COUPLED FLUID MODELS OF SMOOTHED PARTICLE HYDRODYNAMICS AND FINITE DIFFERENCE METHOD FOR SIMULATING DYNAMIC PENGUIN HUDDLES

by

WEN GU

(Under the Direction of Jason Christian)

ABSTRACT

A coupled numeric model of finite difference method (FDM) and smoothed particle hydrodynamic (SPH) is utilized for the simulation of dynamic penguin huddles. In this coupled fluid model, the full Navier-Stokes equations are used to solve a wind field using a finite difference method and simultaneously model penguin huddling through a smoothed particle hydrodynamics method. The FDM method is a common Eulerian numerical approach based on application of a local Taylor expansion and is used to estimate wind flowing in two dimensions around complex and dynamic huddle shape on a rectangular computational grid. The SPH method is a mesh-free Lagrangian method driven by local interactions between neighboring fluid particles and their environment allowing particles to act as free ranging "penguins" unconstrained by a computational grid. These coupled fluid numerical models are recomputed simultaneously as the huddle evolves over time to update individual particle positions, redefine the fluid properties of the developing huddle (i.e., shape and density), and redefine the wind field flowing through and around the dynamic huddle. This study shows the ability of a coupled model to predict the dynamic properties of penguin huddles, to quantify biometrics of individual particle "penguins" and to attempt an explanation of communal penguin huddling behavior as observed in nature.

INDEX WORDS: Coupled fluid models, Finite difference models, Smoothed particle hydrodynamics, Penguin huddling, Navier-Stokes Equation

COUPLED FLUID MODELS OF SMOOTHED PARTICLE HYDRODYNAMICS AND FINITE DIFFERENCE METHOD FOR SIMULATING DYNAMIC PENGUIN HUDDLE

by

WEN GU

B.E., Ocean University of China, China, 2010

M.S., University of Florida, 2012

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2018

COUPLED FLUID MODELS OF SMOOTHED PARTICLE HYDRODYNAMICS AND FINITE DIFFERENCE METHOD FOR SIMULATING DYNAMIC PENGUIN HUDDLE

by

WEN GU

Major Professor:	Jason Christian
Committee:	Catherine Edwards
	David Stooksbury
	Brock Woodson

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
May 2018

ACKNOWLEDGEMENTS

First and foremost, I want to deeply thank my adviser Jason Christian. It has been an honor to be his first Ph.D. student. Dr. Christian provide me with every bit of guidance, assistance, and expertise that I needed during my first few semesters; then, when I felt ready to venture into research on my own and branch out into new research areas, Dr. Christian gave me the freedom to do whatever I wanted, at the same time continuing to contribute valuable feedback, advise, and encouragement. In addition to our academic collaboration, I greatly value the close personal rapport that Dr. Christian and I have forged over the years. I quite simply cannot imagine a better adviser.

I would like to thank my committee members at the University of Georgia, Dr. David Stooksbury, Dr. Brock Woodson and Dr. Catherine Edwards, for the influence that their courses and great comments have had on my research. I gratefully acknowledge them for their time and valuable feedback on a preliminary version of this thesis.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Fluid dynamic models provide a powerful deterministic technique in a variety of application areas. Fluid mechanics models have been employed to characterize many ecological problems, such as animal locomotion when animals move through a fluid (Peskin. 1988), and even non-fluid problems, such as transportation (Chakraborty and Srinivasan. 2016) and dynamic pricing (Kachani and Perakis. 2006). These dynamic phenomena in diverse application domains share similar characteristics that mimic fluid properties. Experimentally, fluid model analysis can be an effective way to provide insights and understandings into the nature of dynamic phenomena existing outside traditional fluid disciplines (Kachani and Perakis. 2006).

Huddling behavior in penguins is an example of a biological phenomenon that may lend itself to modeling as a fluid. The formation and movement of huddles share similar properties to fluids including having deformable boundaries and variable density. In the application to penguin huddles, both shape and density of the huddle are responsive to environmental inputs of wind, temperature, and physical obstacles. Because of these fluid properties of penguin huddles, it is theorized that fluid mechanic models would be particularly well suited for simulating this communal behavior. Emperor penguins (*Aptenodytes forsteri*) are the only birds that breed during the Antarctic winter on fast-ice (sea ice that is "fastened" to the coastline) and are known to huddle (an active and close aggregation of animals) through extended stressful periods without food in the severe conditions of Antarctica winters. Huddling allows penguins to minimize heat loss, lower their energy expenditure and reallocate the saved energy to other functions by thermoregulation

(Gilbert, et al. 2010). Huddles are discontinuous events corresponding to severe storm events and the density in a huddle at a colony may be as high as 10 bird/m^2 (Waters et al. 2012). Biologists observed that penguins rotate positions constantly when they are exposed to the wind during huddling, although the moving penguins remained together as single unit and the huddle as a whole appears semi-static (Gilbert et al. 2006). This repositioning of individual penguins has been well documented through observation, but the status of individual penguins (i.e. how warm or cold they are individually and how their positions within the huddle change over time) has been difficult to assess and quantify in the field.

There are few numerical studies on penguins' huddling. Waters et al. (2012) suggested that a model of Penguins' huddling could be established based on fluid dynamics using the Navier-Stokes advection-diffusion equation to quantify the temperature profile around the huddle. There are several numerical methods for solving the Navier-Stokes equation, including the Finite Difference Method (FDM), the Finite Element Method (FEM), the Finite Volume Method (FVM) and Smoothed Particle Hydrodynamics (SPH). FDM is a Eulerian numeric method based on the application of a local Taylor expansion to approximate the differential equation. It is easy to implement and optimizes the approximation for the differential operator of the considered patch. More recently, a meshless Smoothed Particle Hydrodynamics (SPH) method has developed and become widely adopted in modeling of dynamic fluid systems. SPH is a Lagrangian particle method that is well suited to transient fluid dynamics problems that involve complex free surface behavior and moving and deforming boundaries of complicated shape. In this study, FDM was applied to simulate two-dimensional wind field, and simultaneously SPH was applied to simulate penguin huddling.

CHAPTER 2

LITERATURE REVIEW

2.1 Reviews on Huddling

Huddling can be defined as active aggregation of individuals to share benefits from the warmth of colony, and is a very important social behavior for birds and mammals to survive low temperature environments (Gilbert et al. 2010; Ancel et al. 2015). Through huddling, penguins can maximize energy saving by reducing their body surface area exposed to the cold and decreasing individual heat loss through warming local ambient temperature. The local ambient temperature is raised as the heat released by individual penguins accumulates within huddles, and the benefits can be shared by all penguins in the huddling group (Haig. 2008; Ancel et al. 2015). Gilbert et al. (2006) suggested that all individuals may get the same benefits from huddling; therefore, penguins seek to join or leave huddle based on their individual need of warmth.

Huddling behavior in social animals is considered by biologists to be a self-organized system (simple local interactions between individual animals collectively give rise to a complex group-level behavior) coupled with an external driving parameter (Canals and Bozinovic. 2011). Several studies investigated the effects of thermoregulatory huddling behaviors by mammals, birds and other social animals, and suggested that this self-organizing huddling behavior may guide natural selection (Glancy et al. 2016; Wilson. 2017; Canals and Bozinovic. 2011). There are several studies on mammals' huddling behaviors to quantify the ecological and physiological benefits of these behaviors, including bats (Boratynski et al. 2014), mice (Eto et al. 2014), rats (Schank and Alberts. 1997) and other social rodents (Sanchez and Solis. 2015). One of the major

benefits of huddling comes through a reduction of individual energy expenditure. Therefore, individual survival is increased, required food intake is lowered, reduction in body mass is decreased, growth rate is increased, water loss is decreased, and metabolic rate is significantly reduced (Hayes et al. 1992; Willis and Brigham. 2007; Gilbert et al. 2010). By establishing numerical model of this group behavior, the status of individuals can be quantified, benefits to individual animal can be estimated, and the dynamics of social interactions, group behavior and developmental changes can be investigated by biologists (Schank and Alberts. 1997).

Observations and simulations are the main methods to study penguin huddles. The average breeding male emperor penguin weighs from 22 to 45 kg and is 122 cm in height. The females lay eggs, which are incubated by the males while the females return to the sea to feed. During the 4-month fast associated with pairing and egg incubation, male emperor penguins rely completely on their body reserves of fat to fuel ongoing metabolism. The warmth created inside huddles is a necessary mechanism explaining energy savings observed in huddling emperor penguins (Gilbert et al. 2008). By close packing in tight huddles, penguins' cold-exposed body surface can be reduced greatly as penguins inside the huddle are shielded from wind. Also, the combined heat loss of all huddling penguins increases local ambient temperature and helps stabilize huddles within this comfortable TNZ (Gilbert et al. 2007).

The dynamic of huddles is related to environmental conditions. Previous observations showed how external conditions affect penguins, suggesting that huddling behavior is enhanced by unfavorable weather conditions, such as low ambient temperature and high wind speed (Gilbert and Robertson. 2008). A series of environmental factors were tested for their influence on huddle patterns, such as the number of huddles and mean number of individuals per huddle. Both the number of huddles and the number of individuals per huddle increased when air temperature

decreased. When wind speed increased, or solar radiation decreased, the number of huddles decreased, and huddles got larger (Ancel et al. 2015).

Determining individual benefits for emperor penguins' huddling is a significant challenge as almost all their colonies are difficult to observe given their remote location and forbidding environment. In 2012, the first global census of penguins was published based on an automated analysis of satellite images by the British Antarctic Survey (Fretwell et al. 2012). Previous studies show that males are huddled on average 38% of the time (Gilbert et al. 2006), and experience ambient temperatures within their comfortable thermo-neutral zone (TNZ), which varies from -10 to 20 ºC (Gilbert et al. 2008). To understand huddling as a key behavior enabling penguins to survive the long winter fast, new models are needed to identify individual benefits of this group behavior. Internal environmental conditions of penguin huddles and how they interact with external conditions, such as changes in wind speed direction, ambient temperature, and wind chill affecting huddling behaviors should be quantified to achieve full understanding of huddle dynamics.

There are few theoretical models on penguins' huddling. From previous observations, the movement of penguin huddles is slow but continuous (Le Maho. 1977). The observation of a medium-size Emperor penguin colony (~2000 animals) near the Neumayer Antarctic Research Station (70º39S 8º15W) with air temperature varying from -33 to -43ºC and a maximum 8.3 m/s wind speed showed that penguin huddle may be a self-organizing event. Penguins move small steps and the small movements lead to large-scale reorganization of the huddle (Zitterbart et al. 2011). Additionally, there are several studies on other animal huddle behavior to investigate how huddling behavior is triggered by low temperatures as a self-organizing event (Canals and Bozinovic. 2011; Wilson. 2017).

Biologists attributed heterogeneity of the huddle shape to ensuring penguins have equal access to warm center, but without providing details about how equality is achieved (Gilbert et al. 2006). A systematic and quantitative mathematical model on penguin huddles was established by Waters et al. (2012) based on the warmth equality assumption (each individual penguin in the huddle seeks only to reduce its own heat loss). They used fluid dynamics equations to calculate wind flow and temperature around the huddles, with each penguin moving or staying stationary to minimize individual heat loss. Penguins on the huddle boundary were assumed to lose more heat because they were exposed to a large temperature gradient due to wind chill. However, the model presented was simplistic and potentially misleading as penguins' movement is not replicated as observed in nature. In their model, only the penguins on the leading edge of the huddle were relocated behind the huddle, while the penguins in the interior of the huddle didn't move. Additionally, the status of individual penguin is not quantified in this previous model (Waters et al. 2012).

2.2 Reviews on Finite Difference Method (FDM)

The Finite Difference Method is a Eulerian numeric solution of differential equations derived from Taylor series expansions. It provides solutions at discrete points in a continuum domain as grid points and is the most commonly used numerical method because of its computational efficiency and relative ease of implementation. Finite difference solutions for solving parabolic partial differential equations can be classified as explicit or implicit. An explicit scheme predicts all quantities at a new advanced time step from known values at the current time step, whereas implicit schemes require a system of equations including the boundary condition equations to be solved simultaneously to determine quantities at each new advanced time step (Jeppson. 1972). Truncation error reflects the fact that a finite part of a Taylor series is used in the

approximation. Central difference, back and front difference approximation are used in this model and have a second order truncation error.

FDM models are widely used and have been one of the dominant numerical methods in the simulation of engineering applications. They require a computational mesh across the problem domain and can present stability problems with complex or dynamic boundary problems. Recently, coupled models with FDM and meshless method (Zhang et al. 2018), or meshless generalized finite difference method (Gu et al. 2017) have been shown to reduce problems associated with boundary or domain meshing.

In computational fluid model analysis, FDM is commonly used to solve the Navier-Stokes equation. A challenge solving the Navier-Stokes Equation is that no explicit equation can be used for pressure of flow. Therefore, for most finite difference solution for incompressible flow, the pressure field is obtained from a pressure Poisson equation, which is derived from momentum equation and continuity equation (Rusli et al. 2011).

2.3 Reviews on Smoothed Particle Hydrodynamics (SPH) Method

Numerical methods are indispensable in successful simulation of physical problems. As mentioned earlier, conventional mesh-based numerical methods, such as the Finite Element Method (FEM) and Finite Volume Method (FVM), require connectivity of the nodes, give solutions only at nodal points, and remeshing is usually required when dealing with large boundary deformation problems (Fries and Matthies. 2004). Fries and Matthies (2004) provided a thorough review of all meshless methods that do not include these mesh-based constraints. Several features of mesh-free methods were summarized as:

1) Mesh is absent, which means no mesh generation at the beginning of the

calculation and no remeshing during simulations.

2) The requirement on conservation of mass is readily fulfilled, and the shape functions (the function which interpolates the solution between the discrete values obtained at the mesh nodes) may easily be constructed.
3) Mesh-free methods are usually more computationally expensive as solution of shape functions are more complex than mesh-based methods.

The Smoothed Particle Hydrodynamics (SPH) method is a mesh-free method for solving the Navier-Stokes equations in fluid mechanics. It is a Lagrangian model that uses motions of discrete particles to represent a flow of bulk fluid and to solve the continuum hydrodynamic equations (Hosseini and Feng. 2011, Ellero. et al., 2007). These hydro-particles carry their own fluid physical quantities, such as mass, volume, temperature, velocity and density, and the interaction with other particles is defined by a kernel function. This method was introduced initially to solve astrophysical problems where Eulerian mesh-based methods were insufficient. The SPH method has been extensively applied in science and engineering fields, especially in fluid mechanics and solid mechanics, as it provides advantages compared with both element-based and grid-based numerical methods.

Because it is a mesh-free method, SPH inherently solves many of the issue caused by the grid-based modeling, such as avoiding problems relating to mesh distortion and tracing convection-diffusion terms (i.e., numeric solutions become unstable or oscillatory) (Fuller. 2010; Gu and Liu. 2006). Unlike mesh-based methods, there is unstructured and dynamic topological connectivity between traditional SPH particles, as they interact with all the neighbors according to their radial separation mediated by an interpolating kernel function. This particle-nature shows advantage for multi-phase fluid problems as each particle can be assigned to a different phase.

Similarly, interfacial or free-surface flows do not require the explicit tracking of the interface or free-surface of interest, as it will be implicitly defined by the positions of the particles (Fries and Matthies. 2003). Therefore, SPH shows great potential in solving problems with complex geometries, moving bodies, flexible boundaries or large deformation problems, such as surface erosion by impact (Dong et al. 2016; Leroch et al. 2015; Takaffoli and Papini. 2012; Wang and Yang. 2009), flow-like landslide propagation by earthquake (Dai et al. 2014), debris flows and avalanches (McDougall and Hungr. 2004), Lahars (a destructive mudflow on the slopes of a volcano) (Haddad et al. 2010), geo-disasters (Huang and Dai. 2014), machining process (Spreng and Eberhard. 2015) , strong explosions (Sirotkin and Yoh. 2013), shear panel dampers (Chen et al. 2013), metal manufacturing (Cleary et al. 2006), rock caving (Karekal et al. 2011), rock shapes on brittle fracture (Das and Cleary. 2010), buried structures (Lu et al. 2005), liquifaction (Huang et al. 2011), and military applications including quantifying shape charge jet (Feng et al. 2013). Additionally, SPH modeling has been widely explored on many hydraulic applications, such as free surface flows (Shao et al. 2012; Amicarelli et al. 2015; Saunder et al. 2014), viscous flow (Takeda et al. 1994), high velocity impacts (Randles and Libersky. 1996), geophysical flows (Shao and Lo. 2003), and multi-phase flows (Hu and Adams. 2007; Tartakovsky and Meakin. 2005; Fries and Matthies. 2003).

In fluid mechanics, SPH has been used in solving the Navier-Stokes Equation (NSE), such as propagation of solitary waves in shallow water over varying bottom geometries (Li et al. 2012). Forces on individual fluid particles can be calculated by the information carried from particles and interpolation kernel helps to get the smoothed approximations to the physical properties of the domain from particle information (Fries and Matthies. 2003). Recently, several studies applied SPH in the biomechanics field, such as penetrating impacts on ballistic gelatin (Taddei et al. 2015),

as well as aquatic sports context to simulate elite swimming and dolphin kicks (Cleary et al. 2013; Cohen et al. 2009; Cohen et al. 2010; Cohen et al. 2015; Cohen et al. 2012).

Despite these remarkable numeric properties, SPH also has some disadvantages. To achieve satisfactory resolution accuracy and stability, SPH requires high demands for CPU time with a corresponding large number of particles. There is much less information in the literature available on SPH errors and instabilities for disordered particles than mesh-based methods (Monaghan. 2005). However, this resource-intensive issue is likely to see major improvement in the near future due to the increased availability of high-end computing capability and better coding. Recently, a parallel version of the SPH which takes full advantage of widely used multi-core resources GPU version of SPH employed in popular graphic processing techniques has been developed and proved highly efficient in certain simulations (Li, et al. 2012). There are several successful examples on GPU-based SPH method, such as simulations of automotive fuel cell sloshing (Longshaw and Rogers. 2015), simulations of free-surface flows (Valdez-Balderas et al. 2013), and solving shallow-water equations (Xia and Liang. 2016).

CHAPTER 3

RESEARCH METHOD AND MODEL SET-UP

In this chapter, governing equation, basic equations of the model and model set-up are discussed. The SPH method is a mesh-free method that solves the Navier-Stokes equations in fluid mechanics. It uses a system of Lagrangian particles to represent a flow of bulk fluid through solution of the continuum hydrodynamic equations (Hosseini And Feng. 2011; Ellero et al. 2007). These hydro-particles carry their own physical quantities, such as mass, volume, temperature, velocity and density, and the interaction with other particles is defined by a kernel function. For its advantages of Lagrangian, meshless, and particle properties, SPH has been extensively applied in fluid dynamics. In contrast, the FDM Euler grid-based method is based on the application of a local Taylor expansion to approximate the non-linear differential fluid equations. FDM is comparatively easy to implement and has a relatively low computational cost. A coupled method with SPH and FDM is established for penguin huddling simulation, with the "penguins" resolved by SPH particles and wind fields solved with FDM. These two numeric solutions are coupled through ambient temperature and position of penguins at sufficient time intervals, making it possible to develop an algorithm that retains the best properties from each numerical solution approach when simulating dynamic penguin huddling.

