<table>
<thead>
<tr>
<th>Title</th>
<th>The Integration of Bounce and Resuspension into a Lagrangian Simulation of Particle Movement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Fergus, Alan</td>
</tr>
<tr>
<td>Publication Date</td>
<td>2010-09-30</td>
</tr>
<tr>
<td>Item record</td>
<td><a href="http://hdl.handle.net/10379/4885">http://hdl.handle.net/10379/4885</a></td>
</tr>
</tbody>
</table>

Downloaded 2018-12-29T12:30:14Z

Some rights reserved. For more information, please see the item record link above.
The Integration of Bounce and Resuspension into a Lagrangian Simulation of Particle Movement.

By

Alan Fergus, BSc.

September 30, 2010

School of Physics,
National University of Ireland, Galway.

Supervisors
Dr Miriam Byrne, School of Physics & Dr Aaron Golden, School of Mathematics, Statistics and Applied Mathematics; National University of Ireland Galway
Science is a wonderful thing if one does not have to earn one's living at it.

~Albert Einstein
Abstract

The phenomenon of particle transport in turbulent flows has been studied extensively using Direct Numerical Simulation (DNS) over the past few decades. One of the most useful DNS approaches is Lagrangian simulation. Lagrangian simulations of particles in turbulent flows have provided much information regarding particle deposition to surfaces and particle-turbulence interactions. Because individual particle trajectories are calculated in Lagrangian simulations, details of particle motion are accessible that are unattainable by experiment, thus making Lagrangian Modelling a highly powerful tool in particle transport simulation.

A Computational Fluid Dynamic program called OpenFOAM[1] is used as a base and modified to incorporate particle-surface interactions, bounce and re-entrainment. To model such behaviour, particles’ potential functions for a population of particles are developed, governing sticking and re-entrainment of particles. When a particle strikes a surface using most existing models, the particle adheres to the surface (perfect sink). In this work, there are two outcomes that occur which are dependent on the velocity of the particle when its impacts with the surface. At a low impact velocity the particle adheres, but as the impact velocity increases, the particle may bounce if the velocity of impact is greater than a critical velocity $V_{cr}$ defined as

$$V_{cr} = \left[ \frac{2E_a}{m} \right]^{\frac{1}{2}}$$

where $m$ is the mass of the particle, $E_a = E_m + E_s$ is the surface adhesion as detailed by Johnson et al. (1971)[2], and is the sum of the mechanical potential energy $E_m$ and the surface adhesion energy $E_s$. 
While the phenomenon of resuspension has been studied extensively on its own, very little work has been carried out that included resuspension in a standard Lagrangian model. In the present work, the critical velocity is calculated and if the critical velocity is less than the velocity of the flow at the position of the particle, the particle is resuspended into the flow at a velocity dependent on the particle properties and energy remaining after the particle has overcome the adhesion energy.

To validate the presented theory, three test cases were chosen. The first test case was based on the experimental deposition data for 33-36 micron particles, collected by Paw and Braaten (1992)[3]. It is evident that the simulated $V_{cr}$ is well within the values of ideal deposition and net deposition as presented by Paw and Braaten (1992)[3]. The second test case was based on the simulated work of Li and Ahmadi (1993)[4], who examined bounce for 1-10 micron particles and obtained a critical approach velocity above which particles bounced. When the critical approach velocity is compared with $V_{cr}$, it is clear that there is very good agreement for particle sizes above 4 µm. The final test case chosen was a sand transport simulation. When the sand in freefall test case is examined, it is found that the settling velocity agrees well with those values presented by Fentie et al. (2004)[5] and the spatial distribution in this test case, in a 5 m s$^{-1}$ wind, agrees well with the spatial distribution as presented by Kang and Guo (2006)[6], who modelled aeolian sand transport.
Acknowledgements

I would like to thank everyone who contributed to this work. I would like to thank the staff and my colleagues in the School of Physics, who were always on hand to aid me and to discuss problems as they occurred. In particular I would like to single out my supervisors Dr Miriam Byrne and Dr Aaron Golden who gave me constant encouragement, guidance and sponsorship.

I would also like to thank my parents, Padraig & Ann Fergus, my family and friends – without their support over the years I would have never reached this point. I cannot thank them enough for their unwavering support and patience. I would like to thank my partner Jenny Spollen whom I met during this work, for the unending encouragement and push to finish it.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>Contents</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
<tr>
<td>List of Code excerpts</td>
<td>xi</td>
</tr>
<tr>
<td>Nomenclature</td>
<td>xi</td>
</tr>
<tr>
<td>I Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Particle Deposition</td>
<td>2</td>
</tr>
<tr>
<td>1.1.1 Brownian Diffusion</td>
<td>2</td>
</tr>
<tr>
<td>1.1.2 Drag Force</td>
<td>3</td>
</tr>
<tr>
<td>1.1.3 Gravitational Force</td>
<td>3</td>
</tr>
<tr>
<td>1.1.4 Thermophoresis</td>
<td>4</td>
</tr>
<tr>
<td>1.1.5 Electrophoresis</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Modelling Approaches</td>
<td>5</td>
</tr>
<tr>
<td>1.2.1 Empirical Equations Method</td>
<td>5</td>
</tr>
<tr>
<td>1.2.2 Eulerian Modelling</td>
<td>6</td>
</tr>
<tr>
<td>1.2.3 Sublayer Modelling</td>
<td>6</td>
</tr>
<tr>
<td>1.2.4 Lagrangian Simulation</td>
<td>6</td>
</tr>
<tr>
<td>1.3 Particle Surface Interactions</td>
<td>7</td>
</tr>
<tr>
<td>1.3.1 Bounce</td>
<td>7</td>
</tr>
<tr>
<td>1.3.2 Resuspension</td>
<td>8</td>
</tr>
<tr>
<td>1.4 OpenFOAM</td>
<td>9</td>
</tr>
<tr>
<td>1.5 Outline of Objectives and Layout of Thesis</td>
<td>10</td>
</tr>
<tr>
<td>2 Overview</td>
<td>11</td>
</tr>
<tr>
<td>2.1 Lagrangian Simulation of Particle Transport</td>
<td>11</td>
</tr>
<tr>
<td>2.2 Particle Surface Interactions</td>
<td>18</td>
</tr>
<tr>
<td>2.3 Conclusions</td>
<td>25</td>
</tr>
</tbody>
</table>
5.2.7 Conclusions on the comparison of simulated particle behaviour vs. Paw and Braaten's experimental data. 85

5.3 Particle deposition: Li and Ahmadi test case. 86

5.3.1 Comparison of simulated data and the data of Li and Ahmadi (2 m s\(^{-1}\) flow velocity). 87

5.3.2 Comparison of simulated data and the data of Li and Ahmadi (3 m s\(^{-1}\) flow velocity). 92

5.3.3 Comparison of simulated data and the data of Li and Ahmadi (5 m s\(^{-1}\) flow velocity). 96

5.3.4 5 m s\(^{-1}\) Li and Ahmadi test case simulation with 'perfect sinks'. 102

5.3.5 5 m s\(^{-1}\) test case simulation comparing resuspension and no resuspension. 103

5.3.6 15 m s\(^{-1}\) simulation where bounce dominates. 104

5.3.7 Conclusions on the comparison of simulated particle behaviour vs. Li and Ahmadi's simulated data. 106

5.4 Sand 108

5.4.1 Test case 3: Sand sedimentation. 109

5.4.2 Comparison of simulated data with sand transport values for a flow velocity of 5 m s\(^{-1}\). 112

5.4.3 Conclusions. 118

6 Conclusions 119

6.1 Deposition 120

6.2 Bounce 122

6.3 Resuspension 122
### List of Figures

1.1 Diagram of the drag and gravitational forces acting on a particle. 4
1.2 A depiction of particle deposition and bounce phenomenon 8
1.3 A depiction of particle resuspension. 9
2.1 Coordinate system for solving two-dimensional particle equation of motion, from Kallio and Reeks (1989)[15]. 12
2.2 Particle concentration field from Brooke et al. (1994)[14] 14
2.3 Deposition velocity on plain glass plates compared to the terminal settling velocity for particles in still air, (a) wind velocity 2 m s\(^{-1}\), (b) wind velocity 1 m s\(^{-1}\), (c) wind velocity 0.5 m s\(^{-1}\), (d) terminal settling velocity of particles in still air (from Steen (1986)[30]). 19
2.4 Particle-surface for ‘Rock-n-Roll’ model as published by Reeks and Hall (2001)[34]. 20
2.5 Measured spatial aerosol particle velocity along one pitch length of a rib for 4.5 µm particle: (a) With paraffin oil applied to the ribs, and (b) without oil applied (from Lai (1997)[38]). 23
2.6 Variations of critical velocities for rolling detachment of silica-coated particles with particle diameter during the stepping down process at varying particle roughness, Δc 24
3.1 A diagram of the bounce process as presented in this work. 27
3.2 Diagram of a sphere impacting on a flat surface, with the contact radius shown. 28
3.3 A schematic of the outcomes after a particle collision with a flat surface. 30
3.4 A schematic of resuspension with 1) the particle resting on the surface and 2) the particle when V\(_{cr}\) is less than U\(_{posp}\). 31
4.1 The 3D mesh for Paw test case and the block structure. 34
4.2 Steps involved in adding Hard ball particles to the existing Lagrangian models in OpenFOAM[1]. 37
4.3 Comparison of the number of particles in the simulation under 'icoLagrangianFoam' and the presented work with respect to time. 43
4.4 An example of the visualisation produced by Paraview[47] and a graph of the deposition velocity of the particle, V\(_{elp}\) vs. the diameter of the particle, d\(_p\) produced by SigmaPlot[48]. 45
4.5 A flowchart of the implementation of bounce, deposition and Resuspension (part 1). 46
4.6 A flowchart of the implementation of bounce, deposition and Resuspension (part 2). 47
4.7 Development steps used to implement the code. 54
4.8 Execution times for the development steps of the present work. 59
4.9 A plot of the execution times of a test case with varying particle numbers simulation in serial and parallel. 62
4.10 Execution time and clock time for Li and Ahmadi test case with 5 m s\(^{-1}\) flow on a serial machine. 64
4.11 Execution time and clock time for Li and Ahmadi test case with 5 m s\(^{-1}\) flow in parallel on 2 processors. 64
5.1 Particle transport and flow for the Paw and Braaten test case in a 0.1 m x 0.1 m x 0.1 m box with a flow in the box induced by 5 m s\(^{-1}\) flow across the top. 68
5.2 A plot of the deposition velocity, V\(_{elp}\) vs. particle diameter, d\(_p\), showing the area bounded by the upper value of the critical velocity, V\(_{cr}\). 69
5.3 A plot of critical velocity, \( V_{cr} \) vs. particle diameter, \( d_p \). Included on the plot is the ideal deposition velocity, \( V_i \) and the net deposition velocity, \( V_d \) for a flow velocity of 2.82 m s\(^{-1}\) from Paw and Braaten (1992)\(^{[3]}\).

5.4 A plot of the particle deposition velocities for particles that bounced due to their \( V_{elp} \) being greater than \( V_{cr} \) and it shows the upper boundary of the \( V_{cr} \) values.

5.5 A plot of \( U_{posp} \) vs. \( d_p \). \( V_{cr} \) is shown on the plot.

5.6 A plot of the deposition velocity, \( V_{elp} \) vs. particle diameter, \( d_p \), showing the area bounded by the upper value of the critical velocity, \( V_{cr} \).

5.7 A plot of critical velocity, \( V_{cr} \), vs. particle diameter, \( d_p \). Included on the plot is the ideal deposition velocity, \( V_i \) and the net deposition velocity, \( V_d \) for a flow velocity of 3.41 m s\(^{-1}\) (from Paw and Braaten (1992)\(^{[3]}\)).

5.8 A plot of \( V_{elp} \) vs. particle diameter, \( d_p \) with \( V_{cr} \) values shown as an area plot.

5.9 A plot of the position velocity of the particle, \( U_{posp} \) vs. particle diameter, \( d_p \). Included on the plot is the \( V_{cr} \) region above which a particle resuspends.

5.10 A plot of the deposition velocity, \( V_{elp} \) vs. particle diameter, \( d_p \), showing the area bounded by the upper value of the critical velocity, \( V_{cr} \) for the 4.59 m s\(^{-1}\) simulation.

5.11 A plot of critical velocity, \( V_{cr} \), vs. particle diameter, \( d_p \). Included on the plot is the ideal deposition velocity, \( V_i \) and the net deposition velocity, \( V_d \) for a flow velocity of 4.59 m s\(^{-1}\) from Paw and Braaten (1992)\(^{[3]}\).

5.12 A plot of deposition velocity, \( V_{elp} \) vs. particle diameter, \( d_p \), with \( V_{cr} \) values shown as an area plot.

5.13 A plot of the velocity of the fluid at the position of the particle, \( U_{posp} \) vs. particle diameter, \( d_p \). Included on the plot is the \( V_{cr} \) region above which a particle resuspends.

5.14 A plot of the deposition velocity, \( V_{elp} \) vs. particle diameter, \( d_p \) for 'perfect sink' simulation. Included is an area plot of critical velocity, \( V_{cr} \).

5.15 \( V_{elp} \) for 2.82, 3.41 and 4.59 m s\(^{-1}\) when bounce occurs and at 4.59 m s\(^{-1}\) with perfect sinks.

5.16 Plot of \( U_{posp} \) from resuspension checks. With an area plot of \( V_{cr} \).

5.17 \( V_{elp} \) of bounce for Paw and Braaten test case.

5.18 Particle transport and flow for the Li and Ahmadi test case in a 0.1 m x 0.1 m x 0.1 m box with a flow in the box induced by 5 m s\(^{-1}\) flow across the top.

5.19 A plot of the deposition velocity, \( V_{elp} \) vs. particle diameter, \( d_p \), for flow rate of 2 m s\(^{-1}\). Included is an area plot of \( V_{cr} \).

5.20 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity, \( V_c \) vs. particle diameter, \( d_p \).

5.21 A plot of the deposition velocity, \( V_{elp} \) vs. particle diameter, \( d_p \), when bounce occurred. Included is an area plot of \( V_{cr} \).

5.22 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity, \( V_c \) vs. particle diameter, \( d_p \) for bounce at a flow velocity of 2 m s\(^{-1}\).

5.23 A plot of \( V_{elp} \) vs. \( d_p \) for resuspension at a flow velocity of 2 m s\(^{-1}\).

5.24 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity, \( V_c \) vs. particle diameter, \( d_p \), for resuspension at a flow velocity of 2 m s\(^{-1}\).

5.25 A plot of the deposition velocity, \( V_{elp} \) vs. particle diameter, \( d_p \) for deposition at a flow velocity of 3 m s\(^{-1}\).

5.26 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity, \( V_c \) vs. particle diameter, \( d_p \) for deposition at a flow velocity of 3 m s\(^{-1}\).
5.27 A plot of the deposition velocity, \( V_{dep} \) vs. particle diameter, \( d_p \) when bounce occurred at a flow velocity of 3 m s\(^{-1}\).

5.28 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity, \( V_c \) vs. particle diameter, \( d_p \) for bounce at 3 m s\(^{-1}\).

5.29 A plot of \( U_{poop} \) vs. \( d_p \) when resuspension occurred at a flow velocity of 3 m s\(^{-1}\).

5.30 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity, \( V_c \) vs. particle diameter, \( d_p \) for resuspension at 3 m s\(^{-1}\).

5.31 A plot of the deposition velocity, \( V_{dep} \) vs. particle diameter, \( d_p \) for deposition at a flow velocity of 5 m s\(^{-1}\).

5.32 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity calculated by Li and Ahmadi, \( V_c \) vs. particle diameter, \( d_p \).

5.33 A plot of the deposition velocity, \( V_{dep} \) vs. particle diameter, \( d_p \) when bounce occurred at a flow velocity of 5 m s\(^{-1}\).

5.34 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity, \( V_c \) vs. particle diameter, \( d_p \) for bounce at a flow velocity of 5 m s\(^{-1}\).

5.35 A plot of \( U_{poop} \) vs. \( d_p \) when resuspension occurred at a flow velocity of 5 m s\(^{-1}\).

5.36 A plot of the critical velocity, \( V_{cr} \) and critical approach velocity, \( V_c \) vs. particle diameter, \( d_p \) for resuspension at a flow velocity of 5 m s\(^{-1}\).

5.37 A plot of the deposition velocity, \( V_{dep} \) vs. particle diameter, \( d_p \) for perfect sink simulation.

5.38 A histogram of \( V_{dep} \) for flow velocities of 2, 3 and 5 m s\(^{-1}\) when bounce is included and at 5 m s\(^{-1}\) with 'perfect sinks'.

5.39 Plot of \( U_{poop} \) from resuspension checks, with an area plot of \( V_{cr} \).

5.40 \( V_{dep} \) of bounce for Li and Ahmadi test case.

5.41 Simulation of sand particles in freefall.

5.42 Settling velocity plotted against sediment size from Fentie et al. (2004)\[5\].

5.43 Plot of \( V_{dep} \) values for particles that deposit.

5.44 Plot of \( V_{dep} \) of particles that deposited with corresponding \( V_{cr} \).

5.45 Sand particles in a 5 m s\(^{-1}\) velocity flow.

5.46 A plot of the critical velocity, \( V_{cr} \) and deposition velocity, \( V_{dep} \) vs. particle diameter, \( d_p \).

5.47 A plot of the deposition velocity, \( V_{dep} \) vs. particle diameter, \( d_p \) when bounce occurred.

5.48 A plot of the \( V_{cr} \) and \( U_{poop} \) vs. \( d_p \) for resuspension.

5.49 Spatial distribution of sand particles at 3.96 s from Kang and Guo (2006)\[6\].

5.50 Spatial distribution of sand particles at 4.05 s from Kang and Guo (2006)\[6\].

5.51 The 'y' position data of the Sand particle at 3.96 s.

5.52 The 'y' position data of the Sand particle at 4.05s.

6.1 A screenshot from the Paw and Braaten test case showing near-wall concentration of particles.

6.2 A sequence of screenshots showing a particle bouncing, depositing and resuspending.
List of Tables.

4.1. Summary of Decomposition Methods. 36
4.2. Summary of CPU times for development steps test cases. 57
4.3. Execution times for the development steps for the presented work. 60
4.4. Execution times of a test case with varying particle numbers simulation in serial and parallel. 61
5.1. Summary of bounce, deposition and resuspension for the Paw and Braaten test case. 85
5.2. Summary of bounce, deposition and resuspension for the Li and Ahmadi test case. 106
5.3. Energy of adhesion values used in the calculation of $V_{cr}$ and $V_c$. 107
5.4. A summary of sand test case simulations. 118

List of Code excerpts.

4.1. Example of 'decomposeParDict' file. 35
4.2. Hard ball particle definition. 40
4.3. Injector code. 41
4.4. Track function that handles particle transport. 42
4.5. Original Boundary collisions code. 42
4.6. Code to output to a text file when bounce occurred. 44
4.7. 'icoLagrangianFoam' original handling of particles during an impact. 48
4.8. Modification made to keep particles in the simulation after an impact. 48
4.9. Original particle velocity calculation from 'icoLagrangianFoam'. 49
4.10. Modification made to account for particles which have deposited, when calculating the velocity of the particle. 49
4.11. Modification made to add resuspension to 'HardBallParticle'. 50
4.12. Code to calculate $V_{cr}$. 51
4.13. Implementation of bounce and deposition for the presented work. 53

Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS</td>
<td>Direct Numerical Simulation</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
</tr>
<tr>
<td>$V_{cr}$</td>
<td>critical particle velocity for bounce</td>
</tr>
<tr>
<td>$E_a$</td>
<td>energy of adhesion</td>
</tr>
<tr>
<td>$E_m$</td>
<td>mechanical potential energy</td>
</tr>
<tr>
<td>$E_s$</td>
<td>surface adhesion energy</td>
</tr>
<tr>
<td>$V_d$</td>
<td>deposition velocity</td>
</tr>
<tr>
<td>J</td>
<td>deposition flux</td>
</tr>
</tbody>
</table>
C∞  particle concentration
J_B  Brownian diffusive particle flux
D  Brownian diffusivity
k  Boltzmann constant \(1.38 \times 10^{-23}\) J K\(^{-1}\)
\(\mu\)  the viscosity of the fluid
\(d_p\)  the diameter of the particle
C_C  Cunningham slip correction factor
T  absolute temperature
Kn  Knudsen number
F_D  Drag force
\(\rho_f\)  density of the fluid
C_D  drag coefficient
u  fluid velocity
v  particle velocity
\(\vec{r}\)  the position vector of the particle
\(u_g\)  particle gravitational settling velocity
\(\tau_p\)  particle relaxation time
\(g\)  acceleration due to gravity
F_C  coulomb's force on a particle
q  charge on a particle
E  electric field strength
\(V_p\)\(^i\)  initial particle velocity
\(V_p\)\(^f\)  final particle velocity
\(U_{posp}\)  the velocity of the fluid at the position of the particle at the time of resuspension
\(V_{crit}\)  critical fluid velocity for resuspension
E  surface potential energy
\(V_c\)  critical approach velocity
\(c_r\)  coefficient of restitution
m  mass of the particle
r  radius of the particle
A  Hamaker constant
\(y_0\)  equilibrium separation of the particle
\(V_i\)  ideal deposition velocity
\(V_d\)  net deposition velocity
F  fraction of rebound
U  wind speed
\(\beta\)  resuspension rate
N  number of particles on the surface at time t
\(N_0\)  initial number of particles on a surface
dG  Van der Waals force
\(D_1/D_2\)  spherical particle diameters
h  separation distance
A_1  Hamaker constant for the particle
A_2  Hamaker constant for the surface
A_3  Hamaker constant for the fluid
\(V_y\)  elastic limiting velocity
K  mechanical constant
M_e  Young's modulus
\(y_1\)  elastic stress limit
E_{eel} energy stored in the elastic deformation
R_e contact radius
R_p plastic deformation contact radius
G_c contact geometry
F_1 sum of all the surface forces
F_0 gravitational force
G universal gravitational constant
3D three-dimensional
LAM Local Area Multicomputer
Velp the deposition velocity of the particle
$\mathbf{y}$ a unit vector in the y direction,
1. **Introduction**

This thesis introduces a method for the incorporation of bounce and resuspension into a Lagrangian simulation of particle transport. This chapter begins with a general introduction to particle deposition, then an introduction to particle transport modelling approaches and Lagrangian Simulation. This is followed by an introduction to particle surface interactions of bounce and resuspension. The CFD used in this work is introduced in the next section. The chapter concludes with an outline of the objectives of the presented work and the layout of the rest of the thesis.
1.1 Particle deposition

In order to model particle deposition to surfaces, the mechanisms of deposition must first be examined. Model inputs for examining deposition include particle density, diameter, fluid density, and viscosity and domain dimensions. Particle deposition is described in terms of deposition flux, $J$, at a point, or in terms of a deposition velocity $V_d$. Deposition velocity is defined as the net flux of particles to a surface divided by the concentration in the fluid and was first described by Chamberlain (1967):[7]

$$V_d = \frac{J}{C_\infty} \quad (1.1)$$

where $C_\infty$ is the particle concentration at a large distance from the surface.

