r), we then know $F_f = \mu_s N$ and $M_R = \mu_r d_p N/2$. Using the curvilinear coordinate system (ζ, η, ξ) conformable with the rough surface, the corresponding particle velocities

are denoted by u^c, v^c and w^c. Hence, Equation (4.66) becomes:

$$\frac{d\zeta_p(t)}{dt} = u_p^c(\zeta_p(t),t) + \frac{d_p}{2}\omega_p^c(\zeta_p(t),t), \tag{4.67a}$$

$$\frac{du_p^c(\zeta_p(t),t)}{dt} = \frac{\cos\varphi}{m_p}(F_D - G\sin\theta - \mu_{se}(G\cos\theta + F_C - F_L)), \tag{4.67b}$$

$$\frac{d\omega_p^c(\zeta_p(t),t)}{dt} = \frac{d_p\cos\varphi}{2I_p}(F_D - G\sin\theta - \mu_{re}(G\cos\theta + F_C - F_L)), \tag{4.67c}$$

where $\zeta_p(t)$ is the particle position on the rough wall, $u_p^c(\zeta_p(t),t)$ the particle velocity, $\omega_p^c(\zeta_p(t),t)$ the particle angular velocity, $\mu_{se} = \tan(\varphi_s + \varphi)$ the equivalent sliding friction coefficient with $\varphi_s = \arctan\mu_s$, $\mu_{re} = \tan(\varphi_r + \varphi)$ the equivalent rolling friction coefficient with $\varphi_r = \arctan\mu_r$ and θ the local bed slope, respectively. I_p denotes the inertial moment of a sphere to the pivot point O as shown in Figure 4.21, *i.e.*, $I_p = m_p d_p^2/10 + m_p d_p^2/4 = 7m_p d_p^2/20$. Note that Equation (4.67) is satisfied if $F_L \leq G\cos\theta + F_C$, whereas Equation (4.57) is used to describe the particle motion if $F_L > G\cos\theta + F_C$.

Assuming $U_p^c(\zeta_p(t),t) = u_p^c(\zeta_p(t),t) + \frac{1}{2}d_p\omega_p^c(\zeta_p(t),t)$ and from Equation (4.67), the governing equations of bed-load particle motion for the mixed mode — rolling-sliding are deduced as:

$$\frac{d\zeta_p(t)}{dt} = U_p^c(\zeta_p(t),t), \tag{4.68a}$$

$$\frac{dU_p^c(\zeta_p(t),t)}{dt} = \frac{1}{m_{p,eff}}\left(F_D - G\sin\theta - \mu_{eff}(G\cos\theta + F_C - F_L)\right), \tag{4.68b}$$

where $m_{p,eff} = 12\cos\varphi/(7m_p)$ and $\mu_{eff} = (7\mu_{se} + 5\mu_{re})/12$ denote the effective mass and the effective friction coefficient of the roll-sliding motion. respectively. It is known that $m_{p,eff}$ and μ_{eff} depend on the unknown angle φ, which is determined by the arrangement of solid particles and by the position of the moving particle on the rough surface. In our numerical simulations, since the grid size is much larger than the particle diameter d_p, the local surface geometry at the scale of d_p is not able to be described. Therefore, it is impossible to evaluate μ_{eff} and $m_{p,eff}$. In this case, the rough surface is assumed to be regarded as a smooth one without accounting for the geometries at small scales. Hence, we have $m_{p,eff} = 12/(7m_p)$ as $\varphi = 0$, and μ_{eff} is considered as the dynamic friction coefficient for a rough surface. Quartier et al. (2000) found from his experiments that the dynamic angle of repose is 8.5° for a rough surface composited by sand piles. Hence, we suppose $\mu_{eff} = \tan 8.5°$. Actually, we will conduct some testing cases with different values of μ_{eff} and investigate the influence of the coefficient μ_{eff} on the sand dune deformation in the numerical simulation of wind erosion.

4.3.3 Lagrangian stochastic model

In the equation of particle motion, *e.g.*, in Equation (4.58), the fluid velocity $u_i(x_p(t),t)$ is exactly the summation of the filtered velocity $\tilde{u}_i(x_p(t),t)$ supplied by the LES and the velocity of sub-grid scales $u_i''(x_p(t),t)$ introduced by the models, that is:

$$u_i(x_p(t),t) = \tilde{u}_i(x_p(t),t) + u_i''(x_p(t),t). \tag{4.69}$$

According to Vinkovic et al. (2006), $u_i''(x_p(t),t)$ is evaluated by a Lagrangian stochastic model:

$$du_i''(x_p(t),t) = \left(-\frac{1}{T_L^p} + \frac{2}{k_{sgs}}\frac{dk_{sgs}}{dt}\right) u_i''(x_p(t),t)dt + \sqrt{\frac{4k_{sgs}}{3T_L^p}}dW_i, \tag{4.70}$$

where k_{sgs} is the subgrid kinetic energy, T_L^p the time scale of Lagrangian velocity correlation following the particle motion, and dW_i the Wiener process vector with $\langle dW_i dW_j \rangle = \delta_{ij}dt$, respectively. Equation (4.70) is a simple Langevin equation, which describes the diffusion process of $u_i''(x_p(t),t)$ by a stochastic differential equation in terms of k_{sgs} and T_L^p. k_{sgs} is calculated by Equation (2.33), and the timescale T_L^p following the particle motion is given by:

$$T_L^p = \frac{T_L}{\alpha_{grav} + \alpha_{iner}}, \tag{4.71}$$

where α_{grav} and α_{iner} are two coefficients accounting for gravitational and inertial effect on decorrelating the particle motion from the fluid particle's, and T_L is the timescale of Lagrangian velocity correlation following fluid particle motion:

$$T_L = \frac{4k_{sgs}}{3C_\kappa \tilde{\varepsilon}}, \tag{4.72}$$

where C_κ is the Kolomogorov constant and $\tilde{\varepsilon}$ is the filtered dissipation rate. The further details of the Lagrangian stochastic model can be found in the thesis of Aguirre (2005) and Vinkovic (2005).

4.3.4 Two-way coupling

The two-way coupling is a numerical technique accounting for the effect of solid particles on the carried flow. Firstly, an additional drag force is introduced to the fluid momentum equation (Equation (2.25b)), that is:

$$\vec{f}_{u_i,drag} = -\frac{1}{\rho V_{grid}} \sum_{i=1}^{N_p} \rho_p V_p f(Re_p) \frac{\vec{u}(\vec{x}_p(t),t) - \vec{u}_p(\vec{x}_p(t),t)}{\tau_p}, \tag{4.73}$$

where $V_{grid} = \Delta x \Delta y \Delta z$ is the grid cell volume, V_p the solid particle volume, N_p the number of solid particle in the grid cell, respectively. Secondly, the corresponding source term is introduced in the transport equation of sub-grid kinetic energy (Equation (2.33)):

$$s_{sgs} = -\frac{1}{\rho V_{grid}} \sum_{i=1}^{N_p} \rho_p V_p f(Re_p) \frac{2k_{sgs}}{\tau_p + T_L}. \tag{4.74}$$

As noted in Elghobashi (1994), the interaction between solid particles and the turbulent flows can not be neglected, and two-way coupling should be applied in the numerical simulations when the volume fraction, defined by $\Phi_p = \sum_{i=1}^{N_p} V_p / V_{grid}$, is larger than 10^{-6}.

4.4 Particle surface interaction

When the particle falls and makes the collision with the surface composed by particles, two events occur: firstly, if its momentum is large, then it rebounds and continues its motion in the fluid flow with a new velocity, and some of the particles in contact with the bouncing one may begin their incipient motion; secondly, if its momentum is too small to achieve the rebound process, it may continue to slide or roll on the surface until the deposition occurs due to the friction effect.

4.4.1 Rebound processes

The rebound process is obviously a stochastic process, which is usually simulated by an empiric model based on the experimental observations. Commonly, the particle-surface collision process is finished in a very short time, hence, it is considered to be independent on the influence of the surrounding flow. Figure 4.22 shows the sketch of the three-dimensional rebound process $P-O-R$ of a spherical particle. The impacting particle P with the incident angle θ_i and the speed v_{imp} falls down and makes the collision with the particle surface at the origin O. After the collision, the rebounding particle gains a new speed v_{reb} and a new direction determined by the rebound angle θ_r and the $y-$ deviation $\theta_{r,y}$. In practice, due to the difficulty of capturing the three-dimensional trajectories of solid particles by a fast video camera, we usually analyzed the two-dimensional rebound process $P-O-R_y$ on the incident plane Oxz, in which the rebound speed and angle are expressed by v_{reb}^{xz} and θ_r.

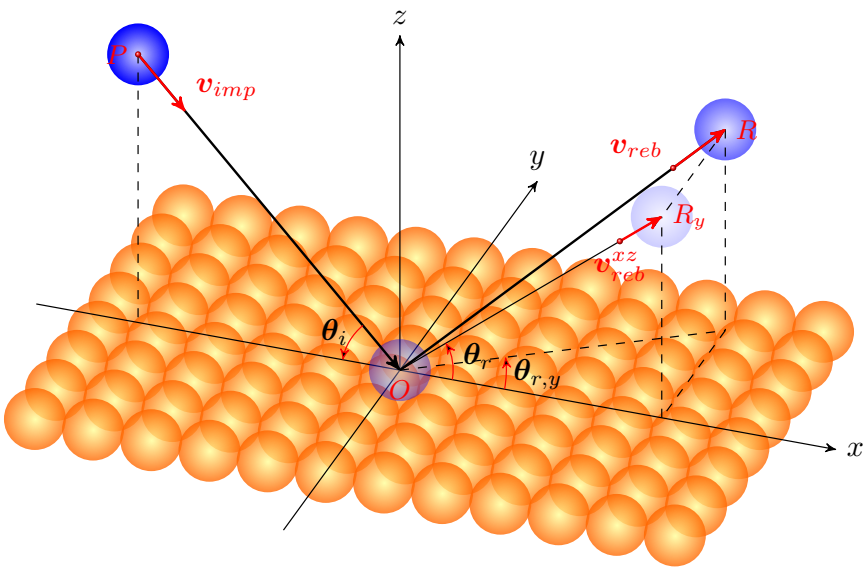


Figure 4.22 Sketch of the rebound process. $P-O-R$ describes the three-dimensional rebound process with the incident angle θ_i, the impacting speed v_{imp}, the rebound speed v_{reb}, the rebound angles θ_r and $\theta_{r,y}$. $P-O-R_y$ describes the two-dimensional rebound process projected on the incident plane Oxz with its rebound speed v_{reb}^{xz} and the rebound angle θ_r.

Probabilistic rebound model: In a saltating layer, the rebound probability proposed by Anderson and Haff (1991) is given by:

$$P_{reb}(v_{imp}) = 0.95(1-\exp(-\beta_{imp}v_{imp})), \tag{4.75}$$

where v_{imp} is the impacting particle speed and β_{imp} is an empirical parameter of order $2\mathrm{s}\cdot\mathrm{m}^{-1}$. Equation (4.75) indicates that the rebound probability monotonically decreases with the impacting speed v_{imp}, and that the null probability is obtained when v_{imp} is equal to 0.

When the rebound processes is achieved, the velocity norm v_{reb} and angles θ_r and $\theta_{r,y}$ of the rebounding particle must be specified. Experimental results shown that:

- The remained kinetic energy $(v_{reb}^{xz})^2$ of rebounding particles in the incident plane approximately obeys to a normal distribution [Wang et al. (2008)].
- The rebounding angle θ_{reb} almost satisfies an exponential distribution [Kang et al. (2008); Rice et al. (1996); Willetts and Rice (1986)].

Unfortunately, it is difficult to investigate probabilistic characteristic of the deviated angle θ_y in y- direction, due to the technical restriction from an experimental point of view. In the numerical steady state saltation model of Kok and Renno (2009), the remained

kinetic energy is approximately equal to $45 \pm 22\%$ of the impacting kinetic energy, *i.e.*, $(v_{reb}^{xz})^2 = (0.45 \pm 0.22) v_{imp}^2$; The rebounding angle obeys an exponential distribution with a mean value 40° from the ground, whereas the deviated angle is not taken into account.

Beldajine's model: A three-dimensional particle-surface collision experiment was carried out by Beladjine et al. (2007). In this experiments, both the incident speed and the incident angle varied in a wide range: $74\sqrt{gd_p} \leq v_{imp} \leq 161\sqrt{gd_p}$ and $0^\circ \leq \theta_i \leq 90^\circ$. Let's denote the incident velocity $\vec{v}_{imp} = (v_{imp,x}, 0, v_{imp,z})$ and the rebound velocity $\vec{v}_{reb} = (v_{reb,x}, v_{reb,y}, v_{reb,z})$ as shown in Figure 4.22, the two-dimensional mean restitution coefficients $\bar{e}_z$ and $\bar{e}_{xz}$ are obtained by fitting with the experimental data:

$$\bar{e}_{xz} = \left\langle \frac{v_{reb}^{xz}}{v_{imp}} \right\rangle = A - B \sin \bar{\theta}_i, \tag{4.76a}$$

$$\bar{e}_z = \left\langle \frac{v_{reb,z}}{v_{imp,z}} \right\rangle = \frac{A_z}{\sin \bar{\theta}_i} - B_z \approx \bar{e}_{xz} \sin \bar{\theta}_r, \tag{4.76b}$$

where $v_{reb}^{xz} = \sqrt{v_{reb,x}^2 + v_{reb,z}^2}$ denotes the projection of the rebounding velocity on the incident plan Oxz, θ_i is the incident angle, θ_r is the rebound angle, $A_z = 0.30$, $B_z = 0.15$, $A = 0.87$, and $B = 0.72$. Note that the rebounding velocity can be reconstructed by Equation (4.76), except for the y- component $v_{reb,y}$. According to Dupont et al. (2013), the horizontal deviated angle $\theta_{reb,y}$ is assumed to be normally distributed with a zero mean and 10° variation, then we have $v_{reb,y} = v_{reb,x} \tan \theta_{reb,y}$.

In ARPS, the two restitution coefficients are supposed to obey a Gaussian probabilistic distribution. The mean values are evaluated by Equation (4.76), and the standard variations are assumed to be:

$$\sigma_{\theta_r} = \bar{\theta}_r, \tag{4.77a}$$

$$\sigma_{e_{xz}} = \left(2 - \frac{\theta_r}{\bar{\theta}_r}\right) \bar{e}_{xz}. \tag{4.77b}$$

It is remarked that Equation (4.77) indicates that the rebound velocity is smaller when the rebound angle takes a larger value, which is consist with the experimental observation of Anderson et al. (1991).

4.4.2 Splash entrainment

The splash entrainment is caused by the rebound: when a particle with a large velocity impacts on the surface composited by solid particles, some other particles can be ejected

due to this strong collision. In practice, a splash model is used to specify the number and velocity of the ejected particles. According to Ungar and Haff (1987), Kok and Renno (2009) and Dupont et al. (2013), the number of ejected particles is considered as a function of the impacting velocity:

$$N_{ejc} \approx \frac{a}{\sqrt{gd_p}} v_{imp} \tag{4.78}$$

where a is a dimensional constant, which is suggested to be in range of $0.01 - 0.05$. One drawback of Equation (4.78) is that the incident angle of impacting particle is not taken into account. By fitting with experimental data, Beladjine et al. (2007) obtained a power law as follows:

$$N_{ejc} \approx 0.02(1 - A^2 + 2AB\sin\theta_i - B^2\sin^2\theta_i)\left(\frac{v_{imp}}{\sqrt{gd_p}}\right)^{2-p} \tag{4.79}$$

where $p \approx 0.5$, θ_i is the incident angle, A, B are identical to the values found in Equation (4.76a).

Similar to the rebound model, both the velocity norm v_{ejc} and angle θ_{ejc} from the ground need to be modeled. Anderson et al. (1991) suggested that v_{ejc} follows an exponential distribution:

$$P(v_{ejc}) = \frac{1}{\langle v_{ejc}\rangle}\exp\left(-\frac{v_{ejc}}{\langle v_{ejc}\rangle}\right) \tag{4.80}$$

where $\langle v_{ejc}\rangle$ denotes the mean ejected speed. Rice et al. (1995) suggested $\langle v_{ejc}\rangle = 0.08 v_{imp}$. Using the conservation of momentum, the momentum lost during the rebound process is totally spent on ejecting other particles, Kok and Renno (2009) found:

$$\frac{\langle v_{ejc}\rangle}{\sqrt{gd_p}} = \frac{\alpha_{ejc}}{a}\left(1 - \exp\left(-\frac{v_{imp}}{40\sqrt{gd_p}}\right)\right) \tag{4.81}$$

where α_{ejc} is the momentum fraction lying in the range of $0.14 - 0.20$ [Rice et al. (1995)]. In the numerical model of saltation, Kok and Renno (2009) choses $\alpha_{ejc} = 0.15$. The ejection angle θ_{ejc} follows a Gaussian distribution with a mean of 60° and a variation of 15°. As for the rebound process, the horizontal deviated angle $\theta_{ejc,y}$ is also assumed to be normally distributed with a mean value of 0° and a variation of 10°.

4.4.3 Particle deposition

Particle deposition describes the action that a particle slows down on a surface, lastly stops and becomes part of the surface. It is completely different from the deposition process, which means the process of falling down of an airborne particle. The particle will be deposited on

the surface if its velocity is theoretically null. In numerical simulation, solid particles either stop immediately after colliding with the surface or deposit after rolling a small distance. The deposition criterion in ARPS is that the particle velocity is too small to move on the surface.

4.5 Sand dune deformation

4.5.1 Temporal evolution of dune height

According to the conservation of the mass, the evolution of dune height can be written as

$$\frac{\partial h(\vec{x},t)}{\partial t} = E(\vec{x},t) + D(\vec{x},t) \tag{4.82}$$

where $h(\vec{x},t)$ is the dune height, $E(\vec{x},t)$ the emission rate per unit area, and $D(\vec{x},t)$ the deposition rate per unit area, respectively.

The emission and the deposition estimated during a small time step δt are defined by:

$$E(\vec{x},t) = -\frac{1}{\rho_p S(\vec{x})}\frac{1}{\delta t}\sum_{n}^{N_e(\vec{x},t)} m_p^n, \tag{4.83a}$$

$$D(\vec{x},t) = \frac{1}{\rho_p S(\vec{x})}\frac{1}{\delta t}\sum_{n}^{N_d(\vec{x},t)} m_p^n, \tag{4.83b}$$

where ρ_p is the particle density, $S(\vec{x},t)$ the cell area at the position $\vec{x}$ and at the time t, $N_e(\vec{x},t)$ (or $N_d(x,t)$) the number of entrained (or deposited) particles during t and $t+\delta t$, m_p^n the mass of the n-th entrained (or deposited) particle, respectively.

In this thesis, we study the deformation in the stream-wise direction of a three-dimensional dune, which is composed by mono-diameter particles. Hence, using $m_p = \pi\rho_p d_p^3/6$, Equation (4.82) and Equation (4.83) can be rewritten as follows:

$$\frac{\partial h(x,t)}{\partial t} = E(x,t) + D(x,t) \tag{4.84a}$$

$$E(x,t) = -\frac{\pi N_e(x,t) d_p^3}{6S(x)\delta t}, \quad D(x,t) = \frac{\pi N_d(x,t) d_p^3}{6S(x)\delta t}. \tag{4.84b}$$

In numerical simulations, $S(x) = \Delta x \Delta y$ where Δx and Δy are the horizontal grid increments, and $\delta t = \Delta t$ where Δt is the big time step. $N_e(x,t)$ (or $N_d(x,t)$) is the number of entrained (or deposited) particles between t and $t+\Delta t$.

4.5.2 Avalanche model

The angle of repose is the maximal bed slope of a modelled dune. It is determined by the intrinsic characteristic of sand particles. The avalanche is a process in which sand particle slides or rolls down the steepest slope when the angle of the local slope exceeds the angle of repose. In our simulations, the one-dimensional avalanche model has been implemented and its algorithm is given by the following steps:

1. We locate the zones where the bed slope between two grid nodes exceeds the angle of repose, for example, Location A denotes the upper of those two grids nodes and B the lower one.

2. We lower the bed level at the position A to position B such that the bed slope between A and B becomes equal to the angle of repose.

3. We determine the volume of sediment, which is removed by this process (Step 2)).

4. We raise the bed level at the position A and B according to the mass conservation law.

5. We repeat Steps 1) to 4) until the local bed slope everywhere on the dune is less than the angle of repose.

The avalanche function needs many iterations to achieve a smooth dune form. This may turn out to be time consuming. In our simulation, since the big dune deforms slowly, the avalanche function is not applied at each time step during the total process of dune deformation.

4.6 Conclusions

In this chapter, physical modeling of wind erosion is described. Based on the forces acting on an individual particle, integrated wind erosion models at the scale larger than particle diameter and smaller than the saltation length, including particle entrainment models, particle transport equations, particle deposition criterion and particle-surface interaction models, are proposed or introduced in detail.

We firstly review the forces acting on an airborne particle, especially on an individual particle contacting on the wall. For a small heavy particle submerged in a boundary layer, it is found that the forces induced by the disturbance such as the added-mass force, the Basset force and the acceleration by the fluid, are negligible compared with the aerodynamic drag. For solid particles far away from the wall, the aerodynamic drag and the gravity are dominant. For particles contacting on the wall, the aerodynamic lift, the cohesion and the wall friction should be taken into account.

Based on the balance of the forces acting on an individual particle contacting on the wall, we theoretically investigate the particle incipient motions, including lifting, sliding and rolling motions, and then obtain their corresponding analytical formulas of the threshold friction velocity $u_{*,c}$. According to the dimension analysis, we propose a general formula of the critical Shield number: $Sh_c = F(d_p^+)(H(\theta) + G(d_p^-))$, where $Sh_c = u_{*,c}^2/(\gamma_\rho g d_p)$, $d_p^+ = u_{*,c} d_p/\nu$ and $d_p^- = (\gamma_\rho g)^{1/3} d_p/\nu^{2/3}$. The function $F(d_p^+)$ related to the drag and lift coefficients accounts for the aerodynamic effect, $H(\theta)$ for the bed slope effect, $G(d_p^-)$ for the inter-particle cohesion effect, respectively. Using the Shield diagram, it is shown that Sh_c of rolling and sliding agrees well with the measured critical Shield number in experiments and Sh_c of lifting seems much larger (Figure 4.15), and that the bed slope effect on the lifting mode is different from that on the rolling and sliding modes (Figure 4.16). Moreover, inspired by the works of Diplas et al. (2008), we propose an instantaneous entrainment model with not only accounting for the magnitude and duration of the forces but also for the numerical grid spacing effect. Due to the high correlation between the particle incipient motion and the sweep events observed in many experiments, the instantaneous aerodynamic forces in this model are supposed to be the product of the mean force and the Reynolds stress related to the sweep actions.