For this study, the penguin huddle is assumed to be situated on a flat plane with dimensions of 50m x 100m and no obstacles impeding penguin movement other than other penguins (Fig. 1). Penguins are initially randomly located in this flat plane and will interact with the wind field according to their distribution across the finite grids. The wind-field in this study is calculated in

two dimensions, but wind only enters from the *Y* direction. Initially, wind is assumed to develop from 0 to 5 m/s until it reaches steady state, while penguins are randomly located. Turbulence is introduced as wind interacts with, and is impeded by, the distribution of penguins across the field's computational grid.

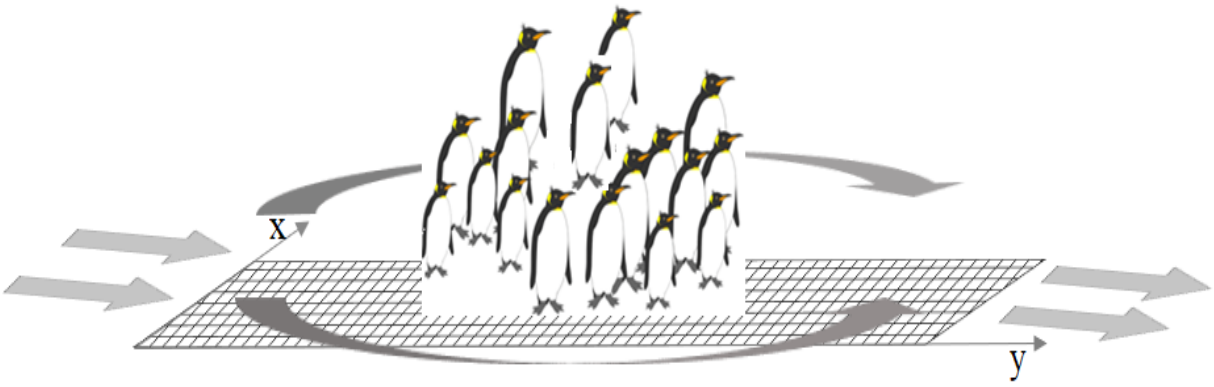

Figure 1 Models' Setting (wind field is 2D and calculated in FDM, which wind enters in y direction and interact with penguin huddle. Penguins' positions are updated in SPH method. Wind velocity also contributes to the wind chill effect)

3.1 Governing equations

The movement of wind and evolution of the penguin huddle are governed by the Navier-Stokes equations (NSE) of fluid motion in a continuum as described below (also see Appendix A).

$$\frac{\partial}{\partial t}\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \mu\nabla^2\boldsymbol{u} + \boldsymbol{b} \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{2}$$

Where $\boldsymbol{u}$ is the velocity vector of the fluid at a point, ∇ is the del operator defining spatial gradients, ρ is density, p is the fluid pressure, μ is the viscosity of the fluid, ∇^2 is the Laplacian operator. This equation includes a dynamic term ($\frac{\partial}{\partial t}\boldsymbol{u}$)**,** a convective term ($\boldsymbol{u}\cdot\nabla\boldsymbol{u}$), a dispersive term ($\mu\nabla^2\boldsymbol{u}$), a pressure term ($-\frac{1}{\rho}\nabla p$), and an external acceleration term ($\boldsymbol{b}$). Each term in

Equation 1 has units of acceleration (m/s^2), and Equation 2 enforces conservation of mass. When resolving the wind field, equations 1 and 2 are solved for the dynamic term at the current time, after which the velocity field is updated for the next time step. The Pressure Poisson equation (PPE) is considered to solve the pressure term to calculate the wind field (Eq. 4 and 5). The implementation method is summarized by (Seibold. 2008).

$$\nabla \cdot \left(\frac{\partial}{\partial t}\boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) = \nabla \cdot (-\frac{1}{\rho}\nabla p + \mu \nabla^2 \boldsymbol{u} + \boldsymbol{b}) \quad (3)$$

$$\nabla^2 p = \nabla \cdot (\boldsymbol{b} - (\boldsymbol{u} \cdot \nabla) \cdot \boldsymbol{u}) \quad (4)$$

3.2 Heat flux and estimation of wind chill

Heat fluxes through the wind field can be considered as a combination of convective and diffusive movement of heat within the wind field. The governing equations of change of temperature is defined by the divergences of the conductive and advective fluxes.

$$\frac{\partial}{\partial t}T + (\boldsymbol{u} \cdot \nabla)T - \mu \nabla^2 T = 0 \quad (5)$$

Wind flow around and through the huddle affects individual penguins differently through the windchill effect depending on the wind velocity where penguins are positioned (Waters et al. 2012). Windchill for each particle penguin is calculated by a standard windchill index equation Eq. 6 (Grant. 2008).

$$T_{wc} = 13.12 + 0.6215T_a - 11.37u^{0.16} + 0.3965T_a u^{0.16} \quad (6)$$

Where T_{wc} is the windchill temperature (ºC) felt by each computational particle, T_a is the local air temperature (ºC) and u is the the magnitude of wind vector $\boldsymbol{u}$ solved in Eq. 3 and 4.

3.3 Penguin metabolic heat release equations

From biologists' observation, penguin cold-exposed body surface can be reduced by huddling, which then affects their metabolic rate. Prior studies developed models on rodent huddling which incorporate an exponential decay relationship between surface area and the number of animals in a huddle (Wilson. 2017). In these studies, metabolic heat production is calculated based on Fourier's Law of heat flow (Gilbert et al. 2007; Glancy et al. 2016; Canals and Bozinovic. 2001; Wilson. 2017).

$$MR = AC(T_b - T_a) \tag{7}$$

Where *MR* is metabolic heat lost to the environment (in Watts, W), *C* is the whole body thermal conductance (W^oC^{-1}), T_b is individual body temperature (ºC), T_a is ambient air temperature (ºC) and A is the proportion to the body surface area that is exposed to the ambient temperature. Eq. 7 indicates that internal thermal conductance of penguins and temperature gradients between body and ambient temperatures determine total heat loss. When penguins experience thermo-neutral environmental temperatures (TNZ), they produce minimum and constant metabolic heat to maintain their physiological activities (Gilbert et al. 2010). For penguins, the TNZ ranges from the lower critical temperature (LCT) of -10ºC to the upper critical temperature (UCT) of 20 ºC (Gilbert et al. 2007). For ambient temperatures higher than the UCT, heat loss is increased, and huddles disperse; and for temperatures lower than the LCT, metabolic rate is increased prompting huddles to form (Gilbert. 2010). Huddle behavior allows a reduction in the body surface (A) exposed to wind through tight packing and provides for local heating of the surrounding environment (Gilbert et al. 2010). In previous models on mammals, researchers suggested that the

average reduced proportion of animal's surface area that is exposed depends on the number of aggregated animals (Canals and Bozinovic. 2001), which can be estimated as

$$A = n^{-\frac{1}{4}} \tag{8}$$

where *n* is the number of huddling individuals.

3.4 Formulation of particle solution using SPH

In SPH method, the fluid field is discretized and solved by a series of particles representing individual penguins. Using this method, the interpolated value of fluid parameter (*A)* at any position (*r)* can be expressed as (Gingold and Monaghan. 1977):

$$A(\boldsymbol{r}) = \int A(\boldsymbol{r}')W(\boldsymbol{r} - \boldsymbol{r}', h)d\boldsymbol{r}' \tag{9}$$

Where the integration is over the entire domain corresponding to the continuous medium, (h) is a smoothing length, and (W) is a weight function, called an interpolation kernel. For the simulation model, the integral interpolant in Eq.8 can be approximated by a summation interpolant shown as:

$$A(r) = \sum_d m_d \frac{A_d}{\rho_d} W(r - r_d, h) \tag{10}$$

Where m_d and ρ_d are the mass and density of particle *d* and the sum is over all particles within a radius of *2h* to the current particle *a*. In this application, the smoothing kernel *W (r, h)* is specified to be a C^2 spline-based interpolation (a special type of piecewise polynomial) with radius *2h*, which approximates the shape of a Gaussian function (Fig. 2 and Eq. 11).

$$W(r,h) = \frac{7\pi}{10h^2}\begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & if\ 0 \le q \le 1 \\ \frac{1}{4}(2-q)^3 & if\ 1 \le q \le 2 \\ 0 & otherwise \end{cases} \tag{11}$$

where $= \frac{r}{h}$.

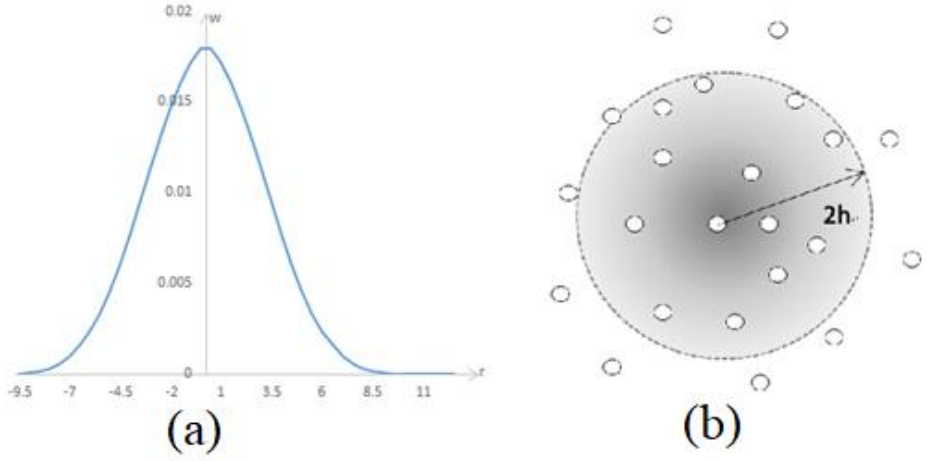

Figure 2 Kernel value (a) smooth kernel value depends on distance (b) only particles within smooth length contribute because W falls off rapidly for r≥h and the interactions are zero for r>2h

The spatial gradient of the function (*A)* is given by differentiating the interpolation equation as (Gingold and Monaghan. 1977):

$$\nabla A(\boldsymbol{r}) = \sum_d m_d \frac{A_d}{\rho_d} \nabla W(\boldsymbol{r} - \boldsymbol{r}_d, h) \tag{12}$$

The increased temperature due to released metabolic heat can by calculated by

$$\Delta T = \frac{\sum_d MR_d \frac{A_d}{\rho_d} W(r - r_d, h)}{C_{air} \rho_{air} V_{air}} \tag{13}$$

Ambient temperature is then updated by adding background air temperature and increased temperature ΔT from released metabolic heat.

Kernel value plays a significant role in deciding the contribution of neighboring particles to the center particle. By Eq. 7 and Eq. 13, to allows penguins to stay in their thermo-neutral zone (-10 to 20 ◦C) when huddle is formed, a 1.5m of kernel value is chosen for local ambient temperature calculation from accumulated metabolic heat by penguins in SPH model.

3.5 Continuity and momentum equations

An SPH-approximation of the continuity equation (Monaghan. 2009) is used for penguin huddle model as shown in Eq. 14 and 15:

$$\rho_a = \sum_d m_d W_{ad} \tag{14}$$

$$\frac{d\rho_a}{dt} = \sum_d m_d (v_d - v_a) \nabla W_{ad} \tag{15}$$

Where ρ_a is the density of particle a with velocity v_a, and m_d is the mass of particle d. The position vector $\boldsymbol{r}$ from particle d to particle a is calculated by $\boldsymbol{r_{ad} = r_a - r_d}$. $W_{ad} = W(r_{ad}, h)$ is the interpolation kernel with smoothing length h evaluated by the distance $|r_{ad}|$.

In the SPH method, the pressure gradient is estimated using:

$$\rho_a \nabla P_a = \sum_d m_d \, (P_d - P_a) \nabla_a W_{ad} \tag{16}$$

Momentum Equation is usually described as:

$$\frac{D\boldsymbol{u}}{Dt} + \frac{\nabla P}{\rho} = 0 \tag{17}$$

By rewriting $\frac{\nabla P}{\rho}$ as:

$$\frac{\nabla P}{\rho} = \nabla \left(\frac{P}{\rho}\right) + \frac{P}{\rho^2} \nabla \rho \tag{18}$$

The interpolated value of pressure gradient ($\frac{\nabla P}{\rho}$) can be expressed using SPH smoothing as:

$$\nabla(\frac{P}{\rho}) = \sum_d m_d \frac{P_d}{\rho_d^2} \nabla W \tag{19}$$

Combining Eq.16 and Eq.18, the momentum equation for particle a becomes

$$\frac{D\boldsymbol{v_a}}{Dt} = -\sum_d (m_d (\frac{P_d}{\rho_d^2} + \frac{P_a}{\rho_a^2}) \nabla_a w_{ad} \quad (20)$$

The contribution force from particle *d* to particle *a* when the kernel is a Gaussian is summarized as

$$F = \frac{2m_a m_d}{h^2} (\frac{P_d}{\rho_d^2} + \frac{P_a}{\rho_a^2})(\boldsymbol{r_a} - \boldsymbol{r_d}) W_{ad} \quad (21)$$

Finally, penguin particles are moved throughout the simulation using

$$\frac{d\boldsymbol{r}_a}{dt} = v_a \quad (22)$$

3.6 Penguin behavior model

This numeric experiment attempts to mimic huddling behavior and possibly provide clues explaining the process behind it through an algorithm seeking to minimize heat lost from individual penguins. In other words, the experimental model considers huddling not as an empathetic behavior where penguins cooperate for the good of the colony, but as an individual strategy to minimize energy expenditures with no knowledge or concern of the state of neighbors. The algorithm implemented considers possible factors compelling penguin movement using a modified form of the Navier-Stokes equation.

$$\frac{\partial \boldsymbol{v}}{\partial t} - \boldsymbol{b} + \frac{\nabla P}{\rho} = 0 \quad (23)$$

In Eq. 23, $\boldsymbol{v}$ is the velocity vector of each individual penguin; $\boldsymbol{b}$ is assumed an acceleration term (m/s^2) that compels "cold" penguins toward the mass center of the penguin colony and 'hot' penguins away from the huddle center; *p* is a repulsive pressure between individual penguins that prevents particles from overlapping. In this SPH huddle model, the convection term (i.e., $\boldsymbol{v} \cdot \nabla \boldsymbol{v}$)

is not represented as there is no bulk fluid flow of penguins in the time frame of interest, nor is the dispersion term needed (i.e., $\mu\nabla^2\boldsymbol{v}$) as the huddle does not develop internal eddies. The contribution to the pressure force p on one particle from its neighboring particles is driven by the SPH momentum equation (Eq. 21), and the body acceleration parameter $\boldsymbol{b}$ is a function of ambient temperature which accelerates cold penguin particles towards the huddle center of mass. The empirical $\boldsymbol{b}$ value is decided by considering the largest pressure force generated between penguins when huddle is formed. Linear and second order polynomial equations between $\boldsymbol{b}$ and ambient temperature are tested, and there is no significant difference on penguin moving direction. Linear relation is chosen for its simplicity and easy implementation. As the ambient temperature increases, this $\boldsymbol{b}$ parameter diminishes. As the ambient temperature increases further, b becomes negative and accelerates particles away from the colony center (Fig.3).

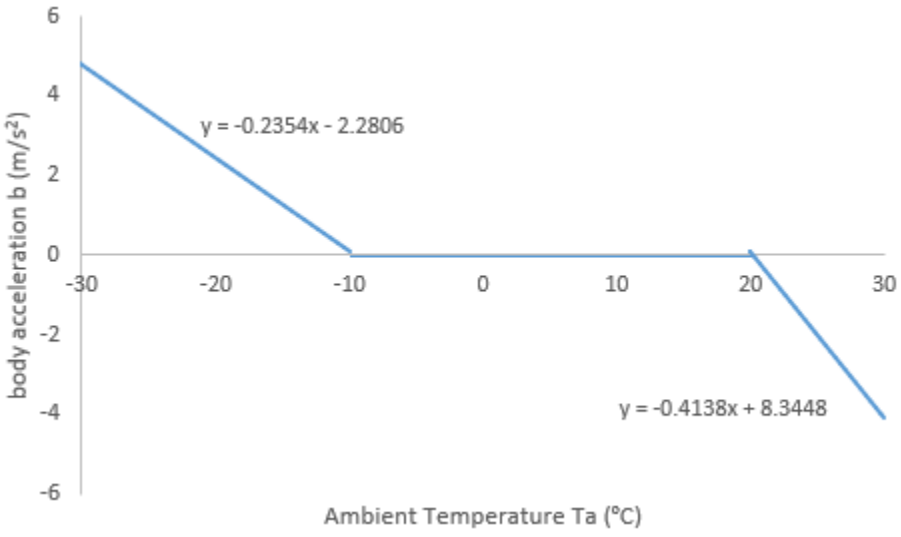

Figure 3 Relationship between ambient temperature (T_a) and particle acceleration ($\boldsymbol{b}$)

After calculating individual penguin's metabolic heat from Eq. 7, the accumulated heat released from penguins is calculated by Eq. 10 in SPH method. Ambient temperatures at grid

locations are calculated by summing air temperature and the temperature increased by accumulated heat from penguins nearby. Incorporated with wind velocity from FDM, the wind chill temperature is used to identify if penguins are within their TNZ. If penguins are not in their TNZ, the moving direction is determined by the body force acceleration $\boldsymbol{b}$ and pressure force acceleration from neighboring penguins.

3.7 SPH-FDM coupling

FDM is implemented in central difference approximations for interior nodes; in forward and back difference approximations for grid points on the boundary; both are with an expected second order of numeric error. In FDM, the mass within the zone changes due the mass fluxes and may cause error. However, SPH shows advantages in eliminating truncation and round-off errors for conservation of conservative properties, such as heat, mass, momentum. The properties of SPH let particles carry their own mass, and therefore, mass is always conserved.

Penguin particles move in response to ambient temperature as determined by the individual body temperature, the aggregate heat given off by neighboring penguins, the effects of wind chill, and the background ambient environmental temperature. As the trajectory of a particle is computed, the quantities of heat release and momentum gained or lost by all particles are incorporated in subsequent calculations. For the SPH-FDM coupled method, the penguin positions and resultant ambient temperature are the primary parameters exchanged between the models, and the wind velocity field is updated every 30 seconds (simulation time) as individual penguins move and the huddle shape forms, deforms or breaks apart. The flow chart of SPH-FDM coupled method is shown as Fig 4.