1.1.1 Brownian Diffusion

Brownian Motion is the seemingly random movement of particles suspended in a fluid as a result of molecular collisions between particles and the fluid. The movement of the particles due to Brownian diffusion is calculated by Fick’s Law of diffusion and can be expressed as:

$$J_B = -D \frac{\partial C}{\partial x} \quad (1.2)$$

where $J_B$ is the Brownian diffusive particle flux in the x-direction and $\partial C/\partial x$ is the x component of the particle concentration gradient. $D$ is termed the Brownian diffusivity and can be calculated by using the Stokes-Einstein expression:

$$D = \frac{kTC_c}{3\pi\mu d_p} \quad (1.3)$$

where $k$ is Boltzmann’s constant, $\mu$ is the viscosity of the fluid, $d_p$ is the particle diameter, $C_C$ is the Cunningham correction factor and $T$ is the absolute temperature. The Cunningham correction factor is expressed as:

$$C_C = 1 + Kn \left[ A + Be^{-C/kn} \right] \quad (1.4)$$
A, B and C are empirical constants and Kn is the Knudsen number. The Knudsen number is a dimensionless number defined as the ratio of the molecular mean free path length to a representative physical length scale. Brownian diffusion can be the dominant transport mechanism of very small particles of a few micrometres over very small distances (due to the fact that the diffusion coefficient is inversely proportional to the radius of a particle, or the cube root of the volume).

1.1.2 Drag Force

Drag refers to forces that oppose the relative motion of an object through a fluid. Drag forces act in a direction opposite to the oncoming flow velocity and depend on the relative velocity of the particle as follows:

\[
F_D = \frac{1}{2} \rho C_D |u - v|(u - v)
\]  

(1.5)

where \(\rho\) is the fluid density, \(C_D\) is the drag coefficient dependent on the Reynolds number of the particle, \(u\) is the fluid velocity and \(v\) is the particle viscosity.

1.1.3 Gravitational force

Particles suspended in a fluid are subject to the effects of gravity. Fig. 1 represents the drag and gravitational forces acting on a particle. A balance of the drag force with the gravitational force on a particle leads to an expression for particle gravitational settling velocity of:

\[
V_g = \tau_p g
\]  

(1.6)

where \(\tau_p\) is the particle relaxation time and \(g\) is the acceleration due to gravity.

\[
\tau_p = \frac{\rho_p d_p^2}{18 \mu}
\]  

(1.7)

As noted by Li and Ahmadi (1993)[8], the gravitational force becomes significant for particles with diameters greater than 2 µm.
1.1.4 Thermophoresis.

Thermophoresis is a force resulting from a temperature gradient established in the gas medium. A particle in that temperature gradient experiences a force in the direction of decreasing temperature. The motion of the particle is a result of the fluid molecules on the warmer side striking the particle with a greater average momentum than those on the cooler side. It is a more complicated for larger particles as a temperature gradient is created in the particle itself and this affects the temperature gradient of the fluid. The net result is that the particle still receives more momentum on the hotter side than the colder side but the thermal force is influenced by the thermal conductivity of the particles. As noted by Nazaroff and Sippola (2002)\textsuperscript{9}, the thermophoretic velocity for particles
decreases with increasing particle size while for particles smaller than 1 µm the thermophoretic velocity is at a maximum and nearly independent of particle size.

1.1.5 Electrophoresis

A charged particle in a fluid is influenced by an electric field. Electrophoresis occurs because particles dispersed in a fluid almost always carry an electric surface charge. An electric field exerts an electrostatic Coulomb force on the particles through these charges. The Coulomb force on the particle is

\[ F_C = qE \]  

where \( q \) is the charge on the particle and \( E \) is the electric field strength. \( q \) is calculated by multiplying the number of charges \( n \) by the charge on an electron, \(-1.6 \times 10^{-19} \) C. Li and Ahmadi (1993) examined the effect of electrostatics on particle deposition rates. Their results showed that the use of a mean number of charges was not sufficient and that the actual charge distribution needed to be used.

1.2 Modelling Approaches

Particle transport in a turbulent multiphase flow is of great importance in a number of fields and has been widely studied. Traditional simulations and models of particle transport rely on the solution of the Reynolds averaged Navier-Stokes equations. The difficulty with these methods is that they rely on the Gradient Transport hypothesis, which does not accurately model particle transport, especially near a wall. The most common methods for modelling particle transport are Empirical Equations, Eulerian Modelling, Sub-layer Modelling and Direct Numerical Simulation (DNS).

1.2.1 Empirical Equations Method

Empirical equations methods are simply best fits to experimental data. There is good agreement with experimental data because adjustments are made to fit the results,
Introduction

however this adds little to the understanding of particle transport and the fit does not hold if the experiment parameters are outside the original fit parameters.

1.2.2 Eulerian Modelling

Eulerian Modelling treats the particles and the fluid as separate continuous phases. The behaviour of a cluster of particles is then predicted. Eulerian models predict the trends seen in experimental data of the entire range of particles but they are limited to applications in a fully developed turbulent flow and fail to show changes in particle deposition velocities due to turbulent eddies.

1.2.3 Sublayer Modelling

Sublayer models use a Lagrangian format to calculate a single limiting or critical trajectory in the near-wall region for a particle that just impacts the surface. Predicted deposition velocities are based on the percentage of the trajectories that would bring particles into closer contact with the wall than the limiting trajectory. Since deposition velocities are calculated based on limiting a particle trajectory and not on statistics of a large ensemble of particles, the computational requirements are substantially lower than fully Lagrangian models. Sub-layer models have shown the same shape of the deposition curve as observed in experiments with physically satisfying assumptions but, for particles where Brownian diffusion is significant, this method is invalid and a different approach that includes Brownian diffusion is necessary.

1.2.4 Lagrangian Simulation

The DNS method accurately models particle behaviour in turbulent flows. It is a simulation in computational fluid dynamics in which the Navier-Stokes equations are numerically solved without any turbulence model. Work such as that of McLaughlin (1989 & 1992)\cite{11,12} and Brooke et al. (1992 & 1994)\cite{13,14} have increased the understanding of particle behaviour but have shown that to accurately represent particle
behaviour, flow most be accurately modelled. One of the most useful and highly robust DNS methods is Lagrangian Simulation.

Lagrangian Simulation involves two steps when modelling particle behaviour. First, the flow is mathematically described and evaluated. Next the motion of each particle obtained by integrating the particle equation of motion. Lagrangian Simulations assume there is one-way coupling i.e. particles are affected by the fluid but the fluid is not affected by the particles. Due to the fact that individual particle trajectories are tracked, a level of knowledge on particle behaviour is achievable that is not accessible by experimentation. As indicated by Nazaroff and Sippola (2002)\cite{9}, there is less scatter seen in Lagrangian Simulation data than in data taken experimentally. This is due to the fact that there is minimal error in determining particle size or deposition flux in Lagrangian Simulation, unlike in experimental work. Lagrangian Simulation is an extremely useful CFD tool which agrees well with experimental data and trends seen in other simulation work but its use has been limited by the high computational cost.

1.3 Particle - Surface interactions

While particle transport is an extensively studied field, overall there has been limited study into particle-surface interactions within particle transport studies. Most simulations, models and experiments assume perfect sinks on boundaries when dealing with particles striking surfaces and there has been very limited inclusion of resuspension or particle bounce.

1.3.1 Bounce

Any particle impacting on a surface can either bounce or deposit. There are many factors (particle size, particle density, deposition velocity etc.) which determine whether the particle deposits or bounces. At low deposition velocities, particles deposit but as this velocity increases it becomes more likely that a particle will bounce. Bounce occurs if
the kinetic energy of the depositing particle is greater than the adhesion energy $E_{ad}$ at the surface. Fig. 1.2 depicts the particle deposition and bounce phenomenon.

**Fig. 1.2. A depiction of particle deposition and bounce phenomenon.**

### 1.3.2 Resuspension

Particle resuspension from a surface into a fluid is an important wide-ranging phenomenon affecting many disciplines. The resuspension of particles involves complex interactions of forces such as fluid drag, lift and adhesion of particles to a surface. These forces are hinged on mean flow, surface roughness and physical proprieties of the particle. For resuspension to occur the velocity of the flow at the position of the particle, $U_{posp}$ must be greater than the critical velocity of adhesion, $V_{cr}$ as illustrated in Fig. 1.3
Introduction

1.4 OpenFOAM

To model particle transport by a Lagrangian simulation a Computational Fluid Dynamic program (CFD) was needed. OpenFOAM\textsuperscript{[1]} was chosen for this task. As described in the literature that accompanies OpenFOAM\textsuperscript{[1]}, the OpenFOAM CFD Toolbox can simulate a wide range of scenarios, from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics, electromagnetics and the pricing of financial options. OpenFOAM\textsuperscript{[1]} is produced by OpenCFD Ltd and is freely available and open source, licensed under the GNU General Public Licence. The core technology of OpenFOAM\textsuperscript{[1]} is a flexible set of efficient C++ modules. These are used to build a wealth of solvers, to simulate specific problems in engineering mechanics and utilities, to perform pre- and post-processing tasks ranging from simple data manipulations to visualisation and mesh processing. While OpenFOAM\textsuperscript{[1]} can be ported to a wide range of architectures simulating differing operating systems, OpenFOAM\textsuperscript{[1]} is usually operated in a Linux based operating system.
1.5 Outline of Objectives and Layout of the Thesis

The goal of the presented work is to develop a model that accurately and cost-effectively incorporates bounce and resuspension into a Lagrangian simulation of particle movement. To this end, a theory is developed that indicates the critical velocity needed for a particle to bounce and the critical velocity needed for a particle to resuspend from a surface after it deposits there. This theory is then evaluated and reviewed by simulating test cases and comparing them with expected results from previous experimental and simulated work. The presented work begins with an overview of previous experimental and simulated research in Chapter 2. Chapter 3 contains the theory needed to calculate the critical particle velocity for bounce, $V_{cr}$, and the critical fluid velocity for resuspension, $V_{crf}$. Chapter 4 reviews the computational implementation of Chapter 3. Chapter 5 contains the results of the test cases that were simulated and Chapter 6 details the main conclusions of the presented work and contains a short introduction to future work which could be undertaken to further develop the presented model. Chapter 7 is the bibliography.
This chapter will give an overview of the most relevant work that has been done in this area of research. First, earlier work on Lagrangian simulation will be reviewed and then an overview of experiments and models investigating particle-surface interactions.

### 2.1 Lagrangian Simulation of Particle Transport

An extensive review of the developments in Lagrangian simulation of particle transport has been carried out by Nazaroff and Sippola (2002)\(^9\). Over the next few pages, the relevant aspects of their review will be reproduced and expanded on, with additional paragraphs added by the present author to reflect further developments since 2002.

As discussed by Nazaroff and Sippola (2002)\(^9\), Kallio and Reeks (1989)\(^15\) were the first to carry out a fully Lagrangian simulation of particle transport and deposition from a turbulent flow. Each individual particle trajectory was tracked and the effects of pressure gradient force, virtual mass, Stokes drag, gravitational settling and Brownian diffusion were ignored. The turbulent flow was modelled as a two-dimensional random velocity field as seen in Fig. 2.1.
Particle trajectories were determined from repeated interactions with discrete turbulent eddies, which had a random wall-normal velocity component drawn from a Gaussian distribution and a random time scale drawn from an exponential distribution. This work was in good agreement with experimental data for deposition velocities, the deposition curves were similar and particle relaxation times were of the same magnitude as observed in experimentation.

McLaughlin (1989)[11] modelled the trajectories of rigid spherical particles in a turbulent flow. He assumed that the channel flow was vertical to ratify the assumption that gravity did not affect particle deposition. The particle equation of motion did not account for wall effects. Particles were released at random points within the computational domain and the trajectories were obtained by integrating equations 2.1 and 2.2 forward in time using a flow field that was generated using DNS.

\[
\frac{dv}{dt} = -\frac{1}{\tau_p} (v - u) - \left(\frac{0.0762r}{\tau_p}\right) \left|\frac{du_x}{dy}\right|^{1/2} (v_x - u_x)^f \quad (2.1)
\]
Overview

\[ \frac{d \vec{r}}{dt} = v \]  (2.2)

where \( \vec{r} \) is the position vector of the particle, \( \vec{S} \) denotes a unit vector in the y direction, and \( u_x \) and \( v_x \) denotes the velocity of the fluid and the particle in the x direction, respectively. The observed deposition velocities were similar to those observed in experimental data. McLaughlin (1989) \[11\] concluded that particles were brought to the wall by strong fluid motions and that the lift force on the particle trajectory was an important effect. This work gave information on particle velocities and near-wall concentrations that had not been obtained by earlier experimental work or by Eulerian Modelling.

Ounis et al. (1991) \[17\] introduced a method to account for Brownian particle transport in Lagrangian simulations. They simulated the particle transport of 0.01 µm to 0.1 µm diameter particles, using a velocity flow field that was generated with DNS of the Navier-Stokes equations; the Brownian force was modelled by using a Gaussian white noise process. Particle-wall interactions were neglected in the simulation as well as the effects of surface forces. They concluded that Brownian diffusion is an important mechanism for the deposition of submicrometre particles and affects the dispersion of submicrometre particles in the near-wall region. The simulation results were in close agreement with earlier empirical equation models.

Brooke et al. (1992 & 1994) \[13,14\] calculated the trajectories of particles in a DNS of turbulent flow in a vertical channel. They used the same methodology as McLaughlin (1989) \[11\] to solve the Navier-Stokes equation. The particle equation of motion accounted for the effects of the Stokes drag force and wall effects. The simulation showed an order of magnitude agreement with the experimental deposition rates of Aggarwal and Liu (1974) \[16\], as well as an accumulation of particles in the near-wall region and the importance of near-wall eddies in particle deposition. The paper of Brooke et al. (1994) \[14\] was an extension of their earlier work. The particles were assumed to be small, rigid spheres with a density much greater than the fluid itself. The work again reported a build-up of particles in the near-wall region.
Li and Ahmadi (1993)[4] simulated the deposition of aerosol particles in a 2 cm wide vertical channel and a 2 cm wide horizontal channel. The particle size range was 0.01 µm to 10 µm with a mean air velocity of 5 m s\(^{-1}\). The initial locations of the particles were chosen at random within 30 wall units of the boundary. Gravitational settling was neglected in the vertical channel and a lower wall region was used so that gravitational sedimentation would not increase the particle deposition rate. Bounce was examined as part of this work and the results showed good agreement with experimental data and the work of McLaughlin (1989)[11]. To account for bounce Li and Ahmadi (1993)[4] proposed a critical approach velocity, \(V_c\) as

\[
V_c = \sqrt{\frac{2E}{m} \left(1 - \frac{c_r^2}{c_e^2}\right)}
\]

(2.3)

Where \(c_r\) is a coefficient of restitution and \(E\) is the surface potential energy given by

\[
E = \frac{Ad}{12y_0}
\]

(2.4)

where \(A\) is the Hamaker constant, \(y_0\) is the equilibrium separation of the particle and the surface (the distance between the particle and the surface at which the force on each is zero) and \(d\) is the diameter of the particle.
Fan and Ahmadi (1993)\cite{18} added the effect of rough surfaces to the published work of Li and Ahmadi (1993)\cite{8}. The particle equation of motion included Brownian and Saffman lift forces, in addition to the turbulent dispersion effect and gravity. The instantaneous fluctuating turbulent velocity field was modelled by a modified version of the Gaussian field proposed by Kraichnan (1970)\cite{19}. Brownian motion was simulated using a white noise process. Unlike the previous work, bounce was not studied in this model. Fan and Ahmadi (1993)\cite{18} reported that an increase in wall roughness increases the particle deposition velocity and that the simulation was in good agreement with empirical equation results and previous experimental data.

Chen and McLaughlin (1995)\cite{20} modelled the motion of spherical polystyrene particles in a vertical channel by the use of a modified Maxey and Riley equation. It included elements to account for Stokes drag force, wall-induced and shear-induced lift forces and Brownian diffusion from a Gaussian white noise process. The particles modelled had a diameter of 1 µm to 24.3 µm. The simulation indicated that the deposition rate depends more strongly on particle radius than expected from previous experiments and simulated data. They also suggested that polydispersity has a big effect on particle deposition.

Uijttewaal and Oliemans (1996)\cite{21} modelled particles in turbulent flow in a vertical cylindrical tube by means of DNS. The results of the study showed that small particles easily follow the large scale turbulent motions and are most sensitive to turbulence in the near-wall region of the tube flow. While the motion of large particles was dominated by the overall turbulence, decreasing particle deposition with increasing particle relaxation time was observed. As seen in numerous experiments and simulations, a build-up of particles in the near-wall region was also observed.

Chen and Ahmadi (1997)\cite{22} modelled the effects of flow direction, nonlinear drag and a corrected lift force on particle deposition rates in a turbulent flow with a maximum mean velocity of 3.5 m s\(^{-1}\) in a pipe. The particle diameter range studied was 0.01 µm to 100 µm. The effects of turbulent dispersion, Brownian diffusion and lift force were included. A Gaussian random field was used to model the instantaneous turbulent fluctuation and a
Gaussian white noise process was used to model the Brownian force. The effects of gravity were studied and this data was compared to a case where gravity was ignored. The simulation results were in good agreement with experimental data and showed the V-shaped curve expected in the dimensionless deposition velocity vs. non-dimensional particle relaxation time.

Wang et al. (1997)\cite{23} modelled particle trajectories in a turbulent flow incorporating an optimum lift force. The optimum lift force was formulated by Wang et al. (1997)\cite{23} to accurately represent the lift force acting on a particle in a wall-bounded shear flow. Since the calculations were in a vertical channel, gravity was not included as it does not affect deposition. The work found that overall the effect was negligible, since only a slight reduction in the deposition rates was seen. The magnitude of the deposition velocity was smaller than expected from experiments but the particle relaxation time was in reasonable agreement with the expected values.

Zhang and Ahmadi (2000)\cite{24} studied aerosol particle transport and deposition in vertical and horizontal duct flows. The fluid velocity field was generated by DNS of the Navier-Stokes equations. The particle equation of motion included Stokes drag, Brownian diffusion, lift and gravitational forces. The study found that for large particles lift and drag forces played a dominant role in particle transport while the Brownian force significantly affected the dispersion of submicrometre particles. It was concluded that in the vertical duct, gravity enhanced the non-dimensional deposition rate and reduced it when gravity opposed the flow. In the horizontal duct the non-dimensional deposition velocity increased sharply due to the sedimentation effect and the non-dimensional deposition velocity was much higher to the horizontal surface than seen in the vertical channel. The simulation results agreed well with empirical equation models, previous simulation and experimental data.