In the Lagrangian description, the governing equations of particle transport motion are given. For the airborne particles, the particle motion is determined by the aerodynamic drag and the gravity. The resulted equations are usually used to track the trajectories of solid particles in numerical simulations. In particular, the motion of the bed-load particles, which is supposed to be important for sand dune deformation, is firstly studied and the corresponding governing equations are deduced. Since the bed-load particles slide or roll on the surface, not only the aerodynamic forces and gravity but also the wall friction and the cohesion are responsible for their motions. In addition, a Lagrangian stochastic model and a two-way coupling strategy are briefly described.

For the particle-surface interaction, we describe the probabilistic properties of the rebound and splash process and introduce two empirical models of them: the model of Kok and Renno (2009) and the model of Beladjine et al. (2007). In the following simulations of wind erosion, the rebound model of Beladjine et al. (2007) and the splash model of Kok and Renno (2009) are used.

According to the conservation of mass, the governing equation of sand dune deformation is deduced. For a two-dimensional dune or a three-dimensional dune homogeneous in the spanwise direction, the temporal evolution of the dune height is determined by the local erosion rate and deposition rate. During the deformation, an avalanche model is applied to control the local bed slope, which should be smaller than the angle of repose.

Chapter 5

Numerical simulation of wind erosion

In the view of fluid dynamics, wind erosion is a two-phase complex problem with moving boundaries. It induces many topographical, environmental and agricultural issues, such as dune migration and ripple formation; air pollution and sand storms; desertification and agricultural land degradation. Recently, numerical simulations of wind erosion have been performed: Nabi et al. (2012) attempted to simulate the dynamic process of underwater ripple and dune deformation; Lopes et al. (2013) conducted the simulation of sand dune deformation using a empirical wind erosion model based on the saturated transport flux; Durán et al. (2014) employed the discrete element method to directly simulate the process of ripple formation in an atmospheric boundary layer. However, the simulations of Nabi et al. (2012) are underwater, and Lopes et al. (2013), Durán et al. (2014) used the RANS and conducted two-dimensional simulations. To our knowledge, few three-dimensional numerical simulations of wind erosion have been carried out. Hence, in this thesis, we conduct a three-dimensional numerical simulation of sand dune deformation in an atmospheric boundary layer using integrated wind erosion models for solid particles and large-eddy simulation (LES) coupled immersed boundary method (IBM) for turbulent flows with moving boundary.

In this chapter, the developed numerical solver of integrated wind erosion models coupled with IBM-ARPS is preliminarily validated by comparing the simulated results of particle transport over a fixed Gaussian dune with the experimental data of Simoëns et al. (2015) in Section 5.1. Then, in Section 5.2, we use this numerical solver to conduct the numerical simulation of wind erosion with sand dune deformation and compare the simulated dune shapes at different times with the rare experimental results of Ferreira and Fino (2012). In this wind erosion case, flow characteristics during the deformation is presented, quantitative investigation of sand dune deformation is done and the influence of some parameters in the integrated wind erosion models is studied. Moreover, a new splash model to account for the bed slope effect is proposed and used for the numerical simulation of sand dune deformation.

5.1 Validation case: particle transport over a fixed Gaussian dune

As noted in Section 3.4, the developed IBM-ARPS solver for simulating boundary layers over obstacles has been validated. In this section, integrated wind erosion models are validated through comparison between the numerically simulated and experimental results of particle transport in a turbulent boundary layer over a fixed small Gaussian dune [Simoëns et al. (2015)].

5.1.1 Experimental configuration

In the experiment of Simoëns et al. (2015), the wind-tunnel setup is illustrated in Figure 3.20. Sand particles with a density of 1000 $\mathrm{kg \cdot m^{-3}}$ are placed in a box 20 cm $\times$ 10 cm located at the beginning of the fluid domain. The particle diameter has a mean value of 200 μm and ranges from 170 μm to 250 μm. In order to obtain the static rough-wall conditions similar to the realistic sand bed, particles of same characteristic are uniformly and closely spread and glued on the ground and over the Gaussian dune. Hence, the process of particle rebound occurs in a natural way but the splash entrainment caused by the rebound is artificially eliminated.

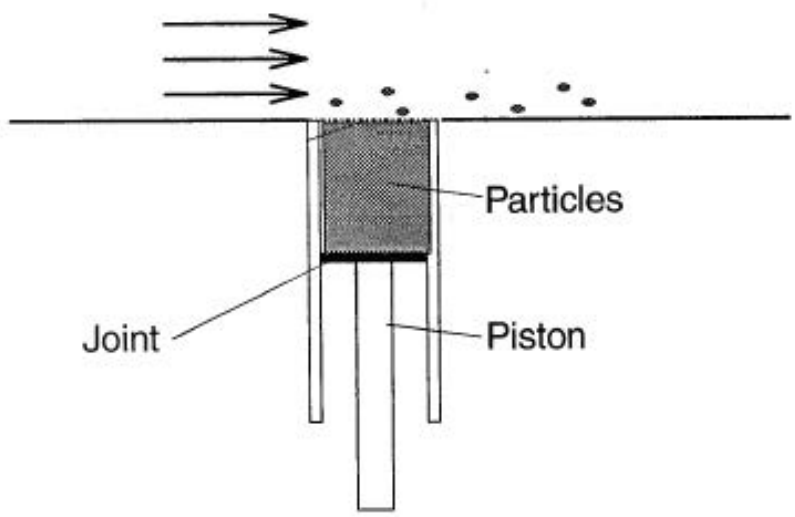


Figure 5.1 Sketch of sandbox-type particle-feeding device. Figure adapted from Taniere et al. (1997).

Figure 5.1 schematically shows the particle-feeding device used in this experiment. This device is installed at the upstream position of the wind tunnel flow. Powered by an upward-moving piston, solid particles are entrained into the boundary layer flow. Particles supplied

from the sandbox device generate an relatively natural inlet condition for particle transport in a boundary layer. When the state of particle transport becomes equilibrium, the particle concentration and velocity field are measured by digital image treatment.

The characteristic parameters of solid particles are shown in Table 5.1. The Shields number $Sh = 0.14$ reveals that saltation and suspension are two main modes of the particle motion. As the particle relaxation time scale τ_p satisfies $\tau_\eta \ll \tau_p < \tau_L$, the particle motion is hardly sensitive to the turbulent dissipative scale structures, but can be easily influenced by the energetic scale structures. The gravity parameter $\gamma_g > 1$ indicates that the modified saltation mode is dominant.

$\rho_p(\mathrm{kg}\cdot\mathrm{m}^{-3})$	$d_p(\mu\mathrm{m})$	$g(\mathrm{m}^2\cdot\mathrm{s}^{-1})$	$\tau_p(\mathrm{s})$	Sh	St_L	St_η	γ_g
1000	200	9.81	0.12	0.14	0.75	35.01	2.56

Table 5.1 Parameters of solid particles.

5.1.2 Numerical configuration

In numerical implementation, a sand box is installed at the beginning of the computational domain as shown in Figure 5.2. The box size takes the same value as those in the experiment

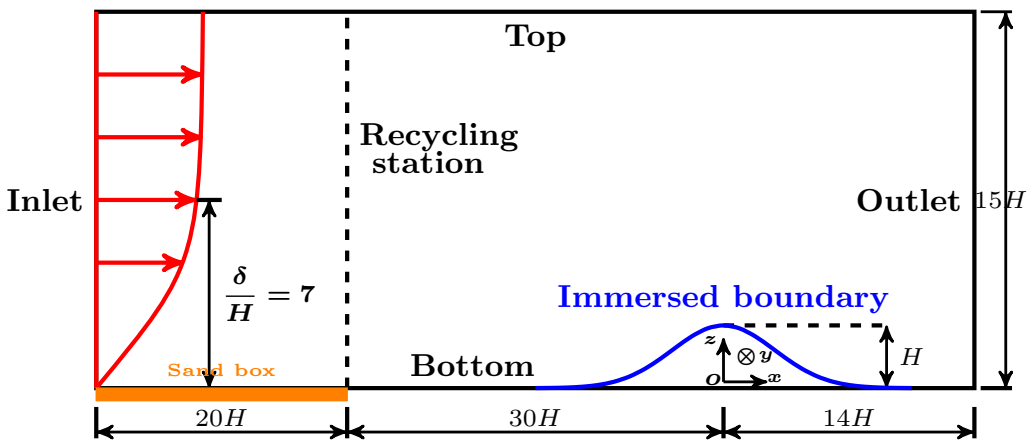


Figure 5.2 Sketch of numerical configuration of particle transport over a Gaussian dune.

and solid particles are of a uniform diameter $d_p = 200\ \mu$m. In numerical simulations, the particle incipient motions at the sand box are determined by the aerodynamic entrainment model implemented in the ARPS code. The initial particle velocity depends on the momentum obtained from the nearby fluid flow. This implementation is compatible with the experimental set-up. Particles supplied by the sand box forms an exponential profile of particle concentration, which agrees well with the experimental result. In addition, the

physical and numerical parameters are the same as given in Table 3.1 and Table 3.2, and more details of numerical set-up are found in Subsection 3.4.1.

In practice, two numerical cases are studied as illustrated in Table 5.2. Case A was carried out using the ARPS with a terrain-following grid [Huang (2015)] and Case B is performed using the developed IBM-ARPS solver with a Cartesian grid. In particular, the sand dune surface in Case B is considered as the immersed boundary, on which no-slip boundary condition is imposed by introducing the direct forcing method. In both cases, an instantaneous entrainment model is applied to initialize the particle movement while particle/soil interaction is taken into account by the rebound model. Since the bed surface is static, the splash model is useless and then is not used in these two simulations. The details of these models are introduced in Chapter 4. It should be noted that the Lagrangian stochastic model and two-way coupling are not used in these simulations.

Case	ARPS or IBM-ARPS	Particle entrainment model	Rebound model	Splash model
Case A [Huang (2015)]	ARPS	Yes	Yes	Non
Case B	IBM-ARPS	Yes	Yes	Non

Table 5.2 Numerical cases of particle transport in a TBL over a small Gaussian dune.

5.1.3 Results and analysis

In this subsection, the results of particle transport over a fixed Gaussian dune obtained by LES coupled with integrated wind erosion models are shown and analyzed. Concretely, visualization of particle transport in the carried flow is done by showing particle trajectories and quantitative description of particle transport is conducted by plotting the particle concentration and velocity profiles. In order to validate the integrated wind erosion models, comparisons of the particle concentration profiles between simulated results and experimental data are carried out.

Particle trajectories: Figure 5.3 shows several sand particle trajectories around the Gaussian hill. Sand particles are aerodynamically entrained at the location of the sandbox. Once lifted-off, sand particles are transported by the carried flow. By the action of gravity they might impact the wall and rebound. According to the rebound model proposed by Beladjine et al. (2007), particles that impact the dune on the upstream side are highly dispersed in the turbulent flow.

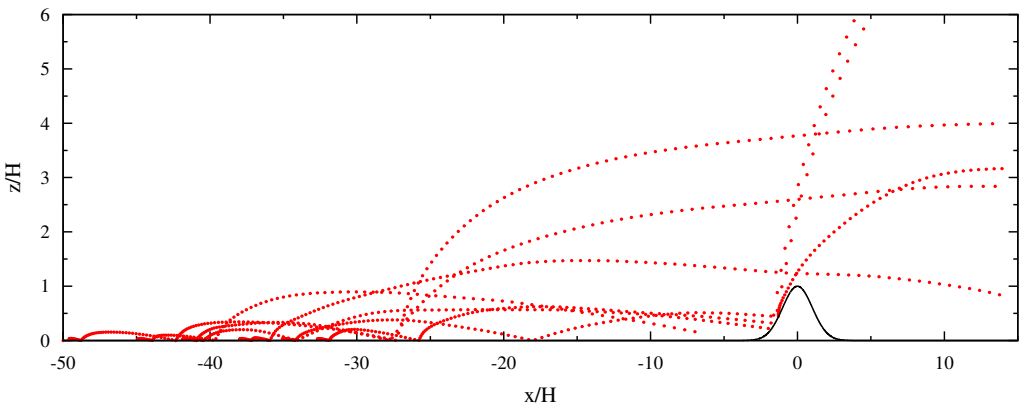


Figure 5.3 Particle trajectories over a fixed Gaussian dune. Results obtained by the IBM-ARPS simulation coupled with integrated wind erosion models (Case B).

Particle velocity profiles over a Gaussian dune: We study the kinetic aspects of particle transport starting from the particle velocity profiles. Here, the particle velocity is defined as the average value of particle velocities within a unit volume V in an Eulerian description. In our case, the motion of solid particles is homogeneous in the spanwise direction. The unit volume is given by $\mathrm{V} = \Delta x_p L_y \Delta z_p$ at a given position (x,z). In practice, we set $\Delta x_p = 0.1H$ and $\Delta z_p = 0.1H$ and L_y is equal to the spanwise length of the computational domain. Thus, the formula of Eulerian particle velocity is given by:

$$\langle U_p \rangle (x,z) = \frac{1}{\Delta x_p L_y \Delta_p} \sum_{\mathrm{V}} u_p, \tag{5.1}$$

where u_p is the Lagrangian particle velocity within the unit volume V.

The comparison between the particle and fluid longitudinal velocity is shown on Figure 5.4. The velocity is normalized by the external one U_e. Due to the presence of zero concentration as shown in Figure 5.5, an apparent discontinuity exists on the particle velocity profile after the dune. On the upstream side, the fluid and particle velocities are close. Over the dune, the particle velocity is obviously smaller than the fluid. Due to the presence of the dune, the fluid velocity increases more rapidly than the particle's because of the particle response time to fluid solicitations (τ_p).

Particle concentration profiles over a Gaussian dune: The particle concentration $\langle C \rangle$ is defined as the mean volume fraction of all the solid particles within a unit volume $\mathrm{V} = \Delta x_p L_y \Delta z_p$. It is written as:

$$\langle C \rangle (x,z) = \frac{1}{\Delta x_p L_y \Delta_p} \sum_{\mathrm{V}} V_p, \tag{5.2}$$

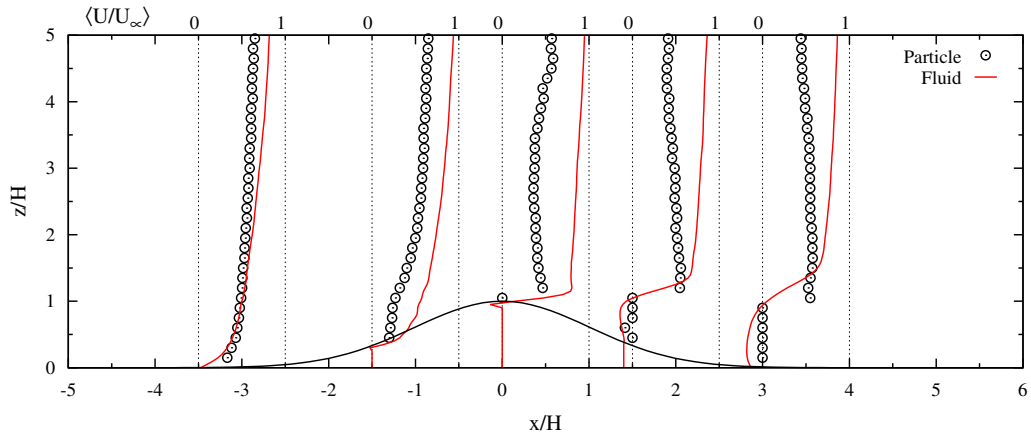


Figure 5.4 Comparison between particle longitudinal velocity (points) and fluid velocity (lines) over a Gaussian hill. Results obtained by the IBM-ARPS simulation coupled with integrated wind erosion models (Case B).

where V_p is the volume of individual particles within V.

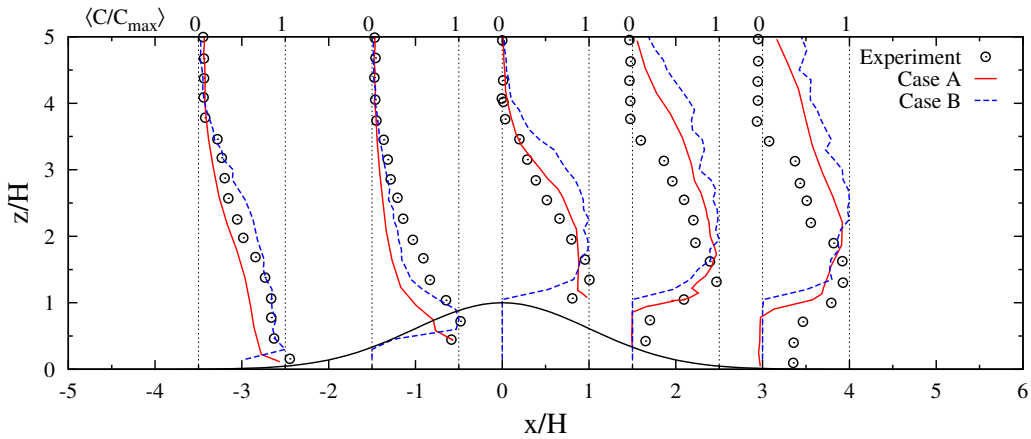


Figure 5.5 Particle concentration profiles over a Gaussian dune. Points: experiments of Simoëns et al. (2015). Dashed lines: ARPS simulation coupled with integrated wind erosion models (Case A). Solid lines: IBM-ARPS simulation coupled with integrated wind erosion models (Case B).

The profiles of particle concentration over a fixed Gaussian dune are shown in Figure 5.5. Each profile is normalized by its maximal concentration. High values of the concentration are obtained close to the surface and in the mixing layer at the interface of the recirculation region and the outer flow. Behind the dune, in the recirculation region, particle concentration is low and drops to zero. On the windward side, a good agreement is obtained between simulation results and experimental data. On the lee side, the locations of the concentration

peak are well predicted by both simulations (Case A and Case B), even if the simulated concentration profiles differ from the experimental ones.

In conclusion, the integrated wind erosion models have been implemented into the IBM-ARPS code. Globally, a good agreement between the simulated results and the experimental data preliminarily demonstrates the ability of the IBM-ARPS solver coupled with integrated wind erosion models.

5.2 Wind erosion case: particle transport over a deformable sinusoidal dune

As the integrated wind erosion models are preliminarily validated in the previous section 5.1, an attempt of simulating wind erosion with sand dune deformation is carried out using these models for particle motions coupled with IBM-ARPS for calculating turbulent flows with deformable immersed interface. Based on the simulated results, quantitative investigation of dune deformation is conducted and the influence of some parameters in integrated wind erosion models is studied.

5.2.1 Estimation of the duration of sand dune deformation

Supposing the saturated saltation flux $\vec{q}_s$ ($\mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}$), from a simple flux-divergence model, the temporal evolution of the dune height h is governed by:

$$\frac{\partial}{\partial t} h(\vec{x},t) = -\frac{1}{\rho_p} \nabla \cdot \vec{q}_s(\vec{x},t). \tag{5.3}$$

where ρ_p is the average density of sand particles. Equation (5.3) indicates that the local divergence of particle transport flux is responsible for the evolution of local dune elevation.

Considering a two-dimensional dune or a three-dimensional dune homogeneous in the spanwise direction in a boundary layer, the dune height and saltation flux are thus simplified to functions of the streamwise coordinate x, *i.e.*, $h = h(x,t)$ and $q_s = q_s(x,t)$. Integrating Equation (5.3) from x_1 to x_2, we obtain:

$$\frac{\partial}{\partial t} \int_{x_1}^{x_2} h(x,t) dx = -\frac{1}{\rho_p} \int_{x_1}^{x_2} \frac{\partial q_s(x,t)}{\partial x} dx, \tag{5.4}$$

For the simplification, assuming that the saltation flux is nearly unchanged during the dune deformation, for example, q_s takes the initial value and only depends on the position, *i.e.*

$q_s(x,t) = q_{s,0}(x)$, then Equation (5.4) becomes:

$$\frac{\partial}{\partial t}\int_{x_1}^{x_2} h(x,t)dx = -\frac{1}{\rho_p}\int_{x_1}^{x_2}\frac{\partial q_{s,0}(x)}{\partial x}dx = -\frac{1}{\rho_p}(q_{s,0}(x_2) - q_{s,0}(x_1)). \tag{5.5}$$

As $q_{s,0}$ is a increasing function of the wall shear stress, *i.e.*, $q_s \sim (u_* - u_{*,c})^3$, where u_* and $u_{*,c}$ is the flow and critical friction velocity. As shown in Figure 3.26, the friction velocity increases on the windward side of the dune, takes the maximal value near the crest. Assuming that the interval $[x_1, x_2]$ signifies the windward side of the sand dune, Equation (5.6) is then approximately equal to:

$$\frac{\partial}{\partial t}\frac{1}{2}S(t) \approx -\frac{1}{\rho_p}q_{s,max}, \tag{5.6}$$

where $S(t)$ denotes the dune surface and $q_{s,max}$ is the maximal saturated flux. Here, we note $q_{s,0}(x_2) = q_{s,max}$ with the maximal friction velocity u_* at the position x_2 and $q_{s,0}(x_1) = 0$ since u_* at the position x_1 is smaller than the critical one. Supposing that $q_{s,max}$ is a constant during the sand deformation, from Equation (5.6), it is known that the dune surface is decreasing at a constant rate $q_{s,max}/\rho_p$. Hence, the duration T for a sand dune erosion is approximated by:

$$T \approx \frac{\rho_p S_{dune}}{2q_{s,max}}. \tag{5.7}$$

For the wind erosion in a boundary layer over a large sinusoidal dune, we have $u_* = 0.42\ \mathrm{m\cdot s^{-1}}$, $d_p = 500\ \mu\mathrm{m}$, $\rho_p = 2650\ \mathrm{kg\cdot m^{-3}}$, $\rho = 1.25\ \mathrm{kg\cdot m^{-3}}$, $H = 0.06$ m. The saturated flux q_s is estimated by the formula of Bagnold (1941), *i.e.*, $q_s = c_0\sqrt{d_p/D}\rho u_*^3/g$, where $c_0 = 1.5$, $D = 250\ \mu\mathrm{m}$, $g = 9.8\ \mathrm{m\cdot s^{-2}}$. Using Equation (5.7), the deformation duration of a large sinusoidal dune is about:

$$T \approx \frac{2650\times 1.5\times 6\times 10^{-2}\times 6\times 10^{-2}}{2\times 1.5\times\sqrt{500/250}\times 1.25\times(0.05\times 11.2)^3/9.8} \approx \frac{6.36}{0.05} \approx 127.2\ \mathrm{sec}. \tag{5.8}$$

Hence, the duration is about 2 min, which is consistent with the experimental result of Ferreira and Fino (2012).

