

Evaluation of Particle Clustering Algorithms in the Prediction of Brownout Dust Clouds

Bharath Govindarajan

The
Institute for
Systems
Research



A. JAMES CLARK
SCHOOL OF ENGINEERING

ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.

ISR is a permanent institute of the University of Maryland, within the A. James Clark School of Engineering. It is a graduated National Science Foundation Engineering Research Center.

www.isr.umd.edu

ABSTRACT

Title of thesis: **EVALUATION OF PARTICLE CLUSTERING
ALGORITHMS IN THE PREDICTION OF
BROWNOUT DUST CLOUDS**

Bharath Govindarajan, Master of Science, 2011

Thesis directed by: **Professor J. Gordon Leishman
Department of Aerospace Engineering**

A study of three Lagrangian particle clustering methods has been conducted with application to the problem of predicting brownout dust clouds that develop when rotorcraft land over surfaces covered with loose sediment. A significant impediment in performing such particle modeling simulations is the extremely large number of particles needed to obtain dust clouds of acceptable fidelity. Computing the motion of each and every individual sediment particle in a dust cloud (which can reach into tens of billions per cubic meter) is computationally prohibitive. The reported work involved the development of computationally efficient clustering algorithms that can be applied to the simulation of dilute gas-particle suspensions at low Reynolds numbers of the relative particle motion. The Gaussian distribution, k-means and Osipov's clustering methods were studied in detail to highlight the nuances of each method for a prototypical flow field that mimics the highly unsteady, two-phase vortical particle flow obtained when rotorcraft encounter brownout conditions. It is shown that although clustering algorithms can be problem dependent and have bounds of applicability, they offer the potential to significantly re-

duce computational costs while retaining the overall accuracy of a brownout dust cloud solution.

EVALUATION OF PARTICLE CLUSTERING ALGORITHMS IN THE PREDICTION OF BROWNOUT DUST CLOUDS

by

Bharath Madapusi Govindarajan

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2011

Advisory Committee:
Professor J. Gordon Leishman, Chair/Advisor
Professor Roberto Celi
Associate Professor James Baeder

Acknowledgments

A Masters degree is conferred upon one in an instant but takes a good few years to work towards it. In this regard I would like to thank my advisor, Professor J. Gordon Leishman, for granting me the opportunity to work under his guidance and for making it a fun experience. It has been an absolute joy working for Dr. Leishman and I look forward to working under his tutelage towards my doctoral degree.

I would also like to thank my committee members, Dr. Roberto Celi and Dr. James Baeder for their valuable insights towards this project. I would also like to thank Dr. Nail Gumerov from the computer science department for making me look at this problem from a different light.

I would specially like to thank Monica Syal for guiding me both academically and for making my transition into the United States a comfortable one. The lab would just not have been the same if it were not for the “Team Leishman!” camaraderie and spirit. I want to thank Ajay Baharani, Anish Sydney, Joe Milluzzo, Ben Hance, Joe Ramsey, Jaime Reel, Juergen Raulerder, John Tritscheler, David Mayo and Mark Glucksman-Glaser for making everyday a memorable one. Warm thanks to everyone in the “big-office” or the “cube-farm” for all the wonderful times.

None of this would ever have seen the light of day had it not been for the unrelenting support of my loving parents and sister. Words cannot express the deepest gratitude and love I feel for them, hence I will leave it at a simple ‘Thank you for always being there’.

Table of Contents

List of Tables	vi
List of Figures	vi
Nomenclature	xii
1 Introduction	1
1.1 The Problem of Brownout	1
1.2 The Physics of Brownout	5
1.2.1 Flow Physics	6
1.2.2 Dust Particle Dynamics	8
1.3 Computational Modeling	13
1.3.1 Flow Field Modeling	14
1.3.2 Particle Physics Modeling	17
1.3.3 Eulerian Particle Models	18
1.3.4 Lagrangian Particle Tracking Models	20
1.3.5 Reducing Computational Costs	22
1.3.6 Particle Clustering Techniques	23
1.4 Objectives of the Present Work	25
1.5 Organization of Thesis	26
2 Methodology	28
2.1 Introduction	28

2.2	Gaussian Method	29
2.3	k-means Method	31
2.4	Osipsov's Method	34
2.5	Error Estimations	38
2.6	Prototypical Flow	41
2.6.1	Particle Convection	42
2.6.2	Particle Bombardment on Bed	45
2.7	Summary	46
3	Results and Discussion	47
3.1	Introduction	47
3.2	Gaussian Method	50
3.2.1	Effect of Number of Particles per Cluster	52
3.2.2	Error Estimation in Two-Dimensions	58
3.2.3	Gaussian Method: Application to Three-Dimensional Problem . .	61
3.3	k-means Method	71
3.3.1	Introduction	71
3.3.2	Declustering and Reclustering	71
3.3.3	Computational Time Gain and Accuracy	75
3.3.4	Effect of Velocity Gradients	77
3.4	Osipsov's Method	89
3.4.1	Introduction	89
3.4.2	Results for a Potential Flow	90

3.4.3	Results for the Prototypical Flow	90
4	Conclusions	96
4.1	Suggestions for Future Work	98
4.1.1	Improvements in Estimating Accuracy	99
4.1.2	Improvements in the Gaussian Method	99
4.1.3	Improvements in the k-means Method	100
4.1.4	Improvements in Osipsov's Method	101
A	Derivation of Radial Basis Function	103
	Bibliography	105

List of Tables

3.1	Ratio of clustered time to exact time for the current problem as a fraction of total number of clusters to the number of particles.	80
-----	---	----

List of Figures

1.1	A helicopter encountering brownout conditions during a landing maneuver. (Courtesy of Optical Air Data Systems LLC.)	2
1.2	Schematic showing the different modes of sediment particle motion and the resulting dust field generated by the rotor wake from a helicopter hovering over a surface covered with loose material.	4
1.3	Flow visualization image of a 2-bladed rotor operating near a ground plane [1].	7
1.4	Schematic of the flow field below a rotor when it is operating near a ground plane.	7
1.5	Concentration of particles of different diameters in a dust cloud [2,3] . . .	9
1.6	Mass density of particles of different diameters in a dust cloud [2,3] . . .	10
1.7	Forces acting on a particle while in suspension and on a mobile sediment bed	11
1.8	Schematic showing the different modes of sediment particle motion and the fundamental uplift mechanisms seen in the near-wall region.	13
1.9	Schematic showing the modeling of the Free-Vortex Method	15

1.10	VTM/PTM simulation of the dust cloud that is formed in the air surrounding a generic tandem helicopter during its approach to a desert landing [4].	19
1.11	Contours of instantaneous streamwise particle velocities. Top: Initial seeding both cases; Middle: Coherent vortex; Bottom: Turbulent boundary layer. [5].	20
1.12	Dust cloud along with the associated velocity contours using a Lagrangian particle tracking methodology [6].	21
2.1	Schematic showing the Gaussian method of particle clustering.	30
2.2	Schematic showing the k-means method of clustering and declustering.	32
2.3	Schematic showing the change in shape of the particle fluid element. The dotted lines represent particle pathlines, which would normally be curved [7].	36
2.4	Spherical coordinate system centered around the pilot. Grid size of each “bin” increases when moving away from an observer.	40
2.5	Schematic showing the passage of vortices over an initial saltation layer, and uplifting the dust [1].	42
3.1	Exact solution. Initial layer = 15, particles per layer = 75.	49
3.2	Exact solution. Initial layers = 15, particles per layer = 375.	51
3.3	Gaussian solution. Initial layers = 15, clusters per layer = 75, particles per cluster = 5, $k_1 = 1$ and the corresponding density contour plots.	53
3.4	Gaussian solution. Initial layers = 15, clusters per layer = 75, particles per cluster = 5, $k_1 = 2$ and the corresponding density contour plots.	54

3.5	Gaussian solution. Initial layers = 15, clusters per layer = 15, particles per cluster = 25, $k_1 = 1$ and the corresponding density contour plots. . . .	55
3.6	Gaussian solution. Initial layers = 15, clusters per layer = 15, particles per cluster = 25, $k_1 = 2$ and the corresponding density contour plots. . . .	56
3.7	Error estimation using Pearson's coefficient for Gaussian solution for both $k_1 = 1$ and $k_1 = 2$	60
3.8	A solution obtained from brownout dust field computations when using the Gaussian clustering distribution method.	64
3.9	Another example of a brownout dust field computation using the Gaussian clustering distribution method.	65
3.10	Actual and clustered solution during a landing maneuver using the Gaussian clustering distribution method.	66
3.11	Development of the dust cloud during the landing approach at time $t = 3$ and 6 seconds.	67
3.12	Comparison of the dust cloud between a brute-force calculation and Gaussian clustered solution. Lateral view.	68
3.13	Comparison of the dust cloud between a direct or brute-force calculation and Gaussian clustered solution. Isometric view.	69
3.14	Comparison of the dust cloud between a direct or brute-force calculation and Gaussian clustered solution. Isometric view.	70
3.15	Schematic showing the process of reclustering and formation of new clusters.	72