Choa et al. (2005)\cite{25} modelled a two dimensional channel flow of a fluid with many particles by DNS. The model was simulated over varying volume fractions (0.05 to 0.40) and for varying particle numbers (20 to 5120), the particle size was chosen to be large
enough to allow Brownian motion to be neglected. The model agreed with previous experimental work and it was found that as the volume fraction increases the fluid velocity decreases and the effective viscosity of the suspension increases.

Kang and Guo (2006)\cite{6} modelled aeolian sand transport with an Eulerian-Lagrangian Model. Volume-averaged Navier-Stokes equations were used to model the gas phase and the particle equation of motion was obtained by solving Newton's second law of motion. The model consisted of 2 cases; 1) a sand bed with 3000 0.5 mm particles and 2) a sand bed of 5000 0.4 mm particles with particle-particle collisions at 5 m s$^{-1}$. Particle-particle collisions were handled by use of a soft shell model from Xu and Yu (1997)\cite{26}. The soft sphere model for quasi-static deformation of a particle bed has the Voigt model like interaction consisting of a springs and dash pot at a contact point to take into account both the multiple particle contacts on one particle and the delayed force propagation. The Eulerian-Lagrangian model gave detailed information on the motion of the particle; it showed that sand flux decays exponentially with height for heights greater than 0.02 m and that particle-particle interactions have a key role in the sand transport.

Liu and Ahmadi (2006)\cite{27} modelled particle transport, dispersion and deposition near a building using a Lagrangian particle tracking computational procedure. The model accounted for drag and lift forces, Brownian force and gravitational sedimentation. The simulated particle range used was 0.1 µm to 10 µm. The Brownian force was simulated using a white noise process and the turbulent flow was evaluated using the Reynolds Stress Transport model. As noted by Liu and Ahmadi (2006)\cite{27}, this work was developed from the experimental data of Mirzai et al. (1994)\cite{28}. Solid surfaces were assumed to be a 'perfect sink', so that when a particle touched a surface, it would stick with no chance of rebound. The model agreed well with the experimental data of Mirzai et al. (1994)\cite{28} and they came to the conclusion that gravitational sedimentation affects the deposition of particles larger than 10 µm and Brownian motion was important for particles smaller than 0.1 µm and becomes negligible for particles larger than a few micrometres.
Hryba et al. (2009) modelled particle transport using both a Lagrangian and a Eulerian formulation. In the Lagrangian formulation the dynamic equation for the particle was solved and a discrete random walk model was used to account for turbulent effects. The Eulerian formulation calculated the particle concentration with the convection-diffusion equation using the terminal velocity of the particle and turbulent diffusion. Both account for the effects of buoyancy, inertial added mass and drag forces. Though the results for both showed good agreement with each other and previous experimental data, the Lagrangian approach gave a better insight into particle transport and this allowed for an easy inclusion of particle interactions.

2.2 Particle Surface Interactions

While deposition to surfaces has been extensively studied over the last few decades, little of this research has examined the effect of particle bounce or resuspension. The prevalence of deposition gauges/capture filaments led to perfect sinks, therefore particles cannot bounce or resuspend after being deposited. Research has shown that particles will bounce and resuspend off most surfaces.

Steen (1986) investigated dry deposition of dolomite particles (>50 µm) from a small plume onto various kinds of surfaces. A circular moving inlet was used perpendicular to the flow, to give an even distribution in the flow. The input was placed 10-15 m from the testing area. This aided in having all the large agglomerates deposited before they reached the testing area while not affecting the size distribution of the dolomite powder used. Samples were then taken from the grass, moss, twigs and glass plates close to the deposition gauges for comparison.

The research of Steen (1986) showed that the ratio of deposition velocity on the gauges to that obtained on the glass plates depended on the flow velocity and showed an increase in the ratio of deposition on the deposition gauges to that on the glass plates with an increase in flow velocity. The deposition velocity onto the gauges, plain glass and twig samples increased with particle size at a lower rate than terminal settling velocity of the
particles in still air and was sometimes lower than the settling velocity as shown by Steen’s Figure, reproduced in Fig. 2.2.

![Deposition velocity on plain glass plates compared to the terminal settling velocity for particles in still air](image)

**Fig. 2.3.** Deposition velocity on plain glass plates compared to the terminal settling velocity for particles in still air, (a) wind velocity 2 m s\(^{-1}\), (b) wind velocity 1 m s\(^{-1}\), (c) wind velocity 0.5 m s\(^{-1}\), (d) terminal settling velocity of particles in still air (from Steen(1986)[30]).

Steen (1986)[30] indicated, in comparison with deposition data obtained by Sehmel (1980)[31], that the values of deposition velocity were somewhat lower than expected. This was credited to the use of non-adhesive particles and heavier turbulence, which both increase bounce and resuspension.

As discussed by Ziskind et al. (1995)[32], a different approach to model the way small particles were resuspended, was developed by Reeks et al. (1988)[33]. The Reeks, Reed and Hall (RRH) model proposed that turbulent energy is transferred to the particle from the flow and this energy builds up in the particle in a surface adhesive potential well. The particle oscillates vertically in the potential well and when the built-up energy exceeds the potential well, the particle resuspends. This transfer of energy from the flow to the particle is most efficient, when the force frequency of the lift force is close to the natural frequency of vibration of the particle. This allowance for resonant energy transfer allows
for particles to resuspend before they would in an equivalent balance of force model. As reported by Ziskind *et al.* (1995)\[32\] and seen in the experimental work of Reeks and Hall (2001)\[34\], the RRH model, while being similar to experimental data on resuspension rates, gives lower values than expected and underestimates the resuspension rate from rough surfaces.

Reeks and Hall (2001)\[34\] published an extension to their RRH model to overcome the above underestimation and to account for drag as well as lift. In the RRH model the particles oscillated in the surface adhesive potential well while in the ‘Rock-n-Roll’ model the particle oscillates about a pivot point, P as seen in Fig. 2.3. From the experimental work from this publication, it is clear that the simpler but more exact ‘Rock-n-Roll’ model gave better results while still being similar to the results for the RRH model.

Paw and Braaten (1992)\[3\] carried out a study to examine the processes of deposition, bounce and resuspension individually and then compared them to relevant theory. Bounce was shown to be the dominant process limiting net deposition of large natural particles (pollen and spores) on natural surfaces. Experiments were conducted to determine resuspension and ideal deposition and net deposition of pollen and spores using leaf surfaces. Ideal deposition ($V_i$) and net deposition ($V_d$) were determined experimentally in a wind tunnel with maximum dispersion. Ideal deposition ($V_i$) was
Overview
determined with petroleum-jelly coated glass cover slips mounted flush on the wind
tunnel. They were positioned between 3 rows of 10 leaf disks which were used to
determine net deposition. The fraction of particles that must be accounted for by the sum
of resuspension and rebound processes according to Paw and Braaten (1992)[3] was:
\[
\frac{V_i - V_d}{V_i} \quad (2.5)
\]
To calculate resuspension, samples of leaf surfaces were coated with lycopodium spores
(34.6 µm diameter) and ragweed pollen (20.2 µm diameter). These were placed in a
wind tunnel for 4 minutes and then the particles were re-counted manually with a
microscope. The cycle was repeated in under a 2 hr. period for a range of ascending
wind speeds from 0.5 m s\(^{-1}\) to 8.0 m s\(^{-1}\).

Paw and Braaten (1992)[3] indicated that resuspension of particles from natural surfaces
was difficult to obtain and that the linear trend of ideal deposition with \(U\) (wind velocity)
matches theoretical models. At flow velocities of 3-4 m s\(^{-1}\) there is a marked decrease in
\(V_d\) with values of zero or close to zero being reached by 6 m s\(^{-1}\). Net deposition was
shown to be reduced significantly by rebound at high wind velocities. Resuspension
processes are more important at lower wind velocities, while at higher speeds, bounce is
more prevalent as there are fewer particles available for resuspension. This work clearly
showed that rebound and resuspension were major factors in limiting net deposition.
In 1995, Paw and Braaten (1995)[35] expanded on their previous work by simulating
incomplete particle bounce. The general hypothesis was that particles striking the surface
at velocities less than the critical rebound speed would still bounce from the surface, but
be trapped by the adhesion energy well. The drag force was found to be 2-4 orders of
magnitude smaller than adhesion forces and thus was neglected in the simulation on the
order of magnitude basis. Again the work showed that resuspension appears to bypass
rebound at moderate to high airflow rates.

Wu et al. (1992)[36] used micro-video camera techniques to observe particle bounce off
and resuspension in wind tunnel experiments. These observations were supplemented by
quantitative data from counting individual particles under the microscope and using bulk
analysis of fluorescing particles. The particles used were observed to have diameters in range 7-42 µm and were roughly spherical. Bounce off was determined by the comparison of ungreased and greased Teflon surfaces. By setting the run time to approximately one hour, Wu *et al.* (1992)[36] tried to negate the effects of resuspension. 

F, the fraction of rebound, was determined from the ratio of the deposition rate on the ungreased surface to that on the greased surface. Wu *et al.* (1992)[36] observed that typically F increased with wind speed and that bounce-off occurred for those particles that had sufficient kinetic energy to overcome the adhesion energy between the particle and the surface in equation 2.4; for 5.4, 8.6 and 10 µm particles n was determined to be 1.1, 4 and 0.6.

\[ F \propto U^n \]  

Wu *et al.* (1992)[36] calculated the resuspension rate, \( \beta \) as

\[ \beta = -\frac{1}{t} \ln \left( \frac{N}{N_0} \right) \]  

where N is the number of particles on the surface at time t and \( N_0 \) is the initial number of particles on the surface. Wu *et al.* (1992)[35] reported the finding of Farchild and Tillery (1982)[37] that resuspension rate from smooth surfaces increases according to \( U^{1.3} \) and that Sehmel (1980)[31] reported the exponent of wind speed ranges from 1.1 to 6.4. Wu *et al.* (1992)[36] concluded that the fraction of rebound is strongly dependent on the free stream velocity, particle size and relative humidity and at higher wind speeds that there are less particles for resuspension as bounce-off becomes the dominant process.

Lai (1997)[38] carried out an experimental study of the deposition of aerosol on rough surfaces and discussed the implications for indoor air quality control. To examine the effect of resuspension on the aerosol deposition velocity from ribbed surfaces, a preliminary experiment was carried out at an average velocity inside the duct of 4.4 m s\(^{-1}\). Fig. 2.2 shows the measured spatial deposition velocity for the case run by Lai where no paraffin oil was used to trap particles as in his other cases. From Fig. 2.2, it can be clearly seen that particle resuspension reduced particle deposition. Lai (1997)[38] observed that, except at the top of the surface rib, the time-averaged reduction was approximately 50%. However at the top of the surface, the percentage reduction was
approximately 80%. In this work the maximum reduction in the particle deposition due to resuspension was greatest at the top surface of the rib due to the high air speed over this surface. A key conclusion Lai drew from his work was that particle resuspension substantially reduces aerosol deposition enhancement.

![Graph](image)

*Fig. 2.5. Measured spatial aerosol particle velocity along one pitch length of a rib for 4.5 µm particle. (a) with paraffin oil applied to the ribs, and (b) without oil applied (from Lai (1997)) [38].*

Aguiar *et al.* (2007) [39] examined the influence of particle size on the adhesion force between a particle and a surface by means of a centrifuge technique, which uses imaging analysis to determine the number of adhered particles on the disk surface before and after each centrifugation. The first method of the centrifuge technique involved a pressure force being applied onto the particles to cause particles to resuspend. The second method involved a force being applied to detach the adhered particles (spin-off). As reported by Aguiar *et al.* (2007) [39], it is known that when dealing with dry inert particles, without chemical bonds and/or an external electric field, the force of adhesion is usually due to Van der Waals forces. From their experimental work it can be seen that theoretical models tend to predict lower adhesion force than seen in experimental work and this could be due to surface roughness of particles in experimental work reducing the Van der Waals force.
Kissane et al. (2010)[40] work studied the way small particles attached to a surface and resuspended when exposed to a turbulent flow. The particles studied were less than 5 µm in size. The size range used was chosen so that the principle force holding the particle onto a surface arises from the Van der Waals forces. Kissane et al. (2010)[40] improved on the ‘Rock-n-Roll’ model by using calculated statistics for the fluctuations of both the stream-wise fluid velocity and accelerations close to the wall from a LES of the turbulent channel flow. The modified model gives more resuspension than the original ‘Rock-n-Roll’ model.

Zhang et al. (2011)[41] carried out a study of low density, silica-coated particle resuspension due to human walking in indoor environments. The model accounts for the particle adhesion to the floor using an adhesion model described by Zhang et al. (2008)[42], and the airflow generated by the stepping down and up of the foot during the gait cycle. The rate of particle resuspension and dispersion due to indoor human walking in an experimental chamber was simulated, a critical radius for resuspension was evaluated and the results are compared with that from Arizona Road Dust. Fig. 2.6 shows the variations of critical velocities for rolling detachment of silica-coated particles with diameter during a step down process. Zhang et al. (2011)[41] reported that the critical velocities decreases with the particle diameter, which implies that as particle size increases the probability of resuspension increases and that particles with larger surface roughness are easier to resuspend.

![Critical velocities graph](image)

**Fig. 2.6.** Variations of critical velocities for rolling detachment of silica-coated particles with particle diameter during the stepping down process at varying particle roughness, Δc.
2.3 Conclusions

While there has been extensive research into particle transport by experimentation and Lagrangian simulation, and by examining bounce and resuspension, there has been limited research into accurately modelling bounce and resuspension in particle transport by Lagrangian simulation. From previous research into particle transport by Lagrangian simulation it is clear that there is a need to first accurately model the velocity field of the turbulent flow. When simulating the particle equation of motion, it is applicable to include the effects of drag and gravity when modelling particle deposition in a domain including vertical and horizontal surfaces.

From previous research it is clear when studying dry, inert, hard sphere particles that the main adhesive force capturing the particle in bounce, and keeping the particle attached to the surface in a turbulent flow after it deposits, is the Van der Waals force. When the phenomenon of bounce is examined, it is clear that only incorporating the adhesive force between the surface and the particle when calculating a critical velocity is not enough. To accurately model this process mechanical potential energy, gravitational force due to the mass of the particle, and surface adhesion energy must be included. From research on bounce and resuspension, trends are seen that indicate that, as the velocity of the turbulent field increases, more bounce, resuspension and deposition should be seen to occur (as there are more particles available to deposit), to a point where bounce begins to dominate the process and deposition and resuspension tend to zero (as there are no particles deposited).

In summary, previous research indicates that the probability of bounce increases as particle size increases and the flow velocity increases. While there has been extensive research into particle transport using Lagrangian simulation and the processes of bounce and resuspension, it is imperative that a method to accurately model bounce and resuspension in a Lagrangian simulation of particle transport at a reasonable computational cost is created, as presented in this work.
3. Theory

3.1 Introduction

In this work, an algorithm to incorporate bounce and resuspension in a Lagrangian simulation of a turbulent flow is presented. To model bounce and resuspension, a method needs to be developed to calculate the velocity at which a particle impacting a surface is will bounce, and the fluid flow velocity where a particle resting on a surface will resuspend. As has been seen from previous work, the main attraction force between a smooth particle and a surface is the Van der Waals force. While most methods involve the calculation of the adhesion force using only the Van der Waals force, a highly modified version of the Xu and Willeke (1993) model is used, as it incorporates the Van der Waals force, the effects of elastic deformation, and gravitational effects and also accounts for mechanical potential energy from surface contact.

3.2 Bounce

The Van der Waals force is known to cause the adhesion of particles to a surface or to other particles. The Van der Waals force is caused by molecular interactions between surfaces of the solids. Van der Waals force becomes very important when very smooth surfaces are brought into contact with each other and over very short distances. The surface adhesive energy per unit area of contact dG due to the Van der Waals force is given by
Theory

\[ dG = \frac{A}{12 \pi h^2} \]  

(3.1)

where \( A \) is the Hamaker constant and \( h \) is the separation distance. The Hamaker constant is given by

\[ A_{132} = (\sqrt{A_1} - \sqrt{A_3})(\sqrt{A_2} - \sqrt{A_3}) \]  

(3.2)

where \( A_1 \) is the Hamaker constant of the particle, \( A_2 \) is the Hamaker constant of the surface and \( A_3 \) is the Hamaker constant of the fluid. From Crowe et al. (1998)\[43\] it is known that if the fluid is air, since \( A_3 \) is so small (~ \( 4 \times 10^{-20} \) J), equation 3.2 can be simplified to

\[ A_{12} = \sqrt{A_1 A_2} \]  

(3.3)

In this work, the particles were modelled as hard spheres. Therefore it is assumed that no plastic deformation occurs when the particle impacts the surface with an impact velocity of \( V \) but some of the energy of the collision is stored in elastic deformation of the sphere. After the collision the sphere returns to its original shape as shown in Fig. 3.1.

\[ Fig. 3.1. \textit{A diagram of the bounce process as presented in this work.} \]

Xu and Willeke (1993)\[44\] described an elastic limiting velocity \( V_y \), below which plastic deformation occurs.
where \( K \) is the mechanical constant as described in equations 3.6, 3.7, 3.8, \( y_1 \) is the elastic stress limit (the stress point at which a material, if subjected to higher stress, will no longer return to its original shape) and \( \rho_p \) is the particle density.

\[
K = \frac{4}{3(K_1 + K_2)} \quad (3.5)
\]

\[
K_1 = \frac{1}{M_e_1} \quad (3.6)
\]

\[
K_2 = \frac{1}{M_e_2} \quad (3.7)
\]

where \( M_e_1 \) is Young's modulus for the particle and where \( M_e_2 \) is Young's modulus for the surface. The energy, \( E_{col} \) stored in the elastic deformation is

\[
E_{col} = \frac{1}{2} m V_y^2 \quad (3.8)
\]

The contact radius \( R_e \), as seen in Fig. 3.2 is

\[
R_e = \left( \frac{5E_{col}r^2}{2K} \right) \quad (3.9)
\]

Fig. 3.2. Diagram of a sphere impacting on a flat surface, with the contact radius shown.

Rogers and Reed (1984) defined the contact geometry parameter as

\[
G_e = \frac{K r \left( R_y^2 + R_p^2 \right)^{\frac{3}{2}}}{\pi R_y^2 y_1 + K R_y^2} \quad (3.10)
\]
where \( R_p \) is the plastic deformation contact radius. The presented model assumes that the particles being transported in the flow are hard spheres. To keep this assumption valid, the particles cannot permanently deform, as the particles would no longer be spheres and the assumption of them being hard would not hold. Thus no plastic deformation occurs. This assumption also leads to a computational saving as otherwise a number of iterations of the equation for \( R_p \) would be needed to calculate its value for each particle. Since no plastic deformation occurs, \( R_p \) is equal to zero and the contact geometry can be simplified to

\[
G_c = r
\]  

(3.11)

For models that only use Van der Waals force, the total energy of adhesion would be equal to the Van der Waals adhesion energy but from Johnson et al. (1971)\[2\] it is known that the adhesion energy is

\[
E_a = E_m + E_s
\]  

(3.12)

where \( E_m \) is the mechanical potential energy due to surface contact and \( E_s \) is the surface adhesion energy. \( E_m \) is given as

\[
E_m = F_0 \left\{ \frac{2}{3} F_1^3 + \frac{2}{3} F_0 F_1^{-\frac{1}{3}} \right\} \left( \frac{\pi G_c}{3K^3} \right)^{\frac{1}{3}}
\]  

(3.13)

where \( F_1 \) is the sum of the surface forces and \( F_0 \) is the gravitational force due to the mass of the particle. \( F_1 \) is defined as

\[
F_1 = F_0 + 3(dG)\pi G_c + \sqrt{(F_0 + 3(dG)\pi G_c)^3 - F_0^2}
\]  

(3.14)

From Derjaguin (1975)\[46\] it is known that the surface adhesive energy per unit area can be considered to be dependent on the Van der Waals force given as in equation 3.2. The surface energy is calculated by

\[
E_s = \pi \left( \frac{G_c F_1}{K} \right)^\frac{2}{3} dG
\]  

(3.15)

Xu and Willeke (1993)\[44\] defined the critical velocity for bounce to occur as

\[
V_{cr} = \sqrt{\frac{2(E_a + E_p)}{m}}
\]  

(3.16)
where $E_p$ is the plastic deformation energy. Again, since no plastic deformation occurs equation 3.18 can be simplified to

$$V_{cr} = \sqrt{\frac{2E_a}{m}} = \sqrt{\frac{2(E_m + E_s)}{m}}$$

(3.17)

$V_{cr}$ is calculated using the values of $E_m$ and $E_s$ from equations 3.13 and 3.15. After a particle impacts on the surface, there are only two outcomes, dependent on $V_{cr}$ and the velocity of the particle as it impacts (Fig. 3.3):

1. If the velocity of the particle is less than $V_{cr}$, the particle deposits and the particle velocity is set to zero.
2. If the velocity of the particle is greater than $V_{cr}$, the particle bounces with an elastic collision.