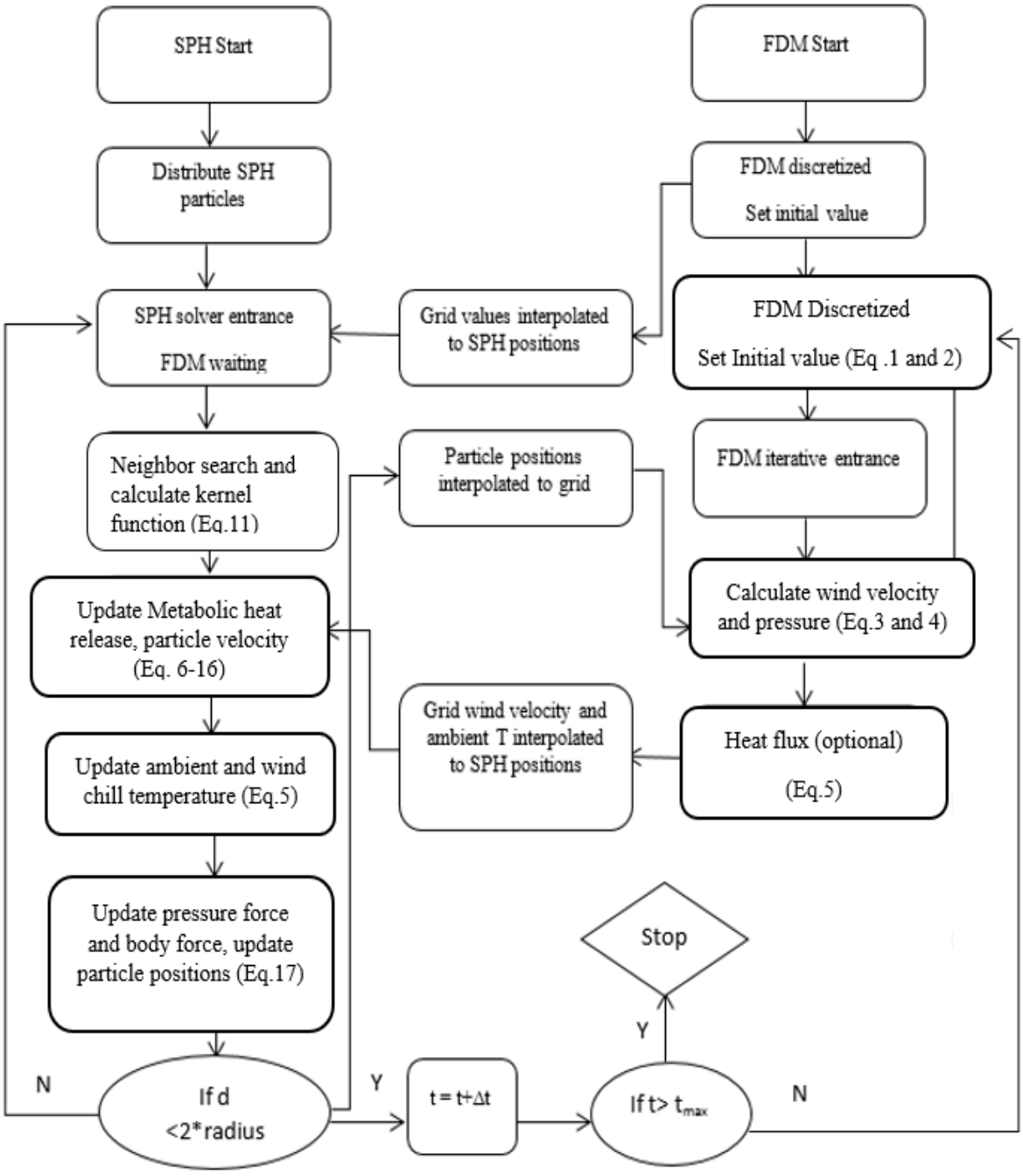

Figure 4 Algorithm of SPH-FDM

Penguin's radius is assumed around 0.25m. The particle size and grid size should be the same magnitude in order to make particle covers at least one grid point for wind field calculation in FDM. Therefore, a small grid size (0.25m) is chosen in order to bridge SPH model and FDM

model. The position of penguins calculated in SPH model need to be updated in FDM model for wind field calculation. Therefore, a small grid size is used to make sure the interaction between the wind and penguins is well-simulated. The small grid size also leads to a small time-step. In this case, the time step is set to 0.02 second. In the process of SPH model, the most time-consuming part is identifying neighboring particles. In a small time-step, the positions of particles are not significantly changed. Therefore, in aim to save computational time, the identifying process is only recalculated once as penguin moves at least one grid size.

3.8 Model Setup

It is important to note that the goal of each penguin is to minimize their individual metabolic rate by finding a comfortable position within its thermo-neutral zone. Penguins are motivated to move or stay stationary depending on the ambient temperature and pressure force from neighboring penguins. Individual penguins in the huddle are assumed to have uniform body temperature, height, volume, and mass. The procedure to simulate huddle is as follows:

1. Randomly locate penguins across the computational FDM grid and resolve the wind field by FDM.
2. Calculate the heat flux across the wind field by FDM.
3. Compute accumulated heat released from penguin particles and its contribution to local temperature by SPH.
4. Update ambient air temperature and calculate particle wind chill temperature by SPH.
5. Compute body force (b) of individual penguin depending on ambient temperature.
6. Compute pressure force of individual penguin from neighboring penguins by SPH.
7. Update particle positions.
8. Simulate huddle break-up by increasing ambient temperature higher than penguin's

thermos-neutral zone and update penguins' moving path

CHAPTER 4

DYNAMIC PENGUIN HUDDLE FORMATION SIMULATION

This chapter presents the simulation of this new method of coupling the smoothed particle hydrodynamic and finite difference methods (SPH-FDM) in penguin huddling formation. With this novel approach, each individual penguin and its information, such as position, velocity, temperature, metabolic heat loss, and path of motion can be traced through implementation of the SPH method, and solutions for wind velocity and direction across a complex and dynamic landscape can be easily quantified using FDM at a reduced computational cost. This coupled method shows benefits in simulating complicated flow dynamic shapes and interactions in dynamic environment.

Penguins are assumed to be randomly situated on a flat plane with dimensions of 50m x 100m. Wind enters from *Y* direction boundary and interacts with the penguins. Penguin movements towards to the group's mass center (i.e., the warmest part of the computational grid) during huddle formation are tracked in SPH method as shown in the time series figures (i.e., upper panes) of Fig. 5 for each of six intervals during the two-hour simulation. Figure 5, panel A is at time zero; panel B is 30 minutes; C is at 60 minutes; D is at 75 minutes; E is at 90 minutes; and F is at time 120 minutes. Wind velocity (lower panes) and the resulting ambient temperature (middle panes) are calculated using the wind field derived by the FDM and SPH model. As a huddle forms, the cumulative metabolic heat released from penguins in a neighborhood accumulates, and the penguin huddle blocks wind from flowing around individuals – thereby eliminating wind chill and elevating local ambient temperature. When wind chill temperatures fall within the thermo-neutral zone

(Typically -10 to 20ºC), penguins are less motivated to move, and the huddle becomes stable. Throughout the simulation, the positions of individual penguins can vary, and wind velocity field is recalculated and updated by FDM model as the shape of the huddle changes.

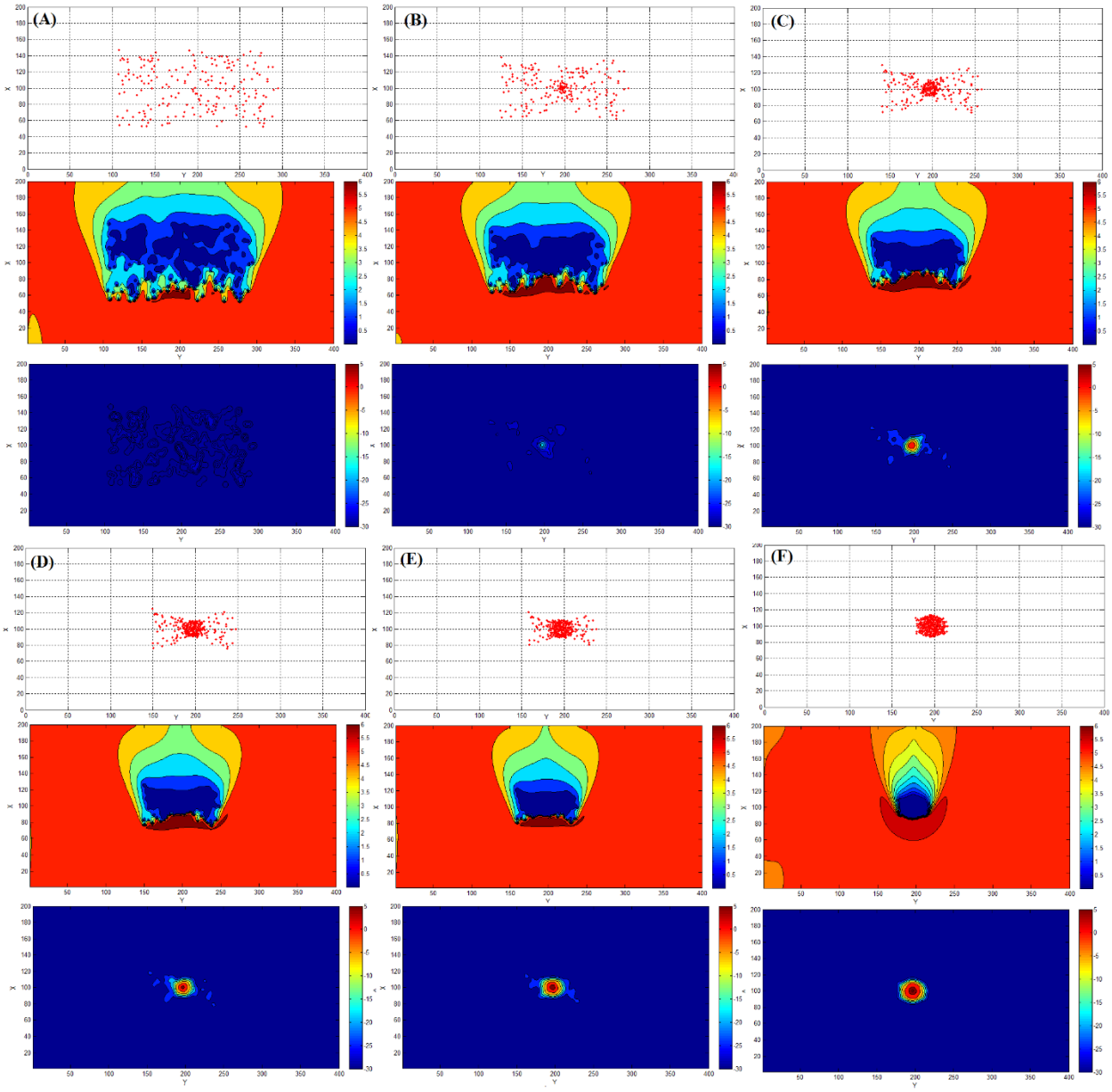

Figure 5 Simulation results of penguin huddling process in 2-hour increments (Upper: penguin position; Middle: Ambient temperature Contour (ºC); Lower: Wind velocity (m/s))

This simulation estimates that individual penguins relocated from loose aggregations to a stable and dense huddle with an interior ambient temperature around 5ºC. Penguins on the huddle perimeter released more metabolic heat than those in the interior because of higher temperature gradient and proximal exposure to wind and associated wind chill effects (Figure 6). Because the ambient temperature of penguins around the perimeter was not in the TNZ, these penguins continued to seek a more comfortable position within the huddle and were not stationary. Table 1 summarizes average temperature, temperature range and the percentage of penguins within TNZ from the six-time points in Figure 5. The average temperature is calculated as the average of all particles, and temperature range is the highest to lowest ambient temperature at these time series. After a 2-hour simulation, approximately 75% of the simulated penguins are in their TNZ, as seen in Figure 6.

Table 1 Summary of descriptive statistics

Figure 5 Pane	**Average ambient Temperature (ºC ± σ)**	**Ambient Temperature Range at the huddle (ºC)**	**% of Penguins in TNZ**	**Average Huddle Density (penguins/m^2)**
A	-26.97±0.92	-28.00~-24.09	0	0.2
B	-23.78±6.07	-27.98~-2.05	5.5	0.3
C	-18.38±9.37	-27.97~2.42	25.5	0.4
D	-14.87±10.00	-27.88~3.80	37.5	0.7
E	-11.42±10.01	-27.67~5.56	53.5	1.2
F	-5.03±6.85	-23.12~7.74	77.5	3.1

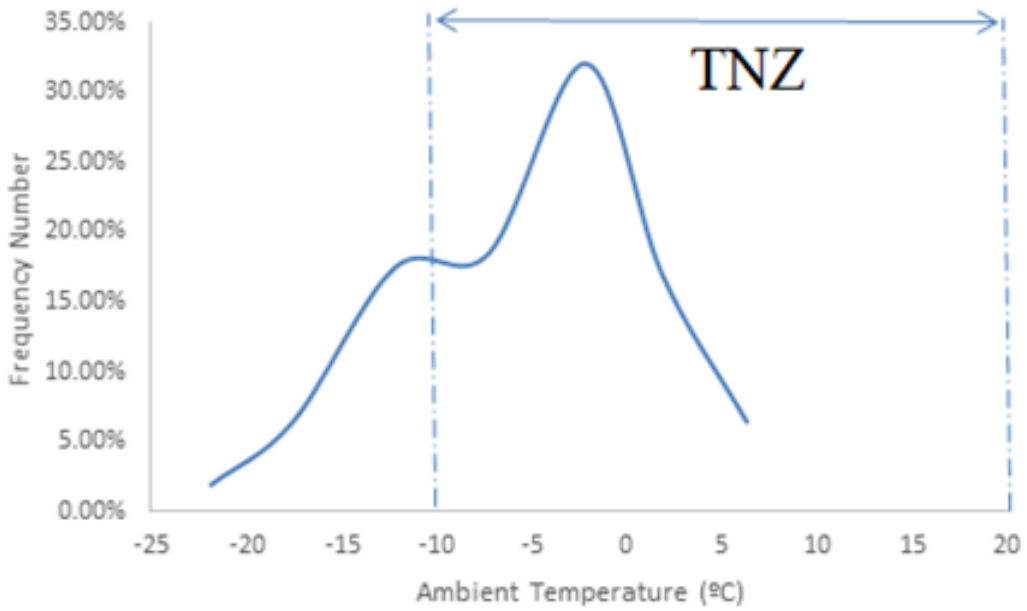

Figure 6 Frequency graph of individual penguin ambient temperature at 2h (the arrow indicates the TNZ)

The ambient temperature of the location where individual particles stand, individual metabolic heat loss are plotted against its distance to the huddle center, which shown through Figure 7 and 8. In Figure 8, the windward quantified penguins on the boundary where exposed to high wind velocity (greater than 3m/s), while leeward quantified penguins on the boundary where exposed to low wind velocity (less than 3 m/s).

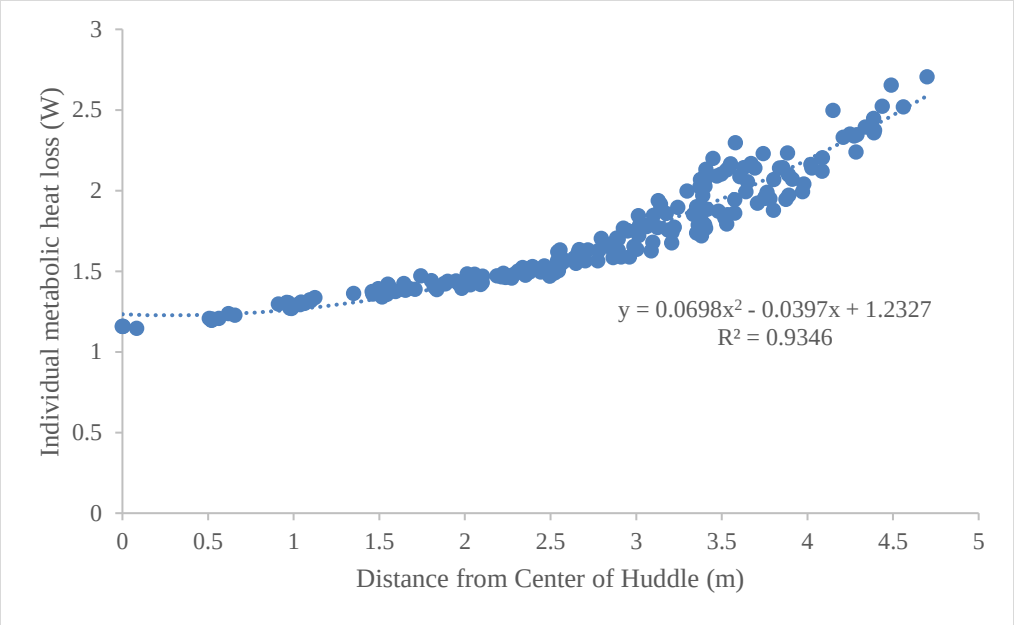

Figure 7 Relationship between individual metabolic heat loss and its distance to huddle center

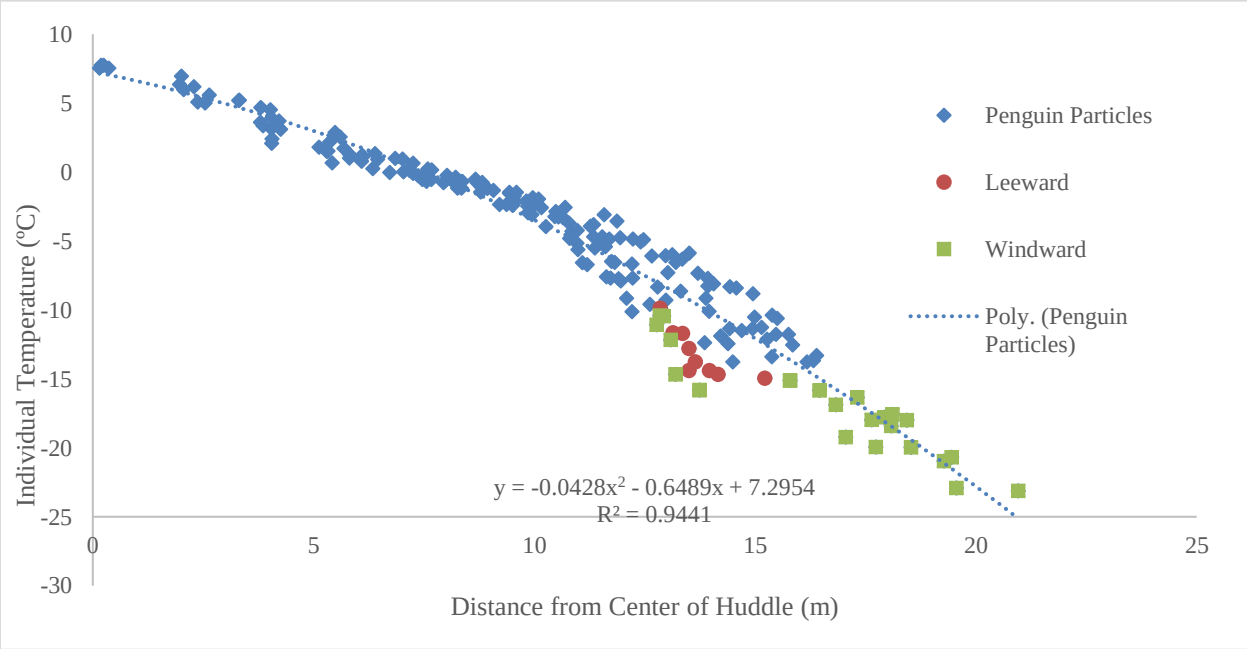

Figure 8 Relationship between individual temperature and its distance to huddle center

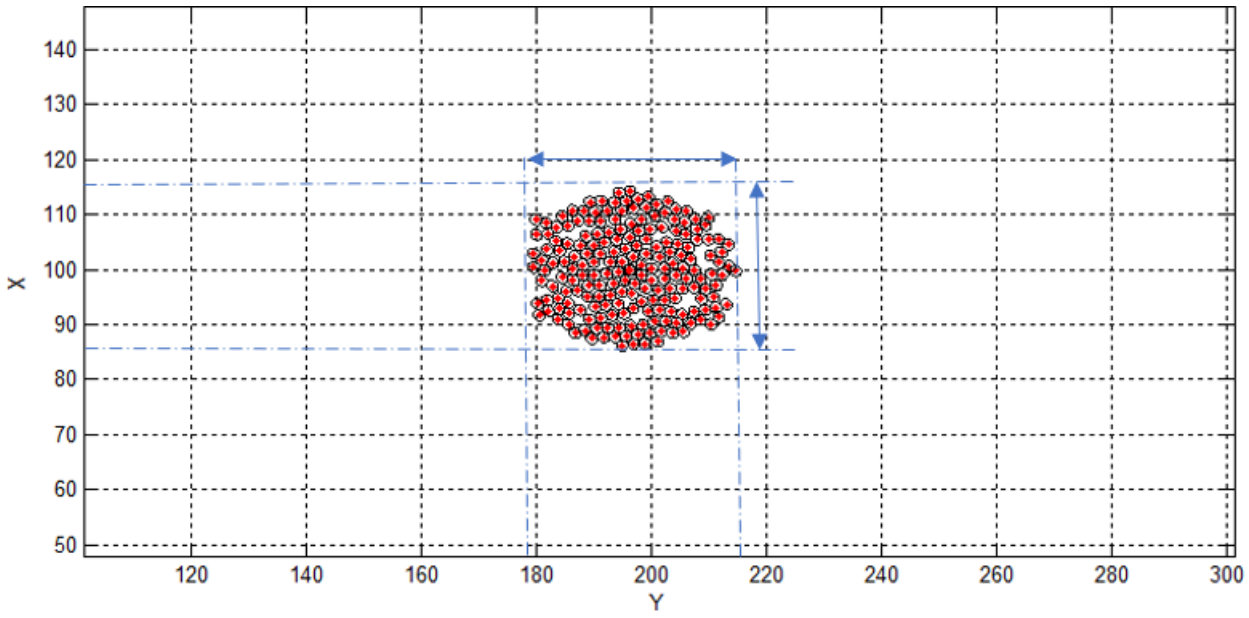

Figure 9 Huddle density

In this simulation model, ambient temperature and wind speed are the main factors affecting penguin huddles. As Figure 5 shows, at the beginning, penguins are affected by both ambient temperature and wind speed; while when the huddle forms, only penguins on the perimeter of the huddle are affected by wind chill effects. These results suggest that both ambient temperature and wind speed are predictors of huddling occurrence, which meet the observations from previous research (Gilbert and Robertson, 2008). This model can then provide a quantified way to calculate the probability of huddling according to ambient temperature and wind speed.