5.2.2 Experimental set-up

The wind-tunnel experiments were carried out by Ferreira and Fino (2012). This wind tunnel is long of 5 m and has a cross-section of 2 m $\times$ 2 m. The sand pile with a height of $H = 0.06$ m is placed $36.7H$ from the input nozzle, and perpendicular to the streamwise

direction as shown in Figure 5.6. For the fully developed boundary layer generated in the

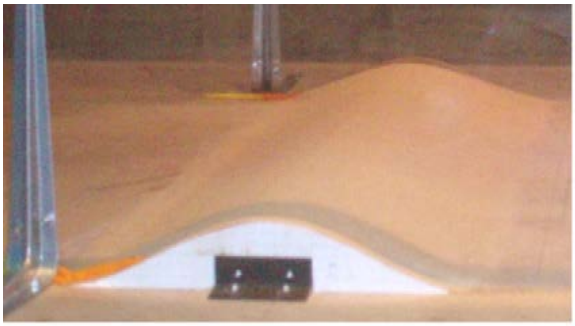


Figure 5.6 View of a single sand pile in the wind tunnel. Taken from Ferreira and Fino (2012)

wind tunnel, the measured external velocity and layer thickness are $U_e = 9.1\ \mathrm{m \cdot s^{-1}}$ and $\delta = 0.1$ m. The Reynold number is 3.6×10^4.

The detailed information of solid particles is presented in Table 5.3. The Shields number $Sh = 0.02$ shows a low transport capacity of sand particles by the turbulent flow. This explains that the dune deformation caused by the particle transport takes a long time. As the particle relaxation time scale τ_p satisfies $\tau_p \gg \tau_\eta$ and $\tau_p > \tau_L$, the particle motion is not only hardly responsive to the turbulent dissipative scale structure, but also can not be influenced by the energetic scale structure. The very large gravity parameter $\gamma \gg 1$ indicates that the modified saltation mode dominates and the creep mode might plays an important role on the dune deformation.

$\rho_p(\mathrm{kg \cdot m^{-3}})$	$d_p(\mu\mathrm{m})$	$g(\mathrm{m^2 \cdot s^{-2}})$	$\tau_p(\mathrm{s})$	Sh	St_L	St_η	γ_g
2650	500	9.81	1.96	0.02	8.24	436.21	45.78

Table 5.3 Parameters of solid particles.

5.2.3 Numerical configuration

The computational domain is illustrated in Figure 5.7. The deformable dune is situated at the origin and is considered as the immersed boundary. In the wind erosion case, sand particles are entrained and deposited according to the corresponding criterion proposed by the integrated wind erosion models. The local particle erosion rate and deposition rate are then evaluated by calculating the mass loss using the countered entrained particles and the mass gain using the deposited particles in a grid cell and in a time step. Therefore, applying Equation (4.84) results in the instantaneous dune deformation. In particular, an avalanche model is used to control the local bed slope, which should be smaller than the angle of repose.

In this simulation, the physical and numerical parameters are given by Table 3.5 and Table

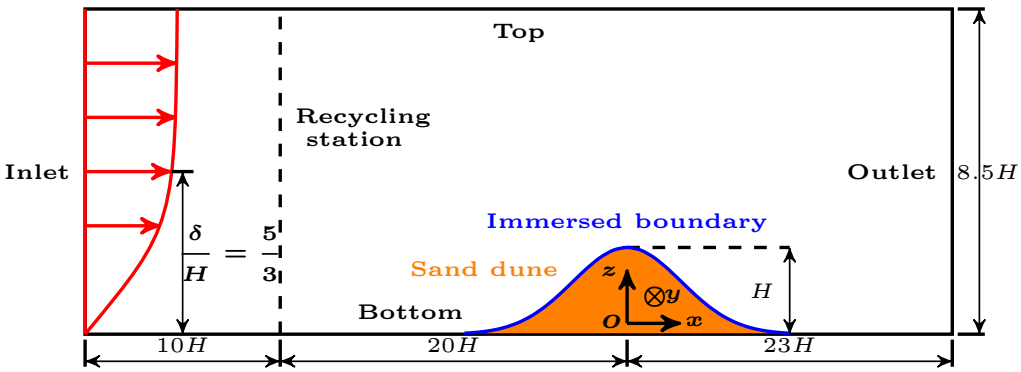


Figure 5.7 Sketch of numerical configuration of wind erosion.

3.6, and more details of numerical set-up are found in Subsection 3.4.3. The characteristics of solid particles are the same with the experiment's as shown in Table 5.3.

Six numerical simulation cases are carried out for the wind erosion with sand dune deformation. The turbulent boundary layer over a deformable dune is simulated by the developed IBM-ARPS solver, and the particle motion is determined by the resolved flow and integrated wind erosion models. Concretely, the particle incipient motions are modeled by an instantaneous lifting or rolling-sliding entrainment model; then the entrained particles are transported in the carried flow using a Lagrangian tracking approach; lastly, due to the gravity effect, solid particles fall back to the dune surface and make a collision with particles on the ground. After the collision, particles either rebound and continue their motions in the flow or roll or slide on the surface until they deposit on the dune, according to the criterion in the rebound model. In addition, owing to the collision, some new particles might be ejected in the flow, which is simulated by a probabilistic splash model. In Case A′, the coefficient in the particle entrainment model is assigned by $\gamma_{eff} = 0.6$, which is obtained by fitting the simulated dune shape with the experimental results at $t = 2$ min; The coefficient in the equation of bed load particle motion is considered as the effective dynamic friction coefficient of a rough surface composed by sand particles, which is measured as $\mu_{eff} \approx \tan(8.5^\circ)$ in Quartier et al. (2000); The rebound model proposed by Beladjine et al. (2007) and the splash model of Kok and Renno (2009) are used. Cases B′ and C′ are conducted to investigate the influence of the coefficient μ_{eff} on the sand dune deformation. Cases D′ and E′ are used to study the effect of the splash entrainment on the wind erosion. Case F′ is an attempt, in which an improved splash model with accounting for the bed slope effect is applied. For numerical simulations, it should be noted that the computing cost is very huge since LES is used and the simulation is three-dimensional. For example, it should take about one month for Case A′ to achieve 6 min of sand dune deformation, even if 128 CPUs are used using

MPI parallelization as shown in Table 5.5. It should be noted that the Lagrangian stochastic model and two-way coupling are not used in these simulations.

Case	Particle entrainment model	Bed-load particle motion	Rebound model	Splash model
Case A′	$\gamma_{eff} = 0.6$	$\mu_{eff} = \tan(8.5°)$	Beladjine et al. (2007)	Kok and Renno (2009) with $a = 0.05$
Case B′	$\gamma_{eff} = 0.6$	$\mu_{eff} = \tan(3.5°)$	Beladjine et al. (2007)	Kok and Renno (2009) with $a = 0.05$
Case C′	$\gamma_{eff} = 0.6$	$\mu_{eff} = \tan(13.5°)$	Beladjine et al. (2007)	Kok and Renno (2009) with $a = 0.05$
Case D′	$\gamma_{eff} = 0.6$	$\mu_{eff} = \tan(8.5°)$	Beladjine et al. (2007)	Kok and Renno (2009) with $a = 0.03$
Case E′	$\gamma_{eff} = 0.6$	$\mu_{eff} = \tan(8.5°)$	Beladjine et al. (2007)	Kok and Renno (2009) with $a = 0.10$
Case F′	$\gamma_{eff} = 0.6$	$\mu_{eff} = \tan(8.5°)$	Beladjine et al. (2007)	Kok and Renno (2009) with bed slope effect

Table 5.4 Numerical cases of particle transport in a TBL over a small Gaussian dune.

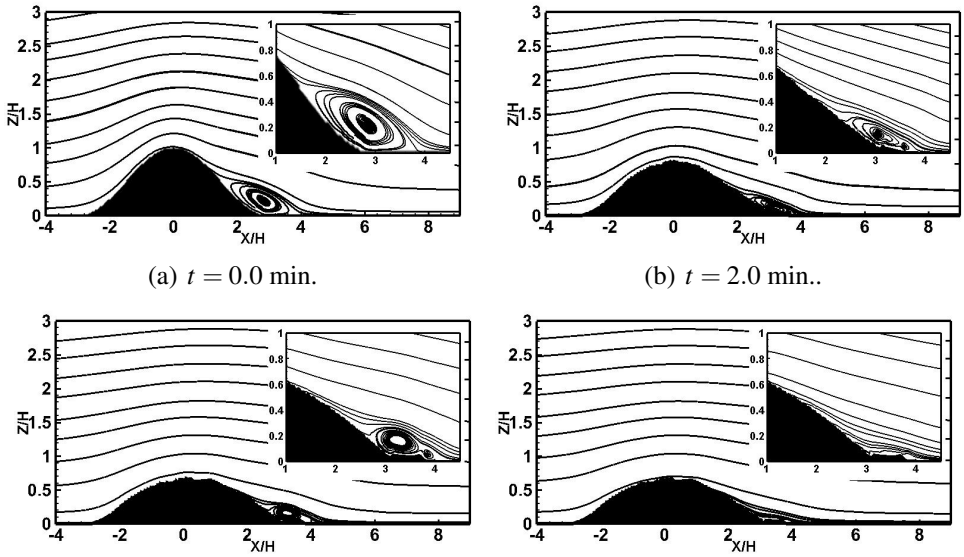
Case	Grid	Δt(s)	Duration (min)	N_p	CPU-H
Case A′	$1063 \times 63 \times 100$	5×10^{-5}	6.0	128	580

Table 5.5 Estimation of computational time for the numerical simulation of Case A′. Δt is the large time step. N_p denotes the number of processors. CPU-H represents CPU hours necessary for 6 min of Case A′.

5.2.4 Flow characteristics

In this subsection, the flow characteristics over a deformable dune is investigated by plotting the recirculation zone, mean longitudinal velocity profiles, Reynolds stress profiles and the friction velocity at different times.

Figure 5.8 shows the recirculation zones obtained by Case A′ at $t = 0$ min, $t = 2$ min, $t = 4$ min and $t = 6$ min. In Figure 5.8, the coordinates are scaled by the initial dune height



(a) $t = 0.0$ min. (b) $t = 2.0$ min.. (c) $t = 4.0$ min.. (d) $t = 6.0$ min..

Figure 5.8 Mean streamlines over a deformable dune and recirculation zones on the lee side. Results obtained by Case A′.

H and the origin is set at the dune center. Table 5.6 presents the values of key parameters characterizing the form of recirculation zone, i.e., x_{sep}, x_{reat}, L_{AB}, L_{CD} and L_{AB}/L_{CD}. As the sand dune deforms, the dune height gradually decreases and the recirculation zone is reduced accordingly. It is observed that the recirculation zone disappears behind the dune at $t = 6.0$ min as shown in Figure 5.8(d). Concretely, the separation position increases from $x_{sep} = 1.67H$ at $t = 0.0$ min to $x_{sep} = 3.28H$ at $t = 6.0$ min while the reattachment position decreases from $x_{reat} = 3.83H$ at $t = 0.0$ min to $x_{reat} = 3.28H$ at $t = 6.0$ min. In particular, the reattachment point moves forward a distance at $t = 4.0$ min as $x_{reat} = 3.92H$ at $t = 4.0$ min is larger than $x_{reat} = 3.63H$ at $t = 2.0$ min. This abnormal process might be caused by the deposition on the lee side, which leads to an increase in the dune area near the recirculation zone. Moreover, both two lengths L_{AB} and L_{CD} also decrease as time goes by.

Time	x_{sep}/H	x_{reat}/H	L_{AB}/H	L_{CD}/H	L_{AB}/L_{CD}
$t = 0.0$ min	1.67	3.83	0.44	2.21	0.199
$t = 2.0$ min	2.47	3.63	0.24	1.18	0.203
$t = 4.0$ min	2.77	3.92	0.20	1.17	0.171
$t = 6.0$ min	3.28	3.28	–	–	–

Table 5.6 Characterization of the recirculation zone behind a deformable dune. Results obtained by Case A′.

The mean longitudinal velocities over a deformable dune at several different streamwise stations and at $t = 0$ min, $t = 2$ min, $t = 4$ min, $t = 6$ min, are shown in Figure 5.9. The mean streamwise velocities are normalized by the external one and located at the corresponding stations, *i.e.*, $x = -1.5H$, $x = 0.0H$, $x = 1.5H$, $x = 3.0H$, $x = 4.0H$, $x = 6.0H$ and $x = 7.5H$. The coordinates are scaled by the dune height H and the origin is set at the dune center. It is shown that the mean longitudinal velocity over the dune decreases as the sand dune deforms, especially for the velocity at the dune crest from $t = 0.0$ min to $t = 2.0$ min. For the stations far away form the dune, there is almost no difference between the mean streamwise velocity profiles. This indicates that the dune deformation only affects nearby flows but has little influence on the external flows.

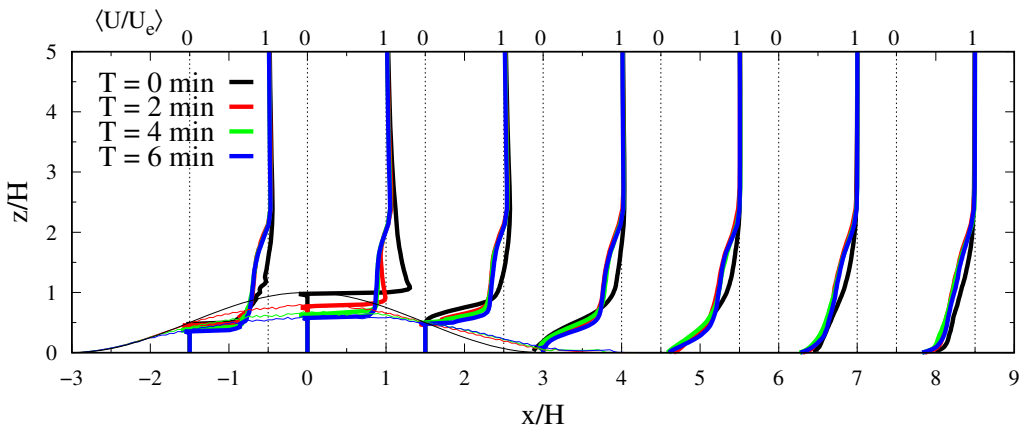


Figure 5.9 Mean longitudinal velocity profiles over a deformable dune. Results obtained by Case A′.

In Figure 5.10, we present profiles of mean Reynolds stress $\langle u'w' \rangle$ over a deformable dune at six different stremwise stations and at $t = 0$ min, $t = 2$ min, $t = 4$ min, $t = 6$ min. Here, u' denotes the longitudinal fluctuation velocity and w' the vertical fluctuation. The Reynolds stress is scaled by the square of the external velocity. It is observed that the Reynold stress behind the dune decreases as the sand dune deforms: a sharp decrease from $t = 0$ min to $t = 2$ min and a slight decline after $t = 2$ min. In particular, even when the recirculation zone disappears at $t = 6$ min, there is also a slight increase on $\langle u'w' \rangle$ behind the dune.

Figure 5.11 gives the friction velocity u_* over a deformable dune at $t = 0$ min, $t = 2$ min, $t = 4$ min, $t = 6$ min. The friction velocity is defined by $u_* = \sqrt{\tau_w/\rho}$, where τ_w is wall shear stress and ρ is the fluid velocity, and it is normalized by the initial value. Within the integrated wind erosion models, the friction velocity is used to calculate the aerodynamic lift and drag forces, which are responsible for the particle incipient motion. It is shown that the friction velocity is reduced at the dune crest and enhanced both on the windward side

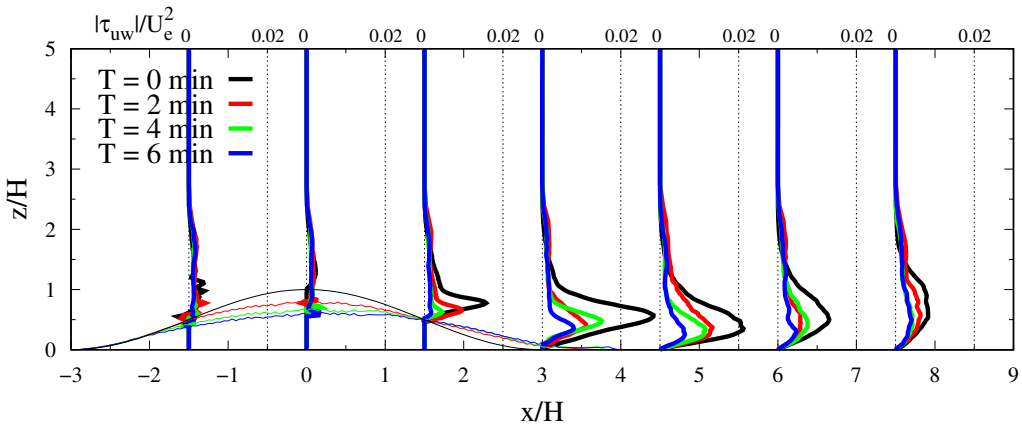


Figure 5.10 Reynold stress $\tau_{uw} = \langle u'w' \rangle$ profiles over a deformable dune. Results obtained by Case A′.

and on the lee side as the sand dune deforms. As the saturated transport flux is commonly a cubic function of local friction velocity, *i.e.*, $q_s \sim u_*^3$, the reduced flux at the crest may be much larger than the increased flux on both sides. Thus, this might lead to a decrease in the erosion rate as time goes by. In addition, it is obviously found that the region of small friction velocities is gradually reduced from $t = 0$ min to $t = 6$ min. This supports the previous point that the recirculation zone gradually decreases during the sand dune deformation. We note that some oscillations are observed on the friction velocity curves over the dune. These are three possible reasons for them. Firstly, the discrete immersed boundary method constructs a sharp representation of the immersed dune surface, which indeed induces some small oscillations on u_* as shown in Figure 3.26. Secondly, as the sand dune deforms, there appear some discontinuities on the dune shape, which might cause some changes on the near-wall flow and thus on the friction velocity. Thirdly, since the computing cost of Case A′ is huge, only one numerical representation is used to evaluate the mean wall shear stress and thus the friction velocity. Hence, lack of rich numerical data might induce some statistic fluctuations.

5.2.5 Sand dune deformation

In this subsection, quantitative investigation of simulated results and experimental data is performed by comparing the sand dune shape after the deformation obtained by both approaches. To evaluate the performance of the proposed integrated wind erosion model coupled with the IBM-ARPS, the mean and root mean square of vertical dune height deviated

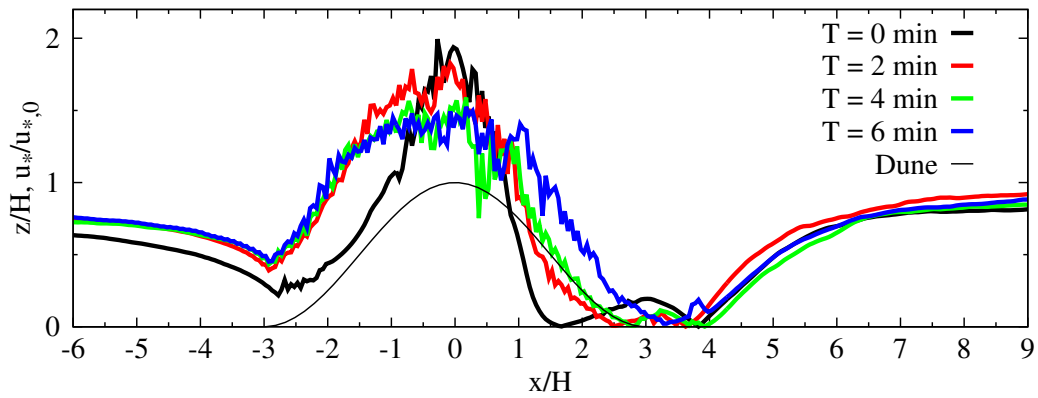


Figure 5.11 Friction velocity at $t = 0$ min, $t = 2$ min, $t = 4$ min, $t = 6$ min. Results obtained by Case A′.

to the experimental results are used. They read:

$$\mathrm{e}_h(t) = \frac{\sum_i h_{i,num}(t) - h_{i,exp}(t)}{\sum_i h_{i,exp}(t)}, \tag{5.9a}$$

$$\sigma_h(t) = \frac{\sqrt{\sum_i (h_{i,num}(t) - h_{i,exp}(t))^2}}{\sum_i h_{i,exp}(t)}, \tag{5.9b}$$

where $h_{i,num}(t)$ is the dune height at the grid index i and at the time t obtained by the numerical simulation and $h_{i,num}(t)$ obtained by the experiment. Equation (5.9a) indicates the difference of simulated area (mass) loss to the experiment during the deformation and Equation (5.9b) is an indicator describing the global derivation of simulated dune shape to the experiment's.