3.16	Difference in the solution between a first-order (blue) and second-order (red) accurate scheme.	74
3.17	Schematic showing the simplified vortical flow field chosen to better illustrate the k-means clustering method.	75
3.18	Comparison of CPU time for clustered and actual simulations for different number of particles using the k-means method.	76
3.19	Comparison of RMS error for clustered and actual simulations for different number of particles using the k-means method.	77
3.20	Comparison of solution for actual and clustered solution using k-means method. The different symbols in (b) represent different clusters and the solid black square represents the position of the vortex.	79
3.21	Comparison of particle displacement for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 10. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.	81
3.22	Comparison of density plots for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 10. Number of clusters = 10. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.	82
3.23	Comparison of particle displacement for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.	83

3.24	Comparison of density plots for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.	84
3.25	Comparison of particle displacement for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.	85
3.26	Comparison of density plots for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.	86
3.27	Comparison of particle displacement for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.	87
3.28	Comparison of density plots for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.	88
3.29	Density contour plot using Osipsov's method around a cylinder in potential flow. (a) Flow velocity of 1ms^{-1} ; (b) Flow velocity of 2ms^{-1}	91

- 3.30 Comparison of Osipov's method with the exact solution. Number of particles = 15 layers, 75 particles per layer. (a) Density field obtained using Osipov's method; (b) Density field obtained using direct counting. 93
- 3.31 Comparison of Osipov's method with the exact solution. Number of particles = 15 layers, 375 particles per layer. (a) Density field obtained using Osipov's method; (b) Density field obtained using direct counting. 94

Nomenclature

C_d	Drag coefficient
d_p	Particle diameter
e	Coefficient of restitution
E_c	Kinetic energy of a cluster
\mathbf{F}	Total force acting on a particle
\mathbf{F}_d	Drag force
g	Acceleration under gravity
J	Jacobian of the particle fluid element
m	Particle mass
N	Number of particles
N_c	Number of clusters
S_{12}	Inter-cluster separation
S_1, S_2	Intra-cluster separations
Re_p	Particle Reynolds number
\mathbf{U}	Carrier flow velocity
U_x, U_y, U_z	Components of carrier fluid velocity
\mathbf{V}_p	Particle velocity
V_{px}, V_{py}, V_{pz}	Components of particle velocity
\mathbf{V}_0	Velocity of impacting particle

Greek Symbols

μ_B	Mean vector for Gaussian distribution
μ_c	Mean vector for clustering
ν	Kinematic viscosity
ρ	Carrier fluid density
ρ_p	Mass of particles in mixture
ρ_p°	Density of the particles
Σ	Covariance matrix for Gaussian distribution
Σ_c	Covariance matrix for clustering
k_1	Principal spread coefficient
k_2	Cross coupling coefficient
τ_p	Particle velocity relaxation time
ω	Derivative of the Jacobian

Chapter 1

Introduction

1.1 The Problem of Brownout

The occurrence of the phenomenon of “brownout” is an ongoing operational problem for most types of rotorcraft when they are operating in dry and arid climates. Brownout occurs when rotorcraft land or take off over surfaces covered with loose material such as sand or dust, where through a series of complex fluid dynamic uplift and mobilization mechanisms, the action of the rotor wake rapidly stirs up a blinding dust cloud. The suspended dust can continue build up in volume and intensity under the continuing influence of rotor wake, and may quickly engulf the helicopter making it difficult for the pilot to see out of the cockpit. An example the problem is shown in Fig. 1.1, which shows the intense dust cloud produced when a helicopter is landing in the desert. In snowy conditions, the uplift of loose snow can produced “whiteout” conditions, which can be an equally concerning problem.

During brownout conditions, volume of dust produced and the resulting density of the dust cloud means that the pilot can lose outward visibility (e.g., to the horizon, peripherally, as well as downward) at a time when it is needed most. The pilot may also experience spurious sensory cues and spatial disorientation from the moving dust cloud, which if severe enough may lead to vertigo. The resulting situation (loss of visibility close to the ground) can then become a serious safety of flight issue, and the pilot may drift into



Figure 1.1: A helicopter encountering brownout conditions during a landing maneuver.
(Courtesy of Optical Air Data Systems LLC.)

obstacles or lose control of the aircraft. In fact, encounters with brownout conditions is currently the leading cause of human factor-related mishaps during military rotorcraft operations [8], and civilian helicopters (e.g., those on MEDEVAC missions) have also suffered from the problem [9]. Besides visibility issues, the suspended dust particles uplifted during brownout conditions can also create a harsh working environment for ground personnel, abrade the rotor blades and cause rapid engine wear, creating serious and costly maintenance issues. In some cases, rotor blades can be abraded to the core in tens of flying hours, and engines may need to be overhauled in just a few hundred hours of operation.

A myriad of technical efforts have now begun to help to reduce the piloting risks posed by brownout occurrences, mainly by focusing on the development of sensors and advanced cockpit displays to “see” through the dust cloud [10]. Another method em-

ployed by pilots to limit the severity of the dust cloud during take off and/or landing, is through careful flight-path management. Although such tactics have helped to reduce the rapidity and/or the severity of the dust cloud development, the resulting flight-paths often involve higher flight speeds closer to the ground to outpace the development of the dust cloud. Such tactics often involve “hard” landings and an attempt to get the aircraft on the ground before it becomes enveloped in the dust, which over time can be injurious to the pilots and crew. While the role of sensor/display technologies and effective piloting strategies cannot be underestimated, there still remains a need to understand the brownout phenomenon at a more fundamental scientific level, as well as to explain how brownout might be affected by the particular design parameters and operational characteristics of a given rotorcraft. A longer term goal must be to try to mitigate brownout more through vehicle design (i.e., treating the problem at its source), which obviously requires a high level of predictive capability for the flow field below the helicopter and for the development of the brownout cloud.

Photographic and videographic observations of actual brownout occurrences for landing rotorcraft suggest that there may be certain (and perhaps even unique) rotor and/or other vehicle design features that can influence the severity, extent, and rapidity of the developing dust clouds. For example, work conducted under the auspices of the DARPA Sandblaster program [2,3] shows the intensity and spatial extent of a developing brownout dust cloud can vary substantially between different types of rotorcraft. For example, some helicopters are known to produce radially expanding toroidal-type dust clouds during landing maneuvers with little in the way of vertical entrainment, leaving zones of good visibility for the pilot within the dust cloud. For other helicopters, the dust seems to

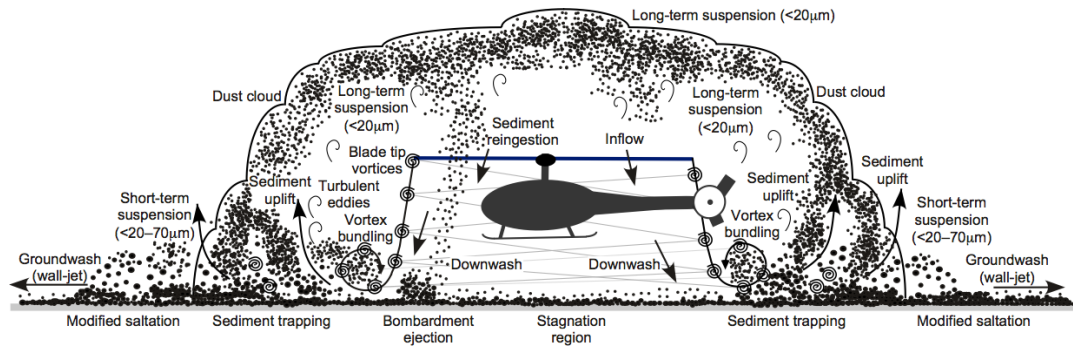


Figure 1.2: Schematic showing the different modes of sediment particle motion and the resulting dust field generated by the rotor wake from a helicopter hovering over a surface covered with loose material.

quickly form into a large, dome-shaped cloud that can engulf the entire vehicle, often with significant recirculation of dust back through the rotor disk. The tendency to form recirculating flows can be accompanied by vertical bombardment of the dust particles back onto the ground, which can stir up more dust through a cascading mechanism. In such cases, the consequences are greatly reduced visibility for the pilot and onset of other visual anomalies. Milluzzo et al. [11] performed an analysis that showed that the intensity of the dust cloud and the rate of its development can be correlated to parameters such as disk loading and blade loading. Furthermore, it was shown that the rotational speed of the rotor and the number of blades combined to produce a characteristic frequency at which the rotor wake vorticity impinges onto the ground. This frequency appeared to correlate with the rapidity of the development of the dust clouds.