The simulation model forms a huddle around 9×7.25 m, as shown in Figure 9. With 200 penguins, the density of huddle is around 3.1 birds m^{-2}, which is very close to 2.8 birds m^{-2} -- the typical mean density shown by emperor huddle in general during winter (Gilbert and Robertson, 2008). While biologists suggest that ground movement may be affected by wind direction, the ambient temperature, however, is the main meteorological factor influencing huddling group density. In our simulation model, the body force term is assumed only by ambient temperature,

thus the density of huddle in this model is mainly affected by this factor. This model can be used to interpret thermal infrared image to predict the occurrence of huddle.

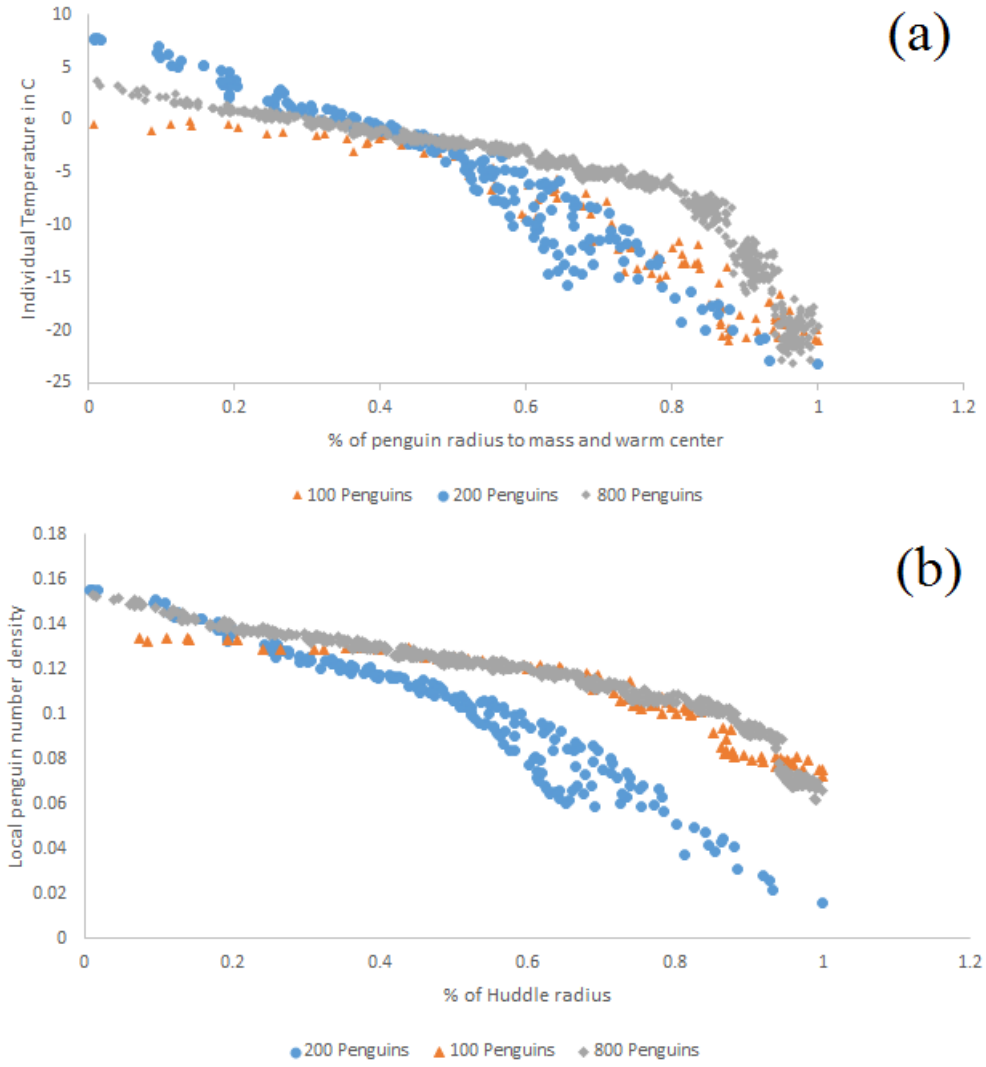

Figure 10 Relationship between % of huddle radius in 3 sizes of huddles and (a) local ambient temperature (b) local penguin number density

From continuity equation in SPH (Eq. 14), local number density within SPH smooth circle is estimated against percentage of huddle radius (Figure 10-b). The typical penguin huddle size under 3 breeding stages (100 penguins, 200 penguins and 800 penguins) are simulated in this model. Figure 10-a shows the relationship between ambient temperature where penguins stand and the percentage of huddle radius. These three penguin huddles show similar trend, where the highest

density and highest ambient temperature are the center of the huddle. It suggests that penguins tend to gather in the warmest place. Penguins on the boundary are affected by wind chill effect. In addition, the trendline of ambient temperature is similar to the trendline of local number density. In SPH calculation, particles within smooth circle contribute to the center particle, which are quantified by kernel function. The denser inside the smooth circle, the more released metabolic heat is contributed to the center particle. These results also suggest that the ambient temperature from the edge to the center of huddle may not increase linearly and fit second polynomial relation (with constrain condition that distance should be positive). Penguin individual temperature can be estimated according to its local number density. Because the body force term is determined by ambient temperature, and pressure force term can be quantified in SPH model, we can then use this model to estimate penguin individual path.

CHAPTER 5

HUDDLE DYNAMICS AND ENVIRONMENTAL FACTORS

Previous observations (Zitterbart et al, 2011) suggest that penguins in a huddle are tightly packed. However, huddle structure is continuously reorganized by the penguins, because each individual needs to maintain warmth and stay in the center part of the huddle. Moreover, penguin huddles are observed in a very slow but continuous movement, and those penguins exposed to the wind move slowly along the sides of the huddle trying to find a warmer locations (Waters et al, 2012; Le Maho, 1977).

In the previous chapter, the model shows the ability to simulate huddle formation. Because SPH model has the advantages in tracking properties of each individual, the dynamic process in a huddle could be well simulated. The assumption is penguins have equal access to the warm center. In this numeric experiment, a circular shape huddle is generated, and all penguins stand besides each other with no empty spaces (Figure 11.a). The total number of penguins is constant during the simulation and the 2D wind field is simulated by FDM model (Figure 11.b). Winds blow from one direction and interact with the huddle. The initial air temperature is set to -30°C, which is a typical temperature in Antarctica's winter.

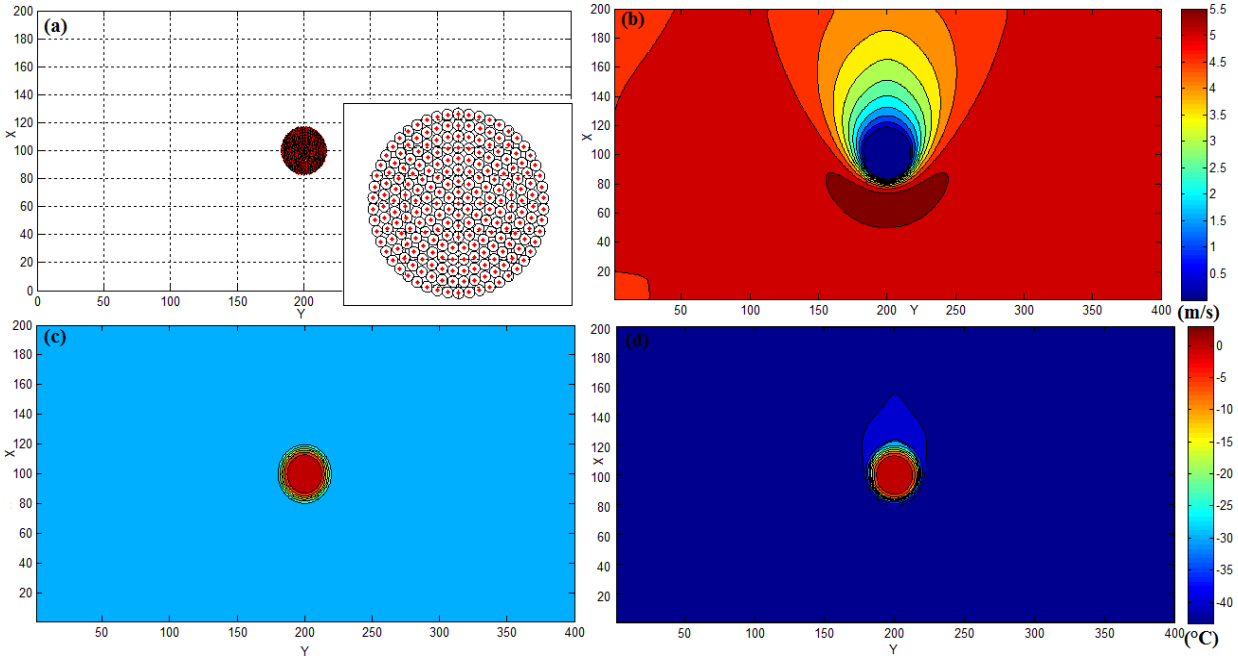

Figure 11 Initial generated huddle of penguins (a) position of each penguin located in a huddle and the huddle arrangement (b) initial wind field calculated by FDM (m/s) (c) ambient temperature (-30°C air temperature plus increased temperature by released metabolic heat) (d) Wind chill temperature

The penguins are stimulated to move by the significant temperature difference between penguin's body and ambient air, while in the same time, SPH model shows that the penguin's metabolic heat also makes the ambient air warmer. Figure 11 shows that the huddle center is the warmest with temperature reaching 3°C (Figure 11.c). With the comparatively high ambient temperature, most penguins within the huddle stay in their thermos-neutral zone; yet the penguins on the perimeter are still in a critical low temperature (-10°C). Penguins on the huddle's windward edge are experiencing higher heat loss and are more motivated to reposition than penguins on the leeward edge of the huddle (Figure 11.d).

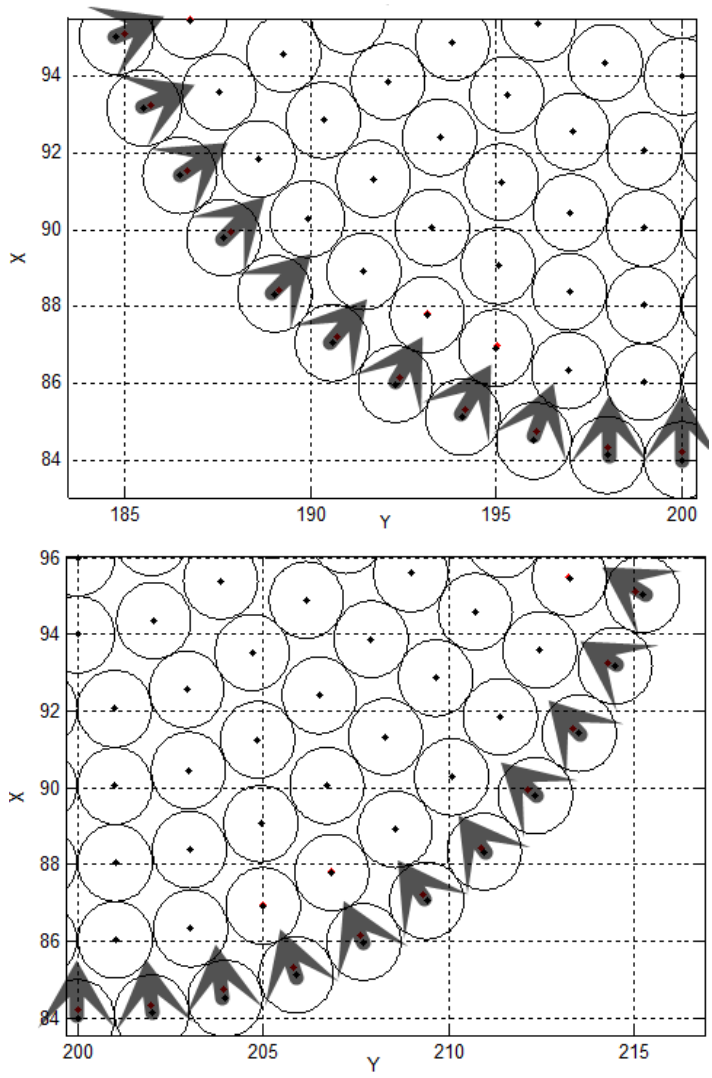

Figure 12 Moving directions of penguins on the perimeter (arrows show the moving direction, x and y are grid numbers)

Because of low ambient temperature and no pressure force from neighboring penguins, body acceleration term ***b*** is the only driven factor that determines the penguin's movement direction on the perimeter, the direction of ***b*** always point towards the warm center, as shown in Figure 12.

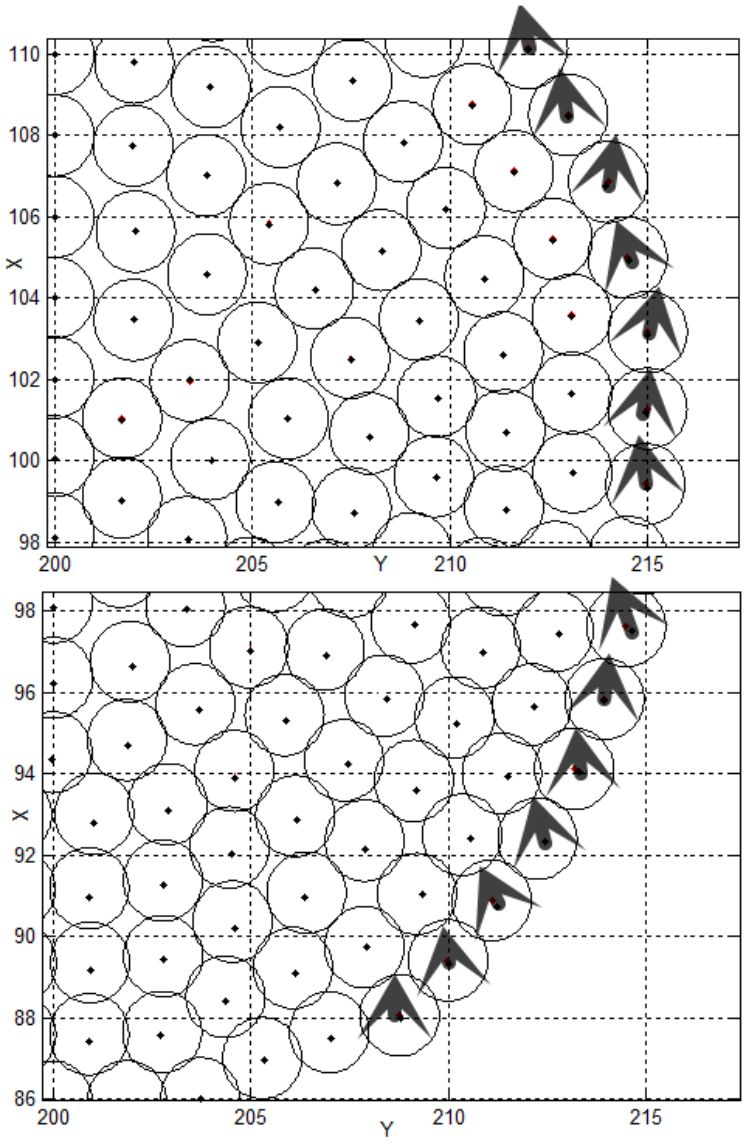

Figure 13 Moving direction of penguins on the perimeter under body force and pressure force (arrows show the moving direction, x and y are grid numbers)

As the penguins on the perimeter move towards the warm center, those on the inner circle gets pushed by their outer companions. The resultant force forces penguins on the perimeter to move along the side from windward edge to the leeward edge (Figure 13), thereby exposing a new layer of penguins to the coldest part of the huddle. From the simulation model, the moving process is continuous.

The huddle dynamics show a strong dependence on the variations of body force term $\boldsymbol{b}$ and pressure force p (Figure 14). Body external accleration is an assumptive condition in this model. For penguins on the perimeter, the body force $\boldsymbol{b}$ is dominant, and penguins tend to move towards the center. Based on SPH kernel function, as penguins continued to move inward and push the interior penguins, the huddle density increase leads to increased pressure force which stops the penguins from further inward movement. At this point, the body forces decrease because ambient temperature is comparatively high. Penguins on the perimeter then move to the leeward edge. With a relative small body force, pressure force is large enough to change or determine moving directions of penguins on the perimeter (Figure 14).