Figure 5.12 shows the numerical results of sand dune deformation at $t = 2$ min using the integrated wind erosion models coupled with IBM-ARPS. The simulated dune shape is compared with the experimental data of Ferreira and Fino (2012). It is observed that a perfect agreement between them is achieved. Both the mean and the root mean square are very small, *i.e.*, $\mathrm{e}_h = -2.98\%$ and $\sigma_h = 0.71\%$ as shown in Table 5.7. In Figure 5.13, the accumulated erosion and deposition rates from $t = 0$ min to $t = 2$ min are presented. Note that the accumulated erosion rate is evaluated by calculating the total area of eroded (or entrained) particles between $t = 0$ min and $t = 2$ min and the accumulated deposition rate by computing the total area of deposited particles. It is found that their summation takes a negative value on the windward side and at the crest, and a positive value on the lee side near the dune foot, precisely around the recirculation zone. This means erosion on the windward side, where particles take off since the local friction velocity exceeds the critical value, and

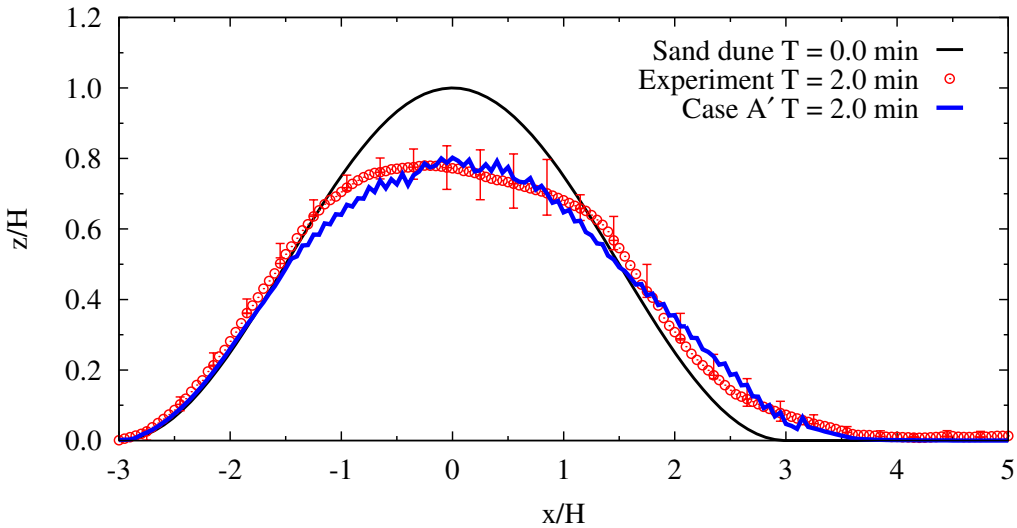


Figure 5.12 Sand dune shape at $t = 2$ min. Red lines: experiment of Ferreira and Fino (2012). Blue lines and points: numerical simulation (Case A′).

deposition on the lee side, where transported particles are trapped into the recirculation zone and then deposited on the surface since the reversed fluid velocities slow down the particle motions. This observation is consistent with the previous conclusion obtained by a priori study of wind erosion over a fixed Gaussian dune [Huang et al. (2018)].

Figure 5.14 gives the different contributions to erosion between $t = 0$ min and $t = 2$ min: the lifting mode, the rolling-sliding mode and the splash mode. Here, we note that the accumulated erosion rate of the lifting mode is defined by the area loss caused by the lifting entrainment between $t = 0$ min and $t = 2$ min, which is equal to the product between the area of an individual particle and the number of lifting particles during this duration. The rolling and splash erosion rates are calculated in the same way. It is observed that the lifting incipient motion occurs on the windward side, the splash entrainment mainly on the lee side and the rolling-sliding incipient motion on both sides. This indicates that the aerodynamic entrainment, including both lifting entrainment and rolling-sliding entrainment, is responsible for the windward side deformation and the splash and rolling-sliding entrainment are important for lee side deformation. It is noticed that non-negligible oscillations are found on the erosion and deposition curves in Figure 5.13 and Figure 5.14. There are two possible causes: firstly, this may be caused by the application of a probabilistic splash model in the simulation. In these oscillated regions, the splash process occurs frequently, which results in many ejected particles by the particle-surface collisions and thus in a very large erosion rate. Most of them get a small initial velocity and then deposit immediately in the same grid

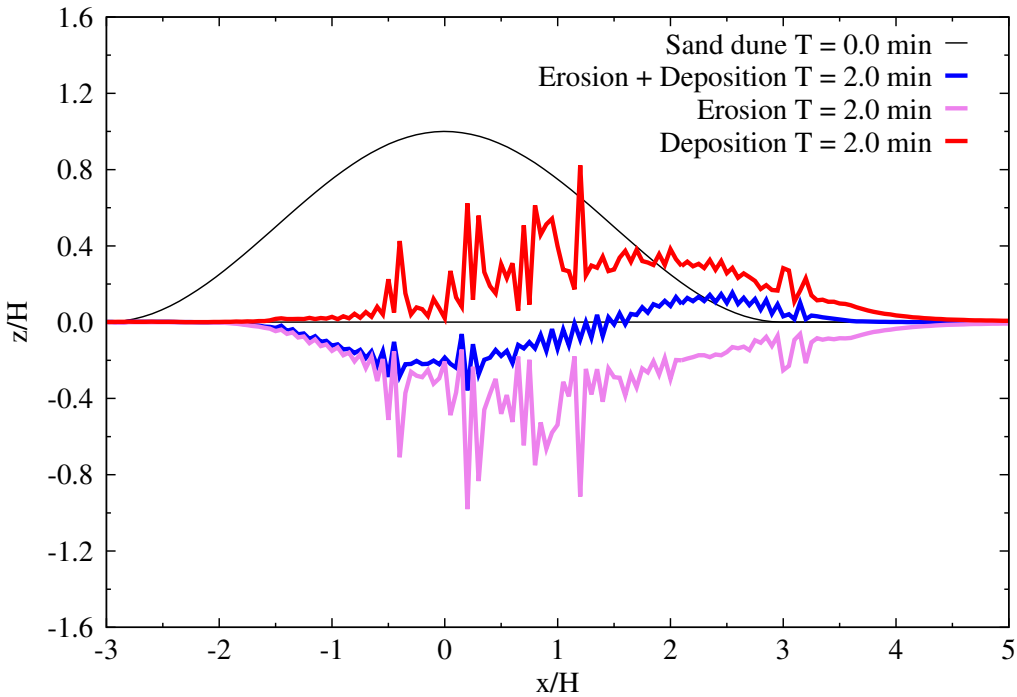


Figure 5.13 Erosion and deposition. Erosion (violet lines and points): total area of particles removed from the sand dune between $t = 0$ min and $t = 2$ min. Deposition (red lines and points): total area of particles deposited on the sand dune between $t = 0$ min and $t = 2$ min.

cell. This explains why the oscillations on the rate curves are nearly situated at the same position. Moreover, due to the limitation of computing ability, the erosion and deposition rates are calculated using only one sample of Case A$'$. Hence, these oscillations might be a representation of the randomness of the splash process. Secondly, the discontinuities on the friction velocity over the dune (shown in Figure 5.11), might also cause some oscillations on the lifting and rolling-sliding erosion rates.

Figure 5.15 and Figure 5.16 show the numerical simulated results of sand dune deformation at $t = 4$ min and $t = 6$ min, respectively. In Figure 5.15(a) and Figure 5.16(a), the dune shapes simulated by Case A$'$ are compared with the experimental data. It is found that an acceptable agreement between them at $t = 4$ min is reached as the mean and root mean square errors are small, *i.e.*, $e_h = 1.50\%$ and $\sigma_h = 1.85\%$. Unfortunately, as shown in Figure 5.16(a), the simulated dune shape at $t = 6$ min differs from the experimental one on the lee side. The mean and root mean square of the dune height are very large, *i.e.*, $e_h = 13.40\%$ and $\sigma_h = 3.91\%$ given in Table 5.7. Comparing Figure 5.15(a) with Figure 5.16(a), we find that there is almost no change in the shape of sand dune from $x = 2H$ to $x = 5H$ between $t = 4$ min and $t = 6$ min. This indicates that the pure deposition rate on the lee side becomes

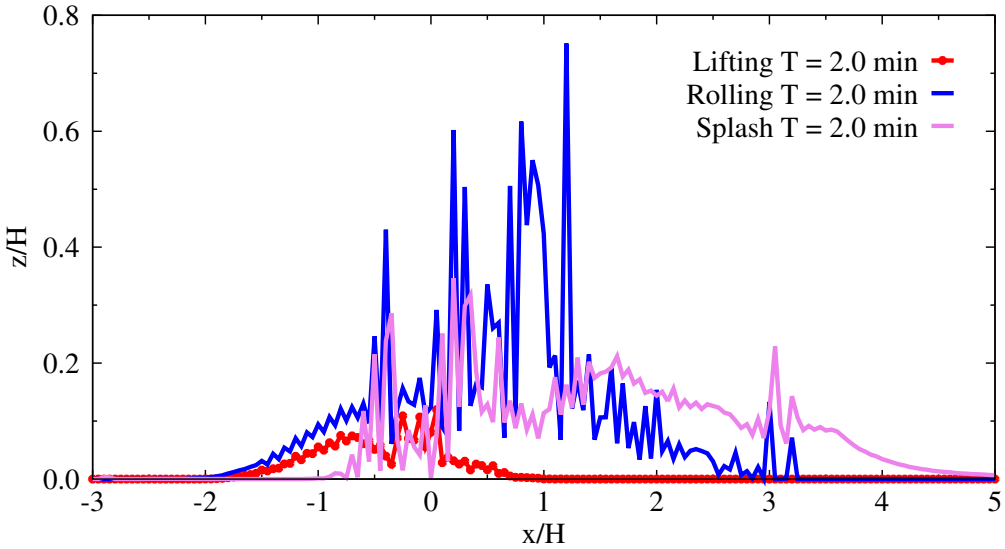


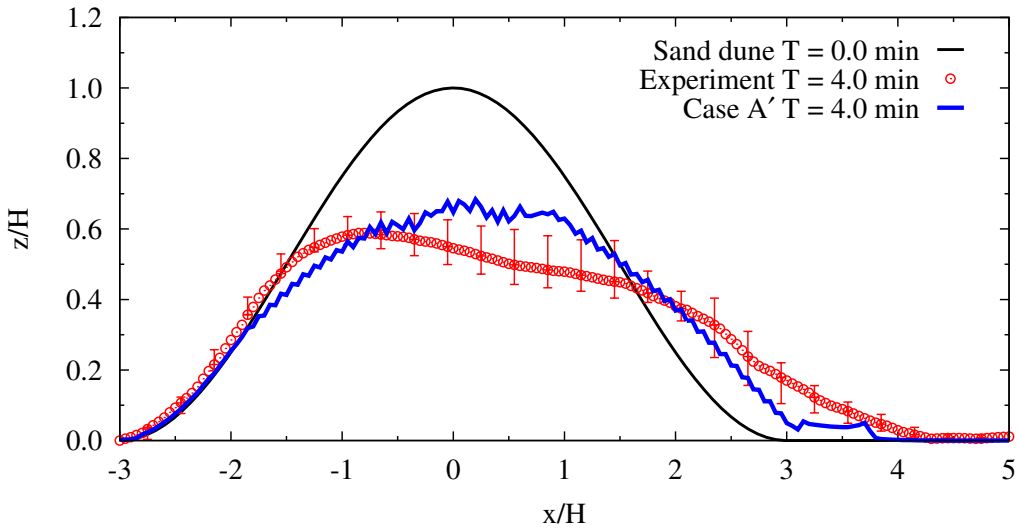
Figure 5.14 Contributions to erosion between $t = 0$ min and $t = 2$min: lifting, rolling and splash. Results obtained by Case A$'$.

negligible when the recirculation zone behind the dune is very small or even disappears. Hence, the recirculation zone plays an important role for the deposition on the lee side. Moreover, it is also observed that erosion occurs on the windward side and deposition on the lee side as shown in Figure 5.15(b) and Figure 5.16(b).

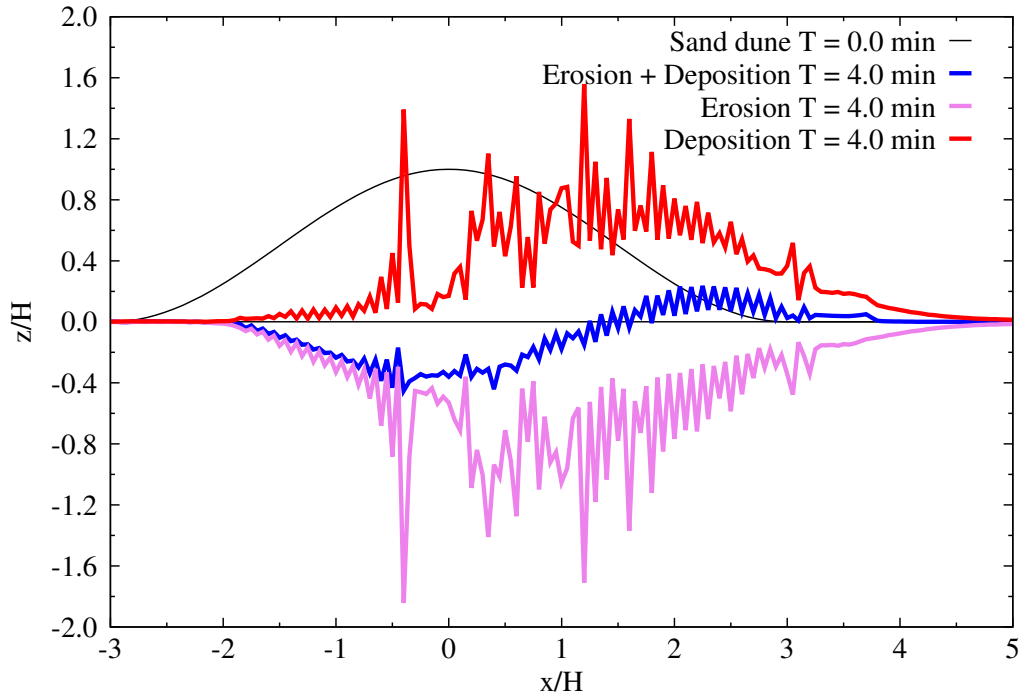
	$t = 2$ min	$t = 4$ min	$t = 6$ min
$e_h(t)$	-2.96%	1.50%	13.4%
$\sigma_h(t)$	0.71%	1.85%	3.91%

Table 5.7 e_h and σ_h between the numerical results (Case A$'$) and the experimental data (Ferreira and Fino (2012)).

Influence of the bed-particle motion model: In order to investigate the influence of the bed-load particle motion model on the sand dune deformation, three numerical cases are conducted with different values of μ_{eff}, *i.e.*, $\mu_{eff} = \tan(8.5°)$ in Case A$'$, $\mu_{eff} = \tan(3.5°)$ in Case B$'$, $\mu_{eff} = \tan(13.5°)$ in Case C$'$. To save the computing time, Case B$'$ and Case C$'$ are initialized using the restart file of Case A$'$ at $t = 1$ min. In Figure 5.17, the dune shapes at $t = 2$ min simulated by Case A$'$, Case B$'$ and Case C$'$ are presented. No big difference between them is observed, except for a slight change at the crest. Therefore, it seems that the parameter μ_{eff} in the bed-load particle motion equation has almost no influence on the numerical simulation of sand dune deformation.

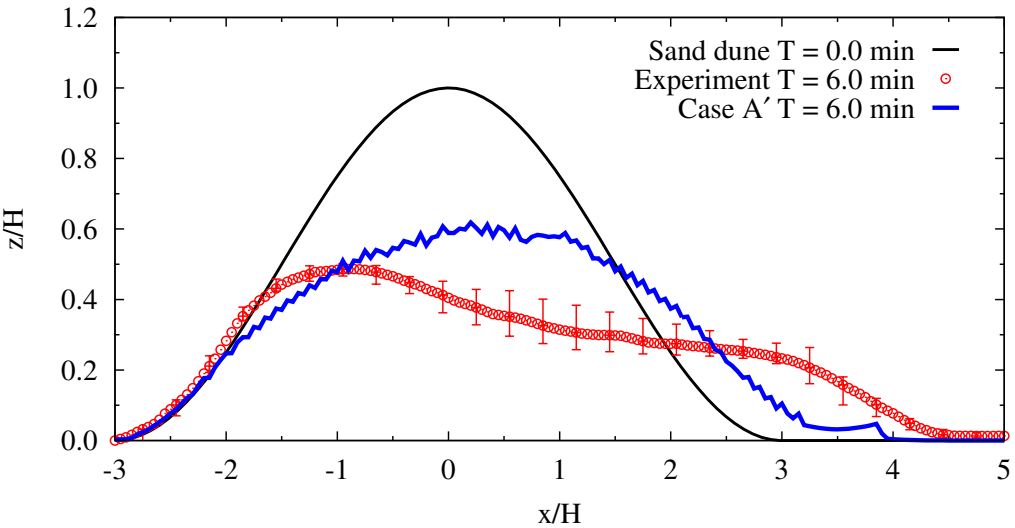


(a) Sand dune shape at $t = 4$ min. Red lines: experiment of Ferreira and Fino (2012). Blue lines and points: numerical simulation (Case A′).

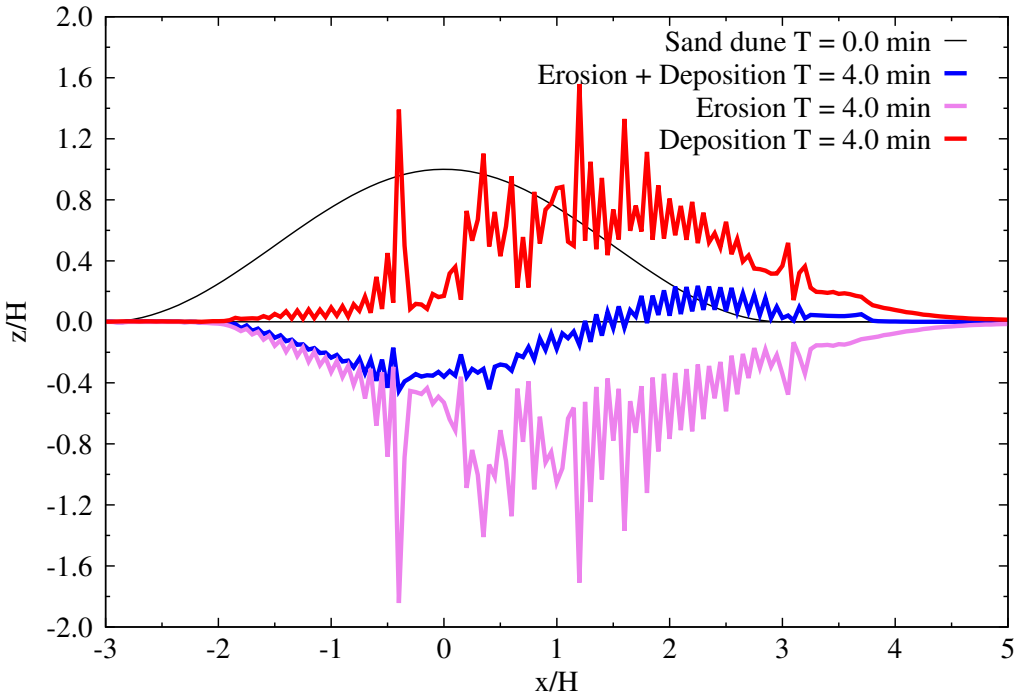


(b) Erosion and deposition. Erosion (violet lines and points): mass of particles removed from the sand dune between $t = 0$ min and $t = 4$ min. Deposition (red lines and points): mass of particles deposited on the sand dune between $t = 0$ min and $t = 4$ min.

Figure 5.15 Sand dune deformation (Case A′) at $t = 4$ min.



(a) Sand dune shape at $t = 6$ min. Red lines: experiment of Ferreira and Fino (2012). Blue lines and points: numerical simulation (Case A′).



(b) Erosion and deposition. Erosion (violet lines and points): mass of particles removed from the sand dune between $t = 0$ min and $t = 6$ min. Deposition (red lines and points): mass of particles deposited on the sand dune between $t = 0$ min and $t = 6$ min.

Figure 5.16 Sand dune deformation (Case A′) at $t = 6$ min.

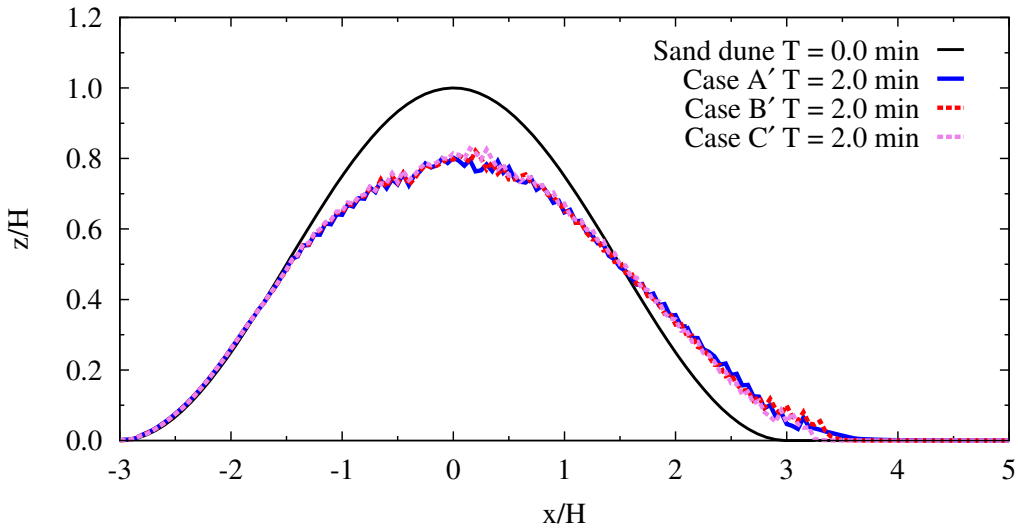


Figure 5.17 Comparison of the dune shape at $t = 2$ min between Case A′, Case B′ and Case C′.

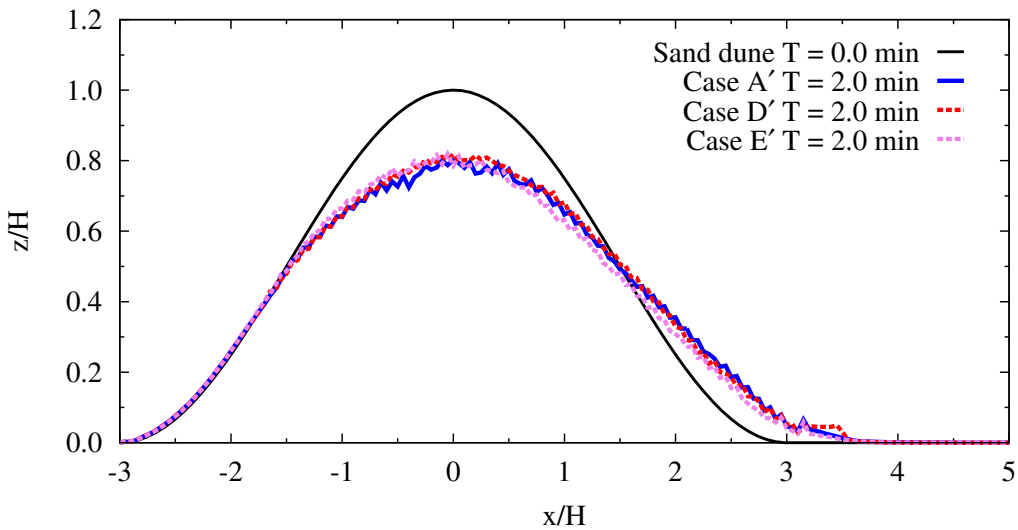


Figure 5.18 Comparison of the dune shape at $t = 2$ min between Case A′, Case D′ and Case E′.