1.2 The Physics of Brownout

A schematic of the general problem of brownout is shown in Fig. 1.2, which identifies some of the fluid mechanics and sedimentology processes involved in brownout. Fundamentally, brownout is an unsteady, two-phase, fluid dynamics problem of considerable three-dimensional complexity. Owing to the complex two-phase nature of the problem, i.e., the fluid (or carrier) phase and the particle (or dispersed) phase, it is obviously imperative to understand the mechanisms and factors that drive each phase to gain an overall understanding of the brownout problem, how it occurs from a fundamental physics perspective, and what could be done (practically) to mitigate its effects.

Clearly, different brownout conditions can also be expected based on the characteristics of the surface sediment, such as the type and size of the particles, their compactness and moisture content, etc. Such an outcome is not unexpected, given that the ultimate reason for the entrainment of loose sediment is from the shear and pressure forces produced by the action of the three-dimensional, unsteady, turbulent flow in the rotor wake as it develops over the ground. Understanding the fluid dynamics of these effects may ultimately lead to means of brownout amelioration at its source, thereby improving visibility through the cloud, as well as reducing abrasion and mechanical wear problems. To this end, the flow and particle physics of brownout must be studied both experimentally and computationally to better understand the phenomenon. In the present work, a more efficient computational approach toward the simulation of brownout dust clouds has been studied using methods of particle clustering.

1.2.1 Flow Physics

The flow around a rotorcraft changes significantly when the vehicle operates in close proximity to the ground. Hence, a prerequisite for understanding brownout is to understand the detailed fluid dynamics of the vortical rotor wake(s) generated by the rotor(s) operating in proximity to an impermeable surface. The problem of “ground effect” aerodynamics in hovering or low speed forward flight has been studied extensively [12–18], but mainly from the perspective of rotor performance. Because an impermeable surface must always be a streamline to the flow, the changes to the flow field by the ground are profound and can drastically affect the blade loads and vehicle performance. Until recently, however, and since brownout has become an operationally acute problem, there have been few reasons to study the flow below the rotor near the ground. To this end, there have been some detailed fluid dynamic studies of the flow at the ground below a rotor [1, 19, 20]. It has been shown in this work that action of the rotor wake on the ground is very complicated because it creates unsteady shear stresses, secondary vortical flows and turbulence, and such effects are ultimately responsible for the uplift of sediment and the eventual development of brownout conditions [1].

In general, the flow structure below the rotor changes considerably for operations in-ground-effect (IGE) versus out-of-ground-effect (OGE). The presence of the ground plane for IGE conditions, requires the streamlines below the rotor to rapidly change their direction and become parallel at the surface, the essence of the problem being shown in flow visualization image in Fig. 1.3. The rotor wake initially contracts below the rotor in a predominantly downward direction, turns ninety degrees to an outward direction, and

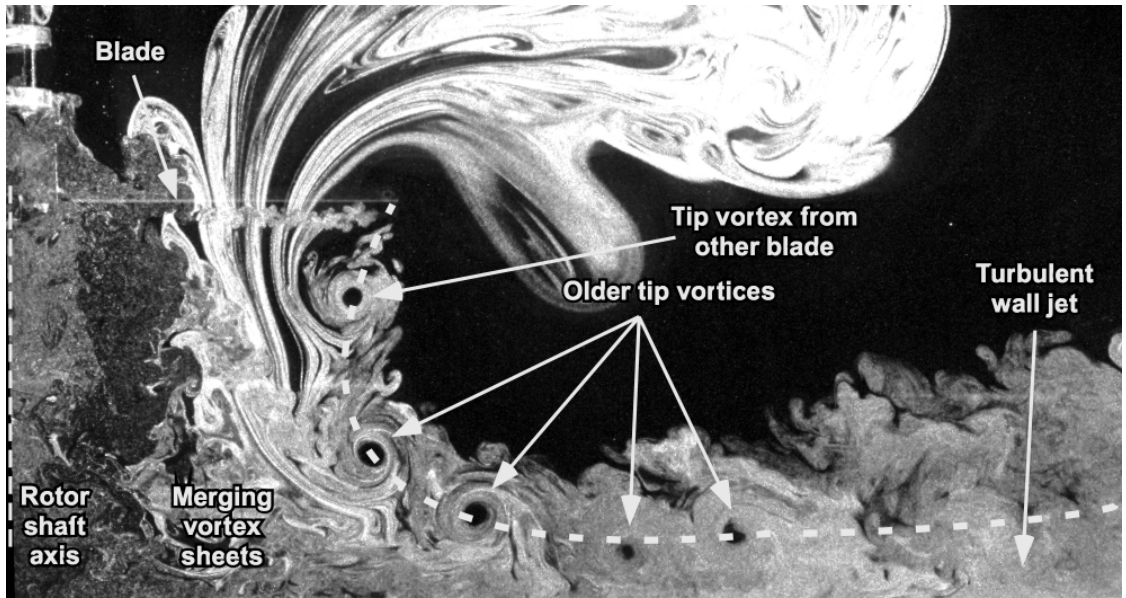


Figure 1.3: Flow visualization image of a 2-bladed rotor operating near a ground plane [1].

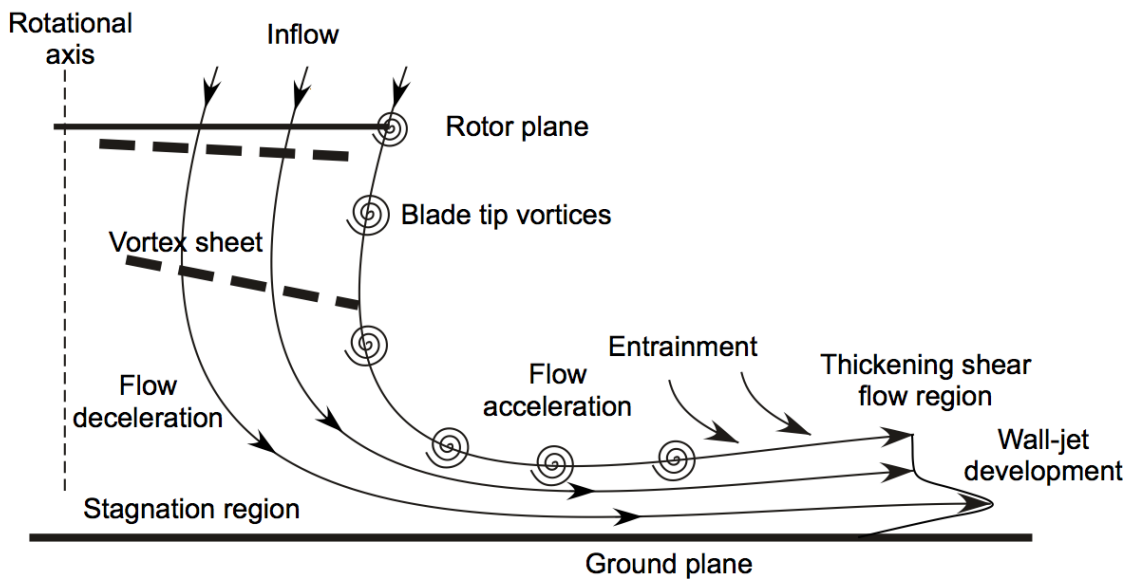


Figure 1.4: Schematic of the flow field below a rotor when it is operating near a ground plane.

then expands outward over the ground plane. The physics of the rotor in ground effect problem are explained in detail by Lee et al. [19], and can be summarized by the schematic shown in Fig. 1.4. Notice that tip vortex filaments, which are of helicoidal form, convect along the slipstream boundary that separates the relatively high velocities inside the rotor wake boundary from the quiescent flow outside this boundary. The vortices in Fig. 1.3 can be identified from the dark voids, which are caused by small Coriolis and centrifugal forces that act on the seed particles that comprise the visualization media. These vortices have primarily laminar cores, devoid of the small eddies and turbulence that are present in the surrounding flow. As they stretch in the flow under the action of the ground they can persist to relatively old wake ages, perhaps up to six to eight rotor revolutions [19, 20]. The flow at the ground also has significant turbulence and aperiodicity, as was observed by both Lee et al. [19] and Sydney et al. [1]. It is clear, therefore, that understanding the details rotor wake structure, and the changes that take place near the ground, is a prerequisite for understanding the problem of brownout.