**Fig. 3.3. A schematic of the outcomes after a particle collision with a flat surface.**

In the case of bounce, the velocity of the particle is set by the use of the perpendicular principle in vector mathematics. For instance, for a collision in the x-y plane with a boundary wall located in the x-z plane, if

$$V_i = (a\vec{x}, b\vec{y}, c\vec{z})$$

(3.18)

then

$$V_f = (a\vec{x}, -b\vec{y}, c\vec{z})$$

(3.19)

and the velocity of the particle is set to $V_f$. 

30
3.3 Resuspension

For resuspension to occur, the velocity of the flow at the position of the particle, $U_{posp}$, needs to be greater than the critical fluid velocity for resuspension, $V_{cr}$. Fig. 3.4 shows a schematic of resuspension.

![Fig. 3.4. A schematic of resuspension with 1) the particle resting on the surface and 2) the particle when $V_{cr}$ is less than $U_{posp}$.]

The procedure to calculate $V_{cr}$ follows the same steps as used to calculate $V_{cr}$ in chapter section 3.2, until the adhesion energy is calculated. From Johnson et al. (1971)[2] (equation 3.12), it is known that the adhesion energy is

$$E_a = E_m + E_s$$

but since there is no mechanical energy as the particle is at rest on the surface $E_m = E_s$ and $V_{cr}$ is defined as

$$V_{cr} = \sqrt{\frac{2E_s}{m}}$$  \hspace{1cm} (3.20)

When a particle resuspends, the velocity is calculated from the energy remaining from the flow after the energy of adhesion is overcome.
3.4 Conclusions

As indicated in the introduction to this section, to incorporate bounce and resuspension into a Lagrangian simulation it is necessary to know at what velocity a particle impacting a surface will bounce and what velocity the flow at the position of the particle will cause it to resuspend from a surface. By the proper implementation of $V_{cr}$ and $V_{crr}$, as detailed in the theory above in a Lagrangian simulation, it is possible to calculate the two necessary velocities and determine if a particle will bounce, deposit or resuspend; this accounts not only for Van der Waals forces but also the effects of mechanical potential energy and gravitational attraction. The computational implementation and development of the above theory will be discussed in the next chapter.
4. **Computational Implementation**

Contents

<table>
<thead>
<tr>
<th>4 Computational Implementation</th>
<th>33</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>33</td>
</tr>
<tr>
<td>4.2 Computational Implementation</td>
<td>46</td>
</tr>
<tr>
<td>4.3 Computational Development</td>
<td>53</td>
</tr>
<tr>
<td>4.4 Computational testing</td>
<td>56</td>
</tr>
<tr>
<td>4.5 Conclusions</td>
<td>65</td>
</tr>
</tbody>
</table>

4.1 Introduction

As introduced in the chapter section 1.4, OpenFOAM\textsuperscript{[1]} is supplied with numerous pre-configured solvers, utilities and libraries and so can be used like any typical simulation package. OpenFOAM\textsuperscript{[1]} allows the flexibility for customisation and add-on developments to answer the need of research users due to its open source nature. Due to OpenFOAM\textsuperscript{[1]} being open source, this Computational Fluid Dynamic (CFD) program is widely used in research, thus a wealth of user knowledge is already available in the form of research papers, model formulation, numerical schemes and linear equation solvers.

OpenFOAM\textsuperscript{[1]} uses the finite volume method to solve a system of partial differential equations ascribed on any 3D unstructured mesh of polyhedral cells. Fig. 4.1 shows the 3D mesh for the Paw and Ahmadi test case and the block structure as detailed in chapter section 5.2. The fluid flow solvers are developed within a robust, implicit, pressure-velocity, iterative solution framework, although alternative techniques are applied to other continuum mechanics solvers.
Domain decomposition parallelism is fundamental to the design of OpenFOAM\textsuperscript{[1]} and integrated at a low level so that solvers can generally be developed without the need for any ‘parallel-specific’ coding. OpenFOAM\textsuperscript{[1]} message passing is usually implemented by the use of Local Area Multicomputer (LAM) of openMPI but if the user desires MPICH can be used instead. To decompose the domain, the user uses the 'decomposePAR' utility built-in to OpenFOAM\textsuperscript{[1]}. The methods and parameters to be used in the decomposition is defined by the user in the 'decomposeParDict' in the system folder of the test case as seen, code excerpt 4.1.

\textbf{Fig. 4.1. The 3D mesh for Paw test case and the block structure.}
Code excerpt 4.1. Example of 'decomposeParDict' file.

When 'decomposePAR' utility is run, the domain is divided into the number of subdomains as specified by the user in the 'decomposeParDict' file. Usually the number of processors available decides this parameter. The goal of 'decomposePAR' utility is to break up the domain with minimal effort but also to guarantee a fairly economic solution. The fields and geometry of the subdomains are defined by the user in the 'decomposeParDict' file. One of four methods as detailed in Table 4.1 is used to achieve this goal.
### Simple Method:
Simple geometric decomposition in which the domain is split into pieces by direction, e.g. two pieces in the x direction, one in y direction etc.

### Hierarchical Method:
Hierarchical geometric decomposition which is the same as simple method except the user specifies the order in which the directional split is done, e.g. first in the y-direction, then the x-direction etc.

### METIS Method:
METIS decomposition which requires no geometric input from the user and attempts to minimize the number of processor boundaries. The user can specify a weighting for the decomposition between processors which can be useful on machines with differing performance between processors.

### Manual Method:
Manual decomposition, where the user directly specifies the allocation of each cell to a particular processor.

<table>
<thead>
<tr>
<th>Table 4.1. Summary of Decomposition Methods.</th>
</tr>
</thead>
</table>

On completion, a set of subdirectories will be created, one for each processor, in the case directory. These represent the subdomains and contain the decomposed field descriptions and a directory containing the decomposed mesh description. To run a decomposed OpenFOAM [1] case in parallel over a network using LAM, a file must be created that contains the host names of the machines and if distributed memory is to be used, this must be defined in the 'decomposeParDict' file and contain a roots entry, that lists the root paths to each of the memories to be used. To run a decomposed case in parallel on a single machine, in which the processor nodes are all local to the user, MPI/MPICH is used. When the command is used, the user must specify the number of processors, the executable and output name for a log file i.e.

```
mpirun -np 3 `which icoFoam` $FOAM run/tutorials/icoFoam cavity -parallel < /dev/null >& log &.
```

To post process a parallel test case, the user has the choice to either post process the subdomains individually or to reconstruct the domain from the subdomains using the built-in utility of ‘reconstructPar’ and then post process the domain.
A variety of fluid-flow problems deal with more than one phase and OpenFOAM\textsuperscript{[1]} 1.3 has a Lagrangian particle tracking library for this kind of problem. The basic Lagrangian library provides an efficient parallelised implementation of tracking and coupling with the Eulerian phase for soft particles.

Fig. 4.2. Steps involved in adding hard ball particles to the existing Lagrangian models in OpenFOAM\textsuperscript{[1]}
Further user derived libraries provide additional functionality for hard particles. One of these user developed libraries is ‘icoLangrangianFoam’ which formed the basis for this thesis. ‘icoLangrangianFoam’ was designed as a demonstration of the ability of OpenFOAM$^{[1]}$ to handle hard particles, which was lacking previously. It was developed from icoFoam, which solves the incompressible laminar Navier-Stokes equations using the Pressure Implicit with Splitting of Operators (PISO) algorithm. The incompressible continuity and momentum equations are given by

\begin{align}
\nabla \cdot \mathbf{u} &= 0 \quad (4.1) \\
\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{u}) - \nabla \cdot (\mu \nabla \mathbf{u}) &= -\nabla \cdot \mathbf{p} \quad (4.2)
\end{align}

where \( \mathbf{u} \) is the fluid velocity vector, \( \mu \) is the viscosity of the fluid and \( \mathbf{p} \) is the pressure. The non-linearity in the convection term \( \nabla \cdot (\mathbf{u} \mathbf{u}) \) is handled using an iterative solution technique, where

\begin{equation}
\nabla \cdot (\mathbf{u} \mathbf{u}) \approx \nabla \cdot (\mathbf{u}^o \mathbf{u}^n) \quad (4.3)
\end{equation}

where \( \mathbf{u}^o \) is the currently available solution and \( \mathbf{u}^n \) is the new solution. The algorithm cycles until \( \mathbf{u}^o = \mathbf{u}^n \). There is no pressure equation, but the continuity equation imposes a scalar constraint on the momentum equation (since \( \nabla \cdot \mathbf{u} \) is a scalar) and therefore the continuity and momentum equations are used to derive a pressure equation. To achieve this, the momentum equation is discretized, keeping the pressure gradient in its original form

\begin{equation}
\alpha_p \mathbf{u}_p + \sum_N a_N \mathbf{u}_N = \mathbf{r} - \nabla \mathbf{p} \quad (4.4)
\end{equation}

Where the subscript \( p \) used in \( a_p \) and \( \mathbf{u}_p \) indicates the value of the variable at the point \( p \) in the centre of the cell. Then the \( \mathbf{H}(\mathbf{u}) \) operator is introduced
Computational Implementation

\[ \mathbf{H}(\mathbf{u}) = \mathbf{r} - \sum_{n} a_{n} \mathbf{u}_{n} \]  \hspace{1cm} (4.5)

so that
\[ a_{p} \mathbf{u}_{p} = \mathbf{H}(\mathbf{u}) - \nabla p \] \hspace{1cm} (4.6)
\[ \mathbf{u}_{p} = (a_{p})^{-1} (\mathbf{H}(\mathbf{u}) - \nabla p) \] \hspace{1cm} (4.7)

The first term of \( \mathbf{H}(\mathbf{u}) \) represents the unsteady term and all the sources except the pressure gradient, while the second part contains the matrix coefficients of the neighbouring cells multiplied by their velocity. Next the continuity equation is discretized

\[ \nabla \cdot \mathbf{u} = \sum_{f} s_{f} \cdot \mathbf{u} = \sum_{f} \mathbf{F} \] \hspace{1cm} (4.8)

where \( \mathbf{F} \) is the face flux, \( \mathbf{F} = s_{f} \cdot \mathbf{u} \). The velocity on the face is obtained by interpolating the semi discretized form of the momentum equation as follows

\[ \mathbf{F} = -(a_{p})^{-1} s_{f} \cdot \nabla p + (a_{p})^{-1} s_{f} \cdot \mathbf{H}(\mathbf{u}) \] \hspace{1cm} (4.9)

By substituting this equation into the discretised continuity equation obtained above, we obtain the pressure equation

\[ \nabla \cdot [(a_{p})^{-1} \nabla p] = \nabla \cdot [(a_{p})^{-1} \mathbf{H}(\mathbf{u})] \] \hspace{1cm} (4.10)

The PISO algorithm is an efficient method to solve the Navier-Stokes equations in unsteady problems. To solve the equations detailed above, the following steps are implemented

1. Set the boundary conditions.
2. Solve the discretised momentum equation to compute an intermediate velocity field.
3. Compute the mass fluxes at the cell faces.
4. Solve the pressure equation.
5. Correct the mass fluxes at the cell faces.
6. Correct the velocities on the basis of the new pressure field.
7. Update the boundary conditions.
8. Repeat from step three for the prescribed number of times.
9. Move on to the next time step and repeat from step one.

The additional functionality needed to handle hard particles, was added by the addition of files to account for the parameters of the particles, their injection into the flow and their behaviour such as their movement and boundary collisions.

'icoLagrangianFoam' adds particles to 'icoFoam' by creating the particles following the procedure in 'HardBallParticle' files using the user defined parameters read in by 'IncompressibleCloud' files (step 1 in Fig. 4.2). The initial set up of the hard ball particles is seen below in code excerpt 4.2.

```cpp
// *************************************************************************//
// Construct from components
HardBallParticle::HardBallParticle(const Cloud<HardBallParticle>& cloud,
const vector& position,
const label celli,
const scalar& d,
const vector &U
)
particle<HardBallParticle>(cloud,position,celli),
d_(d),
mass_(0.),
U_(U),
tEnd_(0.0)
{
  calculateDerived();
}
void HardBallParticle::calculateDerived() {
  mass_=density*4./3.*3.1415*d_*d_*d_/8.;
}
// *************************************************************************//
```


The particles are injected into the flow using the inject( ) member of 'IncompressibleCloud' as defined by the user and read in by 'IncompressibleCloud' (step 2 in Fig. 4.2). inject( ) uses random scalars and vectors to create randomly generated
values for the particles diameter, injection velocity and injection position as seen in code excerpt below.

```c++
// ********************************************************************
void IncompressibleCloud::inject() {
    if(runTime_.time().value()<tStart || runTime_.time().value()>tEnd) {
        return;
    }
    scalar prop=random.scalar01();
    if(prop<thres) {
        vector tmp=(random.vector01()-vector(0.5,0.5,0.5))*2;
        vector pos=center+tmp*r0;
        tmp=vector(random.GaussNormal(),random.GaussNormal(),random.GaussNormal())/sqrt(3.);
        vector vel=tmp*vel0+vel1;
        scalar d=fabs(random.GaussNormal())*d1+d0;
        label cellI=mesh_.findCell(pos);
        if(cellI>=0) {
            HardBallParticle* ptr=new HardBallParticle(*this,pos,cellI,d,vel);
            ptr->tEnd()=runTime_.deltaT().value();
            addParticle(ptr);
            injectedInModel_++;
        }
    }
// *********************************************************************
```

*Code excerpt 4.3. Injector code.*

The particle movement (step 3 in Fig. 4.2) is called by 'moveParticles.H' which uses the 'IncompressibleCloud' member track( ) and which in turn uses 'HardBallParticle' member updateProperties( ) to calculate the velocity of the particle, to calculate its new position and handle any boundary collisions. As previously mentioned, there is no need to model turbulence in the particle phase of a Lagrangian model of particle transport. The turbulence is modelled in the fluid phase. In the presented work, K-omega model is
used for the fluid phase using default values for the turbulence model coefficients. The K-omega model is a two equation model, which means, it includes two extra transport equations to represent the turbulent properties of the flow.

while 'icoLagrangianFoam' was a step in the right direction by adding a method to include hard ball particle transport, many flaws still existed. The treatment of bounce was not realistic, the particle either bounced or deposited. These outcomes were switchable but both outcomes could not occur in the same simulation. When particle deposition was selected, the surfaces acted as 'perfect sinks', the deposition would be counted and the particle would be deleted from the simulation. The reduction in the number of particles in 'icoLagrangianFoam' with deposition outcome selected, over the presented method used is evident in Fig. 4.3. If bounce was turned on, all the particles rebounded from the boundary. Code excerpt 4.5 shows the handling of boundary collisions in the 'icoLagrangianFoam' as detailed above (step 3 in Fig. 4.2).
\[ U() = (1.0 + \text{data.wallElasticity()}) \times \text{Un} \times \text{Sf}; \]

```cpp
ifdef DEBUG_MOVE
    Info << "(" << Sf << " -> " << Un << ") to " << U() << endl;
endif
keepParticle=true;
} else {
ifdef DEBUG_MOVE
    Info << "and dies " << endl;
endif
    keepParticle=false;
```

// ***************************************************************************

**Code excerpt 4.5. Original Boundary collisions code.**
There was no method included to incorporate the resuspension of a particle from the boundary, as the particles were deleted from the simulation after depositing as mentioned above. Due to the treatment detailed previously in this chapter, except for the particle treatment at an inlet or outlet and boundaries of subdomains if the case was simulated in parallel, all other boundary treatments were not handled correctly. While bounce or deposition was included in ‘icoLagrangianFoam’, the only method to look at bounce or deposition was in visualisation of the data in Paraview[47]. To overcome this and add a method to look at the deposition, bounce and resuspension of particles, code was added to output the relevant data to files, so they could be post processed later as seen in code excerpt 4.6.

```cpp
// ******************************************************************//
ofstream bounce_d;
bounce_d.open("sand_bounce_d", ios::out | ios::app);
bounce_d << d_  << "\n";
bounce_d.close();
// ******************************************************************//
```

**Code excerpt 4.6.** Code to output to a text file when bounce occurred.

The visualisation of the data produced by presented work could not be handled by OpenFOAM’s built-in post processing function. To visualise the data, a function called ‘foamToVTK’ was used to convert the data to vtk files which can be viewed Paraview[47]. The flow data and the Lagrangian data are read in separately to Paraview[47]. A sphere glyph is used to represent the particle and arrow glyphs are used to represent the fluid flow. The glyphs used to represent particles can be coloured and sized to reflect the data associated with the particle. The data from the text files was read into SigmaPlot[48] and graphs were produced to illustrate details of the diameter of the particle, the velocity of
the particle and the critical velocity of the particle when a particle bounced or deposited and the diameter of the particle, fluid velocity at the position of the particle and the critical velocity for resuspension when a particle resuspends. Fig. 4.4 is an example of the visualisation produced by Paraview[47] and the graphs produced by SigmaPlot[48].

Fig. 4.4. An example of the visualisation produced by Paraview[47] and a graph of the deposition velocity of the particle, $V_{elp}$ vs. the diameter of the particle, $d_p$ produced by SigmaPlot[48].
4.2 Computational Implementation

Fig. 4.5 and 4.6 are flow charts summarizing the computational implementation of the theory as described in chapter 3.

Fig. 4.5. A flowchart of the implementation of bounce, deposition and resuspension (part).
The first step to modelling bounce and resuspension was to stop particles being deleted from the system when they deposited. This was achieved by modifying the way that deposition was handled. The original code was
If the particle deposited ("dies") the 'keepParticle=false' statement deleted it from the simulation. This was changed to (found in point 7 of Fig. 4.5)

The particle is now kept in the simulation rather than being deleted. With this change, it was important to know whether a particle had deposited or was still in the flow each time 'moveParticles' was called. To this end a flag was created called Flagcheck to indicate whether a particle had deposited and was sitting on a boundary or was still in the flow. The flag was set up as a scalar number and treated in the same manner as other properties of the particle i.e. diameter and mass.
When ‘updateProperties( )’ is called, Flagcheck is set to indicate whether a particle has been deposited or not. Flagcheck is set to one for deposited particles and set to zero if the particles are still in the flow. Originally the particle velocity (as seen in code excerpt 4.9) would just be calculated for all particles in the simulation, as particles which had already deposited would have been deleted from the simulation, therefore there was no need for their velocity to be calculated.

```
// ********************************************************************//
U()=( U() + coeff*Upos + data.g()*dt)/(1. + coeff);
// ********************************************************************//
```

*Code excerpt 4.9. Original particle velocity calculation from 'icoLagrangianFoam'.*

By the use of Flagcheck, the velocity of the particle is set to zero if it has already deposited and calculates it if it had not (at point 2 in Fig. 4.4).

```
// *******************************************************************//
if (flagcheckb==1){
  U()=vector(0.,0.,0.);
}
else{
  U()=( U() + coeff*Upos + data.g()*dt)/(1. + coeff);
}
// *******************************************************************//
```

*Code excerpt 4.10. Modification made to account for particles which have deposited, when calculating the velocity of the particle.*

At this point in code, (point 3 of Fig. 4.4) the critical velocity for resuspension, $V_{crr}$ as detailed in chapter 3.3 is calculated, the flow velocity at the position of the particle, $U_{pos}$ (point 4 of Fig 4.4) and a check for resuspension is run (point 5 of Fig. 4.4). If Flagcheck was set at one and the fluid velocity at the position of the particle was greater than the critical velocity for resuspension, $V_{crr}$, the particle would resuspend and the particle velocity would be calculated, using the remaining energy left after the particle had overcome the surface adhesive energy. If Flagcheck was set at one but $V_{crr}$ was greater
than the fluid velocity at the position of the particle, the velocity of the particle remained at zero. If Flagcheck was set at zero at this point, the particle’s velocity was calculated as the particle was suspended in the flow. This allowed particles to resuspend if they were previously deposited and if the conditions were met for resuspension. If not the particles remained at their positions on the boundary.

```plaintext
//calculate Vcrr
scalar r1=d_/2;
//scalar A1=6.5;   //particle Hamaker constant
//scalar A2=43.5;  //wall Hamaker constant
//scalar A3=4.38;  //fluid Hamaker constant
scalar A=31.6e20;     // hamaker constant for gold-silicon((sqrt(A1)-sqrt(A3))*(sqrt(A2)-sqrt(A3)))
scalar h=0.0000000004;
scalar Me=150;   //modulus of elasticity (Young’s modulus)
scalar k1=1/Me;
scalar dG = A/(12*(22/7)*h*h);
scalar k = 4/(3*k1);
scalar Vy=((((2*(22/7))/(3*k)) *((2*(22/7))/(3*k))) * (sqrt((2/(5*density))))
*pow(y,2.5);
scalar Eeo1=0.5*mass_*Vy*Vy;
scalar need1=(5*Eeo1*r1*r1/2*k);
scalar Ge= (r1);
scalar G=6.6742*(pow(10,11));
scalar F0= mag(data.g())*mass_;
scalar Fone=Fo+3*dG*(22/7)*Gc+ pow((pow(Fo+3*dG*( 22/7)*Gc,2)-Fo*Fo),0.5);
scalar Es=(pow((Gc*Fone/k),(2/3))*dG);
scalar Vcrr=((sqrt(2*Es/(mass_)))/1e4);
double Uposp = fabs(mag(Uposp)) ;
```

// check for resuspension
if (Uposp > Vcrr && flagcheckb==1){
    ofstream resuspension_d;
    resuspension_d.open("liahmadi2_resuspension_d", ios::out | ios::app);
    resuspension_d << d_ << "n";
    resuspension_d.close();
    ofstream resuspension_Vcrr;
    resuspension_Vcrr.open("liahmadi2_resuspension_Vcrr", ios::out | ios::app);
    resuspension_Vcrr << Vcrr << "n";
    resuspension_Vcrr.close();
    ofstream resuspension_Uposp;
    resuspension_Uposp.open("liahmadi2_resuspension_Uposp", ios::out | ios::app);
    resuspension_Uposp << Uposp << "n";
}
As already mentioned in 'icoLagrangianFoam', particles either bounced or deposited depending on a variable set by the user in the test case. This code in the boundary check section was rewritten to allow particles colliding with a surface to bounce or deposit depending on their velocity and a critical velocity for bounce, \( V_{cr} \). The first step in this procedure was to calculate \( V_{cr} \) (point 6 in Fig. 4.5) following the theory as described in chapter 3.2 and code excerpt 4.11.
scalar \( F_{one} = F_{o} + 3dG \times \frac{22}{7} \times G_{c} + \text{pow}(\text{pow}(F_{o} + 3dG \times \frac{22}{7} \times G_{c}, 2) - F_{o}F_{o}, 0.5) \);

scalar \( E_{s} = \text{pow}(G_{c}F_{one}/k, (2/3)) \times dG \);

scalar \( E_{m} = F_{o} \times (\text{pow}(F_{one}, (2/3)) + 2F_{o} \times \text{pow}(F_{one}, (-1/3))/3 \times (\text{pow}(k, (2/3)) \times (\text{pow}(G_{c}, (1/3)))))) \);

scalar \( E_{a} = E_{m} + E_{s}; \) // rewrite to look at eas only
// scalar \( E_{a} = E_{s}; \)
scalar \( V_{cr} = (\sqrt{2E_{a}/(\text{mass}_)}); \)

// ***************************************************************************//

Code excerpt 4.12. Code to calculate \( V_{cr} \).