Influence of the splash model: As shown in Figure 5.14, the splash entrainment plays a crucial role on the lee side deformation. Hence, it is necessary to study the effect of the splash process on the sand dune deformation in numerical simulations. In the IBM-ARPS code, a splash model proposed by Kok and Renno (2009) has been implemented and is used to estimate the number of ejected particles. The velocity norm and direction of ejected

particles are randomly assigned according to an empirical formula. The details of the splash model is introduced in Subsection 4.4.2. In this thesis, we carry out three numerical cases with different values of a in Equation (4.78), *i.e.*, $a = 0.05$ in Case A$'$, $a = 0.03$ in Case D$'$, $a = 0.10$ in Case E$'$. It should be noted that a larger value of a gives a greater number of ejected particle, thus results in a stronger splash process. Figure 5.18 shows the dune shapes at $t = 2.0$ min obtained by Case A$'$, Case D$'$ and Case E$'$. It is observed that no big difference between the dune shapes simulated using $a = 0.03$ and $a = 0.05$. For Case E$'$ with a very strong splash function $a = 0.10$, more erosion on the lee side is obtained than those of a weak splash scheme $a = 0.03$ or $a = 0.05$. Hence, it is concluded that the parameter a in the splash function has a slight influence on the lee side deformation although the splash process is observed to be important for the lee side erosion.

5.2.6 Improved splash model with accounting for bed slope effect

To account for the effect of local bed slope, an improved splash model is proposed by introducing a new function:

$$N'_{ejc} = \underbrace{\frac{a}{\sqrt{gd_p}} v_{imp}}_{N_{ejc}} \underbrace{\left(2 - \cos\theta - \frac{\sin\theta}{\tan\varphi}\right)}_{\text{Bed slope effect}}, \tag{5.10}$$

where θ is the local bed slope and $\varphi = 33^\circ$ equals the angle of repose. In Equation (5.10), N_{ejc} is the number of ejected particle by the splash process, which is estimated by an empirical formula proposed by Kok and Renno (2009) and N'_{ejc} denotes the new number by introducing a function of the local bed slope θ. Note that θ depends on the flow direction: θ takes the dune slope when the flow velocity is positive in the streamwise direction; Conversely, θ equals the negative value of the dune slope if the streamwise velocity is reversed, For example, θ on the lee side is positive rather negative in the recirculation zone.

The idea behind Equation (5.10) is that the splash process is harder if $\theta > 0$ since particles are ejected against the gravity, whereas it is easier if $\theta < 0$ as the gravity is helpful for the ejection, namely, the number of ejected particle is smaller for $\theta > 0$ and is conversely larger for $\theta < 0$. Hence, N'_{ejc} in Equation (5.10) is a strictly decreasing function of the bed slope θ. Moreover, Equation (5.10) should satisfy $N'_{ejc} = N_{ejc}$ if $\theta = 0$ and N'_{ejc} should take a positive value for an arbitrary slope, *i.e.*, $N'_{ejc} > 0, \forall\theta \in [-\pi/2, \pi/2]$. With accounting for the bed slope effect, it is hoped that a stronger splash entrainment will occur on the lee side near the crest, which might be able to improve the simulated dune shapes after a long time where an

unexpected strong lee side deposition near the crest is observed in our previous simulation results as shown in Figure 5.15 and Figure 5.16.

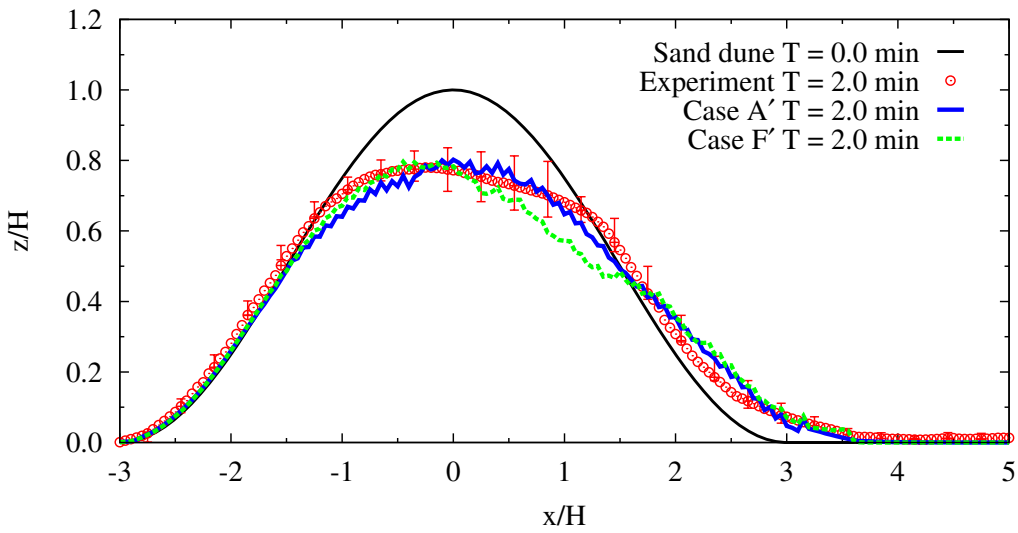


Figure 5.19 Dune shapes at $t = 2$ min obtained by the experiments of Ferreira and Fino (2012), Case A′ and Case F′.

Figure 5.19 gives the simulated dune shape at $t = 2$ min using the improved splash model with accounting for bed slope effect (Equation (5.10)) and compares it with the simulated results using the previous model and the experimental data of Ferreira and Fino (2012). It is observed that the dune height obtained by the new splash model is obviously smaller than the previous simulated result on the lee side near the crest and is slightly larger than that near the dune foot. The mean and root mean square of the new dune height are $e'_h = -4.54\%$ and $\sigma'_h = 2.11\%$, whose absolute value are slightly lager than those of the previous ones $e_h = -2.98\%$ and $\sigma_h = 0.71\%$. This indicates that the improved splash model induces an expected strong lee side erosion near the crest and accordingly brings slightly more deposition near the foot, although the agreement between the new results and the experimental data is not as perfect as the previous ones.

In Figure 5.20, the dune shapes at $t = 4$ min obtained using the new splash model, using the previous model and by the experiments are illustrated. It is observed that the new dune form agrees better with the experimental one than the previously simulated result. This point is also supported by the comparison between the two qualified indicators: the mean and root mean square of the new dune height $e'_h = 1.01\%$ and $\sigma'_h = 1.50\%$ are smaller than those of the previous results $e_h = 1.50\%$ and $\sigma_h = 1.85\%$. This shows that the new splash model

produces a strong erosion on the lee side near the crest as excepted, and thus obtains better results of the dune shape after a long time deformation, although the improvement is not big.

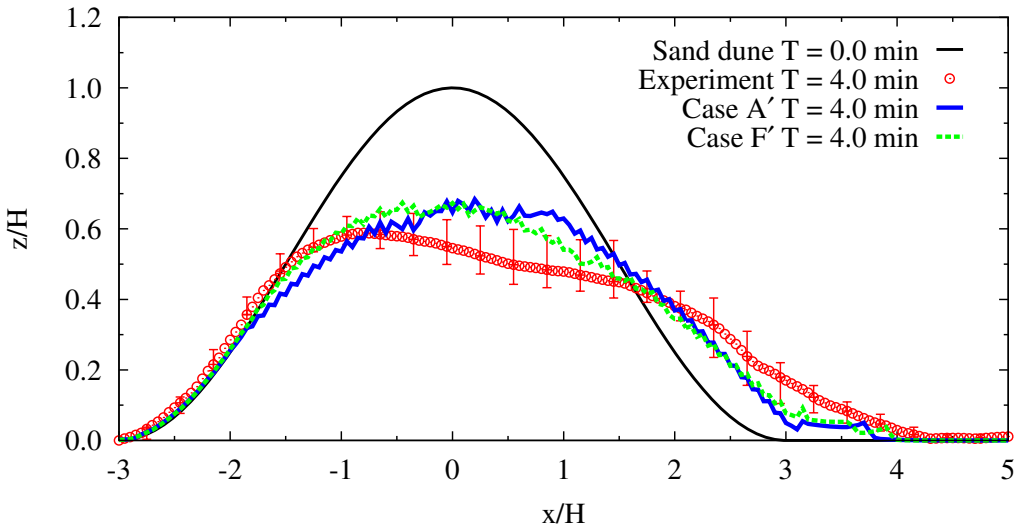


Figure 5.20 Dune shapes at $t = 4$ min obtained by the experiments of Ferreira and Fino (2012), Case A′ and Case F′.

In short, we attempt to propose an improved splash model of Kok and Renno (2009) with accounting for the bed slope effect. By the comparison between the newly simulated results with the previously simulated and experimental ones, it is shown that the new splash model has the ability of producing a stronger erosion on the lee side near the crest and thus brings some improvements to the simulated results of the sand dune shape after a long time deformation. On one hand, this attempt shows that the splash model is important for the lee side erosion and this erosion will become stronger if the bed slope effect is taken into account. On the other hand, as the current splash models are proposed based on the wind-tunnel experiments on a plate surface, this attempt inspires us to carry out the experimental investigation of the bed slope effect on the splash model.

5.3 Conclusions

In this chapter, two numerical simulations of wind erosion are carried out. Firstly, we perform the simulation of particle transport in a TBL over a fixed Gaussian dune, in which the turbulent flow is calculated by the IBM-ARPS and the motion of solid particles are simulated by a Lagrangian tracking approach with integrated wind erosion models. The simulated mean concentration profiles are presented and compared with the numerical results

of Huang (2015) and the experimental data of Simoëns et al. (2015). An overall agreement between them preliminarily validates the ability and accuracy of the developed wind erosion solver of the IBM-ARPS coupled with integrated wind erosion models.

Secondly, we conduct the simulation of wind erosion with sand dune deformation using the original physical models proposed in Chapter 4. For the flow fields, it is shown that the recirculation zone behind the dune is gradually reduced or even disappears as the sand dune deforms, and that the mean longitudinal velocity, Reynolds stress as well as the friction velocity over the dune decreases accordingly as the dune height decreases. For the dune deformation, windward erosion and lee side deposition is observed. In particular, it is shown that the reversed velocity in the recirculation zone is important for the particle deposition and the splash entrainment plays a crucial role on the lee side erosion. It is concluded after testing that the parameter in the equation of the bed-load particle motion has nearly no influence on the dune deformation, and that the parameter in the splash model slightly affects the lee side deformation. By comparing the simulated dune shapes with the experimental ones of Ferreira and Fino (2012), it is observed that a perfect good agreement between them is achieved at $t = 2.0$ min and an unexpected difference on the dune height on lee side near the crest appears after $t = 4.0$ min. Moreover, we attempt to propose an improved splash model with accounting for the bed slope effect and use it in this simulation. It is shown that the improved splash model brings an expected strong lee side erosion near the crest and thus slightly improves the simulated dune shapes at $t = 4.0$ min. We conclude that wind erosion simulations using our developed IBM-ARPS solver coupled with integrated wind erosion models succeed in obtaining a perfect results in a short period but underestimate the dune shapes after a long duration.

Chapter 6

Conclusions and Perspectives

6.1 Conclusions

The goal of this thesis is to conduct the first three-dimensional large-eddy simulation of wind erosion with sand dune deformation. Firstly, a new LES solver has been developed to simulate an atmospheric boundary layer over moving boundaries by introducing the immersed boundary method. Secondly, based on the forces acting on an individual particle, the physical and comprehensive integrated wind erosion models have been proposed and used for the Lagrangian tracking for the trajectories of solid particles carried by turbulent flows. Thirdly, the ability of the developed numerical solver coupled with integrated wind erosion models has been preliminarily validated by comparing the simulated results of particle transport over a fixed Gaussian dune with the experimental data of Simoëns et al. (2015), and then further examined by performing the simulation of wind erosion over a deformable dune and comparing the simulated dune shapes with the experimental ones of Ferreira and Fino (2012).

Developing a numerical method for wind simulation: Due to the presence of a deformable boundary, a new numerical solver (IBM-ARPS) has been developed by introducing the immersed boundary method into the LES code ARPS. A direct forcing approach called ghost-cell finite-difference method has been chosen and implemented. This method constructs a shape representation of the immersed dune surface, on which no-slip boundary condition is imposed through the flow reconstruction on the ghost cells. The accuracy of this solver has been examined through the simulations of a boundary layer over a small Gaussian dune and over a large sinusoidal dune. In the Gaussian case, it is shown that a good agreement on the recirculation zone characteristics, mean longitudinal velocity profiles and Reynolds stress profiles between the simulated results and experimental data is achieved and thus demonstrates the ability of the developed IBM-ARPS solver. It is also shown after

testing that both the interior treatment and the pressure reconstruction have nearly no effect on the standard IBM-ARPS results, whereas the wall-modeling has a slight influence, *e.g.*, weakly reducing the recirculation zone. In the sinusoidal case, a relatively coarse grid is used, *i.e.*, $\Delta z^{+}_{min} = 42$ in the log layer, since the Reynolds number and the blockage ratio are large. It is shown that the results obtained by the IBM-ARPS with wall modeling is much closer to the results simulated by the ARPS with terrain-following grid than those by the standard IBM-ARPS. Particularly, the size of recirculation zone with wall modeling is found to agree well with that in the RANS simulation of Lopes et al. (2013). We conclude that the strategy of wall modeling is important for the IBM-ARPS simulation of a boundary layer over obstacles when a coarse grid is used.

Physical modeling of wind erosion: The forces acting an individual particle submerged within the carried flow or contacting the wall are reviewed. Based on the balance of these forces, particle incipient motions, including lifting, sliding and rolling, have been theoretically studied and their corresponding analytical formulas of the threshold (friction) velocity are obtained. Using the dimension analysis and based on these analytical results, the critical Shield number is supposed to be a combination of three functions: one accounting for the aerodynamic effect, one for the bed slope effect, and the other for the inter-particle cohesion effect. It is shown in the Shield diagram that the modeled values of the critical Shield number of sliding and rolling agree well with the measured ones in different experiments, whereas the values of lifting appears much larger, and that the bed slope effects on the sliding and rolling are similar but different from that on lifting.

To establish integrated wind erosion models, firstly, two instantaneous entrainment models, including lifting for particles detaching far away from the wall and rolling-sliding for particles moving on the wall, are proposed. These models take into account not only the magnitude and duration of the forces acting on the particle but also the numerical grid spacing effect. Secondly, we introduce the governing equations of motion of an airborne heavy particle, which is used for the Lagrangian tracking of the trajectories of solid particles in numerical simulations. Moreover, we firstly focus on the motion of a bed-load particle, which slides or rolls on the wall, and then deduce its equation of motion. Thirdly, a rebound model and a splash model are introduced to account for the particle-surface interactions. Since the movement of bed-load particles are considered, solid particles failing to rebound might continue to slide or roll on the surface and then slow down by the wall friction, or probably deposit immediately on the surface if its velocity is not large enough for the movement. Hence, the criterion of particle deposition in this case is that the particle is located on the surface and its velocity is too small to move. According to the conservation of mass,

the local erosion rate and deposition rate are responsible for the temporal evolution of the dune height. During the deformation, an avalanche model is applied when the local bed slope exceeds the angle of repose.

Numerical simulation of wind erosion: We have conducted three-dimensional numerical simulations of wind erosion, in which turbulent flows are calculated using the IBM-ARPS solver and solid particle trajectories are tracked by a Lagrangian approach. Particle entrainment, particle-surface interaction and particle deposition are accounted for using integrated wind erosion models. In detail, solid particles, are initialized according to the instantaneous particle entrainment models, are then transported by the Lagrangian equations of motion, and lastly collide with the surface or deposit on the wall. During the collision, a rebound model is used to predict the rebounding velocity and angle, and a splash model is applied to estimate the number and velocity of solid particles ejected by the collision.

Firstly, the simulation of particle transport in a TBL over a fixed Gaussian dune is carried out. An overall agreement on the mean particle concentration profiles over the dune between the simulations and the experiment of Simoëns et al. (2015) preliminarily validates the ability of the IBM-ARPS solver coupled with integrated wind erosion models. Secondly, the wind erosion case from the experiment of Ferreira and Fino (2012) is reproduced and studied. For the flow fields, it is observed that the recirculation zone behind the dune is gradually reduced as the dune height decreases, and even disappears at $t = 6$ min, and that the mean longitudinal velocity, Reynolds stress as well as the friction velocity over the dune decreases accordingly. For the dune deformation, the dune shapes are compared with the experimental ones at $t = 2.0$ min, $t = 4.0$ min and $t = 6.0$ min. it is shown that a very good agreement between them is achieved at $t = 2.0$ min and an unexpected difference on lee side near the crest appears at $t = 4.0$ min and $t = 6.0$ min. From the curves of accumulated erosion rate and deposition rate, windward erosion and lee side deposition is observed and the splash entrainment is found to be important for the lee side erosion. By analyzing the simulated dune shapes at different times associated with flow characteristics, it is concluded that the presence of the recirculation zone is important for the particle deposition on the lee side. It is also shown after testing that the value of μ_{eff} in the equation of bed-load particle motion has little effect on the dune deformation, and that a larger value of a in the splash model slightly enhances the lee side erosion. Moreover, It is found from an attempt that a splash model with accounting for the bed slope effect is able to enhance wind erosion near the crest and thus slightly improve the simulated results at $t = 4.0$ min. We conclude that our developed IBM-ARPS solver coupled with integrated wind erosion models shows a good performance

for the numerical simulation of wind erosion in a short period and further improvements need to be proposed for a long time simulation.

6.2 Perspectives

Following the works in this thesis, further studies could be focused on two aspects. On one hand, there is plenty of room for improving the numerical method and the integrated wind erosion models, even though they are currently able to produce an acceptable simulation result of wind erosion:

- In Subsection 3.4.4, an obvious difference on the flow characteristics over a large sinusoidal dune between the IBM-ARPS simulation without wall modeling, the IBM-ARPS simulation with wall modeling and the ARPS simulation with a terrain-following grid is observed. In-depth studies will give us a better understanding and explanation of this observation, and be helpful for improving the accuracy of this developed numerical solver.

- For the immersed boundary method, only two-dimensional interpolation scheme for flow reconstruction has been implemented in ARPS in this thesis. An extension to three-dimensional case will be accomplished in the future.

- The existing rebound and splash models are established based on the experimental data. However, only two-dimensional trajectories of solid particles can be captured by a fast video camera in the experiments. Therefore, they lack some important three-dimensional properties. Moreover, the bed-slope effects on the processes of rebound and splash have never been studied before. Further studies on this subject may bring some improvements on the integrated wind erosion models.

On the other hand, since this numerical modeling of wind erosion has been preliminarily validated, further numerical investigations on the physics of wind erosion will be carried out:

- Numerical simulation of sand ripple formation. It is interesting to find whether the coherent structures in the near-wall regions have an influence on the dynamic formation of sand ripples. Is there a correlation between the characteristic length of sand ripples and the small scales of turbulent flows?

- .....

Appendix A

Similarity theory in turbulent boundary layer flows

A similarity theory describes the dynamic motion by similar functions of non-dimensional physical parameters and scaling coefficients, which may be universal. For turbulent boundary layer flows, a similarity law of the mean velocity is proposed and verified from most of DNS and experimental results. In this appendix, the similar law for boundary layers on a smooth surface is introduced and then the effect of the roughness is discussed.

A.1 Boundary layers on a smooth surface

Considering a simple boundary layer over a plate smooth surface with zero pressure gradient, the mean velocity profile appears similar not only in different streamwise stations, but also for wall flows with different Reynold numbers. This similarity has different mathematic formulations for different regions of boundary layers. They are commonly called the law of the wall in the inner region and the velocity-defect law in the outer layer. From this similarity, the friction law, which describes the relation between the skin-friction coefficient and the Reynold number, is theoretically deduced, and an approximated formula of the mean velocity profile is obtained.

Law of the wall in the inner region ($z/\delta \leq 0.10$)**:** For a fully developed turbulent boundary layer flow at high Reynolds number, the mean velocity profile in the inner layer is expressed as:

$$u^+ = f_w(z^+), u^+ = \frac{U}{u_*}, z^+ = \frac{z}{\delta_\nu}, \tag{A.1}$$

where $U = \langle u \rangle$ is the mean velocity, f_w is the scaling law, u_*, δ_ν are the characteristic velocity and length. In the inner region, u_* is usually given by the friction velocity defined as $u_* = \sqrt{\tau_w/\rho}$ with τ_w the wall shear stress, and δ_ν is then expressed by $\delta_\nu = \nu/u_*$.

For a zero-pressure-gradient boundary layer on a smooth wall, previous studies show that the scaling law f_w is found to be a the linear function in the viscous sublayer, and a logarithmic function in the log region. In the numerical simulation of Huang (2015), an approximated formula of f_w for the smooth case is given by:

$$u^+ = \begin{cases} z^+, & \text{if } z^+ \leq 4; \\ -32.22(z^+)^{-0.3147} + 24.52, & \text{if } 4 < z^+ \leq 90; \\ \frac{1}{\kappa}\ln z^+ + B, & \text{if } z^+ > 90; \end{cases} \tag{A.2}$$

where $\kappa = 0.41$ is the von Kámán constant and $B = 5.5$.

Velocity-defect law in the outer layer ($z^+ > 50$): The similarity in the outer region is given by:

$$\frac{U_e - U}{u_*} = F_d(\chi), \chi = z/\delta, \tag{A.3}$$

where U_e is the external velocity and δ is the thickness of boundary layers.