1.2.2 Dust Particle Dynamics

This dynamic, multi-parameter nature of brownout dust clouds highlights the need for a good predictive capability of the problem, which can be achieved through numerical simulations if they can be suitably validated, both from a numerical perspective and against measurements. It is the dispersed phase, which is induced by the action of the rotor wake, that is responsible for the brownout signature of any given rotorcraft. Hence, before seeking a possible means of mitigation, it is important that the physical phenomena

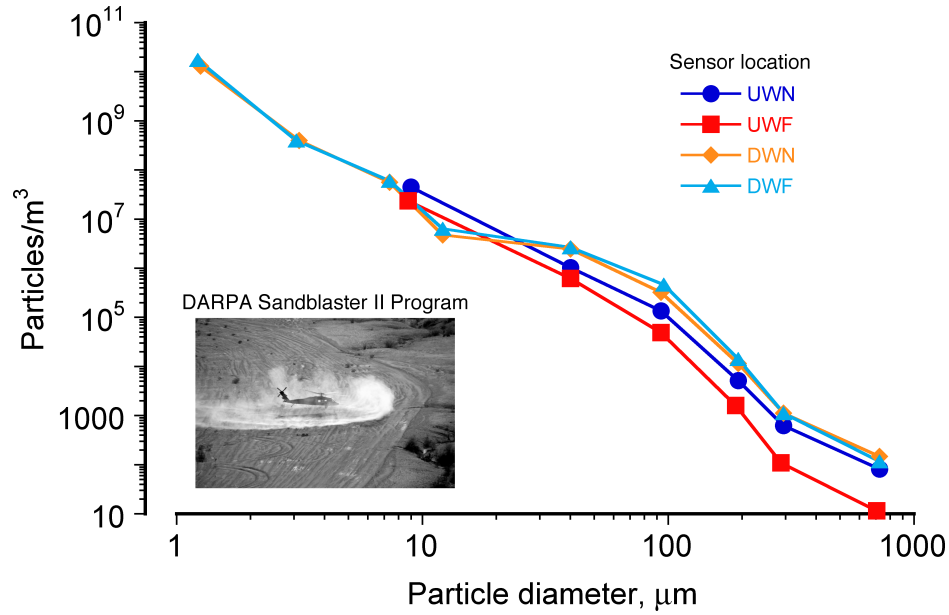


Figure 1.5: Concentration of particles of different diameters in a dust cloud [2,3]

that cause these particles to be mobilized and uplifted are understood. A typical brownout cloud can consist of billions of sediment particles per unit volume with sizes ranging from $1\ \mu\text{m}$ to as much as $1,000\ \mu\text{m}$ in diameter, as shown in Fig. 1.2.2. It is significant that these measurements show high concentrations of small dust particles (diameters of the order of $1\text{--}10\ \mu\text{m}$), which if uplifted are those that tend to remain in suspension for extended periods and are responsible for much of the obscuration effects produced inside the brownout cloud.

The identification of the conditions under which sediment particles become mobilized obviously plays a critical role in correctly predicting the onset of any flow driven particle entrainment process, as well as in the understanding of the specific mechanisms involved in the formation of brownout conditions. Stationary sediment particles on the ground can experience forces from several sources, including shearing from the action

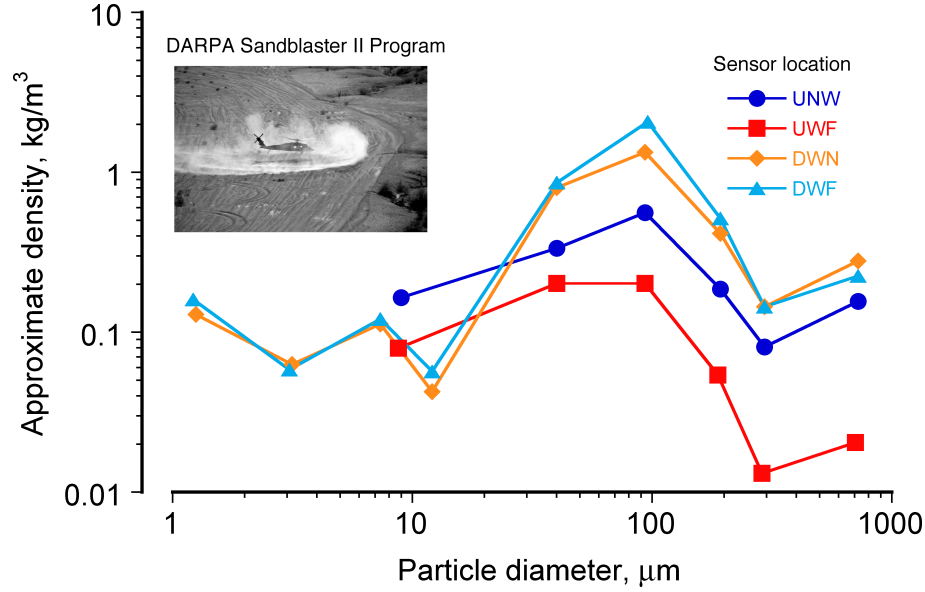
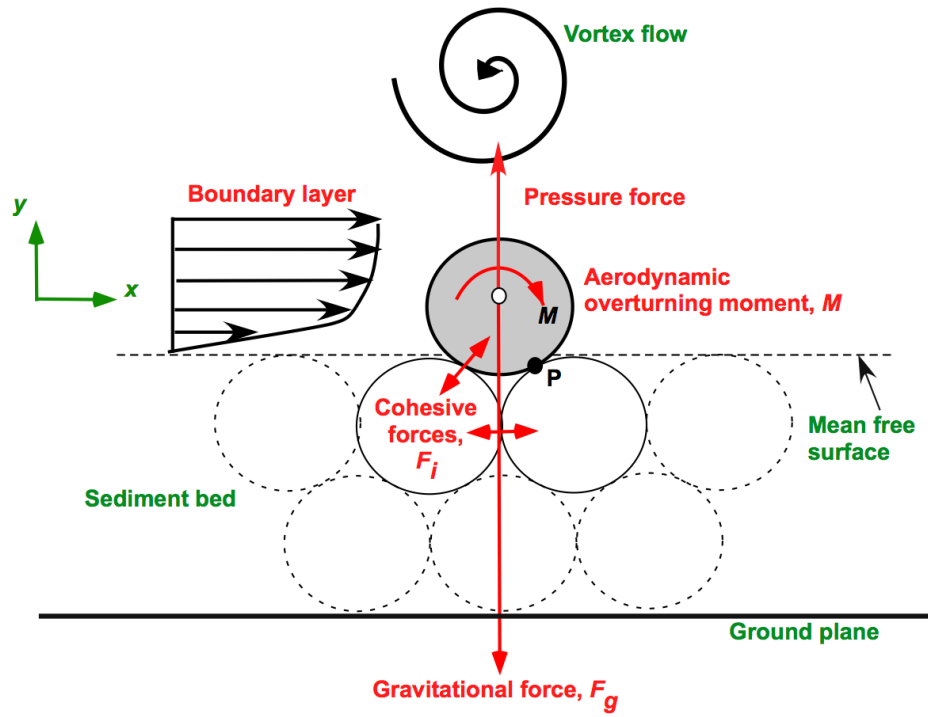
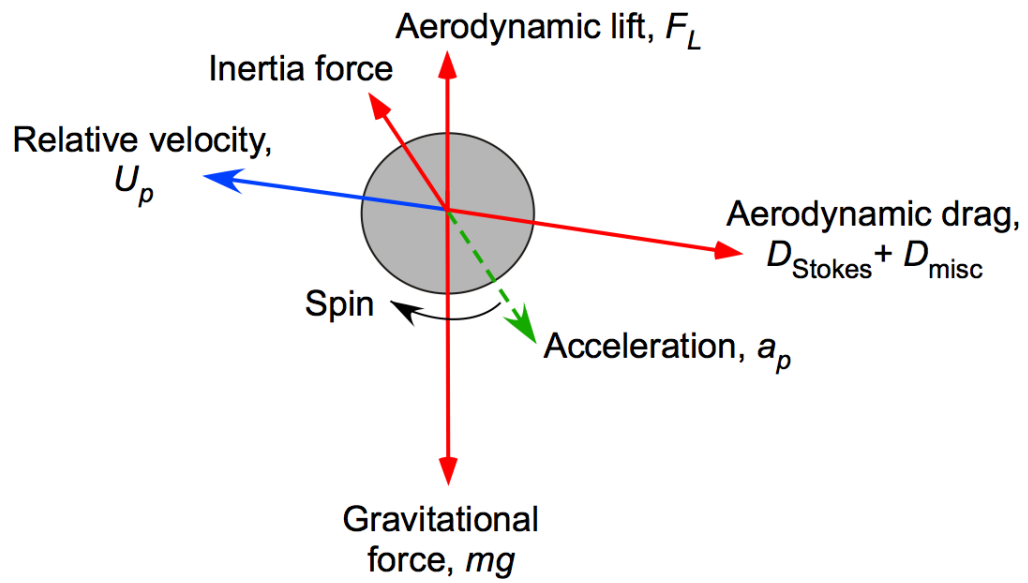