At this point a check was run to see whether a particle that had collided with a surface, bounced or deposited (point 7 in Fig. 4.5). A particle bounces elastically if \( \text{Flagcheck} \) is not equal to one and the particle velocity, \( V_{\text{elp}} \), is greater than the critical velocity for bounce, \( V_{cr} \). If \( \text{Flagcheck} \) was not equal to one and \( V_{cr} \) was greater than \( V_{\text{elp}} \) the particle deposits and its velocity is set to zero. If \( \text{Flagcheck} \) was equal to one, the particle had already deposited and its velocity remained at zero.

// ***************************************************************************//

if (Un > 0) {
  if (velp >= Vcr1 && flagcheckb != 1) {
    U() = (1.0 + data.wallElasticity()) * Un * Sf;
  }
  ofstream bounce_d;
  bounce_d.open("sand_bounce_d", ios::out | ios::app);
  bounce_d << d_ << "\n";
  bounce_d.close();

  ofstream bounce_vcrb;
  bounce_vcrb.open("sand_bounce_vcrb", ios::out | ios::app);
  bounce_vcrb << Vcr1 << "\n";
  bounce_vcrb.close();

  ofstream bounce_velp;
  bounce_velp.open("sand_bounce_velp", ios::out | ios::app);
  bounce_velp << velp << "\n";
  bounce_velp.close();
} else {
  U() = vector(0, 0, 0);
  // U()[0] = 0;
The implementation of the above code in 'icoLagrangianFoam' allowed particle surface interactions to be treated correctly.

4.3 Computational Development

The computational implementation discussed in the previous section was performed on openSUSE 10.3 operating system. The architecture used was a HP Pavilion with 2 AMD Athlon 64 X2 Dual Core Processor 4000+ at a clock speed of 2100 MHz and with 2 GB of RAM. Fig. 4.5 is a flowchart of the development steps used in the implementation of bounce and resuspension in 'icoLagrangianFoam'.
Computational Implementation

To begin with, OpenFOAM\textsuperscript{[1]} was chosen as the CFD to develop the presented code in. The reason it was chosen was that its nature as an open source code program allowed the user to develop user based codes to deal with bounce, deposition and resuspension. As was already discussed in chapter section 4.2, ‘icoLagrangianFoam’ was developed to incorporate hard spheres and allowed deposition or bounce in OpenFOAM\textsuperscript{[1]}.

The first development step in the present work was to become familiar with how ‘icoLagrangianFoam’ dealt with the fluid flow, the particles, particle transport and particle boundary collisions. A basic test case supplied with ‘icoLagrangianFoam’ was

\begin{itemize}
  \item Inner workings of ‘icoLagrangianFoam’ examined.
  \item Particles treatment changed to allow them to remain in the simulation after depositing.
  \item Calculation of $V_{cr}$.
  \item Bounce/Deposition check. Output to terminal added.
  \item Flagcheck added.
  \item Added a method to account for deposited particles when dealing with an impact.
  \item Test cases developed to test the validity of the presented work.
  \item Output to text files added.
  \item Resuspension check added.
  \item Added a calculation for $U_{pos}$ & $V_{cr}$.
  \item Added a method to account for deposited particles when calculating a particle’s velocity.
\end{itemize}

\textbf{Fig. 4.7. Development steps used to implement the code.}
simulated and the author examined how particles were treated and the fluid flow was modelled. This basic test case was used in the following development step simulations. This led to an understanding of how 'icoLangrangianFoam' was implemented.

The next development step was to stop the particles being removed from the simulation, when particles deposited. As detailed in the previous section, the code in ‘HardBallParticle’ was modified to stop the particles being deleted from the simulation, when they collided with the surface.

Development step three involved the addition of a method to calculate $V_{cr}$. This calculation was checked, by outputting the relevant information to the terminal and checking $V_{cr}$ against known values. At this point the implementation of bounce and deposition was rewritten as discussed in chapter section 4.2. This allowed the limitation from 'icoLangrangianFoam' of particles, either bouncing or depositing to be overcome and particles could now bounce and deposit, depending on the velocity of the particle and its calculated $V_{cr}$ during a simulation.

The fifth, sixth and seventh development steps involved the inclusion of Flagcheck as detailed in chapter section 4.2. With these steps included, particles no longer deposited or bounced if they had already deposited. The seventh step allowed the correct calculation of the velocity of the particle depending on Flagcheck.

The eighth development step involved the calculation of $V_{cr}$ and $U_{posp}$. These values were validated by use of the basic test case. At this point the ninth development step involved the inclusion of a resuspension check as detailed in chapter section 4.2.

The tenth development step was the inclusion of an output to text files when a particle bounces, deposits, or resuspends. The eleventh development step involved the creation of test cases to validate and evaluate the developed code. The computational resource usage of the development steps and the full test case is discussed in chapter section 4.4 and the results of the test cases are examined in chapter 5.
4.4 Computational testing.

To test the efficiency of the presented algorithm to incorporate bounce and resuspension, a test case was simulated under each of the development steps and the memory usage and cumulative execution time was recorded. The test case involved the injection of 1000 particles from 0.1 s to 5.1 s into the domain of 0.1 m x 0.1 m x 0.1 m box with a flow in the box induced by 5 m s\(^{-1}\) flow at the top of the box; the test case was simulated with a time step of 0.05 s over 10 s. When memory usage was examined, it was found that the addition of the above approach had no effect on memory usage, when it was simulated on a single processor. Each simulation used the same virtual memory of 23.6 MB and 10.3 MB in resident memory, except the final simulation (in parallel) which had more virtual memory of 57.0 MB. This extra virtual memory requirement was needed due to particles moving between the subdomains during the simulation. When CPU time is examined, there is an increase seen in the CPU time as detailed in Table 4.2.
<table>
<thead>
<tr>
<th>Development step</th>
<th>Description</th>
<th>Cumulative CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>'icoLagrangianFoam'</td>
<td>Original code (with 'perfect sink' boundary condition).</td>
<td>563.94</td>
</tr>
<tr>
<td>'icoLagrangianFoam'</td>
<td>Original code (with bounce on).</td>
<td>619.68</td>
</tr>
<tr>
<td>Step 1</td>
<td>Particles were kept in the simulation after they deposited.</td>
<td>620.46</td>
</tr>
<tr>
<td>Step 2</td>
<td>$V_{cr}$ calculation added.</td>
<td>623.94</td>
</tr>
<tr>
<td>Step 3</td>
<td>Bounce/deposition check and output to terminal.</td>
<td>626.88</td>
</tr>
<tr>
<td>Step 4</td>
<td>Flagcheck included.</td>
<td>626.76</td>
</tr>
<tr>
<td>Step 5</td>
<td>Account for deposited particles added when dealing with bounce/deposition check.</td>
<td>620.7</td>
</tr>
<tr>
<td>Step 6</td>
<td>Account for deposited particles added when dealing with bounce/deposition check.</td>
<td>624.66</td>
</tr>
<tr>
<td>Step 7</td>
<td>Calculation for $V_{cr}$ and $U_{posp}$ added.</td>
<td>625.86</td>
</tr>
<tr>
<td>Step 8</td>
<td>Resuspension check added.</td>
<td>651.72</td>
</tr>
<tr>
<td>Step 9</td>
<td>Output to text files when bounce, deposition and resuspension occur.</td>
<td>751.08</td>
</tr>
<tr>
<td>Step 9 (Rerun in parallel)</td>
<td>Simulation in parallel on 2 processors.</td>
<td>8401.2</td>
</tr>
</tbody>
</table>

Table 4.2. Summary of CPU times for development steps test cases.
It is evident from the table 4.2 that the additional CPU time needed for the inclusion of development step one to seven is only 1%. There is a 4.5% increase in CPU time between the average time for development of step one to seven compared to the development of step eight. There is a 20.4% increase in CPU time between development of step one to seven and the development of step nine. When the execution time is examined as presented in table 4.5 and Fig. 4.6, it is evident that the implementation of the development steps from one to seven have an insignificant effect on execution time from 'icoLagrangianFoam' with bounce on and there is an increase from the execution time of 'icoLagrangianFoam' with deposition on ('perfect sink). The difference can be accounted for by the fact that when a particle deposits in 'icoLagrangianFoam' it is deleted from the simulation to save on computational resources; therefore we see a smaller execution time. There is an increase of 5.5% in the execution time, for development step eight when resuspension is added compared to the execution average time for development step one to seven. The biggest increase in execution time of 21.2% comes from outputting to the text files, when a particle bounces, deposits or resuspends. There was no method to output the information on a boundary collision original in 'icoLagrangianFoam' and therefore, it was imperative to add this, so that bounce, resuspension and deposition could be studied. When development step nine is simulated in parallel, initially there is a decrease in execution time in relation to the execution time seen for simulations up to development step seven. As the number of particles increases and the number of particles moving subdomains increases, this saving in execution time is lost and there is a 13.38% increase in the final execution time over the final execution time of development step 9 on a serial machine.
Fig. 4.8. Execution times for the development steps of the present work.
### Computational Implementation

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Execution time (s)</th>
<th>Execution time (s)</th>
<th>Execution time (s)</th>
<th>Execution time (s)</th>
<th>Execution time (s)</th>
<th>Execution time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>28.83</td>
<td>28.95</td>
<td>29.63</td>
<td>29.37</td>
<td>29.67</td>
<td>19.51</td>
</tr>
<tr>
<td>1</td>
<td>54.39</td>
<td>54.36</td>
<td>55.28</td>
<td>55.97</td>
<td>54.87</td>
<td>38.03</td>
</tr>
<tr>
<td>1.5</td>
<td>80.6</td>
<td>80.65</td>
<td>81.91</td>
<td>83.68</td>
<td>86.87</td>
<td>67.30</td>
</tr>
<tr>
<td>2</td>
<td>107.43</td>
<td>108.17</td>
<td>109.41</td>
<td>112.44</td>
<td>121.16</td>
<td>101.28</td>
</tr>
<tr>
<td>2.5</td>
<td>134.65</td>
<td>136.29</td>
<td>137.38</td>
<td>142.06</td>
<td>157.75</td>
<td>135.77</td>
</tr>
<tr>
<td>3</td>
<td>162.61</td>
<td>165.14</td>
<td>166.89</td>
<td>172.71</td>
<td>196.42</td>
<td>174.20</td>
</tr>
<tr>
<td>3.5</td>
<td>190.71</td>
<td>194.68</td>
<td>196.79</td>
<td>204.34</td>
<td>239.37</td>
<td>220.22</td>
</tr>
<tr>
<td>4</td>
<td>218.91</td>
<td>225.17</td>
<td>227.33</td>
<td>237.05</td>
<td>284.42</td>
<td>264.15</td>
</tr>
<tr>
<td>4.5</td>
<td>247.49</td>
<td>257.16</td>
<td>258.82</td>
<td>270.85</td>
<td>328.17</td>
<td>313.04</td>
</tr>
<tr>
<td>5</td>
<td>276.82</td>
<td>289.52</td>
<td>291.14</td>
<td>304.77</td>
<td>372.42</td>
<td>364.74</td>
</tr>
<tr>
<td>5.5</td>
<td>306.13</td>
<td>322.6</td>
<td>324.17</td>
<td>339.76</td>
<td>416.02</td>
<td>429.21</td>
</tr>
<tr>
<td>6</td>
<td>335.13</td>
<td>355.65</td>
<td>357.2</td>
<td>374.64</td>
<td>460.12</td>
<td>477.96</td>
</tr>
<tr>
<td>6.5</td>
<td>363.93</td>
<td>388.83</td>
<td>390.03</td>
<td>409.48</td>
<td>499.53</td>
<td>521.86</td>
</tr>
<tr>
<td>7</td>
<td>392.68</td>
<td>421.96</td>
<td>422.8</td>
<td>443.99</td>
<td>536.16</td>
<td>562.68</td>
</tr>
<tr>
<td>7.5</td>
<td>421.26</td>
<td>454.94</td>
<td>455.68</td>
<td>478.61</td>
<td>573.97</td>
<td>606.62</td>
</tr>
<tr>
<td>8</td>
<td>449.83</td>
<td>487.91</td>
<td>488.21</td>
<td>513.2</td>
<td>610.7</td>
<td>646.37</td>
</tr>
<tr>
<td>8.5</td>
<td>478.35</td>
<td>520.83</td>
<td>520.53</td>
<td>547.89</td>
<td>646.95</td>
<td>699.34</td>
</tr>
<tr>
<td>9</td>
<td>506.9</td>
<td>553.67</td>
<td>552.84</td>
<td>582.55</td>
<td>681.91</td>
<td>760.48</td>
</tr>
<tr>
<td>9.5</td>
<td>535.33</td>
<td>586.8</td>
<td>585.07</td>
<td>617.01</td>
<td>716.61</td>
<td>809.58</td>
</tr>
<tr>
<td>10</td>
<td>563.83</td>
<td>619.58</td>
<td>617.37</td>
<td>651.67</td>
<td>750.99</td>
<td>851.52</td>
</tr>
</tbody>
</table>

*Table 4.3. Execution times for the development steps for the presented work.*

Next the scaling of the execution time for an increase in the number of particles in the domain was examined, when a test case was simulated in serial and parallel. The test cases used involved the injection of 1, 10, 100 and 1000 particles into the domain of 0.1 m x 0.1 m x 0.1 m box, with a flow in the box induced by 5 m s\(^{-1}\) flow at the top of the box; the test cases were simulated with a time step of 0.005 s over 10 s. The results are presented in Table 4.4 and Fig. 4.9.
### Computational Implementation

<table>
<thead>
<tr>
<th>Execution time for 1 particle</th>
<th>10 particles</th>
<th>100 particles</th>
<th>1000 particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>serial</td>
<td>28.28</td>
<td>48.12</td>
<td>60.66</td>
</tr>
<tr>
<td>parallel</td>
<td>20.19</td>
<td>29.48</td>
<td>29.79</td>
</tr>
<tr>
<td>serial</td>
<td>52.62</td>
<td>65.05</td>
<td>78.75</td>
</tr>
<tr>
<td>parallel</td>
<td>36.21</td>
<td>56.45</td>
<td>78.75</td>
</tr>
<tr>
<td>serial</td>
<td>76.82</td>
<td>80.82</td>
<td>103.09</td>
</tr>
<tr>
<td>parallel</td>
<td>51.54</td>
<td>87.97</td>
<td>91.34</td>
</tr>
<tr>
<td>serial</td>
<td>101.15</td>
<td>95.12</td>
<td>115.96</td>
</tr>
<tr>
<td>parallel</td>
<td>68.11</td>
<td>124.47</td>
<td>129.17</td>
</tr>
<tr>
<td>serial</td>
<td>125.45</td>
<td>109.69</td>
<td>139.96</td>
</tr>
<tr>
<td>parallel</td>
<td>82.41</td>
<td>141.52</td>
<td>166.46</td>
</tr>
<tr>
<td>serial</td>
<td>149.9</td>
<td>125.79</td>
<td>158.53</td>
</tr>
<tr>
<td>parallel</td>
<td>98.06</td>
<td>167.49</td>
<td>206.5</td>
</tr>
<tr>
<td>serial</td>
<td>174.12</td>
<td>141.35</td>
<td>178.09</td>
</tr>
<tr>
<td>parallel</td>
<td>112.47</td>
<td>194.39</td>
<td>251.42</td>
</tr>
<tr>
<td>serial</td>
<td>198.48</td>
<td>156.59</td>
<td>195.87</td>
</tr>
<tr>
<td>parallel</td>
<td>130.69</td>
<td>221.16</td>
<td>298.87</td>
</tr>
<tr>
<td>serial</td>
<td>222.74</td>
<td>171.95</td>
<td>247.14</td>
</tr>
<tr>
<td>parallel</td>
<td>147.12</td>
<td>217.5</td>
<td>344.26</td>
</tr>
<tr>
<td>serial</td>
<td>247.01</td>
<td>187.08</td>
<td>232.64</td>
</tr>
<tr>
<td>parallel</td>
<td>161.71</td>
<td>272.8</td>
<td>390.38</td>
</tr>
<tr>
<td>serial</td>
<td>271.35</td>
<td>202.46</td>
<td>251.52</td>
</tr>
<tr>
<td>parallel</td>
<td>176.15</td>
<td>298.2</td>
<td>436</td>
</tr>
<tr>
<td>serial</td>
<td>295.71</td>
<td>217.36</td>
<td>267.94</td>
</tr>
<tr>
<td>parallel</td>
<td>191.69</td>
<td>323.64</td>
<td>480.6</td>
</tr>
<tr>
<td>serial</td>
<td>320.22</td>
<td>233.61</td>
<td>349.01</td>
</tr>
<tr>
<td>parallel</td>
<td>206.9</td>
<td>324.6</td>
<td>520.18</td>
</tr>
<tr>
<td>serial</td>
<td>344.32</td>
<td>248.71</td>
<td>374.26</td>
</tr>
<tr>
<td>parallel</td>
<td>222.16</td>
<td>303.16</td>
<td>557.54</td>
</tr>
<tr>
<td>serial</td>
<td>368.56</td>
<td>263.47</td>
<td>399.81</td>
</tr>
<tr>
<td>parallel</td>
<td>236.93</td>
<td>399.81</td>
<td>595.81</td>
</tr>
<tr>
<td>serial</td>
<td>392.77</td>
<td>278.95</td>
<td>425.2</td>
</tr>
<tr>
<td>parallel</td>
<td>252.48</td>
<td>337.7</td>
<td>633.56</td>
</tr>
<tr>
<td>serial</td>
<td>416.97</td>
<td>420.54</td>
<td>357.32</td>
</tr>
<tr>
<td>parallel</td>
<td>267.83</td>
<td>450.54</td>
<td>669.89</td>
</tr>
<tr>
<td>serial</td>
<td>441.24</td>
<td>475.81</td>
<td>704.82</td>
</tr>
<tr>
<td>parallel</td>
<td>281.93</td>
<td>475.81</td>
<td>727.59</td>
</tr>
<tr>
<td>serial</td>
<td>465.51</td>
<td>501.05</td>
<td>393.03</td>
</tr>
<tr>
<td>parallel</td>
<td>298.73</td>
<td>501.05</td>
<td>739.37</td>
</tr>
<tr>
<td>serial</td>
<td>489.73</td>
<td>525.65</td>
<td>410.25</td>
</tr>
<tr>
<td>parallel</td>
<td>313.73</td>
<td>773.65</td>
<td>794.22</td>
</tr>
</tbody>
</table>

**Table 4.4.** Execution times of a test case with varying particle numbers simulated in serial and parallel.
The execution times for the 1 and 10 particle simulations in serial are very similar and there is a slight increase in the execution time for the 100 particle simulation in serial. It is evident that there is a saving in execution time for these simulations in parallel. There is a 35.9 % decrease in execution time when the 1 particle test case is simulated in parallel compared to the 1 particle test case simulated in serial. There is a 30.8 %
decrease in execution time when the 10 particle test case is simulated in parallel compared to the 10 particle test case simulated in serial. There is a 22 % decrease in execution time when the 100 particle test case is simulated in parallel when compared to the 100 particle test case simulated in serial. When the 1000 particle test case is examined, there is an increase in the execution time of 2.7% for the simulation in parallel compared to the simulation in serial. The reason for the increase in execution time in the 1000 particle test case, compared to the other test cases, is due to an increase in message passing needed as particles move from one subdomain to another. When a particle moves from one subdomain to another subdomain, all the details of the particle need to be passed to the processor handling the second subdomain. As the number of particles increases, there is an equivalent increase in the number of particles moving from one subdomain to another and therefore an increase in the time used in message passing. The execution time gained by simulating the test case in parallel is lost to the execution time needed to handle the message passing between the processors.