In boundary layers, the velocity departures from the log law is significant for $z/\delta > 0.2$. Hence, the defect law takes a form of:

$$\frac{U_e - U}{u_*} = -\frac{1}{\kappa}\ln(\chi) + \frac{\Pi}{\kappa}(2 - w(\chi)), \tag{A.4}$$

where the first term denotes the log law, and the second term represents the law of the wake. Coles (1956) proposed a normalized wake function $w(\chi) = 2\sin^2(\chi\pi/2)$, where Π is called the wake strength parameter, and its value is flow dependent. $\Pi \approx 0.55$ is recommended for smooth-wall flows, and $\Pi \approx 0.7$ for fully rough flows [Castro (2007)].

Friction law of boundary layers: Addition of Equation (A.1) and Equation (A.4) at $z = \delta$ yields the friction law by introducing Equation (A.2):

$$\frac{U_e}{u^*} = \frac{1}{\kappa}\ln\left(\frac{\delta}{\delta_\nu}\right) + B + \frac{2\Pi}{\kappa} = \frac{1}{\kappa}\ln\left(Re_\delta \frac{u_*}{U_e}\right) + B + \frac{2\Pi}{\kappa}, \tag{A.5}$$

where the Reynolds number is given by $Re_\delta = U_e\delta/\nu$. When Re_δ is given, solving Equation (A.5) yields a solution of u_*/U_e, hence the skin-friction coefficient $C_f = 2(u_*/U_e)^2$. We can

also obtain the friction law on a rough wall through replacing Equation (A.1) by Equation (A.8).

Mean velocity profile: In the region of overlap between the inner and outer layer ($z^+ > 50$ and $z/\delta < 0.1$), it is easy to examined that the asymptotic forms of $f(y^+)$ (for large y^+) and $F(\eta)$ (for small η) are matched if the logarithmic law is taken. In practice, an empirical mean velocity profile of boundary layers is given by:

$$U = \begin{cases} u_* f_w\left(\dfrac{z}{\delta_v}\right) + u_* \dfrac{\Pi}{\kappa} w\left(\dfrac{z}{\delta}\right), & \text{for } z/\delta \leq 1; \\ U_e, & \text{for } z/\delta > 1; \end{cases} \tag{A.6}$$

where the similarity law f_w takes the formula of Equation (A.2). In numerical simulations, δ and U_e are given, we calculate then u_*/U_e from Equation (A.5), namely, get the value of u_*. Therefore, the average velocity field is evaluated using Equation (A.6) and the initialization is achieved.

A.2 Boundary layers on a rough surface

Previous studies show that the wall roughness breaks the near-wall flow characteristic and then decreases the mean velocity of wall flows. A rough wall is generally characterized by the effective height of roughness elements h, which usually takes the value of the root mean square (RMS) of rough element heights. With taking into account the roughness effect, the similar law (Equation (A.1)) in the inner layer becomes:

$$u^+ = f_w(z^+, h^+), u^+ = \frac{U}{u_*}, z^+ = \frac{u_* z}{\nu}, h^+ = \frac{u_* h}{\nu}. \tag{A.7}$$

The presence of rough elements decreases the mean velocity in the inner region, thus a common formula of the log-law of the wall with the roughness correction can be expressed as [Castro (2007)]:

$$u^+ = \frac{1}{\kappa}\ln(z^+ - z_d^+) + B - \Delta u^+, \tag{A.8}$$

with $z_d^+ = u^* z_d/\nu$ and $\Delta u^+ = \Delta u/u^*$. z_d denotes the displacement less than h, and Δu is the shift velocity, which is a function of wall roughness. By fitting with the experimental data of Nikuradse (1933), Ligrani and Moffat (1986) found the empirical formula of Δu^+ by setting

$z_d^+ = 0$:

$$\Delta u^+ = \begin{cases} 0, & \text{if } h^+ \leq 2.25, \\ \left(\frac{1}{\kappa}\ln h^+ - 3.3\right)\sin\left(\frac{\pi}{2}\frac{\ln(h^+/2.25)}{\ln(90/2.25)}\right), & \text{if } 2.25 \leq h^+ < 90, \\ \frac{1}{\kappa}\ln h^+ - 3.0, & \text{if } h^+ \geq 90. \end{cases} \tag{A.9}$$

Equation (A.9) is illustrated in Figure A.1. the mean velocity departure Δu^+ increases as the roughness height h^+ increases. According to the proprieties of Δu^+, the wall roughness effect is classified into three different regimes:

- Hydro-dynamically smooth regime ($h^+ \leq 5$): the rough elements are submerged in the viscous sublayer, and the shift velocity is nearly zero as shown in Figure A.1;
- Transitionally rough regime ($5 < h^+ \leq 70$): the rough elements are located in the buffer layer or at the beginning of the log layer, and Δu^+ is a complex function of h^+;
- Fully rough regime ($h^+ > 70$): the rough elements are completely submerged in the log layer, and Δu^+ is a logarithmic function of h^+.

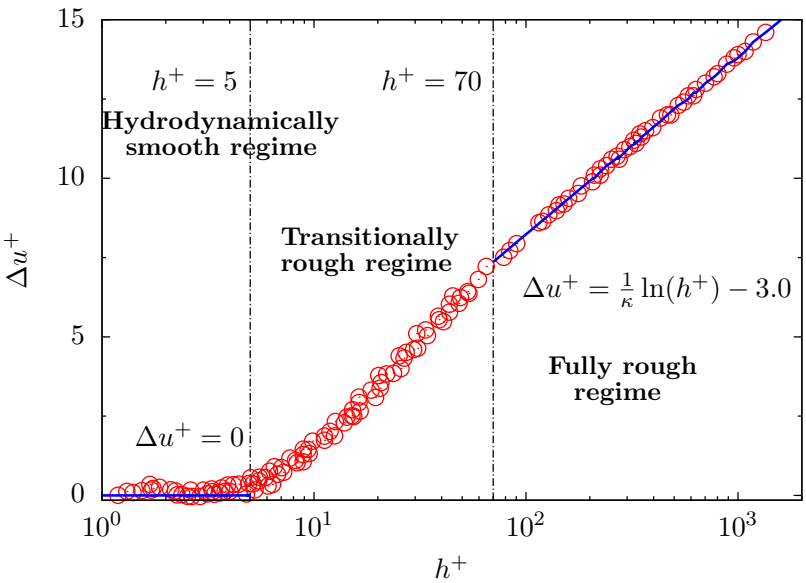


Figure A.1 Experimental results of the shift velocity Δu^+ from Nikuradse (1933) and schematic illustration of different roughness regimes: hydrodynamically smooth regime, transitionally rough regime and fully rough regime.

In particular, for the fully rough regime, one can find $\Delta u^+ = 1/\kappa \ln h^+ + 1/\kappa \ln(z_0/h) + B$ with z_0 representing the roughness height, then we have from Equation (A.8):

$$u^+ = \frac{1}{\kappa} \ln\left(\frac{z - z_d}{z_0}\right). \tag{A.10}$$

Using Equation (A.9), we obtain $z_d = 0$ and $1/\kappa \ln(z_0/h) + B = -3.0$. Taking $\kappa = 0.41$ and $B = 5.5$, we have $z_0 = 0.031h \approx h/30$, which is consistent with the experimental results with spherical grains of Bagnold (1941) and Kamphuis (1974): $z_0 = 0.03\text{-}0.1h$.

According to the roughness configuration shown in Figure A.2, Oke (1988) shown that the near-wall flow can be categorized into three different flow regimes [Huang et al. (2016)]:

- Isolated flow with a small roughness density (≤ 0.15) and $z_d \leq 0.2h$, in which the interaction between the flow and one rough element is completely independent;
- Waked flow with a middle roughness density (0.15-0.35) and $0.75h < z_d \leq h$, in which the interaction between the flow and rough elements is interrelated;
- Skimming flow with a large roughness density (> 0.35) and $0.75h < z_d \leq h$, in which an isolated eddy is produced in each gap between rough elements and the interaction with outer flow is small.

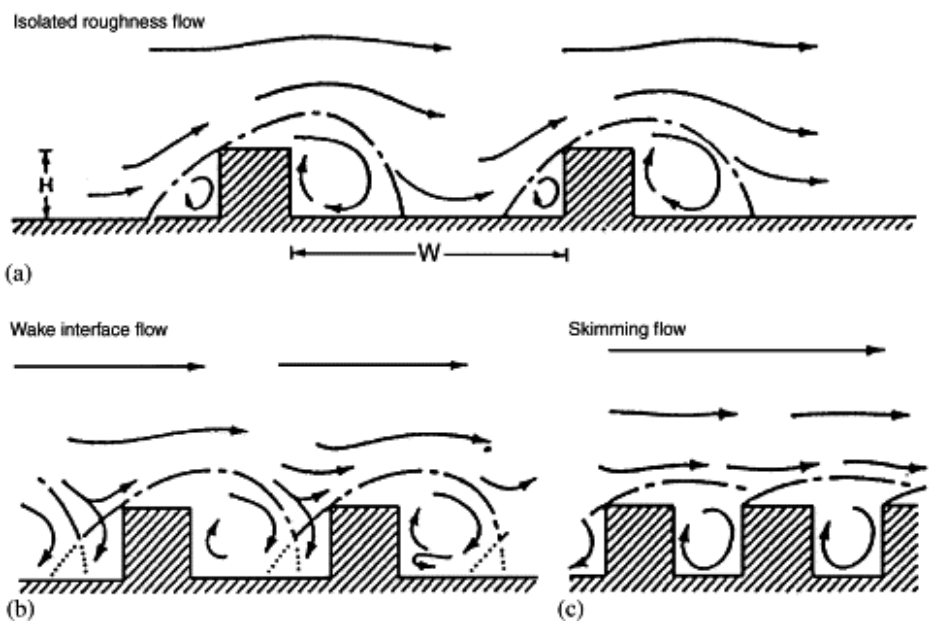


Figure A.2 Sketch of different flow regime depending on the roughness configuration: isolated flow regime, waked flow regime and skimming flow regime. Taken from Oke (1988).

Recently, Huang et al. (2016) introduced a new roughness characteristic scale defined as $\varepsilon = h - z_d$ and the slip velocity u_w as the corresponding average velocity at $z = z_d$, then proposed a new formula of the shift velocity:

$$\Delta u^+ = \frac{1}{\kappa}\ln(\varepsilon^+) + u_w^+. \tag{A.11}$$

By fitting with a collected dataset consisting of DNS results and experimental data, the slip velocity takes a form of:

$$u_w^+ = \frac{1}{\kappa}\ln\left(\frac{z_0}{\varepsilon}\right) + W\exp(4 - 4z/\varepsilon) + B, \tag{A.12}$$

with $W = 1.2$ for both isolated and wake flow regimes, and $W = 4.5$ for the skimming flow regime.

Appendix B

Numerical details in ARPS

B.1 Discretized formulation of governing equations

Using a vector of physical variables $\boldsymbol{\psi}$, which consists of the pressure variation $\Delta\tilde{p}$, the fluid velocity $\tilde{u}_i$, and the potential temperature variation $\Delta\tilde{\theta}$, *i.e.*, $\boldsymbol{\psi} = (\Delta\tilde{p}, \tilde{u}_i, \Delta\tilde{\theta})^t$, the governing equations (Equations (2.25)) simplify to:

$$\frac{\partial \boldsymbol{\psi}}{\partial t} = \mathscr{W}\boldsymbol{\psi} + \mathscr{C}\boldsymbol{\psi} + \mathscr{R}\boldsymbol{\psi} + \mathscr{D}\boldsymbol{\psi}, \tag{B.1}$$

where $\partial\boldsymbol{\psi}$, $\mathscr{W}\boldsymbol{\psi}$, $\mathscr{C}\boldsymbol{\psi}$, $\mathscr{R}\boldsymbol{\psi}$ are detailedly given by:

- $\partial\boldsymbol{\psi}/\partial t$: temporal evolution of $\boldsymbol{\psi}$, such as time partial difference of pressure variation $\partial(\Delta\tilde{p})/\partial t$, of velocity $\partial\tilde{u}_i/\partial t$ and of the potential temperature variation $\partial(\Delta\tilde{\theta})/\partial t$ in Equation (2.25);
- $\mathscr{W}\boldsymbol{\psi}$: acoustic and gravitational wave term, such as $-\rho_r c_s^2 \partial\tilde{u}_j/\partial x_j$ in Equation (2.25a), and $-\rho_r^{-1}\partial(\Delta\tilde{p})\partial x_i + g_i(\Delta\tilde{\theta}/\tilde{\theta}_r - \Delta\tilde{p}/(\rho_r c_s^2))$ in Equation (2.25b);
- $\mathscr{C}\boldsymbol{\psi}$: convective term of $\boldsymbol{\psi}$, always taking form of $-\tilde{u}_j\partial\boldsymbol{\psi}/\partial x_j$ in Equations (2.25) where the components of $\boldsymbol{\psi}$ are $\Delta\tilde{p}$, $\tilde{u}_i$ and $\Delta\tilde{\theta}$;
- $\mathscr{R}\boldsymbol{\psi}$: the rotational term, *e.g.*, $\mathscr{R}\boldsymbol{\psi}$ is represented by $-2\varepsilon_{ijk}\tilde{u}_j\Omega_k$ in Equation (2.25b);
- $\mathscr{D}\boldsymbol{\psi}$: the diffusive term, representing the viscous term in Equation (2.25b), the thermal diffusion term in Equation (2.25c).

In the numerical simulation, the finite difference method is applied to discretize the computational time-space domain. Specially, the mode-splitting time integration approach is used to eliminate the acoustic waves effect. According to this approach, the time domain

is firstly discretized into a set of sub-domains with a large time step, which is then divided into a number of computationally inexpensive small time steps. Only the acoustically active terms are updated in a small time integration while all the other terms are advanced every big time step. As a consequence, only the small time step is limited by the acoustic wave effect.

Before deriving the discrete form of governing equations, several notions are introduced here. Firstly, the fluid velocity will expressed as $\tilde{u}_i = (\tilde{u}, \tilde{v}, \tilde{w})$ with corresponding coordinate system (x, y, z). Secondly, the continuous time-space domain is discretized into the equal sub-domains with large time step Δt, small time-step $\Delta\tau$ and spatial grid sizes Δx, Δy and Δz, which yields the discrete form of a physical variable:

$$\boldsymbol{\psi}_{i,j,k}^{n,m} = \boldsymbol{\psi}((i-1)\Delta x, (j-1)\Delta y, (k-1)\Delta z, (n-1)\Delta t + (m-1)\Delta\tau), \tag{B.2}$$

with i, j, k the position indexes, and n, m the large and small time indexes in the discrete computing domain. The maximum of small time step is $\Delta t/\Delta\tau$. Note that the physical variables of discrete form are staggered on an Arakawa C-grid shown in Figure 2.1, and that $\Delta\tau$ takes a value of Δt divided by a non-zero integer and $1 \leq m < \Delta t/\Delta\tau$. For the simplification in numerical discretization, two spatial increment and average operators are defined as:

$$\delta_j \boldsymbol{\psi}(x_i, t) = \boldsymbol{\psi}\left(x_i + \frac{1}{2}\delta_{ij}\Delta x_j, t\right) - \boldsymbol{\psi}\left(x_i - \frac{1}{2}\delta_{ij}\Delta x_j, t\right), \tag{B.3a}$$

$$\mu_j \boldsymbol{\psi}(x_i, t) = \frac{1}{2}\left(\boldsymbol{\psi}\left(x_i + \frac{1}{2}\delta_{ij}\Delta x_j, t\right) + \boldsymbol{\psi}\left(x_i + \frac{1}{2}\delta_{ij}\Delta x_j, t\right)\right). \tag{B.3b}$$

For example, in the staggered grid, we have $\delta_1 \boldsymbol{\psi}_{i,j,k}^{n,m} = \boldsymbol{\psi}_{i+1,j,k}^{n,m} - \boldsymbol{\psi}_{i,j,k}^{n,m}$ and $\mu_2 \boldsymbol{\psi}_{i,j,k}^{n,m} = (\boldsymbol{\psi}_{i,j,k}^{n,m} + \boldsymbol{\psi}_{i,j+1,k}^{n,m})/2$.

Then the discrete form of Equation (B.1) is obtained in a general way

$$\frac{\boldsymbol{\psi}_{i,j,k}^{n,m+1} - \boldsymbol{\psi}_{i,j,k}^{n,m}}{\Delta\tau} = \beta\mathscr{W}\boldsymbol{\psi}_{i,j,k}^{n,m+1} + (1-\beta)\mathscr{W}\boldsymbol{\psi}_{i,j,k}^{n,m} + \mathscr{C}\boldsymbol{\psi}_{i,j,k}^{n,0} + \mathscr{R}\boldsymbol{\psi}_{i,j,k}^{n,0} + \mathscr{D}\boldsymbol{\psi}_{i,j,k}^{n-1,0}, \tag{B.4}$$

where β is the coefficient between 0 and 1, *i.e.*, $0 \leq \beta \leq 1$, $\mathscr{W}\boldsymbol{\psi}$ uses small time integration m, and the others $\mathscr{C}\boldsymbol{\psi}$, $\mathscr{R}\boldsymbol{\psi}$, $\mathscr{D}\boldsymbol{\psi}$ use the large time integration n. Note that $\beta = 0$ results in an explicit forward scheme, $0 < \beta \leq 1$ yields an implicit scheme, specially, $\beta = 1$ leads to a backward scheme.

Forward scheme in the horizontal direction: Assuming that the base density ρ_r and base potential temperature θ_r are close to constant, applying the forward scheme in the small time integration by setting $\beta = 0$ in Equation (B.4), the momentum equation (Equation (2.25b))

in the horizontal direction becomes after the discretization:

$$\frac{\tilde{u}_{i,j,k}^{n,m+1}-\tilde{u}_{i,j,k}^{n,m}}{\Delta\tau}=-\frac{1}{\rho_r}\frac{\delta_1(\Delta\tilde{p})_{i,j,k}^{n,m}}{\Delta x}+\mathscr{L}\tilde{u}_{i,j,k}^{n,0}, \tag{B.5a}$$

$$\frac{\tilde{v}_{i,j,k}^{n,m+1}-\tilde{v}_{i,j,k}^{n,m}}{\Delta\tau}=-\frac{1}{\rho_r}\frac{\delta_2(\Delta\tilde{p})_{i,j,k}^{n,m}}{\Delta y}+\mathscr{L}\tilde{v}_{i,j,k}^{n,0}, \tag{B.5b}$$

where $\mathscr{L}\tilde{u}$, $\mathscr{L}\tilde{v}$ are the terms of large time integration, containing the convective term $\mathscr{C}\tilde{u},\mathscr{C}\tilde{v}$, the rotation term $\mathscr{R}\tilde{u},\mathscr{R}\tilde{v}$ and the diffusive term $\mathscr{D}\tilde{u},\mathscr{D}\tilde{v}$. Thus, $\mathscr{L}\tilde{u}_{i,j,k}^{n,0}$ and $\mathscr{L}\tilde{v}_{i,j,k}^{n,0}$ are expressed as:

$$\mathscr{L}\tilde{u}_{i,j,k}^{n,0}=\mathscr{C}\tilde{u}_{i,j,k}^{n,0}+\mathscr{R}\tilde{u}_{i,j,k}^{n,0}+\mathscr{D}\tilde{u}_{i,j,k}^{n-1,0}, \tag{B.6a}$$

$$\mathscr{L}\tilde{v}_{i,j,k}^{n,0}=\mathscr{C}\tilde{v}_{i,j,k}^{n,0}+\mathscr{R}\tilde{v}_{i,j,k}^{n,0}+\mathscr{D}\tilde{v}_{i,j,k}^{n-1,0}. \tag{B.6b}$$

Crank-Nicolson scheme in the vertical direction: After the Crank-Nicolson scheme is used in the time discretization of Equation (2.25a) and Equation (2.25b) in vertical direction, it is obtained from Equation (B.4):

$$\begin{aligned}\frac{(\Delta\tilde{p})_{i,j,k}^{n,m+1}-(\Delta\tilde{p})_{i,j,k}^{n,m}}{\Delta\tau}&=-\rho_r g(\beta\mu_3\tilde{w}_{i,j,k}^{n,m+1}+(1-\beta)\mu_3\tilde{w}_{i,j,k}^{n,m})-\rho_r c_s^2\frac{\delta_1\tilde{u}_{i,j,k}^{n,m+1}}{\Delta x}\\&-\rho_r c_s^2\left(\beta\frac{\delta_3\tilde{w}_{i,j,k}^{n,m+1}}{\Delta z}+(1-\beta)\frac{\delta_3\tilde{w}_{i,j,k}^{n,m}}{\Delta z}\right)-\rho_r c_s^2\frac{\delta_2\tilde{v}_{i,j,k}^{n,m+1}}{\Delta y}+\mathscr{L}\tilde{p}_{i,j,k}^{n,0},\end{aligned} \tag{B.7a}$$

$$\begin{aligned}\frac{\tilde{w}_{i,j,k}^{n,m+1}-\tilde{w}_{i,j,k}^{n,m}}{\Delta\tau}&=-\frac{1}{\rho_r}\left(\beta\frac{\delta(\Delta\tilde{p})_{i,j,k}^{n,m+1}}{\Delta z}+(1-\beta)\frac{\delta(\Delta\tilde{p})_{i,j,k}^{n,m}}{\Delta z}\right)+\frac{g}{\theta_r}\mu_3(\Delta\tilde{\theta})_{i,j,k}^{n,m}\\&-\frac{g}{\rho_r c_s^2}(\beta\mu_3(\Delta\tilde{p})_{i,j,k}^{n,m+1}+(1-\beta)\mu_3(\Delta\tilde{p})_{i,j,k}^{n,m})+\mathscr{L}\tilde{w}_{i,j,k}^{n,0},\end{aligned} \tag{B.7b}$$

where $\mathscr{L}\tilde{w}$ and $\mathscr{L}\tilde{p}$ represent the terms of large time integration, consisting of $\mathscr{C}\tilde{w}$, $\mathscr{R}\tilde{w}$, $\mathscr{D}\tilde{w}$ and $\mathscr{C}\tilde{p}$, respectively. Thus, $\mathscr{L}\tilde{w}_{i,j,k}^{n,0}$ and $\mathscr{L}\tilde{p}_{i,j,k}^{n,0}$ are expressed as:

$$\mathscr{L}\tilde{w}_{i,j,k}^{n,0}=\mathscr{C}\tilde{w}_{i,j,k}^{n,0}+\mathscr{R}\tilde{w}_{i,j,k}^{n,0}+\mathscr{D}\tilde{w}_{i,j,k}^{n-1,0}, \tag{B.8a}$$

$$\mathscr{L}\tilde{p}_{i,j,k}^{n,0}=\mathscr{C}\tilde{p}_{i,j,k}^{n,0}. \tag{B.8b}$$

In numerical simulations with complex topology, to avoid the numerical instability caused by the acoustic wave and gravitational wave, an implicit algorithm $\beta\neq 0$ in the vertical direction is usually chosen, *i.e.*, $\beta=0.6$ recommended in ARPS.