Figure 1.6: Mass density of particles of different diameters in a dust cloud [2, 3]

of the boundary layer, unsteady pressure effects, gravitational effects, and inter-particle cohesiveness (which bonds them to each other and to the surface). Schematics of the forces acting on a particle while on the sediment bed and in suspension are shown in Figs. 1.7(a) and 1.7(b), respectively. Much of the knowledge about sediment mobility in response to fluid motion comes from research into the action of turbulent winds on desert sands (i.e., Aeolian transport), the classic works being from Bagnold [21] and Greenly and Iversen [22]. An important result is that below a certain threshold (expressed in terms of a friction velocity) particle entrainment from the sediment bed will not occur unless there are secondary forces acting. The *average* friction velocities below a helicopter rotor are generally too low to mobilize and entrain the smaller dust particles ($\leq 10 \mu\text{m}$) because they are too cohesive, so in the case of a brownout cloud (which as shown in Fig. 1.2.2 contains predominantly smaller particles), other particle uplift mechanisms must be in-



(a) Particle on mobile sediment bed



(b) Airborne particle

Figure 1.7: Forces acting on a particle while in suspension and on a mobile sediment bed

volved.

When sediment particles lie on the ground below a rotor, the action of the rotor wake can be expected to mobilize and entrain the sediment in regions of high shear and pressure forces and uplift (vertical) velocities. An important entrainment mechanism arises from the effects of the blade tip vortices, whose physics have been described in Section ???. These vortices create regions of significant upwash that can potentially uplift sediment particles to great heights. In this case, high surface shear stresses combined with low pressures and the upward flow velocities induced by the tip vortices, lead to “trapping” of the sediment particles, uplifting them away from the ground [1, 6]. Suspended particles are then rapidly convected away from the bed by the highly three-dimensional, turbulent, unsteady flow. The smaller and lighter particles (i.e., what is usually referred to as dust) that are trapped in the stronger vortical flow regions can be recirculated and bombarded back onto the bed with relatively high velocity and momentum, which were observed experimentally by Sydney et al. [1]. The creation of more dust then follows from bombardment ejection, rapidly intensifying the quantity of suspended particles in the flow.

The interplay between these various mechanisms are shown in Fig. 1.8. If sufficient particle concentrations build up, thereby changing the mass density of the flow (especially near the ground), then the carrier itself may be modified (i.e., the problem is now two-way coupled). Factors compounding the transport characteristics of the dust include particle-particle interactions (i.e., four-way coupling) and the morphology of the sediment bed itself (i.e., through the dynamic process of deflation, deposition, and dune formation).

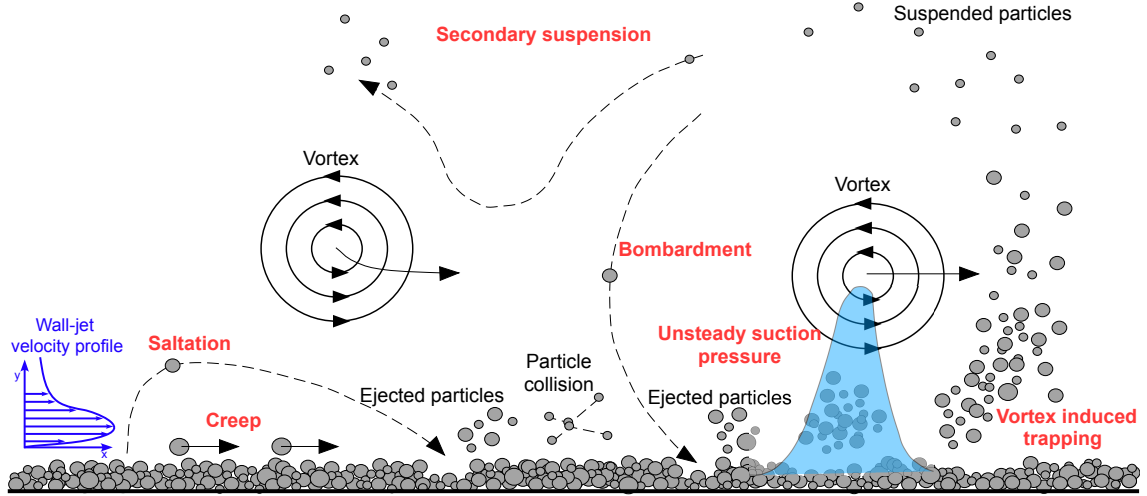


Figure 1.8: Schematic showing the different modes of sediment particle motion and the fundamental uplift mechanisms seen in the near-wall region.

1.3 Computational Modeling

It is clear from the laboratory experiments that have been conducted thus far, as well as from the evidence shown in the field with actual rotorcraft, that there are many interdependent factors involved in the brownout problem. This difficulty suggests that modeling approaches can offer one way of gaining good physical insight into the fundamental parameters that influence the formation of such brownout dust clouds. Such models have value for the brownout problem in terms of their potential predictive capability, but they can also help to understand the individual contributing mechanisms that affect sediment mobility and their overall significance to the brownout problem.

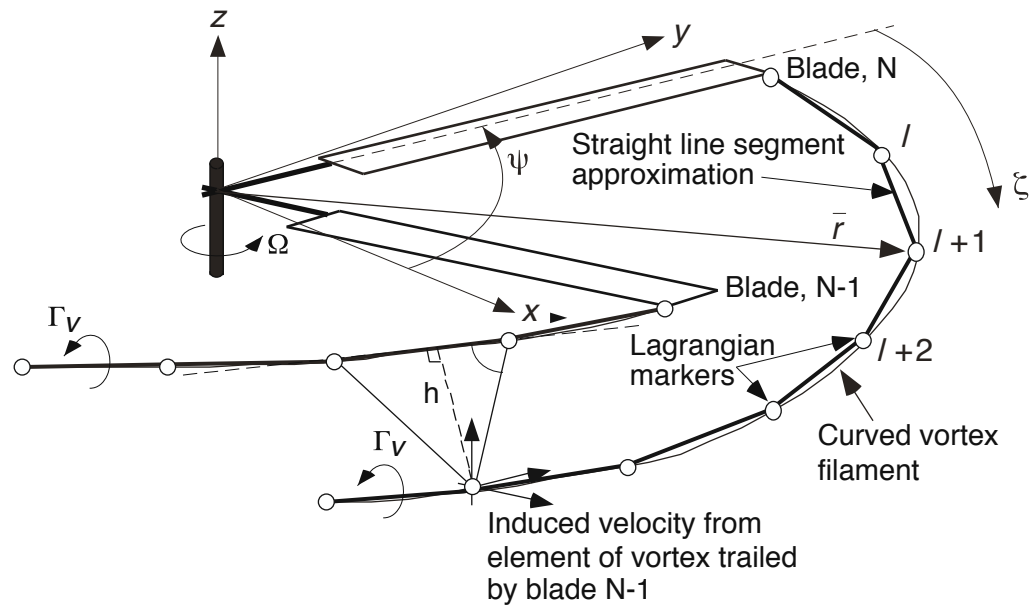
Referring back to Fig. 1.2.2 shown previously, it is clear that most of the particles in suspension are the very small particles, i.e., dust particles with diameters of $1\ \mu\text{m}$ – $10\ \mu\text{m}$ [2,3]. Also of significance is the equivalent mass density of the clouds shown in Fig. 1.2.2, which suggests that simplified “dilute” or one-way fluid coupling assumptions

(i.e., a ratio of density of dispersed phase to the carrier phase of $<10\%$) may be a good modeling assumption, at least for the smaller (and more prominent) dust particles and away from the immediate region at the ground. This observation justifies the development of computational tools for brownout that are based on simpler dilute and one-way coupling assumptions.