The efficiency of the main test cases was examined by comparing a full 150 s simulation of the Li and Ahmadi test case with 5 m s$^{-1}$ flow on a serial machine and in parallel on two processors. The Li and Ahmadi test case involved the injection of 1000 particles from 0.1 s to 5.1 s into the domain of 0.1 m x 0.1 m x 0.1 m box with a flow in the box induced by 5 m s$^{-1}$ flow; the test cases were simulated with a time step of 0.05 s over 150 s. The execution time and clock time for both simulations can be seen in Fig. 4.10 and 4.11.
Fig. 4.10. Execution time and clock time for Li and Ahmadi test case with $5 \text{ m s}^{-1}$ flow on a serial machine.

Fig. 4.11. Execution time and clock time for Li and Ahmadi test case with $5 \text{ m s}^{-1}$ flow in parallel on 2 processors.
It is evident when Fig. 4.9 and 4.10 are compared that the execution time and clock time in the parallel simulation is longer than those for the simulation on a serial machine. There is a 27% increase in the execution time for the parallel simulation compared to the serial simulation. This trend is also seen in the evaluation of the execution time for test cases of 1, 10, 100 and 1000 particles in a domain, simulated in serial and parallel. As previously mentioned, with more particles in the domain, when a test case is simulated in parallel, more particles move between the subdomains. Due to this, more execution time is needed to handle the increased message passing used to pass the particles details between processors. This bottleneck eventually leads to the time saved in the calculations being lost to message passing between the processors.

4.5 Conclusions

By the proper execution of these implementation and development steps, a realistic portrayal of bounce, deposition and resuspension were incorporated into 'icoLangrangianFoam' in a computational cost effective way. While these steps were implemented in 'icoLangrangianFoam', there is no reason that their proper implementation in the other CFDs cannot be undertaken, as most CFDs handle particle deposition and particle movement in a similar manner. With these steps implemented in the correct area of any other CFD's code, bounce and resuspension could be added to the CFD's Lagrangian simulation. Therefore it is clear that this method is a highly robust, computationally inexpensive method to include bounce, deposition and resuspension to any Lagrangian simulation with results that agree well with previous experimental and simulated data as discussed in chapter 5.
5. Results

Contents

5 Results ................................................................. 66

5.1 Introduction ......................................................... 67

5.2 Test Case 1: Deposition of Pollen/spores on leaf surfaces . . . . 68
  5.2.1 Comparison of simulated data and the experimental data for a flow velocity of 2.82 m s$^{-1}$ ................. 69
  5.2.2 Comparison of simulated data and the experimental data for a flow velocity of 3.41 m s$^{-1}$ ................. 73
  5.2.3 Comparison of simulated data and the experimental data for a flow velocity of 4.59 m s$^{-1}$ ................. 77
  5.2.4 4.59 m s$^{-1}$ test case simulation with 'perfect sinks' .......... 81
  5.2.5 4.59 m s$^{-1}$ test case simulation comparing resuspension and no resuspension .......................... 82
  5.2.6 15 m s$^{-1}$ simulation where bounce dominates .............. 84
  5.2.7 Conclusions on the comparison of simulated particle behaviour vs. Paw and Braaten's experimental data. ................................. 85

5.3 Particle deposition: Li and Ahmadi test case. ............... 86
  5.3.1 Comparison of simulated data and the data of Li and Ahmadi (2 m s$^{-1}$ flow velocity) ................. 87
  5.3.2 Comparison of simulated data and the data of Li and Ahmadi (3 m s$^{-1}$ flow velocity) ................. 92
  5.3.3 Comparison of simulated data and the data of Li and Ahmadi (5 m s$^{-1}$ flow velocity) ................. 96
  5.3.4 5 m s$^{-1}$ Li and Ahmadi test case simulation with 'perfect sinks' ................................................. 102

56
5.1 Introduction

To evaluate the effectiveness of the above model and theory, three test cases were chosen. The test cases range over a wide range of particle sizes and flows and come from both experimental and simulated research. Test case one is based on the experimental work carried out by Paw and Braaten (1992)[3], test case two is based on the simulated work of Li and Ahmadi (1993)[4] and finally a model of sand sedimentation is chosen as test case three, to show the versatility of the presented theory. The choice of these test cases shows the robustness of the above theory across a wide range of particle sizes, particle densities and flow conditions.
5.2 Test Case 1: Deposition of pollen/spores on leaf surfaces.

![Particle transport and flow for the Paw and Braaten test case in a 0.1 m x 0.1 m x 0.1 m box with a flow in the box induced by 5 m s\(^{-1}\) flow across the top.](image)

Paw and Braaten (1992)[3] carried out experiments in a low-wind speed tunnel to examine the processes of ideal deposition, net deposition, rebound and resuspension for large particles (lycoperidium spores). The particle diameter range was 33 µm to 36 µm and the particle density was 635 kg m\(^{-3}\). In the simulation, 1000 particles were injected using the injector function as discussed in Chapter 4. The injection lasted 5 s from 0.1 s to 5.1 s and was centred on a position (0.05, 0.05, 0.05) in the domain with a random variance of 0.02 and the flow in the domain is driven by a flow at the top of the domain as seen in Fig. 5.1; the simulation was simulated with a time step of 0.005 s over 150 s. Four different velocities were simulated and compared to the data from Paw and Braaten (1992)[3]. The velocities simulated were 2.82 m s\(^{-1}\), 3.41 m s\(^{-1}\), 4.59 m s\(^{-1}\) and 15 m s\(^{-1}\).
5.2.1 Comparison of simulated data and the experimental data for a flow velocity of 2.82 m s\(^{-1}\).

Fig. 5.2. A plot of the deposition velocity, \(V_{\text{elp}}\) vs. particle diameter, \(d_p\), showing the area bounded by the upper value of the critical velocity, \(V_{\text{cr}}\).

Fig. 5.2 shows the simulated deposition velocity for lycopodium particles ranging in diameter size from 33 to 36 µm. The grey area plot of \(V_{\text{cr}}\) (from equation 3.16) in Fig. 5.2 indicates values of \(V_{\text{elp}}\) for which a particle will deposit; above this area the particle would bounce. The wind speed in the domain for this simulation was 2.83 m s\(^{-1}\) and the plot clearly shows that all the particles deposited below the critical velocity for bounce to occur. In total 474 particles deposited in this simulation. The average deposition velocity for the Lycopodium particles is 0.0397 m s\(^{-1}\), Paw and Braaten's (1992)\[3\] experimental work indicated that the ideal deposition velocity for flow rate of 2.82 m s\(^{-1}\)
was 0.047 m s\(^{-1}\) and that the net deposition velocity was 0.26 m s\(^{-1}\). The decrease in the simulated deposition velocity over the particle diameter range is indicative of the fact that the possibility of bounce increases with diameter size as indicated by Li and Ahmadi (1993)[4]. The average \(V_{cr}\) for the particle range for a flow of 2.83 m s\(^{-1}\) is 0.044 m s\(^{-1}\). This shows a good agreement between the simulation in the presented work and the experimental work carried out by Paw and Braaten (1992)[3]. Fig. 5.3 shows that good agreement is evident across the whole particle range.

![Fig. 5.3. A plot of critical velocity, \(V_{cr}\) vs. particle diameter, \(d_p\). Included on the plot is the ideal deposition velocity, \(V_i\) and the net deposition velocity, \(V_d\) for a flow velocity of 2.82 m s\(^{-1}\) from Paw and Braaten (1992)[3].](image-url)
Paw and Braaten (1992) indicated that rebound can occur at low and intermediate wind speed values of 2.4 m s$^{-1}$ and above. In the simulation, rebound occurs when $V_{elp}$ is greater than the $V_{cr}$ of the particle; the values for which a particle would deposit is indicated by the grey area plot of $V_{cr}$, and above this the particle bounces. This can be seen in Fig. 5.4. This shows that particles in the simulation, above the critical velocity needed for bounce to occur, bounced 478 times.

![Fig. 5.4. A plot of the particle deposition velocities for particles that bounced due to their $V_{elp}$ being greater than $V_{cr}$ and it shows the upper boundary of the $V_{cr}$ values.](image)
For a particle to resuspend, the velocity of the flow at the position of the particle, $U_{posp}$ needs to be greater than the critical velocity for resuspension, $V_{crr}$ (this range of values is indicated by the area plot of $V_{crr}$ (from equation 3.20); to resuspend a particle must be above this area plot in Fig. 5.5). Paw and Braaten (1992)[3] came to the conclusion that reentrainment processes are most important at low to intermediate wind speeds $< 4$ m s$^{-1}$. In the simulation, particles resuspended 458 times. The average $U_{posp}$ is 0.058 m s$^{-1}$, being generated from the stream-wise wind speed of 2.82 m s$^{-1}$ at the top of the simulation domain.

![Fig. 5.5. A plot of $U_{posp}$ vs. $d_p$. $V_{crr}$ is shown on the plot.](image-url)
5.2.2 Comparison of simulated data and the experimental data for a flow velocity of 3.41 m s\(^{-1}\).

For the second simulation of the Paw and Braaten (1992)\textsuperscript{[3]} test case flow velocity at the top of the domain was set to 3.41 m s\(^{-1}\). Previous simulations and experimental work indicate that there should be more bounce and resuspension than in the previous case. An implication of more resuspension is that there are more particles available to deposit. Fig. 5.6 shows that all the particles that deposited had velocities below the \(V_{cr}\) for its diameter. The number of particles that deposited was 524 during the simulation, with a range of \(V_{elp}\) values from 0.0008 m s\(^{-1}\) to 0.045 m s\(^{-1}\) and a \(V_{cr}\) range of 0.042 m s\(^{-1}\) to 0.045 m s\(^{-1}\). This represents an 8.7\% increase in the number of particles deposited. The average deposition velocity for the Lycopodium particles was 0.031 m s\(^{-1}\); Paw and Braaten's (1992)\textsuperscript{[3]} experimental data indicated that the ideal deposition velocity for U of 3.41 m s\(^{-1}\) was 0.056 m s\(^{-1}\) and that the net deposition velocity was 0.027 m s\(^{-1}\). Again there is a good agreement between the experimental data as published by Paw and Braaten (1992)\textsuperscript{[3]} and the average simulated \(V_{cr}\) of 0.044 m s\(^{-1}\). Fig. 5.7 shows the \(V_{cr}\) for the complete range of \(d_p\) and again there is good agreement with the experimental data from Paw and Braaten (1992)\textsuperscript{[3]}.
Fig. 5.6. A plot of the deposition velocity, $V_{elp}$ vs. particle diameter, $d_p$, showing the area bounded by the upper value of the critical velocity, $V_{cr}$.

Fig. 5.7. A plot of critical velocity, $V_{cr}$ vs. particle diameter, $d_p$. Included on the plot is the ideal deposition velocity, $V_i$ and the net deposition velocity, $V_d$ for a flow velocity of 3.41 m s$^{-1}$ (from Paw and Braaten (1992)\cite{3}).
For the 3.41 m s\(^{-1}\) case there was a 6.6% increase in the number of bounces (to 519 bounces), compared to the 2.82 m s\(^{-1}\) case. The average velocity of the particles bouncing was 0.274 m s\(^{-1}\) with a range of 0.045 m s\(^{-1}\) to 0.92 m s\(^{-1}\). The V\(_{cr}\) range was 0.042 m s\(^{-1}\) to 0.045 m s\(^{-1}\) and an average V\(_{cr}\) of 0.044 m s\(^{-1}\). Fig. 5.8 shows that, for the given particle diameter, V\(_{elp}\) was greater than the V\(_{cr}\) for that diameter.

Fig. 5.8. A plot of V\(_{elp}\) vs. particle diameter, d\(_p\) with V\(_{cr}\) values shown as an area plot.
There is a 13% increase in the rate of resuspension in the 3.41 m s\(^{-1}\) simulation. The Lycopodium particles resuspended 520 times in this simulation, with a range for \(U_{\text{posp}}\) of 0.0265 m s\(^{-1}\) to 0.65 m s\(^{-1}\) and an average of 0.102 m s\(^{-1}\). These \(U_{\text{posp}}\) values were above the \(V_{\text{crr}}\) for the corresponding particle size. The range of \(V_{\text{crr}}\) was 0.022 m s\(^{-1}\) to 0.025 m s\(^{-1}\) with an average of \(V_{\text{crr}}\) of 0.024 m s\(^{-1}\) as seen in Fig. 5.9.

Fig. 5.9. A plot of the position of the particle velocity, \(U_{\text{posp}}\) vs. particle diameter, \(d_p\). Included on the plot is the \(V_{\text{crr}}\) region above which a particle resuspends.
5.2.3 Comparison of simulated data and the experimental data for a flow velocity of 4.59 m s$^{-1}$.

The outcome of the simulation with a flow velocity of 4.59 m s$^{-1}$ was that there was an increase in deposition occurrences of 87% compared with the 2.59 m s$^{-1}$ simulation and 70% compared with the 3.42 m s$^{-1}$ simulation. The $V_{\text{elp}}$ range for this simulation was 0.0009 m s$^{-1}$ to 0.045 m s$^{-1}$, with an average value of 0.038 m s$^{-1}$. Paw and Braaten (1992)\cite{3} indicated that at a flow velocity of 4.5 m s$^{-1}$, the ideal deposition velocity, $V_i$ was 0.073 m s$^{-1}$ and the net deposition velocity, $V_d$ was 0.03 m s$^{-1}$. Fig. 5.10 shows that the deposition velocity of the particle was below $V_{cr}$ for the 888 deposition incidents.

**Fig. 5.10.** A plot of the deposition velocity, $V_{\text{elp}}$ vs. particle diameter, $d_p$, showing the area bounded by the upper value of the critical velocity, $V_{cr}$, for the 4.59 m s$^{-1}$ simulation.
Fig. 5.11 is a plot of the critical velocity vs. particle diameter and indicates good agreement between the $V_i$ and $V_d$ values as reported by Paw and Braaten (1992)[3] and the $V_{cr}$ as calculated for each particle size. The range for $V_{cr}$ was 0.042 m s$^{-1}$ to 0.045 m s$^{-1}$, with an average of 0.044 m s$^{-1}$.

![Graph of critical velocity vs. particle diameter](image)

Fig. 5.11. A plot of critical velocity, $V_{cr}$ vs. particle diameter, $d_p$. Included on the plot is the ideal deposition velocity, $V_i$ and the net deposition velocity, $V_d$ for a flow velocity of 4.59 m s$^{-1}$ from Paw and Braaten (1992)[3].
For the number of bounces in the 4.59 m s\(^{-1}\) simulation, there is a 92% increase compared with the 2.83 m s\(^{-1}\) simulation and 80% increase compared with the 3.41 m s\(^{-1}\). 939 particles bounce in the simulation with a \(V_{\text{elp}}\) range of 0.042 m s\(^{-1}\) to 1.05 m s\(^{-1}\) and an average \(V_{\text{elp}}\) of 0.43 m s\(^{-1}\). Fig. 5.12 shows the \(V_{\text{elp}}\) of the bouncing particles and the corresponding \(V_{\text{cr}}\).

![Figure 5.12](image)

*Fig. 5.12. A plot of deposition velocity, \(V_{\text{elp}}\) vs. particle diameter, \(d_p\) with \(V_{\text{cr}}\) values shown as an area plot.*
When resuspension for the 4.59 m s\(^{-1}\) simulation is examined, it is evident that there is a 90% increase from the initial simulation and a 67% increase from the 3.42 m s\(^{-1}\) simulation. 870 particles resuspended in the 4.59 m s\(^{-1}\) simulation. The average \(V_{cr}\) for this simulation was 0.024 m s\(^{-1}\), in the range of 0.022 m s\(^{-1}\) to 0.025 m s\(^{-1}\). The average \(U_{\text{posp}}\) for this simulation was 0.095 m s\(^{-1}\), in the range of 0.023 m s\(^{-1}\) to 0.85 m s\(^{-1}\). Fig. 5.13 illustrates the \(U_{\text{posp}}\) and corresponding \(V_{cr}\) for resuspension in the 4.59 m s\(^{-1}\) simulation.

![Graph](image.png)

**Fig. 5.13.** A plot of the velocity of the fluid at the position of the particle, \(U_{\text{posp}}\) vs. particle diameter, \(d_p\). Included on the plot is the \(V_{cr}\) region above which a particle resuspends.
5.2.4 4.59 m s$^{-1}$ Paw and Braaten test case simulation with 'perfect sinks'.

Fig. 5.14 shows the $V_{\text{elb}}$ and $V_{\text{cr}}$ values for a flow velocity of 4.59 m s$^{-1}$, where the boundaries were assumed to be perfect sinks. From this simulation it is evident that, when bounce is not included, many particles that would have bounced have instead deposited and distorted the average velocity for particles depositing. During the simulation 443 particles deposited. Of these only 19 were below the corresponding $V_{\text{cr}}$ of the particle while 424 were above. The average $V_{\text{elb}}$ for particles that were above $V_{\text{cr}}$ was 0.45 m s$^{-1}$ with a minimum value of 0.042 m s$^{-1}$ (to corresponding $V_{\text{cr}}$ value of 0.0418 m s$^{-1}$) and a maximum value of 1.11 m s$^{-1}$. This clearly indicates that bounce is an important factor in determining the deposition velocity of a particle and should not be neglected. Fig. 5.15 shows the increase in $V_{\text{elb}}$ when bounce is not included.

![Fig. 5.14. A plot of the deposition velocity, $V_{\text{elb}}$ vs. particle diameter, $d_p$, for 'perfect sink' simulation. Included is an area plot of critical velocity, $V_{\text{cr}}$.](image-url)
Fig. 5.15. $V_{el}$ for 2.82, 3.41 and 4.59 m s$^{-1}$ when bounce occurs and at 4.59 m s$^{-1}$ with perfect sinks.

5.2.5 4.59 m s$^{-1}$ test case simulation, comparing resuspension and no resuspension.

Fig. 5.16 shows the $U_{posp}$ when particles were checked to test whether they would resuspend or stay on the boundary. For resuspension to occur the $U_{posp}$ had to be greater than $V_{crr}$. The range of $U_{posp}$ values was 0.007 m s$^{-1}$ to 0.024 m s$^{-1}$ with an average of 0.015 m s$^{-1}$ for the cases where $U_{posp}$ was less than $V_{crr}$; the $V_{crr}$ values ranging from 0.023 m s$^{-1}$ to 0.025 m s$^{-1}$ and an average of 0.024 m s$^{-1}$. For the cases where $U_{posp}$ was greater than $V_{crr}$, $U_{posp}$ range was 0.024 m s$^{-1}$ to 0.86 m s$^{-1}$ with an average of 0.095 m s$^{-1}$. The range of $V_{crr}$ was 0.022 m s$^{-1}$ to 0.025 m s$^{-1}$ with an average of 0.024 m s$^{-1}$. 

82
Fig. 5.16. Plot of $U_{\text{posp}}$ from resuspension check, with an area plot of $V_{cr}$.
5.2.6 15 m s⁻¹ simulation where bounce dominates.

An additional simulation was performed with a flow velocity of 15 m s⁻¹. The results of Paw and Braaten (1992) [3] implied that at very high wind speeds, rebound dominates over deposition and therefore resuspension. Although bounce dominates it is clear from this work and as stated by Paw and Braaten (1992) [3], that deposition and resuspension are not negligible and still occur. It is clear that bounce is the dominant process in this simulation since the sum of the number of depositions and resuspensions only make 1.2% of the total bounce, deposition and resuspension incidents. The average velocity of bounce increases dramatically from 0.43 m s⁻¹ for the 4.59 m s⁻¹ simulation to 3.66 m s⁻¹ in this case. This increase of velocity of bounce as flow velocity increases is clearly seen in Fig. 5.17 for all the Paw and Braaten (1992) [3] test case simulations.

![Fig. 5.17. V_{elp} of bounce for Paw and Braaten test case.](image)
5.2.7 Conclusions on the comparison of simulated particle behaviour vs. Paw and Braaten's experimental data.

Table 5.1. Summary of bounce, deposition and resuspension for the Paw and Braaten test case.