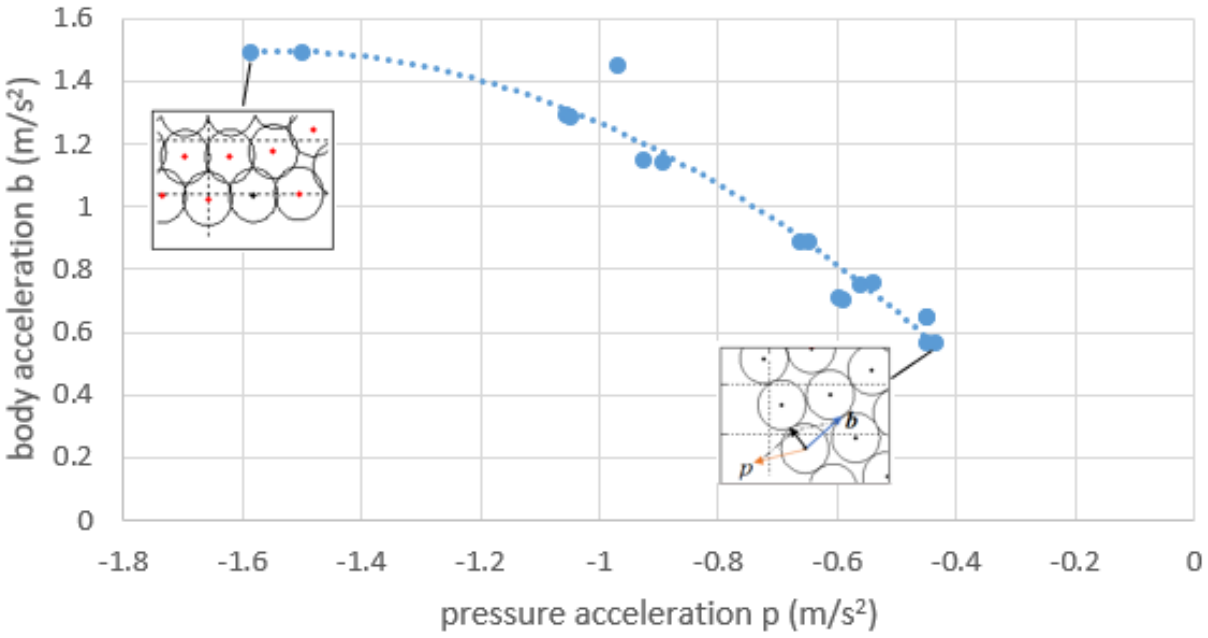

Figure 14 Moving direction under the total force of $\boldsymbol{b}$ and p

In previous chapter, the huddle forms under unfavorable meteorological conditions: both wind speed and ambient temperature affect huddling occurrence. If ambient temperature is higher than penguin's critical temperature 20 °C, huddles start to break up. As Figure 14 shows, $\boldsymbol{b}$

parameter diminishes when the ambient temperature increases; as the ambient temperature increases further, ***b*** becomes negative and accelerates particles to disperse from the colony center.

Two scenarios of different background air temperature are being tested in this model to simulate the process of huddle break-up. Air temperature is increased to 0 °C and 10 °C, respectively, to create local ambient temperature higher than penguin's thermos-neutral zone. Huddle break-up is expected during the process of increasing temperature. Figure 15 and 16 show the huddle break-up happens under the background air temperature of 0°C. Due to released metabolic heat, the highest local ambient temperature reach to 23 °C, which is higher than penguin's thermal neutral zone. Penguins at the warmer part tend to move away under the body acceleration term ***b*** to dissipate excess heat and maintain their thermal-neutral zone. Influenced by pressure force from neighboring penguins, huddles start to break up from edge to the center. This simulation scenario meets the observation (Ancel et al, 2005) that breakups start close to the edge of the huddle, then propagate within a short time to the entire huddle.

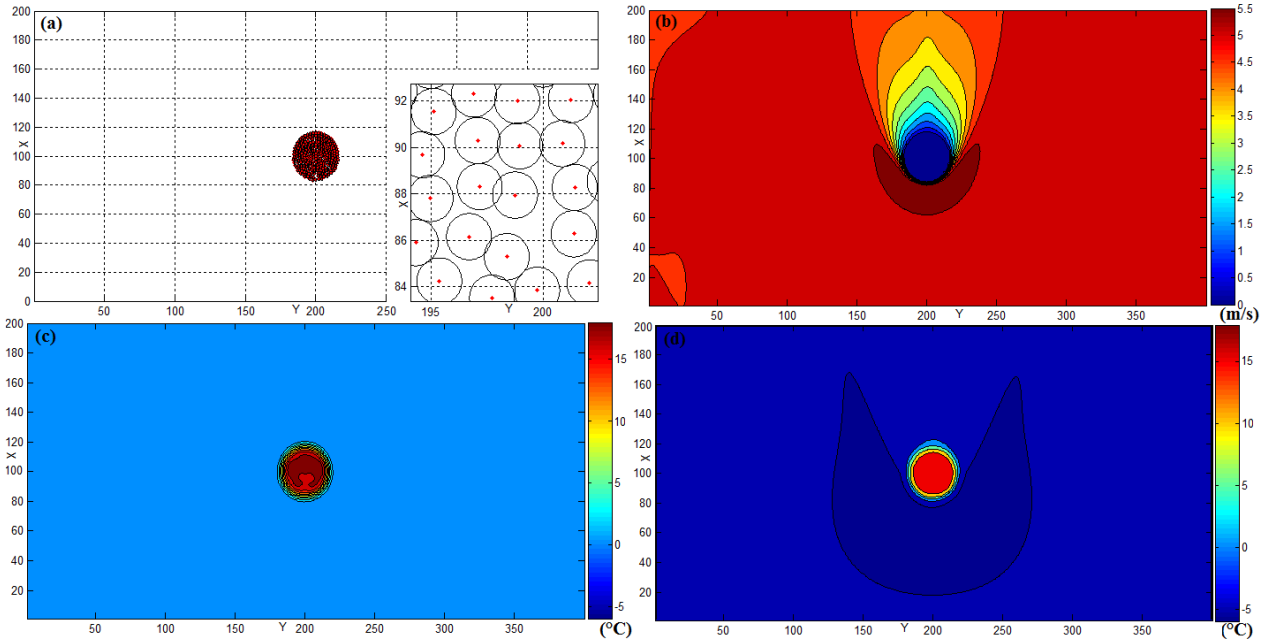

Figure 15 Huddle break-up after air temperature is increased to 0 °C (a) positions of penguins located in a huddle when huddle tends to break up (b) wind profile calculated by FDM (m/s) (d) ambient temperature (calculated by adding air temperature 0 °C and the increased temperature by released metabolic heat (°C) (d) wind chill temperature, calculated by wind chill equation, which incorporated with wind velocity and ambient temperature (°C)

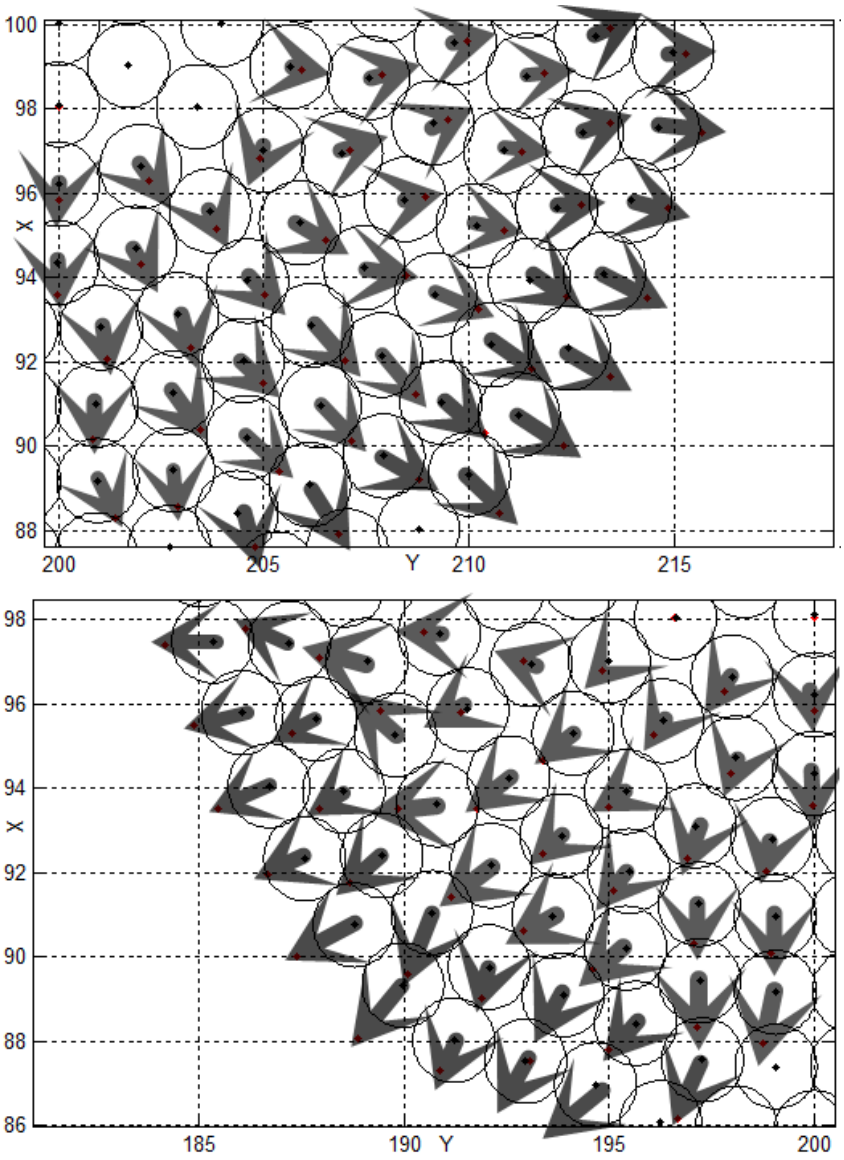

Figure 16 Moving directions of penguins during huddle break-up (Air temperature is 0°C; Arrows show the moving direction and path of the penguins)

When background air temperature is increased to 10 °C (Figure 17 and 18), due to metabolic heat, the local ambient temperature of the whole huddle, with an exception of penguins on the perimeter, is higher than penguin's thermal-neutral zone. With the large body acceleration ***b***, penguins at the warmest part start to move out and push outer loop penguins. From the previous observation, breakup expands until the huddles reach almost twice as large as the initial huddle. Figure 16 indicates penguins' positions during break-up process. Penguins from warmest part can

then release themselves from mutual body contact and dissipate excess heat; while penguins on the perimeter have a large density and will continue to move under the pressure force acceleration and body acceleration ***b***. Biologists observed huddle break-ups happened due to a variety of reasons, such as confliction between two individuals, chick-rearing and even no visible cause. These two scenarios explain how ambient temperature affect penguin break-up while they are under thermoregulation theory, although it cannot explain all types of the huddle break-ups. These simulation results provide a good application of thermoregulation: huddle growth and decay when penguins gain, conserve or lose heat; an individual penguin tends to seek individual need of warmth and main its thermos-neutral zone.

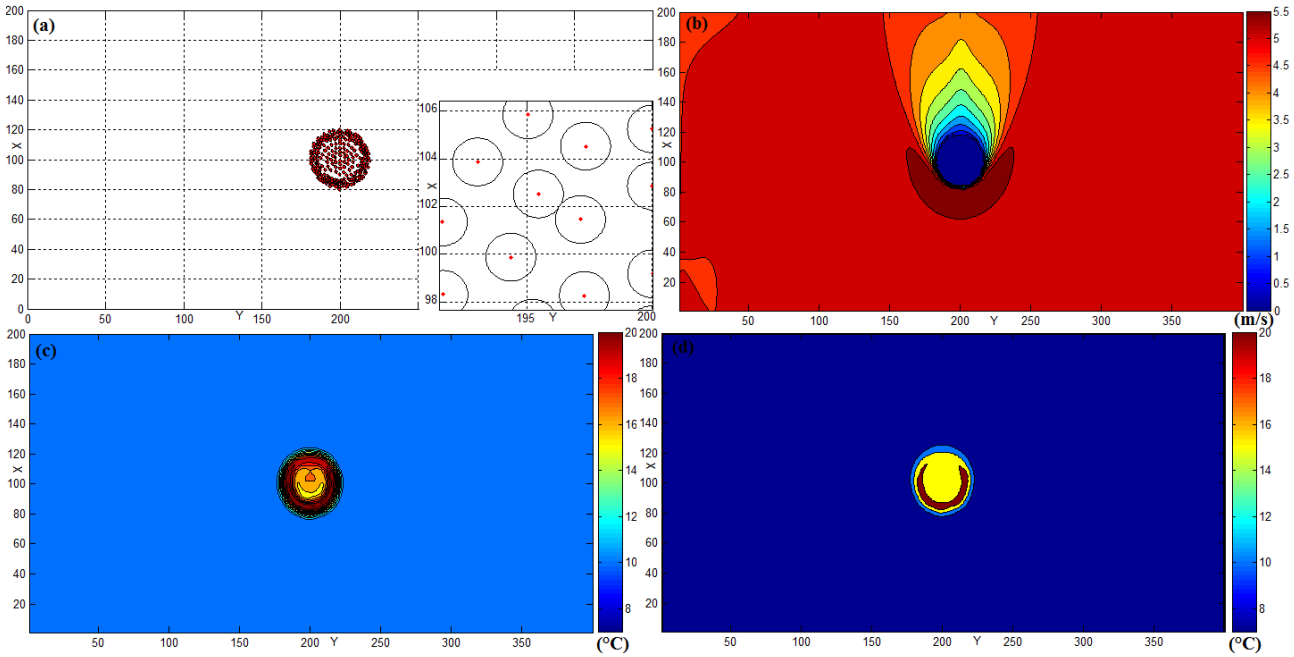

Figure 17 Huddle break-up after air temperature is increased to 10 °C (a) positions of penguins located in a huddle when huddle tend to break up (b) wind profile calculated by FDM (m/s) (d) ambient temperature (calculated by adding air temperature 0 °C and the increased temperature by released metabolic heat (°C) (d) wind chill temperature, calculated by wind chill equation, which incorporated with wind velocity and ambient temperature (°C)

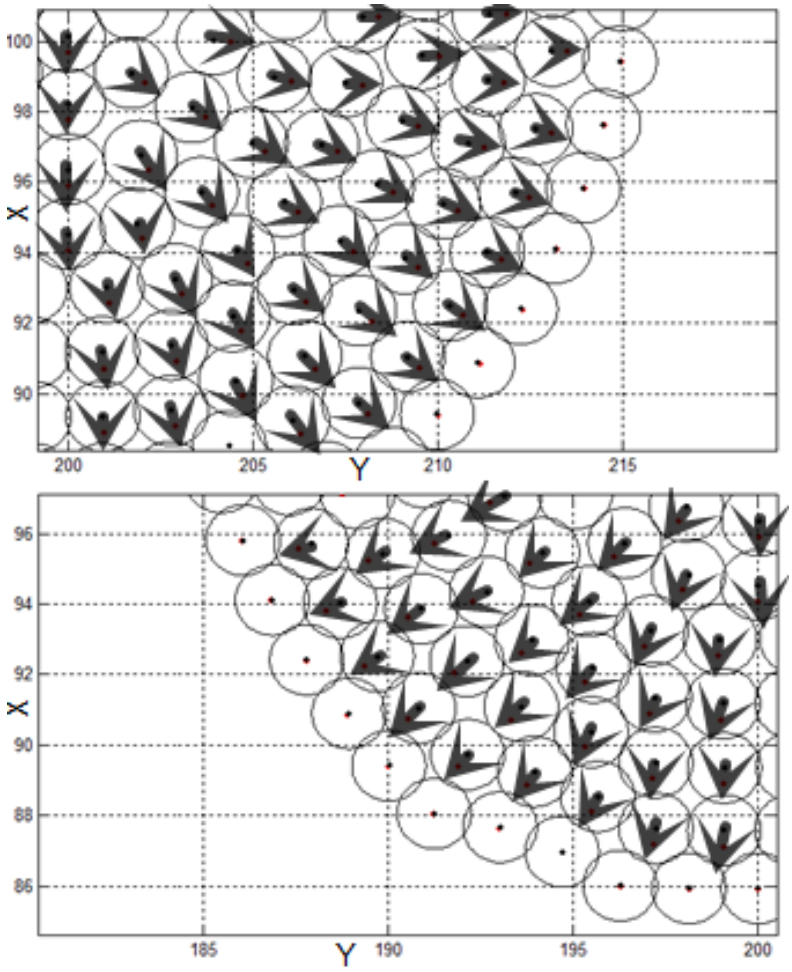

Figure 18 Moving directions of penguins during huddle break-up (Air temperature is 10 °C; Arrows show the moving direction and path of the penguins)

CHAPTER 6

CONCLUSION AND FUTURE APPLICATIONS

For this dissertation, a coupled model is successfully implemented in simulating penguin huddles. Finite Difference Method (FDM) and Smoothed Particle Hydrodynamics (SPH) method are applied to simulate dynamic penguin huddle and simultaneously wind field interacted with the huddle. As the simulation results show, the coupled model can simulate the dynamic huddling formation, track moving directions of penguins when huddle is formed and relatively stable, and simulate the break-up of huddles when ambient temperature is higher than penguin's thermo-neutral zone.

Computational fluid model analysis is widely used in a variety of applications, including ecological problems and non-fluid problems. Previous research suggests that penguin huddles allow individuals to protect individuals from unfavorable environmental conditions, with similar properties as fluid. This behavior can be simulated and implemented with fluid dynamic models. There are few numerical models in penguin huddle simulation. A previous penguin model used fluid dynamic equation to compute the temperature profile around the huddle, with assumptions that penguin will move to positions where the local heat loss is minimal without considering moving path or interactions between penguins. Biologists suggest that penguins may choose to join or leave huddle based on individual need of warmth, therefore, the SPH method was considered in this research because it's mesh-free and can track individual's properties. In addition, SPH shows advantages in solving complex boundaries and moving bodies. Penguins can be treated

as “particles” and represented in SPH method to track their properties, such as moving direction, moving speed, individual temperature.

From previous observations, air temperature and wind velocity are the main environmental factors affecting huddle patterns. To simulate wind field to interact with penguin huddle, FDM is implemented in this research for its easy implementation and calculation. The Navier-Stokes equation is the basic equation used to resolve wind field by using MATLAB. Wind is assumed to enter in one direction and interact with penguins. The most difficult part in wind field simulation is the resolution of pressure term in Navier-Stokes equation. A pressure Poisson equation is considered in the model to deduct pressure gradient term, which make wind bypass penguins when wind field interact with huddle. Moreover, in this research, a small grid size (0.25m) is chosen in order to bridge SPH model and FDM model. A small grid size is used to make sure the interaction between the wind and penguins are well-simulated. The small grid size also requires a small time-step to eliminate numeric oscillations and instability. In the process of SPH model, the most time-consuming part is identifying neighboring particles. In a small time-step, the positions of particles are not significantly changed. Therefore, in aim to save computational time, identifying process is calculated once penguin move at least one grid size.

In severe environmental conditions, penguin tend to huddle to keep warm and reduce energy costs. A 2-hour simulation of the huddle forming process is done by the couple-model. In this simulated huddle, most of the interior penguins reached a thermo-neutral temperature, which kept them stationary and the huddle is stable. However, for the penguins on the huddle perimeter, the ambient temperature was below TNZ and motivated those penguins to move toward a more comfortable location within the huddle (either in the interior or around to the rear of the huddle). This model is then applied to simulate the dynamic huddle in Antarctic’s winter as well as how

the huddle breaks up when air temperature increased. In agreement with previous predictions in Chapter 5, the model simulates how the cold-exposed penguins on the perimeter move during huddling and how the huddle move during breakup when air temperature increase. These results prove the coupled SPH-FDM model can be a calibration in predicting the formation of penguin huddling based upon a "selfish" algorithm seeking only to minimize individual energy expenditures and shows potentials in interpret thermal infrared image of penguins in the future.

In Ancel et al's (2015) observations, several climate factors, such as atmospheric pressure, relative humidity, solar radiation, wind speed and direction, are correlated with dynamics of huddles. Environmental factors affect not only the number of huddles, but also the mean number of individuals per huddle. This research only considers wind velocity and air temperature; therefore, future applications that considering other environmental factors are needed to further study the penguin's behavior equation and track the effect of these factors on huddling patterns. Moreover, biologists suggest that reproductive stages of penguins affect their huddling pattern as well, such as pairing, incubation and chick-rearing. The sizes of huddles are different among these three stages: males and females are present during pairing, while only males in incubation period. Varied sizes of huddles can be then generated and simulated by this coupled method to determine the effecting levels of climate factors and can be used to interpret thermal infrared images of penguin huddle under different reproductive stages. The recorded data for penguin huddles is very limited. Researchers used satellite remote sensing data to track and record penguin distributions. The model can be testified by real penguin huddle data with collaboration of penguin studies.