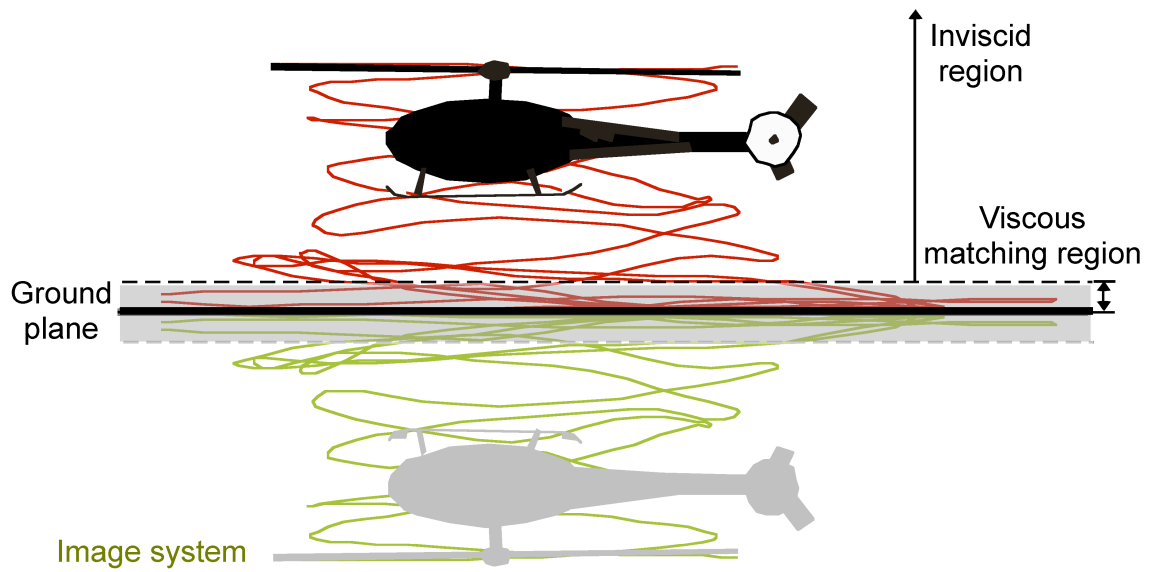
1.3.1 Flow Field Modeling

Several types of brownout modeling approaches have already been developed, including Lagrangian-Lagrangian, Eulerian-Lagrangian, and Eulerian-Eulerian methods. These approaches range from simplified models based on inviscid assumptions and semi-empirical models representing sediment mobility, to the use of CFD including RANS (Reynolds Averaged Navier-Stokes) and DNS (Direct Numerical Simulation) coupled with sediment transport methods. Approaches using RANS methods have been used to study the aerodynamics of the rotor flow in ground effect operations [23]. A RANS-based CFD approach can better resolve velocities near the ground without the use of a sublayer model. However, these methods are computationally very expensive. By taking advantage of the advances in computing technologies, combined with effective use of parallelization techniques, Kalra et al. [23] have shown good success with these high fidelity methods to model the flow at the ground. They used a RANS solver called OVERTURNS [24] and a refined mesh system near the ground to resolve the vortices to longer wake ages, which were validated against the experimental measurements of Lee et al. [19].

Free-Vortex Methods (FVM) have also been developed to model the rotor flow near



(a) Straight line approximation of tip vortices



(b) Free-vortex wake solution at a ground plane using the method of images.

Figure 1.9: Schematic showing the modeling of the Free-Vortex Method

the ground, although the FMV is an inviscid model and so cannot model the boundary layer at the ground. However, using certain approximations the method can be matched to a viscous zone (i.e., a sublayer model) at the ground and the friction velocity estimated there to help calculate sediment mobility and brownout conditions.

The development and validation of FVM has been extensively published [25–29]. Because the FVM has been used in the current study to model the flow field, a brief explanation of the method will be given. The FVM uses Lagrangian markers to track the vortices as they evolve through the rotor flow. In the FVM, the rotor blades are represented using a Weissinger-L type model with a full span near wake. The near wake from each blade is coupled by means of a circulation preserving boundary condition to a far wake consisting of a rolled-up tip vortex. The far wake consists of vortex trailers, usually up to between 10 and 12 rotor revolutions in length, which are discretized into continuous straight-line elements identified by the wake markers. The presence of the ground in the FVM can be modeled using an image rotor system, or with a panel method to account for irregular surfaces.

The governing equation for the wake displacements (written in the form of vorticity transport equation) is further reduced to an advection (wave) equation [30]. This equation is then discretized in space and time, with the velocity source terms appearing on the right-hand side. The solution algorithm used is based on five-point central differencing in space and two-point backward differencing in time [26]. These numerical approximations are second-order accurate, and are consistent with the accuracy obtained with the velocity field reconstruction when using straight-line vortex filament segmentation with the Biot-Savart law [27].

From the wake solutions, the resulting distributions of airloads on each blade are then computed. The individual blade moments, rotor thrust, and torque are obtained by integrating the airloads over the blade span and around the rotor azimuth and are tightly coupled with the blade flapping motion of the rotor wake solution. During the solution process, the rotor is continuously trimmed using collective and cyclic blade pitch inputs to obtain a converged solution at the specified flight condition(s) with the needed rotor thrust.

1.3.2 Particle Physics Modeling

Understanding the phenomenon of brownout, as well as seeking possible means of its mitigation, requires the use of the efficient and accurate particle tracking algorithms. These algorithms need to possess the capability to capture the different fundamental fluid-particle and perhaps even particle-particle interactions, which will all be contributors to the overall brownout footprint. The algorithms developed should also be able to model the sheer number of particles present in a brownout dust cloud. A simple calculation from the measured data presented in Fig. 1.2.2 shows that the number of particles per unit volume in a brownout cloud is of $O(10^9)$. If the total volume of dust entrained into the cloud to be twice the rotor radius in area and one rotor radius in height, then the total number of particles is $O(10^{11})$ – $O(10^{13})$. This is an enormous number of particles to model, which poses many numerical challenges.

1.3.3 Eulerian Particle Models

Different brownout models have emerged over the past years which can be classified into Eulerian and Lagrangian particle techniques. Brown et al. [31] have used an Eulerian-based Vorticity Transport Model (VTM) coupled with a Particle Tracking Model (PTM) to study the evolution of a dust cloud around a rotorcraft. VTM is a class of methods that has been used to model accurate, high-resolution predictions of the dynamics of the wake of a helicopter rotor in ground effect. Like the FVM, the VTM is also an inviscid model, and requires approximate sublayer models to capture the boundary layer on the ground. The VTM was coupled to an Eulerian model for the transport of suspended dust through the flow [4], which aided in the further understanding the fundamental processes at work during the formation of the brownout cloud. A snapshot of a cloud simulated using the Eulerian-Eulerian VTM/PTM model of Brown et al. is shown in Fig. 1.10.

One advantage of the Eulerian, or grid-based approach over the Lagrangian, or particle-based approach to modeling the dust cloud, is that it provides a representation of the dust density distribution that remains linked to the spacing of the underlying computational grid. Brown et al. [31] were able to capture the flow near the ground and also model at least some of the particle entrainment mechanisms that have been observed in experiments. They were able to model and explain the entrainment of particles from vortex-bed interactions, which were associated with “saddle-type” singularities in the velocity field. The Eulerian modeling of the particle cloud, however, prevents visualizing the individual particles in the flow, and only gives a particle density distribution in the brownout cloud. Another disadvantage of this method is its inability to model particle-particle in-

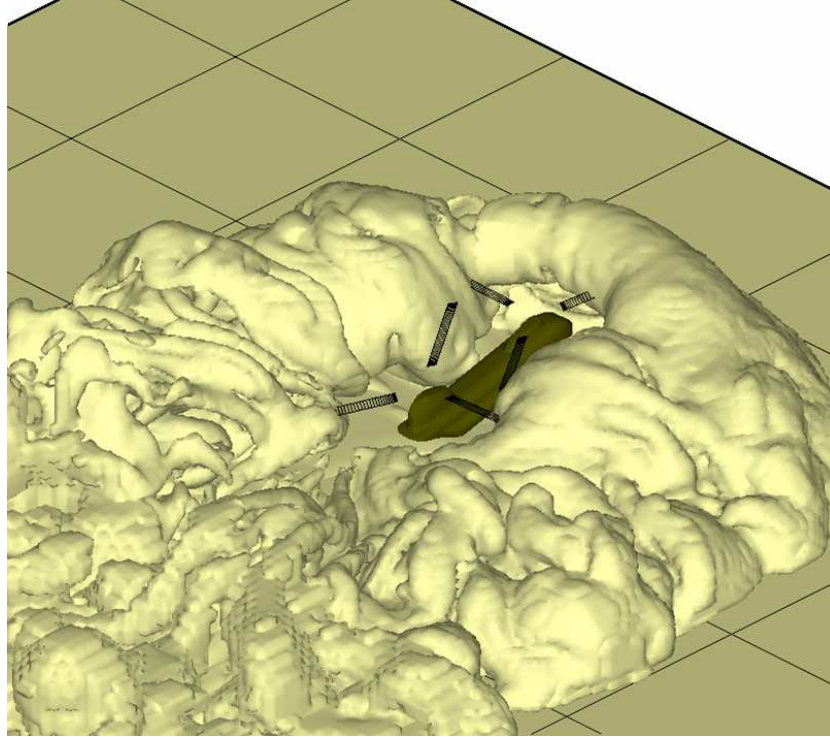


Figure 1.10: VTM/PTM simulation of the dust cloud that is formed in the air surrounding a generic tandem helicopter during its approach to a desert landing [4].

teractions, such as bombardment ejections and collisions, which have been shown to be primary contributors to the development and growth of the dust cloud [1,6].