<table>
<thead>
<tr>
<th>Simulation flow velocity m s(^{-1})</th>
<th>Incident</th>
<th>(V_{cr}) or (V_{crr}) range</th>
<th>(V_{dwp}) or (U_{posp}) range</th>
<th>No. of incidents</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.82</td>
<td>deposition</td>
<td>0.0417 - 0.0455</td>
<td>0.0011 - 0.0455</td>
<td>474</td>
</tr>
<tr>
<td>2.82</td>
<td>bounce</td>
<td>0.0417 - 0.0455</td>
<td>0.0472 - 0.4935</td>
<td>478</td>
</tr>
<tr>
<td>3.41</td>
<td>deposition</td>
<td>0.0417 - 0.0454</td>
<td>0.0008 - 0.0452</td>
<td>524</td>
</tr>
<tr>
<td>3.41</td>
<td>bounce</td>
<td>0.0417 - 0.0454</td>
<td>0.0448 - 0.9214</td>
<td>519</td>
</tr>
<tr>
<td>3.41</td>
<td>resuspension</td>
<td>0.0220 - 0.0250</td>
<td>0.0265 - 0.6457</td>
<td>520</td>
</tr>
<tr>
<td>4.59</td>
<td>deposition</td>
<td>0.0417 - 0.0455</td>
<td>0.0009 - 0.0452</td>
<td>888</td>
</tr>
<tr>
<td>4.59</td>
<td>bounce</td>
<td>0.0417 - 0.0455</td>
<td>0.0424 - 1.0515</td>
<td>939</td>
</tr>
<tr>
<td>4.59</td>
<td>resuspension</td>
<td>0.0220 - 0.0250</td>
<td>0.0235 - 0.8552</td>
<td>876</td>
</tr>
<tr>
<td>15</td>
<td>deposition</td>
<td>0.0371 - 0.0455</td>
<td>0.0002 - 0.0454</td>
<td>674</td>
</tr>
<tr>
<td>15</td>
<td>bounce</td>
<td>0.0371 - 0.0455</td>
<td>0.0378 - 9.7660</td>
<td>113648</td>
</tr>
<tr>
<td>15</td>
<td>resuspension</td>
<td>0.0184 - 0.0250</td>
<td>0.0384 - 4.8332</td>
<td>674</td>
</tr>
</tbody>
</table>

Table 5.1 shows a summary of the test simulation results. From the 2.82 m s\(^{-1}\), 3.41 m s\(^{-1}\) and 4.59 m s\(^{-1}\) simulations it is evident that overall the values of \(V_{cr}\) and \(V_{crr}\) agrees well with the work carried out by Paw and Braaten (1992)\(^{[3]}\) and allow an accurate model of bounce, resuspension and deposition to be simulation and visualised. As is known from previous work, there should be an increase in bounce and resuspension as the velocity of the flow increases. At extremely high velocities bounce becomes the dominate process and short cuts deposition and resuspension. These trends are clearly seen over the four simulations of the Paw and Braaten's test case. From the work carried out on perfect sinks, it is clear that, to truly model particle transport, bounce and resuspension must be included. When perfect sinks were used (bounce neglected), it led to a significant increase in the average deposition velocity over the case where bounce was included.

From Table 5.1 it is evident that, with an increase in flow velocity, there is an increase in the number of incidents and in the corresponding velocity of bounce and resuspension, except for flows that have high flow velocities and thus are bounce dominated.
5.3 Particle deposition: Li and Ahmadi test case.

Li and Ahmadi (1993)[4] simulated the transport of aerosol particles ranging in diameter from 0.01 µm to 10 µm to vertical and horizontal channel surfaces in a mean fluid flow of 5 m s\(^{-1}\). Their simulation studied bounce and included a critical approach velocity \(V_c\) for bounce. In the simulation, 1000 gold particles were injected using the injector function as discussed in Chapter 4. The injection lasted 5 s from 0.1 s to 5.1 s and was centred on a position (0.05, 0.05, 0.05) in the domain with a random variance of 0.02 and the flow in the domain is driven by a flow at the top of the domain as seen in Fig. 5.18; the simulation was simulated with a time step of 0.005 s over 150 s. Three different velocities were simulated and compared to the data from Li and Ahmadi (1993)[4]. The velocities simulated were 2 m s\(^{-1}\), 3 m s\(^{-1}\) and 5 m s\(^{-1}\) and the particle diameter range was 1 µm to 10 µm with a particle density of 19300 kg m\(^{-3}\). 1 µm was chosen as the lower limit for the particle diameter range to allow Brownian motion to be neglected in the particle equation of motion.
5.3.1 Comparison of simulated data and the data of Li and Ahmadi (2 m s$^{-1}$ flow velocity).

![Graph showing deposition velocity vs. particle diameter](image)

*Fig. 5.19. A plot of the deposition velocity, $V_{elp}$ vs. particle diameter, $d_p$ for a flow rate of 2 m s$^{-1}$. Included is an area plot of $V_{cr}$."

Fig. 5.19 shows the deposition velocity for particles ranging in diameter size from 1 µm to 10 µm. The wind speed in the domain for this simulation was 2 m s$^{-1}$ and the plot clearly shows that all the particles deposited below the critical velocity for bounce to occur. During the simulation, 658 particles deposited. The $V_{elp}$ range for the particles was 0.00064 m s$^{-1}$ to 0.067 m s$^{-1}$. Li and Ahmadi (1993) indicated a critical approach velocity of $V_c$ which is compared with the calculated $V_{cr}$ in Fig. 5.20. The range of $V_c$ is 0.0028 m s$^{-1}$ to 0.24 m s$^{-1}$ while $V_{cr}$ is 0.0033 m s$^{-1}$ to 0.093 m s$^{-1}$, with averages respectively of 0.018 m s$^{-1}$ and 0.012 m s$^{-1}$. It can be seen from Fig. 5.20 that below 4 µm, $V_{cr}$ predicts a lower critical velocity than $V_c$, but above 4 µm, there is good agreement between $V_{cr}$ and $V_c$. 
Fig. 5.20. A plot of the critical velocity, $V_{cr}$, and critical approach velocity, $V_c$, vs. particle diameter, $d_p$.

145 particles bounced in this simulation at a flow velocity of 2 m s$^{-1}$. Fig. 5.21 shows the $V_{elp}$ and $V_{cr}$ for the particles, clearing showing that all the particles velocities were above the corresponding $V_{cr}$ when they contacted the surface. The $V_{cr}$ range was 0.0035 m s$^{-1}$ to 0.074 m s$^{-1}$ with an average of 0.021 m s$^{-1}$. The range of velocities at which the particles rebounded was 0.0085 m s$^{-1}$ to 0.25 m s$^{-1}$ with an average $V_{elp}$ of 0.12 m s$^{-1}$. It is again evident that as particle size increases the probability of bounce increases, as indicated by the shrinking of the $V_{cr}$ area plot in Fig. 5.21 as particle sizes increases.
Results (Li and Ahmadi test case)

Fig. 5.21. *A plot of the deposition velocity, $V_{elp}$ vs. particle diameter, $d_p$ when bounce occurred. Included is an area plot of $V_{cr}$.*

In Fig. 5.22, when $V_{cr}$ and $V_c$ are examined, the same pattern is seen as in Fig. 5.20 i.e. good agreement above 4 µm and a lower $V_{cr}$ than $V_c$ below 4 µm. The $V_{cr}$ range was 0.035 m s$^{-1}$ to 0.074 m s$^{-1}$, with an average $V_{cr}$ of 0.021 m s$^{-1}$. The $V_c$ range was 0.031 m s$^{-1}$ to 0.18 m s$^{-1}$, with an average of 0.037 m s$^{-1}$. 
Results (Li and Ahmadi test case)

Fig. 5.22. A plot of the critical velocity, $V_{cr}$ and critical approach velocity, $V_c$ vs. particle diameter, $d_p$ for bounce at a flow velocity of 2 m s$^{-1}$.

142 particles resuspended in this simulation at a flow velocity of 2 m s$^{-1}$. Fig. 5.23 shows the $U_{posp}$ and $V_{cr}$ for the particles, illustrating that $U_{posp}$ was greater than the corresponding $V_{cr}$. The $V_{cr}$ range was 0.0035 m s$^{-1}$ to 0.074 m s$^{-1}$ with an average of 0.0205 m s$^{-1}$. The range of velocities that the particles resuspend at was 0.0085 m s$^{-1}$ to 0.21 m s$^{-1}$ with an average $U_{posp}$ of 0.07 m s$^{-1}$. Fig. 5.24 shows the same trend in $V_{cr}$ and $V_c$ that was seen previously.
Results (Li and Ahmadi test case)

Fig. 5.23. A plot of $V_{elp}$ vs. $d_p$ for resuspension at a flow velocity of 2 m s$^{-1}$.

Fig. 5.24. A plot of the critical velocity, $V_{cr}$ and critical approach velocity, $V_c$ vs. particle diameter, $d_p$, for resuspension at a flow velocity of 2 m s$^{-1}$.
5.3.2 Comparison of simulated data and the data of Li and Ahmadi (3 m s\(^{-1}\) flow velocity).

In this simulation, there is a 7% increase (to 705) in the number of depositions with an increase of the flow velocity to 3 m s\(^{-1}\). Fig. 5.25 clearly shows all the particles deposited below the critical velocity for bounce to occur. The \(V_{\text{elp}}\) range for the particles was 0.00057 m s\(^{-1}\) to 0.1 m s\(^{-1}\) with an average 0.058 m s\(^{-1}\). Li and Ahmadi (1993)\(^{[4]}\) indicated a critical approach velocity of \(V_c\) which is compared with the calculated \(V_{cr}\) in Fig. 5.26. The range of \(V_c\) is 0.0028 m s\(^{-1}\) to 0.27 m s\(^{-1}\) while \(V_{cr}\) is 0.0033 m s\(^{-1}\) to 0.1 m s\(^{-1}\) with averages respectively of 0.021 m s\(^{-1}\) and 0.0013 m s\(^{-1}\). Again, above 5 µm there is very good agreement between \(V_c\) and \(V_{cr}\).

![Graph](image)

*Fig. 5.25. A plot of the deposition velocity, \(V_{\text{elp}}\) vs. particle diameter, \(d_p\) for deposition at a flow velocity of 3 m s\(^{-1}\).*
For the 3 m s\(^{-1}\) simulation, there was a 64% increase in the number of bounces to 239 bounces. The average velocity of the particles bouncing was 0.21 m s\(^{-1}\) with a range of 0.015 m s\(^{-1}\) to 0.56 m s\(^{-1}\). The \(V_{cr}\) range was 0.004 m s\(^{-1}\) to 0.074 m s\(^{-1}\) and an average \(V_{cr}\) of 0.01 m s\(^{-1}\). The \(V_c\) range is 0.0031 to 0.18 m s\(^{-1}\) with an average of 0.037 m s\(^{-1}\). Above 4 \(\mu\)m there is very good agreement between \(V_{cr}\) and \(V_c\) as seen in Fig. 5.28. Fig. 5.27 shows that all the \(V_{clp}\) values were above the \(V_{cr}\) values for the corresponding particle. As expected, and as Li and Ahmadi (1993)\[4]\ indicated, as particle size increases the probability of rebound increases.
Results (Li and Ahmadi test case)

Fig. 5.27. A plot of the deposition velocity, $V_{elp}$ vs. particle diameter, $d_p$ when bounce occurred at a flow velocity of 3 m s$^{-1}$.

Fig. 5.28. A plot of the critical velocity, $V_{cr}$ and critical approach velocity, $V_c$ vs. particle diameter, $d_p$ for bounce at a flow velocity of 3 m s$^{-1}$. 
Results (Li and Ahmadi test case)

There was a 60% increase (to 227) in the number of resuspensions in the 3 m s\(^{-1}\) simulation. Fig. 5.29 shows the \(U_{\text{posp}}\) and \(V_{\text{cr}}\) for the particles, illustrating that \(U_{\text{posp}}\) was greater than the corresponding \(V_{\text{cr}}\). The \(V_{\text{cr}}\) range was 0.0033 m s\(^{-1}\) to 0.1 m s\(^{-1}\) with an average of 0.02 m s\(^{-1}\). The \(V_{c}\) range was 0.003 m s\(^{-1}\) to 0.27 m s\(^{-1}\) with an average of 0.04 m s\(^{-1}\). The range of velocities that the particles resuspend at was 0.009 m s\(^{-1}\) to 0.57 m s\(^{-1}\) with an average \(U_{\text{posp}}\) of 0.11 m s\(^{-1}\). Fig. 5.30 shows the same trend in \(V_{\text{cr}}\) and \(V_{c}\) that has been seen previously.

![Fig. 5.29. A plot of \(U_{\text{posp}}\) vs. \(d_{p}\) when resuspension occurred at a flow velocity of 3 m s\(^{-1}\).](image)

95
Fig. 5.30. A plot of the critical velocity, $V_{cr}$ and critical approach velocity, $V_c$ vs. particle diameter, $d_p$ for resuspension at 3 m s$^{-1}$.

5.3.3 Comparison of simulated data and the data of Li and Ahmadi (5 m s$^{-1}$ flow velocity).

In this simulation the flow velocity was set to 5 m s$^{-1}$ and, as expected, there is again an increase in the number of depositions, bounces and resuspensions. There was a 20% increase in the number of particles deposited compared with the 2 m s$^{-1}$ simulation and 12% increase compared with the 3 m s$^{-1}$ simulation, to 786 occurrences. Fig. 5.31 shows the deposition velocity for particles that deposit and the corresponding $V_{cr}$ value.
Results (Li and Ahmadi test case)

Fig. 5.31. A plot of the deposition velocity, $V_{elp}$ vs. particle diameter, $d_p$ for deposition at a flow velocity of 5 m s$^{-1}$.

The $V_{elp}$ range for the particles is 0.00057 m s$^{-1}$ to 0.097 m s$^{-1}$ with an average of 0.012 m s$^{-1}$. The $V_{cr}$ range is 0.0033 m s$^{-1}$ to 0.1 m s$^{-1}$ with an average of 0.017 m s$^{-1}$. There is an increase in the maximum deposition velocity with the corresponding increase in the flow velocity. This increase is also seen in the average deposition velocity values.

The $V_c$ values calculated by Li and Ahmadi (1993) are compared with the calculated $V_{cr}$ values in Fig. 5.32. The range of $V_c$ is 0.0028 m s$^{-1}$ to 0.27 m s$^{-1}$ while $V_{cr}$ is 0.0033 m s$^{-1}$ to 0.1 m s$^{-1}$, with averages respectively of 0.017 m s$^{-1}$ and 0.03 m s$^{-1}$. Again, above 4 µm there is good agreement between $V_c$ and $V_{cr}$.  

97
Results (Li and Ahmadi test case)

Fig. 5.32. A plot of the critical velocity, $V_{cr}$ and critical approach velocity calculated by Li and Ahmadi, $V_c$ vs. particle diameter, $d_p$.

For the 5 m s$^{-1}$ simulation, there was a 287% increase in the number of bounces to 519 bounces from the 2 m s$^{-1}$ simulation and 135% increase in the number of bounces from the 3 m s$^{-1}$ simulation. The average velocity of the particles bouncing was 0.47 m s$^{-1}$ with a range of 0.008 m s$^{-1}$ to 1.27 m s$^{-1}$ as seen in Fig. 5.33. It is clear that with the increase in the flow velocity, there is a corresponding increase in the velocity of the particles, and therefore, as seen from the data, there is an increase in the number of bounces, the maximum $V_{elp}$ of bounce and the average $V_{elp}$. 

98
The $V_{cr}$ range was 0.003 m s$^{-1}$ to 0.1 m s$^{-1}$, with an average $V_{cr}$ value of 0.02 m s$^{-1}$. The $V_c$ range is 0.003 m s$^{-1}$ to 0.27 m s$^{-1}$ with an average of 0.04 m s$^{-1}$. From Fig. 5.34, it is clear that there is good agreement between the $V_{cr}$ and $V_c$ values above 4 µm, as seen previously.
Results (Li and Ahmadi test case)

Fig. 5.34. *A plot of the critical velocity, $V_{cr}$ and critical approach velocity, $V_c$ vs. particle diameter, $d_p$, for bounce at a flow velocity of 5 m s$^{-1}$.*

There was a 250% increase in the number of resuspensions (to 498) in the 5 m s$^{-1}$ simulation compared with the 2 m s$^{-1}$ simulation and 119% compared with the 3 m s$^{-1}$ simulation. 498 particles resuspended in this simulation. Fig. 5.35 shows the $U_{posp}$ and $V_{cr}$ for the particles, illustrating that $U_{posp}$ was greater than the corresponding $V_{cr}$. The $V_{cr}$ range was 0.0033 m s$^{-1}$ to 0.1 m s$^{-1}$ with an average of 0.021 m s$^{-1}$. The $V_c$ range was 0.003 m s$^{-1}$ to 0.27 m s$^{-1}$ with an average of 0.04 m s$^{-1}$. The range of velocities that the particles resuspended at was 0.008 m s$^{-1}$ to 0.93 m s$^{-1}$ with an average $U_{posp}$ of 0.09 m s$^{-1}$. 
Results (Li and Ahmadi test case)

Fig. 5.35. A plot of $U_{\text{posp}}$ vs. $d_p$ when resuspension occurred at a flow velocity of $5 \text{ m s}^{-1}$.

Fig. 5.36 shows the same trend in $V_{\text{cr}}$ and $V_c$ that was previously seen. There is good agreement seen above 4 µm between $V_{\text{cr}}$ and $V_c$.

Fig. 5.36. A plot of the critical velocity, $V_{\text{cr}}$ and critical approach velocity, $V_c$ vs. particle diameter, $d_p$ for resuspension at a flow velocity of $5 \text{ m s}^{-1}$.
5.3.4 5 m s\(^{-1}\) Li and Ahmadi test case simulation with 'perfect sinks'.

Fig. 5.37 shows the \(V_{\text{elp}}\) and \(V_{\text{cr}}\) of a 5 m s\(^{-1}\) simulation where the boundaries were assumed to be perfect sinks. From this simulation it is evident that, when bounce is not included, many particles deposit that would have bounced and this distorted the average velocity for particles depositing. During the simulation 868 particles deposited. Of these 314 had values below the corresponding \(V_{\text{cr}}\) of the particle while 555 were above. The average \(V_{\text{elp}}\) for particles that were above \(V_{\text{cr}}\) was 0.5 m s\(^{-1}\) with a minimum value of 0.003 m s\(^{-1}\) (corresponding to \(V_{\text{cr}}\) of 0.0025 m s\(^{-1}\)) and a maximum value of 1.3 m s\(^{-1}\). This clearly indicates that bounce is an important factor in determining the deposition velocity of a particle and should not be neglected. Fig. 5.38 shows the increase in \(V_{\text{elp}}\) when bounce is not included.

![Fig. 5.37. A plot of the deposition velocity, \(V_{\text{elp}}\) vs. particle diameter, \(d_p\) for perfect sink simulation.](image-url)
Results (Li and Ahmadi test case)

Fig. 5.38. A histogram of $V_{\text{elp}}$ for flow velocities of 2, 3 and 5 m s$^{-1}$ when bounce is included and at 5 m s$^{-1}$ with 'perfect sinks'.

5.3.5 5 m s$^{-1}$ test case simulation comparing resuspension and no resuspension.

Fig. 5.39 shows the $U_{\text{posp}}$ when particles were checked to determine whether they would resuspend or stay on the boundary. For resuspension to occur, the $U_{\text{posp}}$ had to be greater than $V_{\text{cr}}$. The range of $U_{\text{posp}}$ values was 0.008 m s$^{-1}$ to 0.07 m s$^{-1}$, with an average of 0.03 m s$^{-1}$ for the cases where $U_{\text{posp}}$ was less than $V_{\text{cr}}$ with a range of 0.009 m s$^{-1}$ to 0.27 m s$^{-1}$ and an average of 0.05 m s$^{-1}$. For the cases where $U_{\text{posp}}$ was greater than $V_{\text{cr}}$, $U_{\text{posp}}$ range was 0.008 m s$^{-1}$ to 0.93 m s$^{-1}$ with an average of 0.09 m s$^{-1}$. The range of $V_{\text{cr}}$ was 0.0033 m s$^{-1}$ to 0.1 m s$^{-1}$, with an average of 0.021 m s$^{-1}$. It can be clearly seen from the plot that at the lower part of the particles diameter range (below 4 µm), there is less resuspension due to $U_{\text{posp}}$ being less than $V_{\text{cr}}$ for the particles.
Fig. 5.39. Plot of $U_{posp}$ from resuspension checks, with an area plot of $V_{cr}$.

5.3.6 15 m s$^{-1}$ simulation where bounce dominates.

An additional simulation was carried out with a flow velocity of 15 m s$^{-1}$. The results of Paw and Braaten (1992)$^{[3]}$ implied that at very high wind speeds, rebound dominates over deposition and therefore resuspension. Though bounce dominates it is clear from this work, and as stated by Paw and Braaten (1992)$^{[3]}$, that deposition and resuspension are not negligible and still occur. It is clear that bounce is the dominant process since the number of depositions and resuspensions only make 0.9 % of the total incidents of
Results (Li and Ahmadi test case)

bounce, deposition and resuspension. The average velocity of bounce increases dramatically from 0.47 m s\(^{-1}\) for the 5 m s\(^{-1}\) simulation to 4.65 m s\(^{-1}\) in this simulation. This increase of velocity of bounce as wind speed increases is clearly seen in Fig. 5.40 for all the Li and Ahmadi test case simulations.