When $\beta\neq 0$, the unknown terms are only $(\Delta p)_{i,j,k}^{n,m+1}$ and $w_{i,j,k}^{n,m+1}$ at future time step

in Equation (B.7a) and Equation (B.7b). These two equations constitute a linear equation system with the known coefficients, and should be resolved together. In detail, combining Equation (B.7a) and Equation (B.7b), and eliminating the term $(\Delta\tilde{p})^{n,m+1}_{i,j,k}$, we obtain a linear tridiagonal equation system:

$$A_k\tilde{w}^{n,m+1}_{i,j,k-1}+B_k\tilde{w}^{n,m+1}_{i,j,k}+C_k\tilde{w}^{n,m+1}_{i,j,k+1}=D_k, \tag{B.9}$$

where the coefficients A_k, B_k, C_k and D_k are given by:

$$A_k=\left(\frac{2}{\Delta z}\right)^2-\left(\frac{g}{c_s^2}\right)^2, \tag{B.10a}$$

$$B_k=\frac{4\rho_r^2}{(\beta c_s\Delta\tau)^2}+2\left(\frac{2}{\Delta z}\right)^2+2\left(\frac{g}{c_s^2}\right)^2, \tag{B.10b}$$

$$C_k=\left(\frac{2}{\Delta z}\right)^2+\left(\frac{g}{c_s^2}\right)^2, \tag{B.10c}$$

$$\begin{aligned}D_k=&\frac{2}{\beta\Delta\tau\rho_r c_s^2}\left(\frac{2}{\Delta z}-\frac{g}{c_s^2}\right)\left((\Delta\tilde{p})^{n,m}_{i,j,k-1}+\mathscr{L}\tilde{p}^{n,0}_{i,j,k-1}\right)+\tilde{w}^{n,m}_{i,j,k}+\mathscr{L}\tilde{w}^{n,0}_{i,j,k}\\&-\frac{2}{\beta\Delta\tau\rho_r c_s^2}\left(\frac{2}{\Delta z}+\frac{g}{c_s^2}\right)\left((\Delta\tilde{p})^{n,m}_{i,j,k}+\mathscr{L}\tilde{p}^{n,0}_{i,j,k}\right).\end{aligned} \tag{B.10d}$$

It is found that the coefficients A_k, B_k, C_k are independent on the fluid velocity fields and that only D_k changes with respect to the fluid motion. Combined with the boundary conditions of $\tilde{w}$ described further in Section B.2, Equation (B.9) is solved using an alternating direction implicit (ADI) method based on the Thomas algorithm in ARPS [Roache (1972)]. After $\tilde{w}^{n,m+1}_{i,j,k}$ is obtained, substituting it into Equation (B.7a) yields a solution of $(\Delta\tilde{p})^{n,m+1}_{i,j,k}$.

Scalar discretization: Considering the potential temperature $\Delta\tilde{\Theta}$ in Equation B.4, we obtain the discrete form of Equation (2.25c):

$$\begin{aligned}\frac{(\Delta\tilde{\Theta})^{n,m+1}_{i,j,k}-(\Delta\tilde{\Theta})^{n,m}_{i,j,k}}{\Delta\tau}=&-\mu_1\tilde{u}^{n,m+1}_{i,j,k}\frac{\delta_1(\Theta_r)^{n,m}_{i,j,k}}{\Delta x}-\mu_2\tilde{v}^{n,m+1}_{i,j,k}\frac{\delta_2(\Theta_r)^{n,m}_{i,j,k}}{\Delta y}\\&-\mu_3\tilde{w}^{n,m+1}_{i,j,k}\frac{\delta_3(\Theta_r)^{n,m}_{i,j,k}}{\Delta z}+\mathscr{L}\tilde{\Theta}^{n,0}_{i,j,k},\end{aligned} \tag{B.11}$$

where $\mathscr{L}\tilde{\Theta}$ is the large time integration term, containing the convection term $\mathscr{C}\tilde{\Theta}$ and diffusion term $\mathscr{D}\tilde{\Theta}$. Thus, $\mathscr{L}\tilde{\Theta}^{n,0}_{i,j,k}$ is expressed as:

$$\mathscr{L}\tilde{\Theta}^{n,0}_{i,j,k}=\mathscr{C}\tilde{\Theta}^{n,0}_{i,j,k}+\mathscr{D}\tilde{\Theta}^{n-1,0}_{i,j,k}. \tag{B.12}$$

B.2 Detailed description of boundary conditions

Rigid wall condition: In ARPS, the bottom boundary condition on the ground is the free-slip boundary condition given by:

$$\left.\frac{\partial \tilde{u}}{\partial z}\right|_{z=0} = 0, \left.\frac{\partial \tilde{v}}{\partial z}\right|_{z=0} = 0,. \tag{B.13}$$

Using the incompressibility assumption, taking a $z-$ derivative of the continuity equation (Equation (2.1a)) yields:

$$\left.\frac{\partial^2 \tilde{u}}{\partial x \partial z}\right|_{z=0} + \left.\frac{\partial^2 \tilde{v}}{\partial y \partial z}\right|_{z=0} + \left.\frac{\partial^2 \tilde{w}}{\partial^2 z}\right|_{z=0} = 0. \tag{B.14}$$

Substituting Equation (B.13) into Equation (B.14) yields $\left.\partial^2 \tilde{w}/\partial z^2\right|_{z=0} = 0$, and we have $\tilde{w}|_{z=0} = 0$ from the non-penetrative condition of a rigid wall.

As the staggered grid is used in ARPS, the rigid wall condition is written as:

$$\begin{aligned}
&\text{Streamwise velocity: } \tilde{u}^{n,m}_{i,j,1} = \tilde{u}^{n,m}_{i,j,2}, \\
&\text{Spanwise velocity: } \tilde{v}^{n,m}_{i,j,1} = \tilde{v}^{n,m}_{i,j,2}, \\
&\text{Wall-normal velocity: } \tilde{w}^{n,m}_{i,j,1} = -\tilde{w}^{n,m}_{i,j,3} \text{ and } \tilde{w}^{n,m}_{i,j,2} = 0.
\end{aligned} \tag{B.15}$$

Inlet boundary condition: The inlet boundary condition for boundary layer flow is important and complex. It should supply not only the average velocity fields but also the random fluctuations, which contains the information of coherent structures. Here we introduce two approaches: the random approach, and the recycling approach.

Firstly, In the random method, the mean velocity U_i is given by the scaling law on the smooth or rough wall, and the velocity fluctuations is generated through a series of random number, to match a prescribed Reynold stress T_{ij}. At each time integration in the simulation, the inflow data generated by the random strategy is regarded as inlet boundary condition. The sequence of operations is presented as follows:

- Generating three sequences of random numbers ξ_i, each of which obeys the probabilistic distribution with zero mean, unit variance, and zero covariance with the other two distributions. The numerical strategies to generated ξ_i is found in Le et al. (1997),

- The fluid velocity field is then given by:

$$u_i = U_i + a_{ij}\xi_i \tag{B.16}$$

with the coefficient tensor a_{ij}, which is a function of T_{ij} via

$$a_{ij} = \begin{bmatrix} \sqrt{T_{11}} & 0 & 0 \\ T_{21}/a_{11} & \sqrt{T_{22} - a_{21}^2} & 0 \\ T_{31}/a_{11} & (T_{32} - a_{21}a_{31})/a_{22} & \sqrt{T_{33} - a_{31}^2 - a_{32}^2} \end{bmatrix}. \tag{B.17}$$

Secondly, to provide more information of coherent structures and reduce the flow transition length, Lund et al. (1998) proposed a preliminary boundary layer and developed the recycling method to generate inflow data.

Through the scaling law of average velocity profile (Equation (A.6)), we know:

$$U_{inlt} = \begin{cases} \gamma U_{recy}, & \text{for } \eta \leq 1; \\ U_{recy}, & \text{for } \eta > 1. \end{cases} \tag{B.18}$$

where U_{inlt}, U_{recy} are the mean velocity at the inlet and recycle station, $\gamma = u_{*,inlt}/u_{*,recy}$ with $u_{*,inlt}$, $u_{*,recy}$ the friction velocity at the inlet and recycling station, and $\chi = \chi_{inlt} = \chi_{recy}$.

By assuming the scaling law of the fluctuations $u' = u - U$ as Equation (A.1) and Equation (A.3), we have then the relation between the inlet station and recycling state is

$$u'_{inlt} = \begin{cases} \gamma u'_{recy}, & \text{for } \chi \leq 1; \\ u'_{recy}, & \text{for } \chi > 1, \end{cases} \tag{B.19}$$

with u'_{inlt}, u'_{recy} the fluctuating velocity at the inlet and recycle station.

According to Equation (B.18) and Equation (B.19), one can find a same similarity between the mean velocity U or the fluctuation u' located in inlet station and recycle station, which enables us to generate the inlet velocity by rescaling one at the recycling station. In practice, a weighted composite velocity is used

$$u_{inlt} = u_{inlt}(1 - W(\chi)) + u_{recy}W(\chi) \tag{B.20}$$

with the weight function W defined as

$$W(\chi) = \frac{1}{2}\left(1 + \tanh\left(\frac{a_0(\chi - b_0)}{(1 - 2b_0)\chi + b_0}\right) / \tanh(a_0)\right) \tag{B.21}$$

where $a_0 = 4$ and $b_0 = 0.2$.

Outlet boundary condition: In practice, a radiative boundary condition is usually used as the outlet boundary condition [Orlanski (1976)]:

$$\frac{\partial \boldsymbol{\psi}}{\partial t} + c_{\boldsymbol{\psi}} \frac{\partial \boldsymbol{\psi}}{\partial x} = 0, \tag{B.22}$$

where $c_{\boldsymbol{\psi}}$ the phase velocity of the wave signals. In ARPS, $c_{\boldsymbol{\psi}}$ is approximated locally through applying the same equation at one time step earlier and one grid point interior of the boundary. Miller and Thorpe (1981) analyzed a number of variations of the original radiative formulation of Orlanski (1976). Durran and Klemp (1982) proposed a vertical average estimation on the local phase speed $c_{\boldsymbol{\psi}}$ and applied this averaged one in their numerical simulation of atmospheric flows over mountains.

Another variation of radiative boundary condition suggested by Klemp and Wilhelmson (1978) is:

$$\frac{\partial \boldsymbol{\psi}}{\partial t} + (V + C_{\boldsymbol{\psi}}) \frac{\partial \boldsymbol{\psi}}{\partial x} = 0, \tag{B.23}$$

where V is the flow speed normal to the boundary, and $C_{\boldsymbol{\psi}}$ is a constant phase speed of dominant waves. The studies of Clark (1979) and Lilly (1981) showed that an overestimation of the phase speed $C_{\boldsymbol{\psi}}$ is better that an underestimation of $C_{\boldsymbol{\psi}}$. Hence, $C_{\boldsymbol{\psi}}$ is commonly assigned as the fastest speed of gravitational wave propagating.

Upper boundary condition When the explicit scheme in the vertical direction is used, the zero-gradient boundary condition can be considered as the top boundary conditions. When the implicit scheme is used, the zero-normal gradient boundary condition is imposed for the horizontal velocity and other physique quantities, except for the w and Δp, which are governed by the upper boundary condition.

Klemp and Durran (1983) proposed the upper boundary condition based on an analysis of linear hydrostatic gravity waves. It is a wave-permeable radiation boundary condition, which generates a zero downward energy transport by the hydrostatic gravity waves. Supposing $\hat{w}$, $\Delta\hat{p}$ the vertical velocity and pressure variation after the Fourier transformation, the upper boundary condition is given by:

$$\Delta\hat{p} = \frac{N\rho_r}{k}\hat{w}, \tag{B.24}$$

where ρ_r the density of base state, N the *Brunt-Väisälä* frequency defined as $N = \sqrt{-\frac{g}{\rho}\frac{\partial \rho}{\partial z}}$ and $k = \sqrt{\hat{k}_x^2 + \hat{k}_y^2}$ the horizontal wave-number, respectively.

The upper boundary condition has been implanted for the w and Δp implicit solver in

ARPS. In detail, the horizontal wave-numbers are $\hat{k}_x = \frac{2}{\Delta x}\sin\left(\frac{k_x \Delta x}{2}\right)$ and $\hat{k}_y = \frac{2}{\Delta y}\sin\left(\frac{k_y \Delta y}{2}\right)$ with Δx, Δy the grid increment in horizontal directions and k_x, k_y denoted as the wavenumber of integer type. Neglecting the horizontal variation in the coefficient, and assuming that $\Delta\hat{p}$ is approximated to be the value at the w position in the staggered grid, Equation (B.24) at the time $t = (n-1)\Delta t + m\Delta\tau$ is rewritten as:

$$\Delta\hat{p}^{n,m+1}_{i,j,nz-2} = \frac{N\rho_r}{k}\hat{w}^{n,m+1}_{i,j,nz-1}, \tag{B.25}$$

with nz the grid number in vertical direction, $\Delta\hat{p}^{n,m}_{i,j,nz-2}$ the pressure variation located one-half grid level below the top boundary, and $\hat{w}^{n,m}_{i,j,nz-1}$ the vertical velocity located at the boundary.

From the discrete pressure variation dynamic equation (Equation (B.7a)), we have by setting $k = nz-2$:

$$(\Delta\tilde{p})^{n,m+1}_{i,j,nz-2} = a_z\tilde{w}^{n,m+1}_{i,j,nz-1} + b_z\tilde{w}^{n,m+1}_{i,j,nz-2} + c_z, \tag{B.26}$$

where

$$\begin{aligned}
a_z &= \Delta\tau\beta\left(\frac{g\rho_r}{2} - \frac{c_s^2\rho_r}{\Delta z}\right),\\
b_z &= \Delta\tau\beta\left(\frac{g\rho_r}{2} + \frac{c_s^2\rho_r}{\Delta z}\right),\\
c_z &= (\Delta\tilde{p})^{n,m}_{i,j,nz-2} - \Delta\tau\rho_r c_s^2\frac{\delta_1\tilde{u}^{n,m+1}_{i,j,nz-2}}{\Delta x} - \Delta\tau\rho_r c_s^2\frac{\delta_2\tilde{v}^{n,m+1}_{i,j,nz-2}}{\Delta y} + \Delta\tau\mathscr{L}\tilde{p}^{n,0}_{i,j,nz-2}.
\end{aligned} \tag{B.27}$$

Assuming that the coefficient are slowly varying function of x and y, taking a Fourier transform on Equation (B.26) yields:

$$\hat{p}^{n,m+1}_{i,j,nz-2} = a_z\hat{w}^{n,m+1}_{i,j,nz-1} + b_z\hat{w}^{n,m+1}_{i,j,nz-2} + c_z, \tag{B.28}$$

Eliminating $\hat{p}^{n,m}_{i,j,nz-2}$ by substituting Equation (B.28) into Equation (B.25) yields:

$$\left(a_z - \frac{N\rho_r}{k}\right)\hat{w}^{n,m+1}_{i,j,nz-1} + b_z\hat{w}^{n,m+1}_{i,j,nz-2} + c_z = 0, \tag{B.29}$$

which will be regarded as the top boundary condition after being transformed back into the physical space.

References

Aguirre, C. (2005). *Dispersión et Mélange Atmosphérique Euléro-lagrangien de Particules Fluides Réactives. Application à des cas simples et complexe*. PhD thesis, Université Claude Bernard Lyon 1.

Almeida, G., Durao, D., and Heitor, M. (1993). Wake flows behind two-dimensional model hills. *Experimental Thermal and Fluid Science*, 7(1):87–101.

Anderson, R. S. and Haff, P. (1991). Wind modification and bed response during saltation of sand in air. In *Aeolian Grain Transport 1*, pages 21–51. Springer.

Anderson, R. S., Sørensen, M., and Willetts, B. B. (1991). A review of recent progress in our understanding of aeolian sediment transport. In *Aeolian Grain Transport 1*, pages 1–19. Springer.

Andreotti, B., Claudin, P., and Douady, S. (2002). Selection of dune shapes and velocities part 1: Dynamics of sand, wind and barchans. *The European Physical Journal B-Condensed Matter and Complex Systems*, 28(3):321–339.

Arakawa, A. (1966). Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. part i. *Journal of Computational Physics*, 1(1):119–143.

Aupoix, B. and Cousteix, J. (1982). Modèles simples de tensions de sous-maille en turbulence homogène isotrope. *La Recherche Aérospatiale*, 4:273–283.

Bagnold, R. A. (1941). *The physics of blown sand and desert dunes. London: Methuen*. Menthuen, London.

Balaras, E. (2004). Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations. *Computers & Fluids*, 33(3):375–404.

Basset, A. B. (1888a). On the motion of a sphere in a viscous liquid. *Philosophical Transactions of the Royal Society of London. A*, 179:43–63.

Basset, A. B. (1888b). *A treatise on hydrodynamics: with numerous examples*, volume 2. Deighton, Bell and Company.

Beladjine, D., Ammi, M., Oger, L., and Valance, A. (2007). Collision process between an incident bead and a three-dimensional granular packing. *Physical Review E*, 75(6):061305.

Berselli, L. C., Iliescu, T., and Layton, W. J. (2005). *Mathematics of large eddy simulation of turbulent flows*. Springer Science & Business Media.

Beyer, R. P. and LeVeque, R. J. (1992). Analysis of a one-dimensional model for the immersed boundary method. *SIAM Journal on Numerical Analysis*, 29(2):332–364.

Boussinesq, J. (1885). Sur la résistance qu'oppose un liquide indéfini en repos sans pesanteur, au mouvement varié d'une sphère solide. *Comptes Rendus de l'Académie des Sciences*, 100:985–937.

Briscolini, M. and Santangelo, P. (1989). Development of the mask method for incompressible unsteady flows. *Journal of Computational Physics*, 84(1):57–75.

Buevich, Y. A. (1966). Motion resistance of a particle suspended in a turbulent medium. *Fluid Dynamics*, 1(6):119–119.

Byun, D. W. (1990). On the analytical solutions of flux-profile relationships for the atmospheric surface layer. *Journal of Applied Meteorology*, 29(7):652–657.

Cao, S. and Tamura, T. (2006). Experimental study on roughness effects on turbulent boundary layer flow over a two-dimensional steep hill. *Journal of Wind Engineering and Industrial Aerodynamics*, 94(1):1–19.

Castro, I. P. (2007). Rough-wall boundary layers: mean flow universality. *Journal of Fluid Mechanics*, 585:469–485.

Charru, F., Andreotti, B., and Claudin, P. (2013). Sand ripples and dunes. *Annual Review of Fluid Mechanics*, 45:469–493.

Charru, F. and Hinch, E. (2006). Ripple formation on a particle bed sheared by a viscous liquid. part 1. steady flow. *Journal of Fluid Mechanics*, 550:111–121.

Cheng, N.-S. and Chiew, Y.-M. (1998). Pickup probability for sediment entrainment. *Journal of Hydraulic Engineering*, 124(2):232–235.

Chepil, W. (1945). Dynamics of wind erosion: Ii. initiation of soil movement. *Soil Science*, 60(5):397.

Clark, R. A., Ferziger, J. H., and Reynolds, W. C. (1979). Evaluation of subgrid-scale models using an accurately simulated turbulent flow. *Journal of Fluid Mechanics*, 91(1):1–16.

Clark, T. L. (1979). Numerical simulations with a three-dimensional cloud model: Lateral boundary condition experiments and multicellular severe storm simulations. *Journal of the Atmospheric Sciences*, 36(11):2191–2215.

Coles, D. (1956). The law of the wake in the turbulent boundary layer. *Journal of Fluid Mechanics*, 1(2):191–226.

Corrsin, S. and Lumley, J. (1956). On the equation of motion for a particle in turbulent fluid. *Applied Scientific Research, Section A*, 6(2-3):114–116.

Creyssels, M., Dupont, P., El Moctar, A. O., Valance, A., Cantat, I., Jenkins, J. T., Pasini, J. M., and Rasmussen, K. R. (2009). Saltating particles in a turbulent boundary layer: experiment and theory. *Journal of Fluid Mechanics*, 625:47–74.

Dandy, D. S. and Dwyer, H. A. (1990). A sphere in shear flow at finite reynolds number: effect of shear on particle lift, drag, and heat transfer. *Journal of Fluid Mechanics*, 216:381–410.

Deardorff, J. W. (1970). A numerical study of three-dimensional turbulent channel flow at large reynolds numbers. *Journal of Fluid Mechanics*, 41(2):453–480.

Deardorff, J. W. (1980). Stratocumulus-capped mixed layers derived from a three-dimensional model. *Boundary-Layer Meteorology*, 18(4):495–527.

DeLeon, R., Sandusky, M., and Senocak, I. (2018). Simulations of turbulent flow over complex terrain using an immersed-boundary method. *Boundary-Layer Meteorology*, 167(3):399–420.

Descamps, I. (2004). *Erosion éolienne d'un lit de particules à large spectre granulométrique*. PhD thesis, Valenciennes.

Diplas, P., Dancey, C. L., Celik, A. O., Valyrakis, M., Greer, K., and Akar, T. (2008). The role of impulse on the initiation of particle movement under turbulent flow conditions. *Science*, 322(5902):717–720.

DuBoys, P. (1879). Le rhone et les rivieres a lit affouillable. In *Annales des Ponts et Chaussees, Series*, volume 5, pages 141–195.

Dupont, S., Bergametti, G., Marticorena, B., and Simoëns, S. (2013). Modeling saltation intermittency. *Journal of Geophysical Research: Atmospheres*, 118(13):7109–7128.

Durán, O., Andreotti, B., and Claudin, P. (2012). Numerical simulation of turbulent sediment transport, from bed load to saltation. *Physics of Fluids*, 24(10):103306.

Durán, O., Claudin, P., and Andreotti, B. (2014). Direct numerical simulations of aeolian sand ripples. *Proceedings of the National Academy of Sciences*, 111(44):15665–15668.

Durán, O. and Herrmann, H. (2006). Modelling of saturated sand flux. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(07):P07011.