In another model, Squires et al. [5, 32] performed DNS simulations of a turbulent boundary layer containing coherent vortices (to mimic a rotor wake flow) over a sediment bed. This work uses Lagrangian tracking to modeling of the near-surface dynamics of particle transport near a sediment bed. This Eulerian-Lagrangian model was able to represent the coupling of the dispersed phase to the carrier phase, showing that the carrier phase is modified when high particle density distributions occur. The top frame of Fig. 1.11 shows a snapshot of the predicted particle concentration at the initial instant

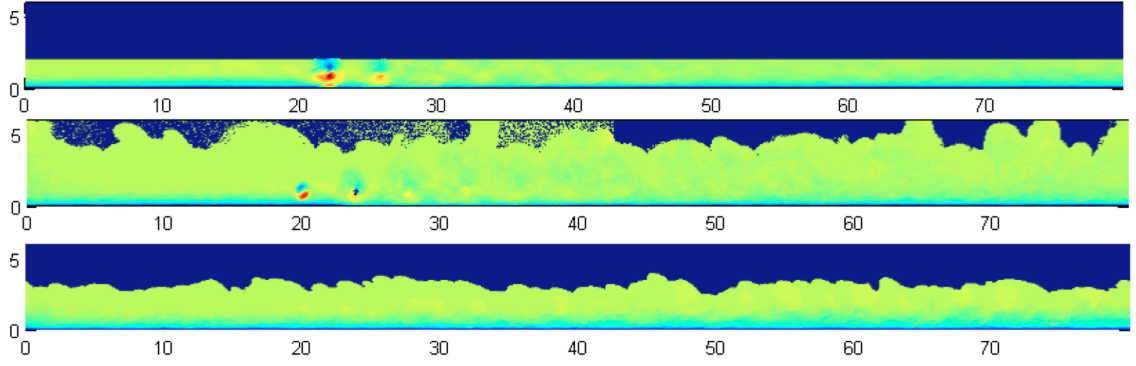


Figure 1.11: Contours of instantaneous streamwise particle velocities. Top: Initial seeding both cases; Middle: Coherent vortex; Bottom: Turbulent boundary layer. [5].

of one such simulation with coherent vortices around a downstream location of 20. The middle and lower frames of Fig. 1.11 show the instantaneous particle concentrations, which include the coherent vortices interacting with the boundary layer and the turbulent boundary layer without the coherent vortices, respectively.

1.3.4 Lagrangian Particle Tracking Models

A Lagrangian particle tracking algorithm was coupled with a FVM by Syal et al. [6]; this method would be classified as a Lagrangian-Lagrangian approach. Various sediment entrainment mechanisms were successfully modeled, including direct entrainment and bombardment entrainment (i.e., saltation and reingestion bombardment). Bombardment entrainment models were based on a Gaussian probabilistic approach. Similar models were used to obtain the volume of the crater after bombardment and the velocity of the ejected particles. The uplift of particles by the unsteady suction pressure on the bed produced by convecting vortices was also modeled. These models were all shown to function very effectively within the context of the entire brownout simulation, and showed

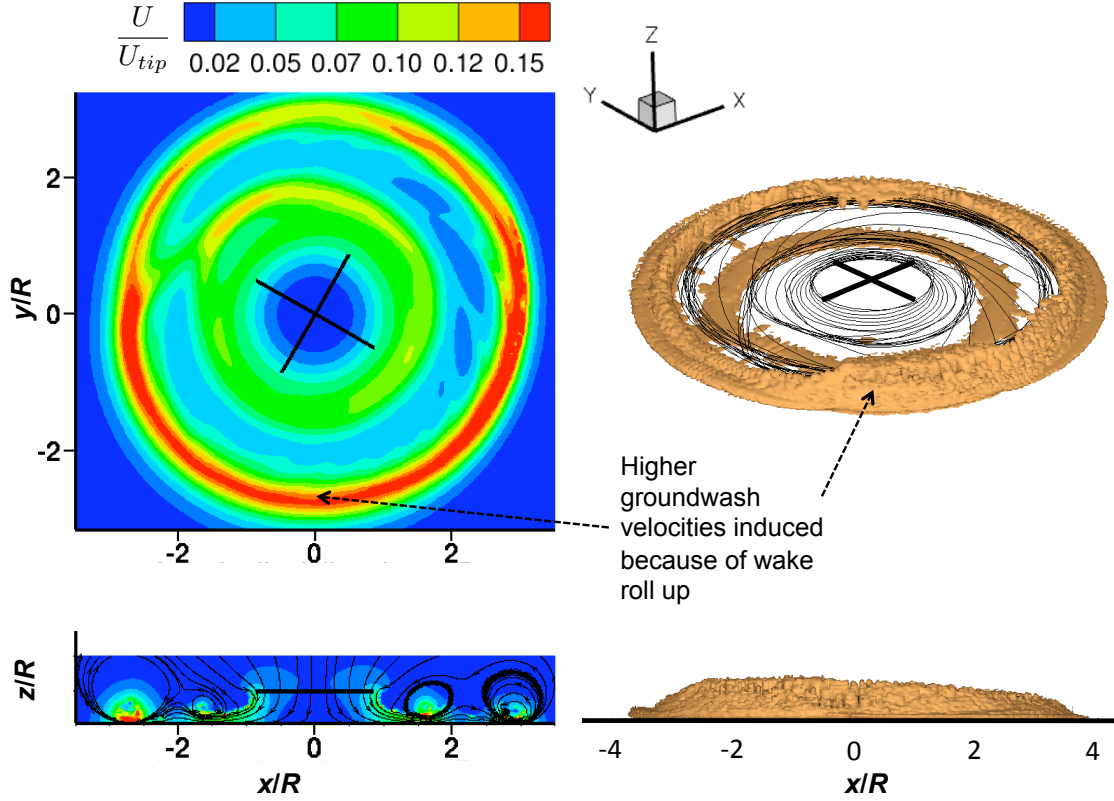


Figure 1.12: Dust cloud along with the associated velocity contours using a Lagrangian particle tracking methodology [6].

physical correlation with the velocity field, a snapshot of which is shown in Fig. 1.12. The method was also used to predict the brownout cloud for a simulated landing maneuver, the results being validated with photogrammetry measurements made by Wong and Tanner [33, 34].

Thomas et al. [35] used the previously described OVERTURNS code along with a Lagrangian sediment tracking algorithm to predict the development of brownout dust clouds. At every time step, the CFD solution was interfaced with the particle tracking algorithm and the velocity of each particle is computed by interpolating the velocities around the CFD grid points that contain it. The advantage of such an Eulerian-Lagrangian

approach is the better prediction of turbulent events, which is known to be important when it comes to the movement and uplift of sediment from an underlying bed [35].

1.3.5 Reducing Computational Costs

A significant advantage that Lagrangian methods offer over Eulerian methods is the ability to accommodate particle level interactions. However, for a high-fidelity realistic brownout simulation, the number of particles required in the simulation are in the order of $O(10^{11})$ – $O(10^{13})$, as previously described, which makes the use of these methods computationally challenging. It is because of the sheer number of particles in the flow that computational costs must be contained if practical simulations of brownout dust clouds are to be performed. For example, for a given Lagrangian-Lagrangian two-phase flow simulation (i.e., using the method of Syal et al. [6]), the total computational cost per time step will be of the order of $O(N^2)$ for the Biot-Savart law computation, and $O(NM)$ for the particle tracking, where N is the number of vortices in the rotor wake and M is the number of particles being tracked.

One approach previously used to contain the cost for each set of calculations has been by using parallel processes and/or multithreaded or multicore CPUs. Recent advancements in computer hardware technology has also enabled the use of Graphics Processing Units (GPU). A GPU is a highly parallel, multi-threaded, many-core processor with significant computational power and high memory bandwidth. GPUs are designed for efficient single instruction multiple data computations [36] and can accelerate the solutions by up to two orders of magnitude.