Fig. 5.40. \(V_{elp}\) of bounce for Li and Ahmadi test case.
5.3.7 Conclusions on the comparison of simulated particle behaviour with the simulation data of Li and Ahmadi.

<table>
<thead>
<tr>
<th>Simulation flow velocity</th>
<th>Incident</th>
<th>$V_{cr}$ or $V_{cr}$ range</th>
<th>$V_c$ range</th>
<th>$V_{elp}$ or $U_{posp}$ range</th>
<th>No. of incidents</th>
</tr>
</thead>
<tbody>
<tr>
<td>m s$^{-1}$</td>
<td>m s$^{-1}$</td>
<td>m s$^{-1}$</td>
<td>m s$^{-1}$</td>
<td>m s$^{-1}$</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>2 deposition</td>
<td>0.0033</td>
<td>0.0928</td>
<td>0.0028</td>
<td>0.2426</td>
<td>0.0006</td>
</tr>
<tr>
<td>2 bounce</td>
<td>0.0035</td>
<td>0.0741</td>
<td>0.0031</td>
<td>0.180</td>
<td>0.0085</td>
</tr>
<tr>
<td>2 resuspension</td>
<td>0.0033</td>
<td>0.0741</td>
<td>0.0031</td>
<td>0.180</td>
<td>0.0080</td>
</tr>
<tr>
<td>3 deposition</td>
<td>0.0033</td>
<td>0.1010</td>
<td>0.0028</td>
<td>0.2715</td>
<td>0.0006</td>
</tr>
<tr>
<td>3 bounce</td>
<td>0.0033</td>
<td>0.1007</td>
<td>0.0028</td>
<td>0.2705</td>
<td>0.0148</td>
</tr>
<tr>
<td>3 resuspension</td>
<td>0.0033</td>
<td>0.1007</td>
<td>0.0028</td>
<td>0.2705</td>
<td>0.0089</td>
</tr>
<tr>
<td>5 deposition</td>
<td>0.0033</td>
<td>0.1011</td>
<td>0.0028</td>
<td>0.2719</td>
<td>0.0005</td>
</tr>
<tr>
<td>5 bounce</td>
<td>0.0033</td>
<td>0.1010</td>
<td>0.0028</td>
<td>0.2715</td>
<td>0.0085</td>
</tr>
<tr>
<td>5 resuspension</td>
<td>0.0033</td>
<td>0.1010</td>
<td>0.0028</td>
<td>0.2715</td>
<td>0.0080</td>
</tr>
<tr>
<td>15 deposition</td>
<td>0.0033</td>
<td>0.1011</td>
<td>0.0028</td>
<td>0.2719</td>
<td>0.0003</td>
</tr>
<tr>
<td>15 bounce</td>
<td>0.0033</td>
<td>0.1011</td>
<td>0.0028</td>
<td>0.2719</td>
<td>0.0301</td>
</tr>
<tr>
<td>15 resuspension</td>
<td>0.0033</td>
<td>0.1011</td>
<td>0.0028</td>
<td>0.2719</td>
<td>0.0125</td>
</tr>
</tbody>
</table>

Table 5.2. Summary of bounce, deposition and resuspension for the Li and Ahmadi test case.

Table 5.2 shows a summary of the simulation results for Li and Ahmadi [4] test case. As expected there is an increase in the number of bounces, resuspensions and depositions as the stream-wise velocity was increased in the simulation. It is evident that there is good agreement between $V_c$ as proposed by Li and Ahmadi (1993) and the $V_{cr}$ value as proposed by this work above 4 µm. Below 4 µm, $V_c$ and $V_{cr}$ are evidently different, as the particle diameter becomes smaller. The reason for this difference is clearly indicated in the energy of adhesion (Table 5.3) which is calculated by inclusion of the Van der Waals forces only for $V_c$, while energy of adhesion for $V_{cr}$ is calculated by the inclusion of the Van der Waals force, mechanical potential energy and gravitational force.
Results (Li and Ahmadi test case)

<table>
<thead>
<tr>
<th>$d_p$ (µm)</th>
<th>$E_{ad}$ $V_{cr}$ (J)</th>
<th>$E_{ad}$ $V_c$ (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.02</td>
<td>5.4861842E-17</td>
<td>1.03E-28</td>
</tr>
<tr>
<td>2.05</td>
<td>5.4862748E-17</td>
<td>2.07E-28</td>
</tr>
<tr>
<td>3.00</td>
<td>5.4863117E-17</td>
<td>3.04E-28</td>
</tr>
<tr>
<td>4.00</td>
<td>5.4862462E-17</td>
<td>4.05E-28</td>
</tr>
<tr>
<td>5.00</td>
<td>5.4862641E-17</td>
<td>5.07E-28</td>
</tr>
<tr>
<td>6.00</td>
<td>5.4862664E-17</td>
<td>6.07E-28</td>
</tr>
<tr>
<td>7.00</td>
<td>5.4862928E-17</td>
<td>7.09E-28</td>
</tr>
<tr>
<td>8.00</td>
<td>5.4862858E-17</td>
<td>8.10E-28</td>
</tr>
<tr>
<td>9.01</td>
<td>5.4862865E-17</td>
<td>9.13E-28</td>
</tr>
</tbody>
</table>

Table 5.3. Energy of adhesion values used in the calculation of $V_{cr}$ and $V_c$.

When bounce is examined, as indicated by Li and Ahmadi (1993)[43], it is clearly evident that as particle size increases so does the probability of rebound and the shape of the curve in figures such as Fig. 5.36 clearly indicate this. 'Perfect sinks' are used in most simulations and experimental work in this area. From this work and research such as Paw and Braaten (1992)[3], it is clear that this is unrealistic and that bounce and resuspension should be included. When a 'perfect sink' was used in this simulation it is clear that this leads to an increase in deposition velocities of the particles impacting a surface and that for a realistic model, such assumptions should not be used.
5.4 Test case 3: Sand sedimentation.

![Simulation of sand particles in freefall.](image)

Sand was chosen as one of the test cases to demonstrate the versatility of the presented theory, i.e. that the theory not only holds true for small particle movement with bounce and resuspension included, but also that the model can account for much larger particles. The diameter range of sand modelled was 250 µm to 2000 µm and the density of the sand was set to 1681 kg m$^{-3}$. This choice of particle range was facilitated by a single value for the density of the sand particle being used in the code. In the simulation, 1000 particles were injected using the injector function as discussed in Chapter 4. The injection lasted 5 s from 0.1 s to 5.1 s and was centred on a position (0.05, 0.05, 0.05) in the domain with a random variance of 0.02 and the simulation was simulated with a time step of 0.005 s over 150 s. In the first simulation the sand was simulated in freefall with no fluid flow as seen in Fig 5.41. The second simulation model is of sand being blown across a surface by 5 m s$^{-1}$ wind flow in the x direction.
5.4.1 Test case 3: Sand sedimentation

Fig. 5.42. Settling velocity plotted against sediment size from Fentie et al. (2004).[5]

Fig. 5.42 is the settling velocity plotted against sediment size as published by Fentie et al. (2004).[5]. There is good agreement between the values seen in Fig. 5.43 area of interest and the calculated free fall velocity from the simulated sand as seen in Fig. 5.43. There is a more gradual rise seen in Fig. 5.43, than in Fig. 5.42, the presented results lie just below the values for smaller particles sizes in Fentie et al. (2004).[5] and there is better agreement seen as particle size increases.
Results (sand test case)

Fig. 5.43. Plot of $V_{\text{elp}}$ values for particles that deposit.

Fig. 5.44 shows that all the particles deposit, as $V_{\text{elp}}$ is less than the corresponding $V_{\text{cr}}$ for that particle. The range of $V_{\text{elp}}$ is 0.004 m s$^{-1}$ to 0.23 m s$^{-1}$ with an average of 0.09 m s$^{-1}$. These values are within the range that is expected for the freefall velocity of sand, as seen in Fig. 5.43.
Fig. 5.44. Plot of $V_{elp}$ of particles that deposited with corresponding $V_{cr}$.
5.4.2 Comparison of simulated data with sand transport values for a flow velocity of 5 m s$^{-1}$.

Fig. 5.45. Sand particles in a 5 m s$^{-1}$ velocity flow.

Fig. 5.45 depicts the domain and flow, into which 1000 sand particles (250 µm to 2000 µm diameter) were injected near the horizontal surface. When a flow with a velocity is introduced to the test case, increased values of deposition and bounce and resuspension are observed. There were 992 particle depositions when the test case was set to freefall. In the simulation with the flow velocity was set to 5 m s$^{-1}$, there were 2628 unique depositions. The $V_{elp}$ range was 0.015 m s$^{-1}$ to 1.73 m s$^{-1}$ and there was a 533 % increase in the average $V_{elp}$ to 0.57 m s$^{-1}$ compared with the freefall case. Fig. 5.46 shows the $V_{elp}$ of the particles that deposited and the corresponding $V_{cr}$. As expected the probability of bounce increases as particle size increases as is evident from the area plot of $V_{cr}$ in Fig. 5.47.
Results (sand test case)

Fig. 5.46. A plot of the critical velocity, $V_{cr}$ and deposition velocity, $V_{elp}$ vs. particle diameter, $d_p$.

Fig. 5.47 shows the $V_{elp}$ and $V_{cr}$ for occasions where bounce occurred. Bounce occurred at the higher end of the particle diameter range due to the $V_{cr}$ calculations and that the probability of bounce increases with particle size. The $V_{cr}$ range was 0.394 m s$^{-1}$ to 1 m s$^{-1}$. The $V_{elp}$ range was 0.395 m s$^{-1}$ to 1.6 m s$^{-1}$. There was 22 unique bounce datasets during the simulation.
Fig. 5.47. A plot of the deposition velocity, $V_{elp}$ vs. particle diameter, $d_p$ when bounce occurred.

During the simulation there was 610 unique resuspensions. The $U_{posp}$ was 5 m s$^{-1}$, and $V_{cr}$ values ranged from 0.395 m s$^{-1}$ to 1.1 m s$^{-1}$. Fig. 5.48 shows $U_{posp}$ and $V_{cr}$ for resuspension incidents. In all the presented cases, $U_{posp}$ was greater than the corresponding $V_{cr}$. 

114
From Kang and Guo (2006)\textsuperscript{[6]} plots of sand particles spatial distribution (in Fig. 5.49 and 5.50) it can be clearly seen that the sand particles concentrate in a region below 0.04 m. This compares favourably with the 'y' position data as seen in Fig. 5.51 and 5.52 and the conclusion of Kang and Guo (2006)\textsuperscript{[6]} that the sand mass flux decays exponentially with height for $h > 0.02$ m. When particles are injected they drop to this region and travel in it by depositing, bouncing and resuspending.

\textit{Fig. 5.48. A plot of the $V_{crr}$ and $U_{posp}$ vs. $d_p$ for resuspension.}
Results (sand test case)

Fig. 5.49. Spatial distribution of sand particles at 3.96 s from Kang and Guo (2006)\cite{6}.

Fig. 5.50. Spatial distribution of sand particles at 4.05 s from Kang and Guo (2006)\cite{6}.
Results (sand test case)

Fig. 5.51. The 'y' position data of the Sand particle at 3.96 s.

Fig. 5.52. The 'y' position data of the Sand particle at 4.05 s.
5.4.3 Conclusions

It is clear from the presented work above that sand transport can be modelled with the theory presented in this work and that the inclusion of bounce and resuspension is of a high priority when modelling the transport of sand in a flow. Again it is evident that as the particle size increases the probability of bounce and resuspension occurring increases. When a comparison is made with particle settling velocity as modelled in the first simulation and those published in recent work, that there is good agreement between the results. The comparison of the presented work with that of Kang and Guo (2006)[6] shows good agreement. Table 5.4 is a summary of the $V_{cr}/V_{crr}$ and $V_{elp}/U_{posp}$ for the sand simulations.

<table>
<thead>
<tr>
<th>Simulation flow velocity</th>
<th>Incident</th>
<th>$V_{cr}$ or $V_{crr}$</th>
<th>$V_{elp}$ or $U_{posp}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>m s$^{-1}$</td>
<td>m s$^{-1}$</td>
<td>m s$^{-1}$</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>freefall deposition</td>
<td>0.3951</td>
<td>8.564</td>
<td>0.0039</td>
</tr>
<tr>
<td>5  deposition</td>
<td>0.3955</td>
<td>8.564</td>
<td>0.0149</td>
</tr>
<tr>
<td>5  bounce</td>
<td>0.3955</td>
<td>1.03</td>
<td>0.3955</td>
</tr>
<tr>
<td>5  resuspension</td>
<td>0.3955</td>
<td>4.91</td>
<td>5</td>
</tr>
</tbody>
</table>

*Table 5.4. A summary of sand test case simulations.*
6. Conclusions

Although there has been an extensive research focus on simulating particle transport using a Lagrangian formulation, there has been no model to account for bounce, deposition and resuspension in a simulation of Lagrangian particle transport. To rectify this, the present work involved the development of a theory to account for bounce and resuspension and the inclusion of this algorithm into an existing CFD code. In this work, when a particle collides with a surface, the only mechanism to adhere the particle to the surface is the adhesion energy of the particle. The energy of adhesion is the sum of the mechanical potential energy and the surface adhesion energy. As indicated by previous experimental and simulation work, the main surface adhesion force is due to the Van der Waals force. The surface adhesion energy used in this work is related to the Van der Waals force by the contact geometry and the mechanical constant. The mechanical potential energy is due to the particle impacting the surface. With the energy of adhesion of the particle to the surface calculated, a critical velocity for bounce could be calculated. A bounce check was undertaken and if the deposition velocity of the particle was greater than the critical velocity for bounce of the particle, the particle bounced. If it was less, the particle deposited. To include resuspension, a critical velocity of resuspension needed to be calculated. The theory for the calculation of resuspension followed much the same procedure as the calculation of the critical velocity for bounce but since the particle is on the surface, there was no mechanical energy in the energy of adhesion sum. After a particle had deposited to a surface in a previous time step, a resuspension check was run and if the flow velocity at the position of the particle was greater than the critical...
Conclusions

velocity for resuspension, the particle resuspended. This is a unique development and leads to a cost effective, highly robust code, that incorporated bounce and resuspension into a Lagrangian simulation of particle transport, whose outputted data agreed well with other published simulated and experimental work.

To examine the validity and versatility of the presented code, in terms of particle size and flow conditions that can be considered, three test cases were developed and investigated. These were

1. Deposition of particles with particle diameter from 33 to 36 µm onto leaf surfaces at flow velocities of 2.82, 3.41, 4.59 and 15 m s\(^{-1}\).
2. Deposition of particles with particle diameter from 1 to 10 µm in a channel at flow velocities of 2, 3, 5 and 15 m s\(^{-1}\).
3. Sand sedimentation and sand blowing across a surface due to a 5 m s\(^{-1}\) flow.

Based on the comparison of the simulated data with the test case data, a number of conclusions were reached, and these are summarised in the sections that follow.

6.1 Deposition

1. The simulation accurately models particle deposition to surfaces. The particle deposition velocity increases with an increase in the flow velocity.
2. With a 'perfect sink' condition used on the surfaces, an increase in the deposition velocity of the particles is seen. This indicates that bounce must be included to accurately model particle transport.
3. With an increase in the flow velocity, there is an increase in the number of depositions. This is due to the fact that, as the flow velocity increases, particles resuspend and therefore there are more particles to deposit.
4. The comparison of the calculated \(V_{cr}\) and with \(V_c\) from Li and Ahmadi (1993)\(^{[4]}\) shows good agreement above 4 µm. The difference seen (for example, in Fig. 5.24) below 4 µm is due to the fact that the \(V_c\) value only includes the Van der Waals force in its calculation, while \(V_{cr}\) is calculated using a surface energy accounting for the Van der Waals force, gravitational effect and the mechanical potential energy from the impact of the particle and the surface.
5. When $V_{cr}$ is examined in relation to the work of Paw and Braaten (1992)\[3\], it is evident that $V_{cr}$ sits well within the upper limit of $V_i$ and the lower limit of $V_d$.

6. When sand sedimentation is examined in test case 3, it is clearly evident that there is good agreement between the calculated freefall deposition velocities and the settling velocity data as published by Fentie et al. (2004)\[5\]

7. As expected from previous simulations and experimental work such as McLaughlin (1989)\[11\] and Brooke et al. (1992 & 1994)\[13,14\], there is clearly a build-up of particles in the near-wall region in the presented work. This build-up is evident in Fig. 6.1.

Fig. 6.1. *A screenshot from the Paw and Braaten test case showing near-wall concentration of particles.*
6.2 Bounce.

8. As mentioned above in point 2, bounce must be included in a simulation of particle transport, as the data for deposition velocity becomes skewed if all the particles deposit on contact with a surface, regardless of their velocity value.

9. The possibility of bounce increases as particle size increases, as is evident from the shape of the $V_{cr}$ curves and area plots seen in chapter 5.

10. Bounce becomes a dominant process and short cuts deposition and therefore resuspension at very high flow velocities.

11. Again, when the calculated $V_{cr}$ value is compared with the $V_c$ value as proposed by Li and Ahmadi (1993)\cite{4}, good agreement is seen above 4 $\mu$m.

12. The simulation of bounce agrees well with previous theoretical and experimental work such as Li and Ahmadi (1993)\cite{4} and Paw and Braaten (1992)\cite{3}.

6.3 Resuspension

13. Resuspension is an important factor in particle transport. As expected resuspension rates increase with an increase in the flow velocity. This is evident until bounce short cuts resuspension at very high flow velocities.

14. The probability of resuspension increases with an increase of particle size, as is evident from the shape of the $V_{cr}$ curves and area plots seen in chapter 5.

15. As noted by Ziskind et al. (1995)\cite{32} resuspension from a surface is not instantaneous but takes place over a period of time. Fig. 6.2 is a sequence of screenshots from the model developed in this thesis which illustrates this phenomenon.

1) The particle (with yellow star) is in the flow.

2) The particle bounces off a surface.

3) The particle has deposited on the surface.

4) It remains on the surface.

5) It is dragged a small distance along the surface.

6) It is again dragged along the surface.

7) It begins to resuspend into the fluid flow

8) It is fully captured by the flow and swept back into the fluid flow.
Fig. 6.2. A sequence of screenshots showing a particle bouncing, depositing and resuspending.
Conclusions

6.4 Future Work

In the work presented, a theory has been developed to incorporate bounce and resuspension into a Lagrangian simulation. There are many important scenarios, where it is not appropriate to model particle transport without including elements to incorporate the bounce and the resuspension of particles. An example of one of these important scenarios is the modelling of the transport of spores such as anthrax. The presence of spores in an environment can lead to major health effects such as infectious diseases and allergies. Paw and Braaten's (1992)\cite{3} experimental work clearly showed that bounce and resuspension were important particle transport mechanisms for spores, and therefore any model of the transport of spores would be incomplete without the inclusion of bounce and resuspension. Another important scenario, where the inclusion of particle resuspension and bounce is a necessity, is the modelling of indoor particle transport. Many studies such as Karlsson et al. (1996)\cite{49} have shown the important effect that resuspension has on the concentration of particles in residential dust. To fully model particle transport in an indoor environment, resuspension must be included.

To fully model bounce and resuspension, extra elements need to be incorporated into the code, such as angle of impact and surface roughness. To extend the theory to include the angle of impact, a method such as that presented by Konstandopoulos (2006)\cite{50} could be used. In the Konstandopoulos (2006)\cite{50} method, a critical angle of impact for bounce is calculated. Even if a particle is below the critical velocity of bounce, if the angle of impact is above the critical angle of impact, the particle will bounce. By the inclusion of critical angle, the ability to model surface roughness, such as detailed by Li and Ahmadi (1993)\cite{8}, is realised. The normal vector perpendicular to the roughness can be calculated, the effect of surface roughness on the bounce and resuspension of a particle could be modelled. Again, this would be a positive step as Li and Ahmadi (1993)\cite{8} noted that an increase in wall roughness leads to an increase in the deposition velocity. By the modification of the presented theory, particle - particle interactions could be modelled. Formulae for contact geometry, contact radius, critical velocity of impact, gravitational force and surface forces would need to be modified to incorporate this. The
hard sphere model would need to be extended to include glancing particles, bouncing and coagulating particles as indicated by the work of Crowe et al. (1997)\cite{43} and Hoffmann and Kosinski (2010)\cite{51}. With particle - particle interactions included and the fact that the particles, when deposited, remain in the calculations, resuspension and deposition from particle collisions with particles on the surface could be modelled.
7. Bibliography

1. OpenFoam, www.openfoam.com


47. Paraview, [www.paraview.org](http://www.paraview.org)


particle collision model to study agglomeration.” Chemical Engineering Science,
65:3231-3239 (2010).