This model platform can be applied to other interesting ecological systems with fluid-fluid properties such as fish shoaling in river currents or bees swarming in a wind field. Also, this model

strategy can be considered to apply to engineering applications such as the behavior of swarms of autonomous drones.

MATLAB is chosen for the model due to its explicit parameters to the methods and straightforward output analysis. However, the running efficiency of loop cycles in MATLAB is relative slow. Future work in improving computational efforts is need to be considered. When greatly increase the number of penguins, SPH presents a high computational cost. High performance computing, such as GPU implementation, parallelization, are common ways to improve model efficiency. In addition, other language may be considered to improve model efficiency, such as Python, which shows advantages in namespaces and introspection.

REFERENCE

Amicarelli, A., Albano, R., Mirauda, D., Agate, G., Sole, A., Guandalini, Ro., 2015. A smoothed particle hydrodynamics model for 3D solid body transport in free surface flow. Computers & Fluids. 16, 205-228

Ancel, A., Gilbert, C., Poulin, N., Beaulieu, M., Thierry, B., 2015. New insights into the huddling dynamics of emperor penguins. Animal Behavior. 110, 91-98, doi: 10.1016/j.anbehav.2015.09.019

Ancel, A., Visser, H., Handrch, Y., Masman, D., Le maho, Y., 1997. Energy saving in huddling penguins. Nature. 385, 304-305

Boratynski, J.S., Willis, C.K.R., Jefimow, M., Wojciechowski, M.S., 2015. Huddling reduces evaporative water loss in torpid Nattere's bas, *Myotis nattereri*. Comparative Biochemistry and Physiology. 179, 125-132

Canals M., Bozinovic F., 2001. Huddling Behavior as critical phase transition triggered by low temperature. Complexity. 17, 35-43

Chakraborty, S., Srinivasan, K.K., 2016. Analysis and application of two-fluid model for mixed traffic conditions. The International Journal of Transportation Research. https://doi.org/10.1080/19427867.2016.1193309

Chen F, Qiang H, Gao W., 2015. Coupling of smoothed particle hydrodynamics and finite volume method for two-dimensional spouted beds. Computational Chemical Engineering. 77, 135-146

Chen, Z., Dai, Z., Huang, Y., Bian, G., 2013. Numerical simulation of large deformation in shear panel dampers using smoothed particle hydrodynamics. Engineering Structures. 48, 245-254

Cleary, P.W., Cohen, R.C., Harrison, S.M., Sinnott, M.D., 2013. Prediction of industrial, biophysical and extreme geophysical flows using particle methods. Engineering Computations. 30 (2), 157-196

Cleary, P.W., Prakash, M., Ha, J., 2006. Novel applications of smoothed particle hydrodynamics in metal forming. Journal of Materials Processing technology. 177(1-3), 41-48

Cohen, R.C., Cleary, P.W., Mason, B., 2009. Simulations of human swimming using Smoothed Particle Hydrodynamics. 7th International Conference on CFD in the Minerals and Process Industries. Commonwealth Scientific and Industrial Research Organization

Cohen, R.C., Cleary, P.W., Mason, B., 2010. Improving understanding of human swimming using smoothed particle hydrodynamics. 6th World Congress of Biomechanics, Singapore, Springer. 174-177

Cohen, R.C., Cleary, P.W., Mason, B., 2012. Simulations of dolphin kick swimming using smoothed particle hydrodynamics. Human Movement Science. 31, 604-619

Cohen, R.C., Cleary, P.W., Mason, B.R., Pease, D., 2015. The role of the hand during freestyle swimming. Journal of Biomechanical Engineering. 137, Article No. 111007

Cornthwaite, J., 2013. Pressure Poisson Method for the Incompressible Navier-Stokes Equations Using Galerkin Finite Elements. http://digitalcommons.georgiasouthern.edu/cgi/viewcontent.cgi?article=1846&context=etd

Cummins, S.J. and Rudman, M., 1999. An SPH Projection Method. Journal of Computational Physics. 152 (2), 584-607

Dai, Z., Huang, Y., Cheng, H., Xu, Q., 2014. 3D numerical modeling using smoothed particle hydrodynamics of flow-like landslide propagation triggered by the 2008 Wenchuan earthquake. Engineering Geology. 180, 21-33

Dalrymple, R.A., and Rogers, B.D., 2006. Numerical modeling of water waves with the SPH method. Coastal Engineering. 53(2-3), 141-147

Dao, M.H., Xu, H., Chan, E.S., Tkalich, P., 2011. Numerical modeling of extreme waves by smoothed particle hydrodynamics. Natural Hazards and Earth System Sciences. 11, 419-429

Das, A.K., Das, P.K., 2015. Modeling of liquid-vapor phase change using smoothed particle hydrodynamics. Journal of Computational Physics. 303, 125-145

Das, R., Cleary, P.W., 2010. Effect of rock shapes on brittle fracture using Smoothed Particle Hydrodynamics. Theoretical and Applied Fracture Mechanics. 53, 47-60

Debroux, F., Prakash, M., Cleary, P., 2001. Three-dimensional modeling of a tsunami interacting with real topographical coastline using Smoothed Particle Hydrodynamics. 14th Australasian Fluid Mechanics Conference.

Di Monaco, A., Manenti, S., Agate, G., Gallati, M., Sibilla, S., Agate, G., Guandalini, R., 2011. SPH modeling of solid boundaries through a semi-Analytic Approach, Engineering Applications of Computational Fluid Mechanics. 5 (1), 1-5. DOI: 10.1080/19942060.2011.11015348

Dong, X.W., Liu, G.R., Li, Z.L., Zeng, W., 2016. A smoothed particle hydrodynamics (SPH) model for simulating surface erosion by impacts of foreign particles. Tribology International. 95, 267-278

Eitzlmayr, A., Koscher, G., Khinast, J., 2014. A novel method for modeling of complex wall geometries in smoothed particle hydrodynamics. Computer Physics Communications. 185, 2436-2448

Ellero, M., Serrano, M., Espanol, P., 2007. Incompressible Smoothed Particle Hydrodynamics. Journal of Computational Physics. 226 (2), 1731-1752

Eto, T., Sakamoto, S.H., Okubo, Y., Koshimoto, C., Kashimura, A., Morita, T., 2014. Huddling facilitates expression of daily torpor in the large Japanese field mouse *Apodemus speciosus*. Physiology & Behavior. 133, 22-29

Fauci, L.J., 1988. A computational Model of Aquatic Animal Locomotion. Journal of Computational Physics. 77, 85-108

Feng, D.L., Liu, M.B., Li, H.Q., Liu, G.R., 2013. Smoothed particle hydrodynamics modeling of linear shaped charge with jet formation and penetration effects. Computers & Fluids. 86, 77-85

Foster, N. and Fedkiw, R., 2001. Pratical Animation of Liquids. http://physbam.stanford.edu/~fedkiw/papers/stanford2001-02.pdf

Fretwell, P., LaRue, M., Morin, P., Kooyman, G., Wienecke, B., Ratcliffe, N., Fox, A., Fleming ,A., Porter, C., Trathan, P., 2012. An Emperor Penguin Population Estimate: The First Global, Synoptic Survey of a Species from Space. DOI: 10.1371/journal.pone.0033751

Fries, T.P., Matthies, H.G., 2003. Classification and overview of meshfree methods. Technical Report Scientific Computing, Technical University, Germany

Fuller, M., 2010. The application of Smooth Particle Hydrodynamics to the Modeling of Solid Materials. https://lra.le.ac.uk/bitstream/2381/8478/1/2010PhD_MDFuller.pdf

Gilbert C, Blanc S, Le Maho Y, Ancel A. Review: Energy saving processes in huddling emperor penguins: from experiments to theory. Journal of Experimental Biology. 2007; 211: 1-8

Gilbert, C., McCafferty, D., Le Maho, Y., Martrette, J., Giroud, S., Blanc, S., Ancel, A., 2010. One for all and all for one: the energetic benefits of huddling in endotherms. Biological Reviews. 85, 545-569

Gilbert, C., Robertson, G., 2008. How do weather conditions affect the huddling behavior of emperor penguins. Polar Biology. 31,163-169

Gilbert. C., Robertson, G., Le Maho, Y., Naito, Y., Ancel, A., 2006. Huddling behavior in emperor penguins: dynamics of huddling. Physiology & Behavior. 88, 479-488

Gingold, R.A., Monaghan, J.J., 1977. Smoothed particle hydrodynamics –Theory and application to non-spherical stars. Monthly Notices of the Royal Astronomical Society. 181, 375-389

Glancy J, Stone J.V., Wilson S.P., 2016. How self-organization can guide evolution. Royal Society open science. 3:160553. http://dx.doi.org/10.1098/rsos.160553

Glancy, J., Grob, R., Stone, J.V., Wilson, S.P., 2015. A Self-Organizing Model of Thermoregulatory Huddling. Plos Computational Biology. 11(9):e1004283

Gotoh, H. and Sakai, T., 2006. Key issues in the particle method for computation of wave breaking. Coastal Engineering Journal. 53,171-179

Gotoh, H., Shao, S.D., Memita, T., 2004. SPH-LES model for numerical investigation of wave interaction with partially immersed breakwater. Coastal Engineering Journal. 46(1), 39-63

Gu, Y.T., Liu, G.R., 2006. Meshless techniques for convection dominated problems. Computational Mechanics. 28:171-182

Gu, Y., Wang, L., Chen, W., Zhang, C., He, X.Q., 2017. Application of the meshless generalized finite difference method to inverse heat source problems. International Journal of Heat and Mass Transfer. 108:721-729

Haddad, B., Pastor, M., Palacios, D., Muñoz-Salinas, E., 2010. A SPH depth integrated model for Popocatépetl 2001 lahar (Mexico): Sensitivity analysis and runout simulation. Engineering Geology. 114 (3-4), 312-329

Hagey T.J., Puthoff J.B., Crandell K.E., Autumn K., Harmon L.J., 2016. Modeling observed animal performance using the Weibull distribution. Journal of Experimental Biology. 219, 1603-1607

Haig, D., 2008. Huddling: Brown Fat, Genomic Imprinting and the Warm Inner Glow. Current Biology. 18(4), 172-174

Hosseini, S.M., Feng, J.J., 2011. Pressure boundary conditions for computing incompressible flows with SPH. Journal of Computational Physics. 230, 7473-7487

Hu, X.Y. and Adams, N.A., 2007. An incompressible multi-phase SPH method. Journal of Computational Physics. 227(1), 264-278

Huang, Y., Dai, Z., 2014. Large deformation and failure simulations for geo-disasters using smoothed particle hydrodynamics method. Engineering Geology. 168, 86-97

Huang, Y., Zhang, W.J., Mao, W.W., Jin, C., 2011. Flow analysis of liquefied soil based on smoothed particle hydrodynamics. Natural Hazards. 59(3), 1547-1560

Ianniello S.,, Di Mascio A., 2010. A self-adaptive oriented particles Level-Set method for tracking interfaces. Journal of Computational Physics. 229, 1353-1380

Jeppson, R.W., 1972. Limitations of Some Finite Difference Methods in Solving the Strongly Nonlinear Equation of Unsaturated Flow in Soils. https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1303&context=water_rep

Jin, H., Ding, X., 2005. On criterions for smoothed particle hydrodynamics kernels in stable field. Journal of Computational Physics. 202 (2), 699-709

Kachani, S., Perakis, G., 2004. Fluid Dynamics Models and their Applications in Transportation and Pricing. European Journal of Operational Research. 170(2),496-517

Karekal, S., Das, R., Mosse, L., Leary, P.W., Application of a mesh-free continuum method for simulation of rock caving processes. International Journal of Rock Mechanics and Mining Science. 48(5), 703-711

Khayyer, A., Gotoh, H., and Shao, S.D., 2008. Corrected Incompressible SPH method for accurate water-surface tracking in breaking waves. Coastal Engineering. 55,236-250

Khorasanizade, S., Sousa, J.M.M., 2015. A two-dimensional Segmented Boundary Algorithm for complex moving solid boundaries in Smoothed Particle Hydrodynamics. Computer Physics Communications. In Press

Kotze, J., Bennett, N.C., Scantlebury, M., 2008. The energetics of huddling in two species of mole rat (*rodentia; Bathyergidae*). Physiology & Behavior. 93 (1-2), 215-221

Lee, E.S., Moulinec, C., Xu. R., Laurence, D., Stansby, P., 2008. Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method. Journal of Computational Physics. 227 (18), 8417-8436

Leroch, S., Varga, M., Eder, S.J., Vernes, A., Rodriguez Ripoll, M., Ganzenmüller, G., 2015. Smoothed particle hydrodynamics simulation of damage induced by a spherical indenter scratching a viscoplastic material, International Journal of Solids and Structures. In press

Li, J., Liu, H., Gong, K., Tan, S.K., Shao, S., 2012. SPH modeling of solitary wave fissions over uneven bottoms. Coastal engineering. 60, 261-275

Longshaw, S.M., Rogers, B.D., 2015. Automotive fuel cell sloshing under temporally and spatially varying high acceleration using GPU-based Smoothed Particle Hydrodynamics. Advances in Engineering Software. 83, 31-44

Losasso, F., 2004. Simulating Water and Smoke with an Octree Data Structure. http://physbam.stanford.edu/~fedkiw/papers/stanford2004-02.pdf

Lu, Y., Wang, Z.Q., Chong, K.R., 2005. A comparative study of buried structure in soil subjected to blast load using 2D and 3D numerical simulations. Soil Dynamics and Earthquake Engineering. 25(4), 275-288

Lucy, L.B., 1977. A numerical approach to the testing of the fission hypothesis. Astronomical Journal. 82, 1013-1024

Macia, F., Antuono, M., Gonzalez, L.M., Colagrossi, A., 2011. Theoretical Analysis of the No-slip Boundary Condition Enforcement in SPH methods. Progress of Theoretical Physics. 125 (6), 1091-1121

Marrone S., Di Mascio A., Letouze D.,2015. Coupling of smoothed particle hydrodynamics with finite volume method for free-surface flows. Journal of Computational Physics. 310,161-180. doi:10.1016/j.jcp.2015.11.059

McDougall, S., Hungr, O., 2004. A model for the analysis of rapid landslide motion across three-dimensional terrain. Canadian Geotechnical Journal. 41(6), 1084-1097

Monaghan, J.J., 2005. Smoothed Particle Hydrodynamics. Reports on Progress in Physics. 68, 1703-1759 doi:10.1088/0034-4885/68/8/R01

Monaghan, J.J., Kajtar, J.B., 2009. SPH particle boundary forces for arbitrary boundaries. Computer Physics Communications. 180, 1811-1820

Napoli, E., Marchis, M.D., Vitanza, E., 2015. PANORMUS-SPH A New Smoothed Particle Hydrodynamics solver for incompressible flows. Computers & Fluids. 106, 185-195

Neethling, S.J., Barker, D.J., 2015. Using SPH to model multiphase mineral processing systems, Mineral Engineering. In Press

Peiro, J., Sherwin, S., 2005. Finite Difference, Finite Element and Finite Volume Methods for Partial Differential equations. Handbook of Materials Modeling. I, 1-32

Putaala, A., Hohtola, E., Hissa, R., 1995. The effect of group size on metabolism in huddling grey partridge (*Perdix perdix*). Comparative Biochemistry and Physiology Part B: Biochemistry and Molecular Biology. 111 (2), 243-247

Randles, P.W. and Libersky, L.D., 1996. Smoothed Particle Hydrodynamics: Some Recent Improvements and Applications. Computer Methods in Applied Mechanics and Engineering. 139, 375-408 http://dx.doi.org/10.1016/S0045-7825(96)01090-0

Rodriguez-Paz, M., Bonet, J., 2005. A Corrected SPH formulation of the shallow-water equations. Computers & Structures. 83(17-18), 1396-1410

Rusli, N., Kasiman, E.H., Hong, A.K.B., Yassin, A.Y.M., Amin, N., 2011. Numerical computation of a Two-Dimensional Navier-Stokes Equation using an Improved Finite Difference Method. 27,1-9

Sauder, K., Prakash, M., Cleary, P.W., Cordell, M., 2014. Application of Smoothed Particle Hydrodynamics for modeling gated spillway flows. Applied Mathematical Modeling. 38, 4308-4322

Schank, J.C., Alberts, J.R., 1997. Self-organized Huddles of Rat Pups by Simple Rules of Individual Behavior. Journal of Theoretical Biology. 189 (1), 11-25

Segal, I.A., 2015. Finite element methods for the incompressible Navier-Stokes equations. Delft University of Technology. http://ta.twi.tudelft.nl/users/vuik/burgers/fem_notes.pdf

Seibold B., 2008. Massachustts Institute of Technology. Available from: http://math.mit.edu/~gs/cse/codes/mit18086_navierstokes.pdf

Shao, S., Ji, C., Graham, D.I., Reeve, D.E., James, P.W., Chadwick, A.J., 2006. Simulation of wave overtopping by an incompressible SPH model. Coastal Engineering. 53, 723-735

Shao, S.D., and Lo, E.Y., 2003. Incompressible SPH Method of simulating Newtonian and non-Newtonian flows with a free surface. Advances in Water Resources. 26, 787-800

Shao, S.D., Gotoh, H., 2005. Turbulence particle models for tracking free surfaces. Journal of Hydraulic Research. 43(3), 276-289

Sirotkin, F.V., Yoh, J.J., 2013. A smoothed particle hydrodynamics method with approximate Riemann solvers for simulation of strong explosions. Computers & Fluids. 88, 418-429

Spreng, F., Eberhard, P., 2015. Machining Process Simulation with Smoothed Particle Hydrodynamics. Procedia CIRP. 31, 94-99

Taddei, L., Awoukeng Goumtcha, A., Roth, S., 2015. Smoothed particle hydrodynamics formulation for penetrating impacts on ballistic gelatin. Mechanics Research Communications. 70, 94-101

Takaffoli, M., Papini, M., 2012. Material deformation and removal due to single particle impacts on ductile materials using smoothed particle hydrodynamics. Wear. 274-275, 50-59

Takeda, H., Miyama, S.M., Sekiya, M., 1994. Numerical Simulation of Viscous Flow by Smoothed Particle Hydrodynamics. Progress of Theoretical Physics. 92, 939-960

Tartakovsky, A.M., Panchenko, A., 2016. Pairwise Force Smoothed Particle Hydrodynamics model for multiphase flow: Surface tension and contact line dynamics. Journal of Computational Physics. 305, 1119-1146

Tofighi, N., Ozbulut, M., Rahmat, A., Feng, J.J., Yildiz, M., 2015. Journal of Computational Physics. 297, 207-220

Tomlinson, S., Withers, P.C., Maloney, S.K., 2014. Huddling behavior and energetics of Sminthopsis spp. (*Marsupialia, Dasyruidae*) in response to environmental challenge. Physiology & Behavior. 128, 9-15

Tong, M., Browne, D.J., 2014. An incompressible multi-phase smoothed particle hydrodynamics method for modeling thermocapillary flow. International Journal of Heat and Mass Transfer. 73, 284-292

Valdez-Balderas, D., Domínguez, J.M., Rogers, B., Crespo, A.J.C., 2013. Towards accelerating smoothed particle hydrodynamics simulations for free-surface flows on multi-GPU clusters. Journal of Parallel and Distributed Computing. 73 (11), 1483-1493

Violeau, D., 2012. Fluid Mechanics and the SPH Method -Theory and Applications. Oxford University Press, Oxford

Wang, Y.F., Yang, Z.G., 2008. A coupled finite element and meshfree analysis of erosive wear. Tribology International. 42 (2), 373-377

Waters, A., Blanchetter, F., Kim, A.D., 2012. Modeling Huddling Penguins. PloS One 7(11):e50277.doi:10.1371/jounal.pone.0050277

Watts, P., Mitchell, E.J., Swartz, S.M., 2001. A computational model for estimating the mechanics of horizontal flapping flight in bats: model description and validation. Journal of Experimental Biology. 204, 2873-2898

Wilson S.P.,2017. Self-organized criticality in the evolution of a thermodynamic model of rodent thermoregulatory huddling. PLoS Computational Biology. 13(1):e1005378. Doi:10.1371/journal.pcbi.1005378

Xia, X., Liang, Q., 2016. A GPU-accelerated smoothed particle hydrodynamics model for the shallow water equations. Environmental Modeling & Software. 75, 28-43

Yildiz, M., Rook, R.A., Suleman, A., 2009. SPH with the multiple boundary tangent method. International Journal for Numerical Methods in Engineering. 77, 1416-1438

Zhang, T., Huang, Y.J., Liang, L., Fan, C.M., Li, P.W., 2018. Numerical solutions of mild slope equation by generalized finite difference method. Engineering Analysis with Boundary Elements. 88, 1-13

Zitterbat, D., Wienecke, B.,Butlr, J.P., Fabry, B., 2011. Coordinated movements prevent jamming in an emperor penguin huddle, PLoS ONE 6(6): e20260.doi:10.1371/journal.pone.0020260

APPENDIX A

The movement and evolution of the wind and the penguin huddling process are governed by the Navier-Stokes equations (NSE) of fluid motion in a continuum.

$$\rho\left(\frac{\partial}{\partial t}\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \mu\nabla^2\boldsymbol{u} + \boldsymbol{b} \tag{24}$$

$$\nabla\cdot\boldsymbol{u} = 0 \tag{25}$$

Wind field simulation is only considered in two dimensions and can be expressed in the FDM as:

$$u_t + p_x = -(u^2)_x - (uv)_y + \gamma(u_{xx} + u_{yy}) \tag{26}$$

$$v_t + p_y = -(v^2)_y - (uv)_x + \gamma(v_{xx} + v_{yy}) \tag{27}$$

$$u_x + v_y = 0 \tag{28}$$

The two momentum equations (Eq. 26, Eq. 27) describe the time evolution of the wind velocity field under inertial and viscous forces, where *u* and *v* are flow velocity (m/s^2) in two directions, u_x, u_y, v_x, v_y are gradients of *u* and *v* in two directions; $u_{xx}, u_{yy}, v_{xx}, v_{yy}$ are divergences of gradient in two directions; and p_x, p_y are gradients of pressure (Pa) in two directions.