Durran, D. R. and Klemp, J. B. (1982). The effects of moisture on trapped mountain lee waves. *Journal of the Atmospheric Sciences*, 39(11):2490–2506.

Einstein, H. A. et al. (1950). *The bed-load function for sediment transportation in open channel flows*, volume 1026. US Department of Agriculture Washington, DC.

Elghobashi, S. (1994). On predicting particle-laden turbulent flows. *Applied scientific research*, 52(4):309–329.

Elhakeem, M., Papanicolaou, A. T., and Tsakiris, A. G. (2017). A probabilistic model for sediment entrainment: The role of bed irregularity. *International Journal of Sediment Research*, 32(2):137–148.

Ernstsen, V. B., Becker, M., Winter, C., Bartholomä, A., Flemming, B. W., and Bartholdy, J. (2007). Bedload transport in an inlet channel during a tidal cycle. *River, Coastal and Estuarine Morphodynamics: RCEM*, pages 351–358.

Fadlun, E., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35–60.

Faxén, H. (1922). Der widerstand gegen die bewegung einer starren kugel in einer zähen flüssigkeit, die zwischen zwei parallelen ebenen wänden eingeschlossen ist. *Annalen der Physik*, 373(10):89–119.

Fernandez Luque, R. and Van Beek, R. (1976). Erosion and transport of bed-load sediment. *Journal of Hydraulic Research*, 14(2):127–144.

Ferreira, A. D. and Fino, M. R. M. (2012). A wind tunnel study of wind erosion and profile reshaping of transverse sand piles in tandem. *Geomorphology*, 139:230–241.

Ferreira, M. C. S. (2017). *Influence of non-erodible particles on aeolian erosion*. PhD thesis, Université Rennes 1.

Fessler, J. R., Kulick, J. D., and Eaton, J. K. (1994). Preferential concentration of heavy particles in a turbulent channel flow. *Physics of Fluids*, 6(11):3742–3749.

Fletcher, B. (1976). The incipient motion of granular materials. *Journal of Physics D: Applied Physics*, 9(17):2471.

Foucaut, J.-M. and Stanislas, M. (1996). Take-off threshold velocity of solid particles lying under a turbulent boundary layer. *Experiments in fluids*, 20(5):377–382.

Gao, T., Tseng, Y.-H., and Lu, X.-Y. (2007). An improved hybrid cartesian/immersed boundary method for fluid–solid flows. *International Journal for Numerical Methods in Fluids*, 55(12):1189–1211.

Gatignol, R. (1983). The faxén formulas for a rigid particle in an unsteady non-uniform stokes-flow. *Journal de Mécanique théorique et appliquée*, 2(2):143–160.

Germano, M., Piomelli, U., Moin, P., and Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765.

Gilmanov, A., Sotiropoulos, F., and Balaras, E. (2003). A general reconstruction algorithm for simulating flows with complex 3d immersed boundaries on cartesian grids. *Journal of Computational Physics*, 191(2):660–669.

Goldstein, D., Handler, R., and Sirovich, L. (1993). Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2):354–366.

Greeley, R. and Iversen, J. (1985). *Wind as a geological process on Earth, Mars, Venus and Titan*.

Grötzbach, G. (1987). Direct numerical and large eddy simulation of turbulent channel flows. *Encyclopedia of Fluid Mechanics*, 6:1337–1391.

Hall, D. (1988). Measurements of the mean force on a particle near a boundary in turbulent flow. *Journal of Fluid Mechanics*, 187:451–466.

Ho, T. D. (2012). *Etude expérimentale du transport de particules dans une couche limite turbulente*. PhD thesis, Université Rennes 1.

Ho, T. D., Valance, A., Dupont, P., and El Moctar, A. O. (2014). Aeolian sand transport: Length and height distributions of saltation trajectories. *Aeolian Research*, 12:65–74.

Hoffmann, P., Muck, K., and Bradshaw, P. (1985). The effect of concave surface curvature on turbulent boundary layers. *Journal of Fluid mechanics*, 161:371–403.

Horiuti, K. (1985). Large eddy simulation of turbulent channel flow by one-equation modeling. *Journal of the Physical Society of Japan*, 54(8):2855–2865.

http://www.idris.fr/formations (MPI). *Training Course Materials, Institut for Development and Resources in Intensive Scientific Computing*.

Huang, G. (2015). *Numerical simulation of solid particle transport in atmospheric boundary-layer over obstacles*. PhD thesis, Ecole Centrale de Lyon.

Huang, G., Le Ribault, C., Vinkovic, I., and Simoëns, S. (2018). Part i: a priori study of erosion and deposition with large eddy simulation of turbulent flow over multiple 2d sandy gaussian hills. *Environmental Fluid Mechanics*, pages 1–29.

Huang, G., Simoëns, S., Vinkovic, I., Le Ribault, C., Dupont, S., and Bergametti, G. (2016). Law-of-the-wall in a boundary-layer over regularly distributed roughness elements. *Journal of Turbulence*, 17(5):518–541.

Hussain, M. and Lee, B. (1980). A wind tunnel study of the mean pressure forces acting on large groups of low-rise buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, 6(3-4):207–225.

Iaccarino, G. and Verzicco, R. (2003). Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Reviews*, 56(3):331–347.

Ibrahim, A., Dunn, P., and Qazi, M. (2008). Experiments and validation of a model for microparticle detachment from a surface by turbulent air flow. *Journal of Aerosol Science*, 39(8):645–656.

Jiménez, J. (2004). Turbulent flows over rough walls. *Annual Review of Fluid Mechanics*, 36:173–196.

Kamphuis, J. (1974). Determination of sand roughness for fixed beds. *Journal of Hydraulic Research*, 12(2):193–203.

Kanda, I., Yamao, Y., Uehara, K., and Wakamatsu, S. (2013). Particle-image velocimetry measurements of separation and re-attachment of airflow over two-dimensional hills with various slope angles and approach-flow characteristics. *Boundary-layer meteorology*, 148(1):157–175.

Kang, L., Guo, L., and Liu, D. (2008). Reconstructing the vertical distribution of the aeolian saltation mass flux based on the probability distribution of lift-off velocity. *Geomorphology*, 96(1-2):1–15.

Khadra, K., Angot, P., Parneix, S., and Caltagirone, J.-P. (2000). Fictitious domain approach for numerical modelling of navier–stokes equations. *International Journal for Numerical Methods in Fluids*, 34(8):651–684.

Klemp, J. B. and Durran, D. R. (1983). An upper boundary condition permitting internal gravity wave radiation in numerical mesoscale models. *Monthly Weather Review*, 111(3):430–444.

Klemp, J. B. and Wilhelmson, R. B. (1978). The simulation of three-dimensional convective storm dynamics. *Journal of the Atmospheric Sciences*, 35(6):1070–1096.

Kok, J. F. and Renno, N. O. (2009). A comprehensive numerical model of steady state saltation (comsalt). *Journal of Geophysical Research: Atmospheres*, 114(D17).

Kuzkin, V. A. (2015). On angular momentum balance for particle systems with periodic boundary conditions. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 95(11):1290–1295.

Lai, M.-C. and Peskin, C. S. (2000). An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160(2):705–719.

Lancaster, N. (2011). Desert dune processes and dynamics. *Arid Zone Geomorphology: Process, Form and Change in Drylands*, pages 487–515.

Lavelle, J. W. and Mofjeld, H. O. (1987). Do critical stresses for incipient motion and erosion really exist? *Journal of Hydraulic Engineering*, 113(3):370–385.

Le, H., Moin, P., and Kim, J. (1997). Direct numerical simulation of turbulent flow over a backward-facing step. *Journal of Fluid Mechanics*, 330:349–374.

Legendre, D. and Magnaudet, J. (1998). The lift force on a spherical bubble in a viscous linear shear flow. *Journal of Fluid Mechanics*, 368:81–126.

Leighton, D. and Acrivos, A. (1985). The lift on a small sphere touching a plane in the presence of a simple shear flow. *Zeitschrift für Angewandte Mathematik und Physik ZAMP*, 36(1):174–178.

Lesieur, M. (1987). *Turbulence in fluids: stochastic and numerical modelling*. Nijhoff Boston, MA.

Lettau, K. (1978). Experimental and micrometeorological field studies of dune migration. *Exploring in the World's Driest Climate*, pages 110–147.

Liao, C. C., Chang, Y. W., Lin, C. A., and McDonough, J. (2010). Simulating flows with moving rigid boundary using immersed-boundary method. *Computers & Fluids*, 39(1):152–167.

Ligrani, P. M. and Moffat, R. J. (1986). Structure of transitionally rough and fully rough turbulent boundary layers. *Journal of Fluid Mechanics*, 162:69–98.

Lilly, D. (1981). Wave-permeable lateral boundary conditions for convective cloud and storm simulations. *Journal of the Atmospheric Sciences*, 38(6):1313–1316.

Lopes, A., Oliveira, L., Ferreira, A. D., and Pinto, J. (2013). Numerical simulation of sand dune erosion. *Environmental Fluid Mechanics*, 13(2):145–168.

Loth, E. (2008). Lift of a spherical particle subject to vorticity and/or spin. *AIAA Journal*, 46(4):801–809.

Lund, T. S., Wu, X., and Squires, K. D. (1998). Generation of turbulent inflow data for spatially-developing boundary layer simulations. *Journal of Computational Physics*, 140(2):233–258.

Lundquist, K. A., Chow, F. K., and Lundquist, J. K. (2010). An immersed boundary method for the weather research and forecasting model. *Monthly Weather Review*, 138(3):796–817.

Mason, P. and Callen, N. (1986). On the magnitude of the subgrid-scale eddy coefficient in large-eddy simulations of turbulent channel flow. *Journal of Fluid Mechanics*, 162:439–462.

Maxey, M. R. and Riley, J. J. (1983). Equation of motion for a small rigid sphere in a nonuniform flow. *The Physics of Fluids*, 26(4):883–889.

Mei, R. (1992). An approximate expression for the shear lift force on a spherical particle at finite reynolds number. *International Journal of Multiphase Flow*, 18(1):145–147.

Meneveau, C. (1994). Statistics of turbulence subgrid-scale stresses: Necessary conditions and experimental tests. *Physics of Fluids*, 6(2):815–833.

Miller, M. and Thorpe, A. (1981). Radiation conditions for the lateral boundaries of limited-area numerical models. *Quarterly Journal of the Royal Meteorological Society*, 107(453):615–628.

Mittal, R., Dong, H., Bozkurttas, M., Najjar, F., Vargas, A., and von Loebbecke, A. (2008). A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10):4825–4852.

Mittal, R. and Iaccarino, G. (2005). Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261.

Moeng, C.-H. (1984). A large-eddy-simulation model for the study of planetary boundary-layer turbulence. *Journal of the Atmospheric Sciences*, 41(13):2052–2062.

Moeng, C.-H. and Wyngaard, J. C. (1989). Evaluation of turbulent transport and dissipation closures in second-order modeling. *Journal of the Atmospheric Sciences*, 46(14):2311–2330.

Mohd-Yusof, J. (1997). Combined immersed boundary/b-spline methods for simulation of flow in complex geometries. *Annual Research Briefs, Center for Turbulence Research*, pages 317–328.

Moin, P. and Kim, J. (1982). Numerical investigation of turbulent channel flow. *Journal of Fluid Mechanics*, 118:341–377.

Mollinger, A. and Nieuwstadt, F. (1996). Measurement of the lift force on a particle fixed to the wall in the viscous sublayer of a fully developed turbulent boundary layer. *Journal of Fluid Mechanics*, 316:285–306.

Muck, K., Hoffmann, P., and Bradshaw, P. (1985). The effect of convex surface curvature on turbulent boundary layers. *Journal of Fluid Mechanics*, 161:347–369.

Nabi, M., Vriend, d. H., Mosselman, E., Sloff, C., and Shimizu, Y. (2012). Detailed simulation of morphodynamics: 1. hydrodynamic model. *Water Resources Research*, 48(12).

Nikuradse, J. (1933). Stromungsgesetze in rauhen rohren. *VDI-Forschungsheft*, 361:1292.

Oke, T. R. (1988). Street design and urban canopy layer climate. *Energy and buildings*, 11(1-3):103–113.

O'Neil, J. and Meneveau, C. (1997). Subgrid-scale stresses and their modelling in a turbulent plane wake. *Journal of Fluid Mechanics*, 349:253–293.

Orlanski, I. (1976). A simple boundary condition for unbounded hyperbolic flows. *Journal of Computational Physics*, 21(3):251–269.

Oseen, C. W. (1927). *Hydrodynamik*, volume 1. Leipzig: Akademische Verlagsgesellschaft.

Ouchene, R., Polanco, J. I., Vinkovic, I., and Simoëns, S. (2018). Acceleration statistics of prolate spheroidal particles in turbulent channel flow. *Journal of Turbulence*, 19(10):827–848.

Owen, P. R. (1964). Saltation of uniform grains in air. *Journal of Fluid Mechanics*, 20(2):225–242.

Peller, N., Duc, A. L., Tremblay, F., and Manhart, M. (2006). High-order stable interpolations for immersed boundary methods. *International Journal for Numerical Methods in Fluids*, 52(11):1175–1193.

Peskin, C. S. (1972). Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10(2):252–271.

Peskin, C. S. (1977). Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252.

Picciotto, M., Marchioli, C., and Soldati, A. (2005). Characterization of near-wall accumulation regions for inertial particles in turbulent boundary layers. *Physics of Fluids*, 17(9):098101.

Pope, S. and Pope, S. B. (2000). *Turbulent Flows*. Cambridge University Press.

Quartier, L., Andreotti, B., Douady, S., and Daerr, A. (2000). Dynamics of a grain on a sandpile model. *Physical Review E*, 62(6):8299.

Raupach, M., Antonia, R., and Rajagopalan, S. (1991). Rough-wall turbulent boundary layers. *Applied Mechanics Reviews*, 44(1):1–25.

Raupach, M., McTainsh, G., and Leys, J. (1994). Estimates of dust mass in recent major australian dust storms. *Australian Journal of Soil and Water Conservation*, 7:20–24.

Raupach, M., Thom, A., and Edwards, I. (1980). A wind-tunnel study of turbulent flow close to regularly arrayed rough surfaces. *Boundary-Layer Meteorology*, 18(4):373–397.

Rice, M. A., Willetts, B. B., and McEwan, I. (1995). An experimental study of multiple grain-size ejecta produced by collisions of saltating grains with a flat bed. *Sedimentology*, 42(4):695–706.

Rice, M. A., Willetts, B. B., and McEwan, I. (1996). Observations of collisions of saltating grains with a granular bed from high-speed cine-film. *Sedimentology*, 43(1):21–31.

Roache, P. J. (1972). *Computational fluid dynamics*. Hermosa publishers.

Rouson, D. W. and Eaton, J. K. (2001). On the preferential concentration of solid particles in turbulent channel flow. *Journal of Fluid Mechanics*, 428:149–169.

Saffman, P. (1965). The lift on a small sphere in a slow shear flow. *Journal of Fluid Mechanics*, 22(2):385–400.

Saiki, E. and Biringen, S. (1996). Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method. *Journal of Computational Physics*, 123(2):450–465.

Sauermann, G., Kroy, K., and Herrmann, H. J. (2001). Continuum saltation model for sand dunes. *Physical Review E*, 64(3):031305.

Schumann, U. (1975). Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *Journal of Computational Physics*, 18(4):376–404.

Shao, Y. (2008). *Physics and modelling of wind erosion*, volume 37. Springer Science & Business Media.

Shao, Y. and Lu, H. (2000). A simple expression for wind erosion threshold friction velocity. *Journal of Geophysical Research: Atmospheres*, 105(D17):22437–22443.

Shields, A. (1936). *Anwendung der Aehnlichkeitsmechanik und der Turbulenzforschung auf die Geschiebebewegung*. PhD thesis, Technical University Berlin.

Simoens, S., Ayrault, M., and Wallace, J. M. (2007). The flow across a street canyon of variable width—part 1: Kinematic description. *Atmospheric Environment*, 41(39):9002–9017.

Simoëns, S., Saleh, A., Leribault, C., Belhmadi, M., Zegadi, R., Allag, F., Vignon, J., and Huang, G. (2015). Influence of gaussian hill on concentration of solid particles in suspension inside turbulent boudary layer. *Procedia IUTAM*, 17:110–118.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review*, 91(3):99–164.

Soldati, A. and Marchioli, C. (2009). Physics and modelling of turbulent particle deposition and entrainment: Review of a systematic study. *International Journal of Multiphase Flow*, 35(9):827–839.

Sørensen, M. (1991). An analytic model of wind-blown sand transport. *Acta Mechanica*, pages 67–81.

Spalart, P. R. (1988). Direct simulation of a turbulent boundary layer up to r θ= 1410. *Journal of Fluid Mechanics*, 187:61–98.

Squires, K. D. and Eaton, J. K. (1991). Preferential concentration of particles by turbulence. *Physics of Fluids A: Fluid Dynamics*, 3(5):1169–1178.

Stokes, G. G. (1851). *On the effect of the internal friction of fluids on the motion of pendulums*, volume 9. Cambridge University Press.

Stull, R. B. (1988). *An Introduction to Boundary Layer Meteorology*, volume 13. Springer Science & Business Media.

Taniere, A., Oesterle, B., and Monnier, J. (1997). On the behaviour of solid particles in a horizontal boundary layer with turbulence and saltation effects. *Experiments in Fluids*, 23(6):463–471.

Tchen, C. (1947). *Mean value and correlation problems connected with the motion of small particles suspended in a turbulent fluid*. PhD thesis, Delft University of Technology.

Tseng, Y.-H. and Ferziger, J. H. (2003). A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623.

Ungar, J. E. and Haff, P. (1987). Steady state saltation in air. *Sedimentology*, 34(2):289–299.

Valyrakis, M., Diplas, P., and Dancey, C. L. (2013). Entrainment of coarse particles in turbulent flows: An energy approach. *Journal of Geophysical Research: Earth Surface*, 118(1):42–53.

Vinçont, J.-Y., Simoens, S., Ayrault, M., and Wallace, J. (2000). Passive scalar dispersion in a turbulent boundary layer from a line source at the wall and downstream of an obstacle. *Journal of Fluid Mechanics*, 424:127–167.

Vinkovic, I. (2005). *Dispersion et mélange turbulents de particules solides et de gouttelettes par une simulation des grandes échelles et une modélisation stochastique lagrangienne. Application à la pollution de l'atmosphère.* PhD thesis, Ecole Centrale de Lyon.

Vinkovic, I., Aguirre, C., Ayrault, M., and Simoëns, S. (2006). Large-eddy simulation of the dispersion of solid particles in a turbulent boundary layer. *Boundary-Layer Meteorology*, 121(2):283.

Vinkovic, I., Doppler, D., Lelouvetel, J., and Buffat, M. (2011). Direct numerical simulation of particle interaction with ejections in turbulent channel flows. *International Journal of Multiphase Flow*, 37(2):187–197.

Wang, D., Wang, Y., Yang, B., Zhang, W., and Lancaster, N. (2008). Statistical analysis of sand grain/bed collision process recorded by high-speed digital camera. *Sedimentology*, 55(2):461–470.

White, B. (1982). Two-phase measurements of saltating turbulent boundary layer flow. *International Journal of Multiphase Flow*, 8(5):459–473.

White, C. (1940). The equilibrium of grains on the bed of a stream. *Proceedings of the Royal Society of London Series A*, 174(958):322–338.

Wiggs, G. and Weaver, C. (2012). Turbulent flow structures and aeolian sediment transport over a barchan sand dune. *Geophysical Research Letters*, 39(5).

Wiggs, G. F. (2001). Desert dune processes and dynamics. *Progress in Physical Geography*, 25(1):53–79.

Willetts, B. and Rice, M. (1986). Collisions in aeolian saltation. *Acta Mechanica*, 63(1-4):255–265.

Williams, J. J. (1986). *Aeolian entrainment thresholds in a developing boundary layer*. PhD thesis, Queen Mary University of London.

Wippermann, F. and Gross, G. (1986). The wind-induced shaping and migration of an isolated dune: a numerical experiment. *Boundary-Layer Meteorology*, 36(4):319–334.

Wu, F.-C. and Lin, Y.-C. (2002). Pickup probability of sediment under log-normal velocity distribution. *Journal of Hydraulic Engineering*, 128(4):438–442.

Wu, J., Le, R., Vinkovic, I., Simoëns, S., et al. (2017). Large-eddy simulation for moving terrain applied to sand dune displacement. In *23ème Congrès Français de Mécanique*. Association Française de Mécanique.

Xue, M., Droegemeier, K., Wong, V., Shapiro, A., and Brewster, K. (1995). Advanced regional prediction system (arps) version 4.0 user's guide. *Center for Analysis and Prediction of Storms, University of Oklahoma*, 380.

Xue, M., Droegemeier, K. K., and Wong, V. (2000). The advanced regional prediction system (arps)–a multi-scale nonhydrostatic atmospheric simulation and prediction model. part i: Model dynamics and verification. *Meteorology and Atmospheric Physics*, 75(3-4):161–193.

Xue, M., Droegemeier, K. K., Wong, V., Shapiro, A., Brewster, K., Carr, F., Weber, D., Liu, Y., and Wang, D. (2001). The advanced regional prediction system (arps)–a multi-scale nonhydrostatic atmospheric simulation and prediction tool. part ii: Model physics and applications. *Meteorology and Atmospheric Physics*, 76(3-4):143–165.

Yang, J. and Balaras, E. (2006). An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics*, 215(1):12–40.

Yoshizawa, A. (1982). A statistically-derived subgrid model for the large-eddy simulation of turbulence. *The Physics of Fluids*, 25(9):1532–1538.

Yoshizawa, A. and Horiuti, K. (1985). A statistically-derived subgrid-scale kinetic energy model for the large-eddy simulation of turbulent flows. *Journal of the Physical Society of Japan*, 54(8):2834–2839.

Yu, W. (2015). *Finite size particle transport in turbulent channel flow*. PhD thesis, Université Claude Bernard Lyon 1.

Zeng, L., Najjar, F., Balachandar, S., and Fischer, P. (2009). Forces on a finite-sized particle located close to a wall in a linear shear flow. *Physics of Fluids*, 21(3):033302.

Zimon, A. D. et al. (1969). Adhesion of dust and powder.

Zingg, A. (1953). Wind tunnel studies of the movement of sedimentary material. In *Proceedings of the 5th Hydraulic Conference Bulletin*, volume 34, pages 111–135. Inst. of Hydraulics Iowa City.