The Pressure Poisson equation (PPE) is used to solve the pressure term. The protocol can be found in Seibold (2008) and Cornthwaite (2013).

At time $n+1$ (the pressure that corresponds with the velocity at $n+1$)

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{n} + \Delta t(-\boldsymbol{u}^{n} \cdot \nabla \boldsymbol{u}^{n} - \frac{1}{\rho}\nabla p^{n+1} + \mu \nabla^{2} \boldsymbol{u}^{n}) \tag{29}$$

$$\nabla \cdot \boldsymbol{u}^{n+1} = \nabla \cdot \boldsymbol{u}^{n} + \Delta t(-\nabla \cdot (\boldsymbol{u}^{n} \cdot \nabla \boldsymbol{u}^{n}) - \frac{1}{\rho}\nabla^{2} p^{n+1} + \mu \nabla^{2} (\nabla \cdot \boldsymbol{u}^{n})) \tag{30}$$

Forcing $\nabla \cdot \boldsymbol{u}^{n+1} = 0$ to satisfy continuity, the passion equation for p at time $n+1$ is

$$\nabla^{2} p^{n+1} = \rho \frac{\nabla \cdot \boldsymbol{u}^{n}}{\Delta t} - \rho \nabla \cdot (\boldsymbol{u}^{n} \cdot \nabla \boldsymbol{u}^{n}) + \mu \nabla^{2} (\nabla \cdot \boldsymbol{u}^{n}) \tag{31}$$

By applying left array division in MATLAB, the velocity field can be updated after pressure correction.

APPENDIX B

```
clear all
close all
clc

% Defining grid in the horizontal direction and vertical direction and
% calculate the x step and y step
xmin = 0.0; xmax = 50.0; xnum = 200.0;
xstp =(xmax-xmin)/xnum;
ymin = 0.0; ymax = 100.0; ynum = 400.0;
ystp =(ymax-ymin)/ynum;
Np = 223;                   % Number of particles
UX = zeros(xnum,ynum);      % Current velocity matrix
UY = zeros(xnum,ynum);
Dist = zeros(xnum,ynum);
V  = zeros(xnum,ynum);      % total wind speed in each grid point
P  = zeros(xnum,ynum);      % Pressure variable matrix
p  = zeros(xnum,ynum);
xp = zeros(1,Np);           % Particle positions
yp = zeros(Np,1);           % Particle positions
dx = 8;                     % Define length between particles
hp = 5;                     % Particle smoothness value
hhp =1;
tol = 1;                    % Check the mass conservation shold  be less than
this tolerance
DistToObj = zeros(1,Np);
load xp;                    % Load saved particle position and initial wind
field
load yp;
% Defining Related Coefficient
D = 1.0;                    % Diffusion coefficient in m^2/s
timestep = 0.002;           % Time step not to exceed 0.2 seconds for 1 m grid
Vel = 5 ;                   % Initial wind velocity in m/s
rhoair = 1.4224;            % Density(kg/m3) of air at -25 degree C
mass = 0.5;                  % Penguin mass (constant)
height = 1.2;               % Average height of adult emperor penguin
C = 0.1040;                 % Thermal conductance of penguin (W/C)
CA = 0.0204 ;               % Thermal conductivity of air (W/mC)at-50 degree C
RadObj = 1;                 % Radius of the impediment object in grid size
k = .08;                    % Ideal gas equation Constance
a = 1;                      % File number
n = zeros(1,Np);            % Initial number of particle within smooth length
circle at the point of each particle


rhoparticle = zeros(1,Np); % Particle density
rhop = zeros(1,Np);         % Density of the particle smooth circle
```

```
rhop0 = zeros(1,Np);       % Initial particle density
rhopr = zeros(xnum,ynum); % Particle density at grid point
Ep = zeros(1,Np);          % Local Metabolic energy within smooth length
Epr = zeros(xnum,ynum);    % Local metabolic energy at grid point
Ei = zeros(1,Np);          % Particle metabolic energy
Eir = zeros(xnum,ynum);    % Particle metabolic energy at grid point
distance = zeros(1,Np);
DistToB1 = zeros(1,xnum); % Distance to boundaries
DistToB2 = zeros(1,xnum);
DistToB3 = zeros(1,ynum);
DistToB4 = zeros(1,ynum);
Tw = zeros(xnum,ynum);     % Wind chill temperature
Tp = zeros(1,Np)+273.15;   % Particle wind chill temperature in K
Pe = zeros(1,Np);          % Particle pressure
fp = zeros(1,Np);          % pressure acceleration
up = zeros(1,Np);          % Particle velocity
r = zeros(Np,Np);          % Distance between each particle
s = zeros(1,Np);           % Particle moving distance
delT = zeros(1,Np);        % Temperature change
delTr = zeros(xnum,ynum); % Temperature change at grid point
zeta = zeros(1,Np);        % Moving angle
coszeta1 = zeros(Np,1);
sinzeta1 = zeros(Np,1);
coszeta2 = zeros(Np,1);
sinzeta2 = zeros(Np,1);
w = zeros(1,Np);            % Kernel function
dw = zeros(1,Np);
ww = zeros(1,Np);           % Kernel function
dww = zeros(1,Np);
Ti = ones(1,Np)*37.7;       % Penguin body temperature is 37.7 degree C
Ta = -30*ones(xnum,ynum);  % Ambient temperature
volqr = 0.345;

% Initialize partile on the grid

xexact=xp;                  % x position of penguins at t=0
yexact=yp;                  % y position  of penguins at t=0

% Defining the maximum number of iterations for convergence
maxiter  = 300000;

for iter = 1:maxiter
    iter                    % iteration numbers
    err = 0.0;

    % Define the boundary condition for the current velocity matrices
    % Set upwind x-boundary to normal velocity of applied wind
    UX0 = UX;                  % Previous time velcoity matrix
    UY0 = UY;
    % Populate the pressure matrix using velocites from previous step
    Lp = kron(speye(ynum),K1(xnum,xstp,1))+kron(K1(ynum,ystp,1),speye(xnum));
    Lp(1,1) = 3/2*Lp(1,1);
    perp = symamd(Lp); Rp = chol(Lp(perp,perp)); Rpt = Rp';
    % upper boundary velocity update

    UX(1,:) = 0;
```

```
    UY(1,:) = Vel;
    Tw(1,:) = 13.12 + 0.6215.*Ta(1,:) - 11.37*Vel^0.16 +
0.3965.*Ta(1,:)*Vel^0.16;
    % Calculate wind velocity at interior points for x and y fr om 2 to (max-
1)
    if (iter ==1 ||iter == maxiter)
        for xn = 2:xnum-1
            for yn = 2:ynum-1

                for ip = 1:Np
                    DistToObj(ip) = sqrt((xn-xp(ip))^2+(yn-yp(ip))^2) ;
%Distance from current grid point to impeding object 1
                    Dist(xn,yn) = min(DistToObj);
                end
            end
        end

        %DistToObj1 = sqrt((xn-ObjX1)^2+(yn-ObjY1)^2);
        for xn = 2:xnum-1
            for yn = 2:ynum-1
                Ta = -30*ones(xnum,ynum);
                if Dist(xn,yn) <= RadObj  % If the grid point is covered by
an object, define the velocity and pressure to be zero
                    UX(xn,yn) = 0;
                    UY(xn,yn) = 0;
                    Tw(xn,yn) = Ta(xn,yn);
                else

                    U2XY = (UX0(xn-1,yn)-
2.0*UX0(xn,yn)+UX0(xn+1,yn))/(ystp)^2; %Central difference of 2nd derivative
of x-velocity in x-direction
                    U2XX = (UX0(xn,yn-1)-
2.0*UX0(xn,yn)+UX0(xn,yn+1))/(xstp)^2; %Central difference of 2nd derivative
of x-velocity in y-direction
                    U1XY = (UX0(xn+1,yn)-UX0(xn-1,yn))/(2*ystp);
%Central difference of 1st derivative of x-velcoity in x-direction
                    U1XX = (UX0(xn,yn+1)-UX0(xn,yn-1))/(2*xstp);
%Central difference of 1st derivative of x-velcoity in y-direction
                    U2YX = (UY0(xn,yn-1)-
2.0*UY0(xn,yn)+UY0(xn,yn+1))/(xstp)^2; %Central difference of 2nd derivative
of y-velocity in y-direction
                    U2YY = (UY0(xn-1,yn)-
2.0*UY0(xn,yn)+UY0(xn+1,yn))/(ystp)^2; %Central difference of 2nd derivative
of y-velocity in x-direction
                    U1YX = (UY0(xn,yn+1)-UY0(xn,yn-1))/(2*xstp);
%Central difference of 1st derivative of y-velcoity in y-direction
                    U1YY = (UY0(xn+1,yn)-UY0(xn-1,yn))/(2*ystp);
%Central difference of 1st derivative of y-velocity in x-direction
                    UX(xn,yn) = UX0(xn,yn) + timestep*(D*(U2XX + U2XY) -
UX0(xn,yn)*U1XX - UY0(xn,yn)*U1XY);
                    UY(xn,yn) = UY0(xn,yn) + timestep*(D*(U2YY + U2YX) -
UY0(xn,yn)*U1YY - UX0(xn,yn)*U1YX);
                    U1XXcurrent = (UX(xn,yn)-UX(xn,yn-1))/xstp;
                    U1YYcurrent = (UY(xn,yn)-UY(xn-1,yn))/ystp;
                    err = max(err,timestep*abs(U1XXcurrent+U1YYcurrent));
```

```
                    V(xn,yn) = sqrt(UX(xn,yn)^2 + UY(xn,yn)^2);

                end

            end % End inner loop

        end %outer loop
        % left and right boudanry velocity update
        for xn = 2: xnum-1
            UX(xn,1) = 0;
            UX(xn,ynum) = 0;
            U2YXy1 = (UY0(xn,1)-2.0*UY0(xn,2)+UY0(xn,3))/(xstp)^2; %Forward
difference of 2nd derivative of y-velocity in y-direction
            U2YYy1 = (UY0(xn-1,1)-2.0*UY0(xn,1)+UY0(xn+1,1))/(ystp)^2;
%Central difference of 2nd derivative of y-velocity in x-direction
            U1YXy1 = (UY0(xn,2)-UY0(xn,1))/xstp;                %Forward
difference of 1st derivative of y-velcoity in y-direction
            U1YYy1 = (UY0(xn+1,1)-UY0(xn-1,1))/(2*ystp);
%Central difference of 1st derivative of y-velocity in x-direction
            U2YXynum = (UY0(xn,ynum)-2.0*UY0(xn,ynum-1)+UY0(xn,ynum-
2))/(xstp)^2; %Back difference of 2nd derivative of y-velocity in y-direction
            U2YYynum = (UY0(xn-1,ynum)-
2.0*UY0(xn,ynum)+UY0(xn+1,ynum))/(ystp)^2; %Central difference of 2nd
derivative of y-velocity in x-direction
            U1YXynum = (UY0(xn,ynum)-UY0(xn,ynum-1))/xstp;
%Back difference of 1st derivative of y-velcoity in y-direction
            U1YYynum = (UY0(xn+1,ynum)-UY0(xn-1,ynum))/(2*ystp);
%Central difference of 1st derivative of y-velocity in x-direction

            UY(xn,1) = UY0(xn,1) + timestep*(D*(U2YYy1 + U2YXy1) -
UY0(xn,1)*U1YYy1 - UX0(xn,1)*U1YXy1);
            UY(xn,ynum) =UY0(xn,ynum) + timestep*(D*(U2YYynum + U2YXynum) -
UY0(xn,ynum)*U1YYynum - UX0(xn,ynum)*U1YXynum);
            V(xn,1) = sqrt(UX(xn,1)^2 + UY(xn,1)^2);
% Calculate average wind speed affecting particle

            if V(xn,1) >= 1.3
                Tw(xn,1) = 13.12 + 0.6215.*Ta(xn,1) - 11.37.*V(xn,1).^0.16 +
0.3965.*Ta(xn,1).*V(xn,1).^0.16;
            else
                Tw(xn,1) = Ta(xn,1);
            end
            V(xn,ynum) = sqrt(UX(xn,ynum)^2 + UY(xn,ynum)^2);
% Calculate average wind speed affecting particle

            if V(xn,ynum) >= 1.3
                Tw(xn,ynum) = 13.12 + 0.6215.*Ta(xn,ynum) -
11.37.*V(xn,ynum).^0.16 + 0.3965.*Ta(xn,ynum).*V(xn,ynum).^0.16;
            else
                Tw(xn,ynum) = Ta(xn,ynum);
            end
        end
        % bottom boundary velocity update
        for yn = 2: ynum-1
```

```
            U2XXxum = (UX0(xnum,yn+1)-2.0*UX0(xnum,yn)+UX0(xnum,yn-
1))/(xstp)^2; %Central difference of 2nd derivative of y-velocity in y-
direction
            U2XYxum = (UX0(xnum-2,yn)-2.0*UX0(xnum-
1,yn)+UX0(xnum,yn))/(ystp)^2; %Central difference of 2nd derivative of y-
velocity in x-direction
            U1XXxum = (UX0(xnum,yn+1)-UX0(xnum,yn-1))/(2*xstp);
%Central difference of 1st derivative of y-velcoity in y-direction
            U1XYxum = (UX0(xnum,yn)-UX0(xnum-1,yn))/(ystp);
%Central difference of 1st derivative of y-velocity in x-direction
            U2YYxum = (UY0(xnum-2,yn)-2.0*UY0(xnum-
1,yn)+UY0(xnum,yn))/(ystp)^2; %Central difference of 2nd derivative of x-
velocity in x-direction
            U2YXxum = (UY0(xnum,yn+1)-2.0*UY0(xnum,yn)+UY0(xnum,yn-
1))/(xstp)^2; %Central difference of 2nd derivative of x-velocity in y-
direction
            U1YYxum = (UY0(xnum,yn)-UY0(xnum-1,yn))/(ystp);
%Central difference of 1st derivative of x-velcoity in x-direction
            U1YXxum = (UX0(xnum,yn+1)-UX0(xnum,yn-1))/(2*xstp);
%Central difference of 1st derivative of x-velcoity in y-direction
            UY(xnum,yn) = UY0(xnum,yn) + timestep*(D*(U2YYxum + U2YXxum) -
UY0(xnum,yn)*U1YYxum - UX0(xnum,yn)*U1YXxum );
            UX(xnum,yn) = UX0(xnum,yn) + timestep*(D*(U2XXxum + U2XYxum) -
UX0(xnum,yn)*U1XXxum - UY0(xnum,yn)*U1XYxum );
            V(xnum,yn) = sqrt(UX(xnum,yn)^2 + UY(xnum,yn)^2);
% Calculate average wind speed affecting particle

            if  V(xnum,yn) >= 1.3
                Tw(xnum,yn) = 13.12 + 0.6215.*Ta(xnum,yn) -
11.37.*V(xnum,yn).^0.16 + 0.3965.*Ta(xnum,yn).*V(xnum,yn).^0.16;
            else
                Tw(xnum,yn) = Ta(xnum,yn);
            end
        end
        % Left Bottom Boundary point and right Bottem boundary point
        UX(xnum,ynum)=0;
        UX(xnum,1) = 0;
        U1XX1 = (UX0(xnum,2)-UX0(xnum,1))/xstp;
        U1YY1 = U1XX1;
        UY(xnum,1) = U1YY1*ystp+ UY(xnum-1,1);
        U1XXnum =(UX0(xnum,ynum)-UX0(xnum,ynum-1))/xstp;
        U1YYnum = U1XXnum;
        UY(xnum,ynum) = U1YYnum*ystp+ UY(xnum-1,ynum);
        V(xnum,ynum) = sqrt(UX(xnum,ynum)^2 + UY(xnum,ynum)^2);
% Calculate average wind speed affecting particle

        if V(xnum,ynum) >= 1.3
            Tw(xnum,ynum) = 13.12 + 0.6215.*Ta(xnum,ynum) -
11.37.*V(xnum,ynum).^0.16 + 0.3965.*Ta(xnum,ynum).*V(xnum,ynum).^0.16;
        else
            Tw(xnum,ynum) = Ta(xnum,ynum);
        end
        V(xnum,1) = sqrt(UX(xnum,1)^2 + UY(xnum,1)^2);
% Calculate average wind speed affecting particle

        if V(xnum,1) >= 1.3
```

```
            Tw(xnum,1) = 13.12 + 0.6215.*Ta(xnum,1) - 11.37.*V(xnum,1).^0.16
+ 0.3965.*Ta(xnum,1).*V(xnum,1).^0.16;
        else
            Tw(xnum,1) = Ta(xnum,1);
        end
        %Pressure correction
        AA = UY(:,1);
        AB = UX(xnum,:);
        rhs =
reshape((diff([UX;AB])/xstp+diff([AA';UY'])'/ystp)*timestep,[],1);
        p(perp) = (-Rp\(Rpt\rhs(perp)));
        P = reshape(p,xnum,ynum);
        P(1,1) = 3/2*P(1,1);
        if iter >5000
            for i = 1:35
                for j = 1:25
                    if P(i,j)>P(36,1)
                        P(i,j)=P(36,1);
                    end
                end
            end
        end
        AC = diff(P);
        AD = diff(P')';
        AC(:,1)=0;
        AC(:,end)=0;
        AD(1,:) = 0;
        UX(2:end,:) = UX(2:end,: )-AC/xstp*0.8;
        UY(:,1:end-1) = UY(:,1:end-1)-AD*0.8;
    end


    if (iter ==1 ||mod(iter,15000)==0)

        Epr = zeros(xnum,ynum);
        Eir = zeros(1,Np);
        rhopr = zeros(xnum,ynum);
        rr = zeros(1,Np);
        drr = zeros(1,Np);
        m = zeros(xn,yn);
        n = zeros(1,Np);
        Pe = zeros(1,Np);          % Particle pressure
        bpX = zeros(1,Np);
        bpY = zeros(1,Np);
        upX = zeros(1,Np);          % Particle velocity in x direction
        upY = zeros(1,Np);          % Particle velocity in y direction
        bp = zeros(1,Np);
        fp = zeros(1,Np);
        fpX = zeros(1,Np);          % pressure acceleration in x direction
        fpY = zeros(1,Np);          % pressure acceleration in y direction
        rhoparticle = zeros(1,Np);
        Ep = zeros(1,Np);
        Tp = zeros (1,Np);
        volq = zeros(1,Np);
        Ei = zeros(1,Np);
        A = zeros(1,Np);
```

```
        Ta = -30*ones(xnum,ynum); % Ambient temperature
        for xn = 1:xnum
            for yn = 1:ynum

        for ip=1:Np
            for iq=1:Np
                if iq==ip
                    continue;
                end
                xpq = xp(ip)-xp(iq) ;
                ypq = yp(ip)-yp(iq);
                r(ip,iq) = sqrt(xpq^2+ypq^2);
                hhpq = hhp;
                [w(ip,iq),dw(ip,iq),stopIter] =
penguin_kernel(hhpq,r(ip,iq));
                if w(ip,iq)>0
                    n(ip) = n(ip)+1;
                end
            end
        end
                for ip = 1:Np
                    xa = xn-xp(ip) ;
                    ya = yn-yp(ip);
                    ha = sqrt(xa^2+ya^2);
                    hpq = hp;
                    [rr(ip),drr(ip),stopIter] = penguin_kernel(hpq,ha);

                    if (rr(ip)>0)
                        A(ip) = n(ip).^(-1/4);
                        Eir(ip) = A(ip).*C*(Ti(ip)-Ta(xn,yn));
                        Epr(xn,yn) = Epr(xn,yn) + (Eir(ip)*volqr*rr(ip));
                    else
                        Epr(xn,yn) = Epr(xn,yn);
                    end
                end
                delTr(xn,yn) = Epr(xn,yn)./(CA*rhoair*pi*hp^2*height);
                Ta(xn,yn) = Ta(xn,yn) + delTr(xn,yn);
            end
        end
        for xn = 2:xnum-1
            for yn = 2:ynum-1
                T1X = (Ta(xn,yn+1)-Ta(xn,yn-1))/(2*xstp);
                T1Y = (Ta(xn+1,yn)-Ta(xn-1,yn))/(2*ystp);
                T2Y = (Ta(xn-1,yn)-2.0*Ta(xn,yn)+Ta(xn+1,yn))/(ystp)^2;
                T2X = (Ta(xn,yn-1)-2.0*Ta(xn,yn)+Ta(xn,yn+1))/(xstp)^2;
                Ta(xn,yn)=  Ta(xn,yn) + timestep*(D*(T2X + T2Y) -
UX(xn,yn)*T1X - UY(xn,yn)*T1Y);
                if V(xn,yn)>1.3
                    Tw(xn,yn) = 13.12 + 0.6215.*Ta(xn,yn) -
11.37.*(3.6*V(xn,yn)).^0.16 + 0.3965.*Ta(xn,yn).*(3.6*V(xn,yn)).^0.16;
                else
                    Tw(xn,yn) = Ta(xn,yn);
                end
            end
        end
```

```
        for ip=1:Np
            Vtem =[V(floor(xp(ip))-2,floor(yp(ip))-2),V(floor(xp(ip))-
2,ceil(yp(ip))+2),V(ceil(xp(ip))+2,floor(yp(ip))-
2),V(ceil(xp(ip))+2,ceil(yp(ip))+2)];
            if max(Vtem)>1.3
                Twtem = [Tw(floor(xp(ip))-2,floor(yp(ip))-
2),Tw(floor(xp(ip))-2,ceil(yp(ip))+2),Tw(ceil(xp(ip))+2,floor(yp(ip))-
2),Tw(ceil(xp(ip))+2,ceil(yp(ip))+2)];
                Tp(ip)=  mean(Twtem)+273.15;
            else
                Tp(ip) =
(Tw(floor(xp(ip)),floor(yp(ip)))+Tw(floor(xp(ip)),ceil(yp(ip)))+Tw(ceil(xp(ip
)),floor(yp(ip)))+Tw(ceil(xp(ip)),ceil(yp(ip))))/4+273.15;
            end
        end
        for ip = 1:Np
            CenterX = sum(xp)/Np;
            CenterY = sum(yp)/Np;
            if xp(ip)-CenterX ~=0
                coszeta1(ip)= (CenterX-xp(ip))/(sqrt(((xp(ip)-
CenterX)^2)+((yp(ip)-CenterY)^2)));
                sinzeta1(ip)= (CenterY-yp(ip))/(sqrt(((xp(ip)-
CenterX)^2)+((yp(ip)-CenterY)^2)));
            end
            if Tp(ip) <263.15
                bp(ip) = (-0.2354*Tp(ip)+62.507)*0.9;
                bpX(ip) = bp(ip)*coszeta1(ip);
                bpY(ip) = bp(ip)*sinzeta1(ip);
            else
               bpX(ip) = 0;
               bpY(ip) =0;
            end
            for iq = 1: Np
                if (iq==ip)
                    continue;
                else
                    if w(ip,iq)>0
                        Pe = ones(1,Np).*0.1;
                        density = mass/(pi*RadObj^2*height);
                        fpX(ip) = fpX(ip) -
mass.*((Pe(ip)/density.^2)+(Pe(iq)./density.^2)).*(xp(ip)-
xp(iq))/abs(r(ip,iq)).*dw(ip,iq)*10;
                        fpY(ip) = fpY(ip) -
mass.*((Pe(ip)/density.^2)+(Pe(iq)./density.^2)).*(yp(ip)-
yp(iq))/abs(r(ip,iq)).*dw(ip,iq)*10;
                    end
                end
            end
            upX(ip) = 50*timestep*(bpX(ip)+fpX(ip));
            upY(ip) = 50*timestep*(bpY(ip)+fpY(ip));
        end
    end
    for ip = 1:Np

        if xp(ip) > xnum-1
            xp(ip) = xp(ip) - xnum +2;
        elseif yp(ip)> ynum-1
```

```
            yp(ip) = yp(ip) - ynum +2;
        elseif xp(ip) < 1
            xp(ip) = xp(ip) + xnum -1;
        elseif yp(ip) < 1
            yp(ip) = yp(ip) + ynum-1 ;
        end
        xp(ip) = xp(ip) + upX(ip).*timestep; % Relocate new position of
particle
        yp(ip) = yp(ip) + upY(ip).*timestep;
        UX(round(xp(ip)),round(yp(ip)))=0;
        UY(round(xp(ip)),round(yp(ip)))=0;
    end

    err
    UX0=UX;% Resetting current values for the next iteration
    UY0=UY;
    if (iter ==1 ||iter == maxiter)
        aberrorX = abs(UX0-UX);
        aberrorY = abs(UY0-UY);
        maxaberrorX = max(max(aberrorX));
        maxaberrorY = max(max(aberrorY));

        while max(maxaberrorX,maxaberrorY)> 0.05
            for xn = 2:xnum-1
                for yn = 2:ynum-1
                    Ta = -30*ones(xnum,ynum);
                    if Dist(xn,yn) <= RadObj  % If the grid point is covered
by an object, define the velocity and pressure to be zero
                        UX(xn,yn) = 0;
                        UY(xn,yn) = 0;
                        Tw(xn,yn) = Ta(xn,yn);
                    else

                        U2XY = (UX0(xn-1,yn)-
2.0*UX0(xn,yn)+UX0(xn+1,yn))/(ystp)^2; %Central difference of 2nd derivative
of x-velocity in x-direction
                        U2XX = (UX0(xn,yn-1)-
2.0*UX0(xn,yn)+UX0(xn,yn+1))/(xstp)^2; %Central difference of 2nd derivative
of x-velocity in y-direction
                        U1XY = (UX0(xn+1,yn)-UX0(xn-1,yn))/(2*ystp);
%Central difference of 1st derivative of x-velcoity in x-direction
                        U1XX = (UX0(xn,yn+1)-UX0(xn,yn-1))/(2*xstp);
%Central difference of 1st derivative of x-velcoity in y-direction
                        U2YX = (UY0(xn,yn-1)-
2.0*UY0(xn,yn)+UY0(xn,yn+1))/(xstp)^2; %Central difference of 2nd derivative
of y-velocity in y-direction
                        U2YY = (UY0(xn-1,yn)-
2.0*UY0(xn,yn)+UY0(xn+1,yn))/(ystp)^2; %Central difference of 2nd derivative
of y-velocity in x-direction
                        U1YX = (UY0(xn,yn+1)-UY0(xn,yn-1))/(2*xstp);
%Central difference of 1st derivative of y-velcoity in y-direction
                        U1YY = (UY0(xn+1,yn)-UY0(xn-1,yn))/(2*ystp);
%Central difference of 1st derivative of y-velocity in x-direction
                        UX(xn,yn) = UX0(xn,yn) + timestep*(D*(U2XX + U2XY) -
UX0(xn,yn)*U1XX - UY0(xn,yn)*U1XY);
```

```
                        UY(xn,yn) = UY0(xn,yn) + timestep*(D*(U2YY + U2YX) -
UY0(xn,yn)*U1YY - UX0(xn,yn)*U1YX);
                        U1XXcurrent = (UX(xn,yn)-UX(xn,yn-1))/xstp;
                        U1YYcurrent = (UY(xn,yn)-UY(xn-1,yn))/ystp;
                        err = max(err,timestep*abs(U1XXcurrent+U1YYcurrent));

                        V(xn,yn) = sqrt(UX(xn,yn)^2 + UY(xn,yn)^2);
% Calculate average wind speed affecting particle

                        if V(xn,yn) >= 1.3
                            Tw(xn,yn) = 13.12 + 0.6215.*Ta(xn,yn) -
11.37.*V(xn,yn).^0.16 + 0.3965.*Ta(xn,yn).*V(xn,yn).^0.16;
                        else
                            Tw(xn,yn) = Ta(xn,yn);
                        end

                    end

                end % End inner loop

            end %outer loop
            % left and right boudanry velocity update
            for xn = 2: xnum-1
                UX(xn,1) = 0;
                UX(xn,ynum) = 0;
                U2YXy1 = (UY0(xn,1)-2.0*UY0(xn,2)+UY0(xn,3))/(xstp)^2;
%Forward difference of 2nd derivative of y-velocity in y-direction
                U2YYy1 = (UY0(xn-1,1)-2.0*UY0(xn,1)+UY0(xn+1,1))/(ystp)^2;
%Central difference of 2nd derivative of y-velocity in x-direction
                U1YXy1 = (UY0(xn,2)-UY0(xn,1))/xstp;                %Forward
difference of 1st derivative of y-velcoity in y-direction
                U1YYy1 = (UY0(xn+1,1)-UY0(xn-1,1))/(2*ystp);
%Central difference of 1st derivative of y-velocity in x-direction
                U2YXynum = (UY0(xn,ynum)-2.0*UY0(xn,ynum-1)+UY0(xn,ynum-
2))/(xstp)^2; %Back difference of 2nd derivative of y-velocity in y-direction
                U2YYynum = (UY0(xn-1,ynum)-
2.0*UY0(xn,ynum)+UY0(xn+1,ynum))/(ystp)^2; %Central difference of 2nd
derivative of y-velocity in x-direction
                U1YXynum = (UY0(xn,ynum)-UY0(xn,ynum-1))/xstp;
%Back difference of 1st derivative of y-velcoity in y-direction
                U1YYynum = (UY0(xn+1,ynum)-UY0(xn-1,ynum))/(2*ystp);
%Central difference of 1st derivative of y-velocity in x-direction

                UY(xn,1) = UY0(xn,1) + timestep*(D*(U2YYy1 + U2YXy1) -
UY0(xn,1)*U1YYy1 - UX0(xn,1)*U1YXy1);
                UY(xn,ynum) =UY0(xn,ynum) + timestep*(D*(U2YYynum + U2YXynum)
- UY0(xn,ynum)*U1YYynum - UX0(xn,ynum)*U1YXynum);
                V(xn,1) = sqrt(UX(xn,1)^2 + UY(xn,1)^2);                %
Calculate average wind speed affecting particle

                if V(xn,1) >= 1.3
                    Tw(xn,1) = 13.12 + 0.6215.*Ta(xn,1) -
11.37.*V(xn,1).^0.16 + 0.3965.*Ta(xn,1).*V(xn,1).^0.16;
                else
                    Tw(xn,1) = Ta(xn,1);
```

```
                end
                V(xn,ynum) = sqrt(UX(xn,ynum)^2 + UY(xn,ynum)^2);
% Calculate average wind speed affecting particle

                if V(xn,ynum) >= 1.3
                    Tw(xn,ynum) = 13.12 + 0.6215.*Ta(xn,ynum) -
11.37.*V(xn,ynum).^0.16 + 0.3965.*Ta(xn,ynum).*V(xn,ynum).^0.16;
                else
                    Tw(xn,ynum) = Ta(xn,ynum);
                end
            end
            % bottom boundary velocity update
            for yn = 2: ynum-1
                U2XXxum = (UX0(xnum,yn+1)-2.0*UX0(xnum,yn)+UX0(xnum,yn-
1))/(xstp)^2; %Central difference of 2nd derivative of y-velocity in y-
direction
                U2XYxum = (UX0(xnum-2,yn)-2.0*UX0(xnum-
1,yn)+UX0(xnum,yn))/(ystp)^2; %Central difference of 2nd derivative of y-
velocity in x-direction
                U1XXxum = (UX0(xnum,yn+1)-UX0(xnum,yn-1))/(2*xstp);
%Central difference of 1st derivative of y-velcoity in y-direction
                U1XYxum = (UX0(xnum,yn)-UX0(xnum-1,yn))/(ystp);
%Central difference of 1st derivative of y-velocity in x-direction
                U2YYxum = (UY0(xnum-2,yn)-2.0*UY0(xnum-
1,yn)+UY0(xnum,yn))/(ystp)^2; %Central difference of 2nd derivative of x-
velocity in x-direction
                U2YXxum = (UY0(xnum,yn+1)-2.0*UY0(xnum,yn)+UY0(xnum,yn-
1))/(xstp)^2; %Central difference of 2nd derivative of x-velocity in y-
direction
                U1YYxum = (UY0(xnum,yn)-UY0(xnum-1,yn))/(ystp);
%Central difference of 1st derivative of x-velcoity in x-direction
                U1YXxum = (UX0(xnum,yn+1)-UX0(xnum,yn-1))/(2*xstp);
%Central difference of 1st derivative of x-velcoity in y-direction
                UY(xnum,yn) = UY0(xnum,yn) + timestep*(D*(U2YYxum + U2YXxum)
- UY0(xnum,yn)*U1YYxum - UX0(xnum,yn)*U1YXxum );
                UX(xnum,yn) = UX0(xnum,yn) + timestep*(D*(U2XXxum + U2XYxum)
- UX0(xnum,yn)*U1XXxum - UY0(xnum,yn)*U1XYxum );
                V(xnum,yn) = sqrt(UX(xnum,yn)^2 + UY(xnum,yn)^2);
% Calculate average wind speed affecting particle

                if  V(xnum,yn) >= 1.3
                    Tw(xnum,yn) = 13.12 + 0.6215.*Ta(xnum,yn) -
11.37.*V(xnum,yn).^0.16 + 0.3965.*Ta(xnum,yn).*V(xnum,yn).^0.16;
                else
                    Tw(xnum,yn) = Ta(xnum,yn);
                end
            end
            % Left Bottom Boundary point and right Bottem boundary point
            UX(xnum,ynum)=0;
            UX(xnum,1) = 0;
            U1XX1 = (UX0(xnum,2)-UX0(xnum,1))/xstp;
            U1YY1 = U1XX1;
            UY(xnum,1) = U1YY1*ystp+ UY(xnum-1,1);
            U1XXnum =(UX0(xnum,ynum)-UX0(xnum,ynum-1))/xstp;
            U1YYnum = U1XXnum;
            UY(xnum,ynum) = U1YYnum*ystp+ UY(xnum-1,ynum);
```

```
            V(xnum,ynum) = sqrt(UX(xnum,ynum)^2 + UY(xnum,ynum)^2);
% Calculate average wind speed affecting particle

            if V(xnum,ynum) >= 1.3
                Tw(xnum,ynum) = 13.12 + 0.6215.*Ta(xnum,ynum) -
11.37.*V(xnum,ynum).^0.16 + 0.3965.*Ta(xnum,ynum).*V(xnum,ynum).^0.16;
            else
                Tw(xnum,ynum) = Ta(xnum,ynum);
            end
            V(xnum,1) = sqrt(UX(xnum,1)^2 + UY(xnum,1)^2);                %
Calculate average wind speed affecting particle

            if V(xnum,1) >= 1.3
                Tw(xnum,1) = 13.12 + 0.6215.*Ta(xnum,1) -
11.37.*V(xnum,1).^0.16 + 0.3965.*Ta(xnum,1).*V(xnum,1).^0.16;
            else
                Tw(xnum,1) = Ta(xnum,1);
            end
            %Pressure correction
            AA = UY(:,1);
            AB = UX(xnum,:);
            rhs =
reshape((diff([UX;AB])/xstp+diff([AA';UY'])'/ystp)*timestep,[],1);
            p(perp) = (-Rp\(Rpt\rhs(perp)));
            P = reshape(p,xnum,ynum);
            P(1,1) = 3/2*P(1,1);
            if iter >5000
                for i = 1:35
                    for j = 1:25
                        if P(i,j)>P(36,1)
                            P(i,j)=P(36,1);
                        end
                    end
                end
            end
            AC = diff(P);
            AD = diff(P')';
            AC(:,1)=0;
            AC(:,end)=0;
            AD(1,:) = 0;
            UX(2:end,:) = UX(2:end,: )-AC/xstp;
            UY(:,1:end-1) = UY(:,1:end-1)-AD;
        end
    end
    if (mod(iter,300000)==0)

        filename = [ '20180316' num2str(a) '.mat' ];
        save(filename);
        a = a+1;
    end

end
plot(xp,yp,'or','MarkerSize',5); grid on; xlim([0 200]);ylim([0 400]);
drawnow;
subplot(1,2,1),surf(UX), xlabel('X'),ylabel('Y');
subplot(1,2,2),surf(UY), xlabel('X'),ylabel('Y');
```

```
x = xmin:xstp:xmax;
y = ymin:ystp:ymax;

set(gcf,'Renderer','zbuffer');
```