Apart from the use of hardware implementation to reduce the computational costs, careful code development leading to algorithmic speed-ups can help to tackle the problem at its source. Furthermore, when coupled with a multithreaded CPU or GPU, further reductions in computational time can be achieved. For the N -body interactions problems, Fast Multipole Methods (FMM) can also reduce the order of computation from $O(N^2)$ to $O(N\log N)$. For example, Hu et al. [36] used the FMM on GPU and coupled it with the sediment tracking algorithm of Syal [6]. In a CPU-only environment, a typical brownout simulation can take days of wall-clock time, whereas if implemented on GPUs the same result can be obtained in only a few hours, and without compromising on the accuracy of the solution.

1.3.6 Particle Clustering Techniques

There exists other avenues to contain the computational costs incurred with the Lagrangian tracking of many particles, aside from parallel processing and use of algorithms such as the FMM. These techniques include particle clustering, where groups of particles are treated as one saving on the computational cost of tracking every individual particle. Although there exists the potential to save on the cost of tracking the particles through the use of such particle clustering methods, very limited work has been done in the past in the development of the most suitable algorithms in terms of their effectiveness and savings in computational cost, e.g., relative to a direct or “brute-force” approach where the trajectories of all particles are computed directly.

Wachpress et al. [37] have claimed that it is possible to perform certain brownout

simulations with the help of a particle clustering method. They assumed that each “cluster” represented a larger number of particles and the particles within a cluster are spread using a Gaussian distribution. The equations of motion were solved by considering the velocity and displacement of the cluster at any given time to be comprised of a mean and a fluctuating component. The final results for the cluster velocities and displacements is based on the mean scale turbulence level and integral scale of mean square turbulence. The fluctuating displacement of the cluster, which was used to compute the position variance, corresponds to the growth of the cluster over time around the mean cluster location. The algorithm used in this approach is detailed by Teske et al. [38]. This model, however, requires the prior knowledge of the mean turbulence and level and the integral scale of the mean turbulence level for specific problems. These parameters were determined by Teske et al. for spray drift for aerial applications such as the dispersion of pesticides from aircraft. However, such parameters are fundamentally more difficult to determine for rotorcraft brownout applications. Although Wachpress et al. [?] showed some results of brownout simulations, the details of the approach were not given and the results were not validated against direct particle solutions.

Other clustering approaches are obviously possible. The k-means algorithm [39] is one of the simplest unsupervised learning algorithms that can be used to divide a given data set into groups or clusters. This algorithm is used in various fields, ranging from market segmentation, computer vision, astronomy, and agriculture. Traditionally, the k-means method is used to help classify data into organized structures, which aids in the further analysis of the problem at hand. Although the method has not been previously applied to the problem of rotorcraft brownout, there is potential for using the algorithm

to divide all the particles to be tracked into clusters, thereby saving on the computational cost of tracking each and every particle. Therefore, the characteristics of the k-means algorithm for brownout problems was explored in the present research.

For the purposes of evaluating the optical characteristics of the dust cloud (i.e., visibility through the cloud from the pilot's perspective), it may be sufficient that the density distribution be computed throughout the spatial region of interest. To this end, Osipov's method [7] helps obtain the density distribution using a few Lagrangian particles used as tracer particles. This method has been applied in predicting the dust density distribution in two-dimensional internal flows through turbine cascades, with promising results. Details of this method are presented in Chapter 2 of this thesis. Healy et al. [7] dealt with mostly internal flows that are much less vortical in nature compared to those seen with rotorcraft wakes.

1.4 Objectives of the Present Work

The present work contributes to a class of algorithms that can help in expediting brownout calculations through smart particle tracking algorithms rather than through hardware implementations. In general, prior work in the field of particle clustering techniques has been limited, and the algorithms used have not been published in detail in the open literature. The advantages and limitations of these methods are not known, especially in regard to the brownout problem, as a consequence of which their applicability to the problem is not fully understood. To this end, the Gaussian, k-means and Osipov's method are studied in the present work with an emphasis on the numerical characteristics

of each algorithm. The algorithms are also studied with an attempt to understand their relative advantages and their applicability to the brownout problem.

In summary, the present research involved the development of generally computationally efficient algorithms that can be applied to the simulation of dilute gas-particle suspensions at low Reynolds numbers of the relative particle motion. The goal was to reduce the computational cost of the dust cloud simulation, while also preserving the accuracy of the solution within acceptable tolerance limits. To this end, the Gaussian distribution, k-means and Osipov's clustering methods were studied in detail for a prototypical flow field that mimics the highly unsteady, two-phase vortical particle flow found during brownout conditions. The regions of applicability of these methods along with their relative advantages were studied in detail and their bounds explored. In the case of the Gaussian method, the algorithm was applied to the full three-dimensional brownout problem. It is shown that although clustering algorithms can be problem dependent and have bounds of applicability, they offer the potential to significantly reduce computational costs while retaining the overall accuracy of a brownout dust cloud solution.

1.5 Organization of Thesis

The present work explores three different particle clustering algorithms to reduce the computational time incurred for brownout dust cloud calculations. The problem of brownout in terms of the flow and particle physics have been discussed in the present chapter. A review of previous work on the brownout problem from a computational standpoint has also been discussed. Chapter 2 gives a comprehensive description of the

algorithms that were implemented and tested. These descriptions include the motivation behind exploring each algorithm and the mathematics behind them. A description of the prototypical flow and the governing flow and particle equations are also discussed. Chapter 3 documents the results in terms of regions of applicability for each of the algorithms, their relative advantages, gains in computational time, and the resulting accuracy of the solutions. The effectiveness of each of the clustering methods is explored for potential use in complete brownout simulations. Finally, Chapter 4 concludes the thesis by discussing the significance of the findings, and suggesting future studies that will help to better understand these algorithms and how they might be applied to modeling rotorcraft brownout problems.

Chapter 2

Methodology

2.1 Introduction

To perform Lagrangian tracking of particles for a full-scale three-dimensional brownout problem, it is clear that with the large number of particles that are involved a direct of “brute force” computation would result in an unreasonable, perhaps prohibitive, computational time. To help overcome this problem, smart particle tracking algorithms have been studied to preserve the total number of particles in the simulation but limiting the computational time. Several different particle-clustering approaches are possible, which may also include the adoption of declustering and reclustering strategies. A disadvantage of a clustering method can be with accumulating inaccuracies in the particle trajectories, which depends on the method used to perform the clustering. However, if it can be shown that a given method incurs only small losses in accuracy of the particle positions for significant gains in computational time in finding a solution, then clustering methods can become very powerful tools in the simulation of brownout dust clouds.

The specific methods developed and implemented in the present work were: (a) the Gaussian Method, (b) the k-means Method, and (c) Osipov’s Method. These algorithms were evaluated with an emphasis on: (a) improvements in computational time as compared to a direct calculation where each the trajectories of individual particles are computed and tracked in time and space, and (b) the accuracy of the solution relative to

the direct method.

2.2 Gaussian Method

The Gaussian method treats all of the particles in the cloud simulation as clusters, each cluster representing a certain set of particles that are spatially distributed around the cluster center. A schematic of the method is shown in Fig. 2.1. If there are n_1 particles assumed in the dust cloud simulation, then each of these particles can be treated as a cluster representing n_2 particles, giving $n_1(n_2 + 1)$ total particles, i.e., $n_1 n_2$ cluster-generated particles and n_1 original clusters.

The n_2 particles around each of the n_1 particle clusters are distributed in a specific Gaussian form. For these clustered particles to mimic the physics of the flow at any given time, the mean vector μ_c and covariance matrix Σ_c for each cluster can be defined as

$$\mu_c = \begin{bmatrix} V_{px} & V_{py} & V_{pz} \end{bmatrix} \quad (2.1)$$

and

$$\Sigma_c = \frac{1}{E_c} \begin{bmatrix} k_1 |V_{px}|^2 & k_2 |V_{px}| |V_{py}| & k_2 |V_{px}| |V_{pz}| \\ k_2 |V_{py}| |V_{px}| & k_1 |V_{py}|^2 & k_2 |V_{py}| |V_{pz}| \\ k_2 |V_{pz}| |V_{px}| & k_2 |V_{pz}| |V_{py}| & k_1 |V_{pz}|^2 \end{bmatrix} \quad (2.2)$$

where V_{px} , V_{py} , and V_{pz} are the instantaneous velocities of each cluster and E_c is a measure of the kinetic energy of the group. Scaling of the covariance matrix by E_c^{-1} is performed where the local flow velocities become large. The generated clusters at worst case (i.e. when the velocities are large) would collapse down to the original particle and will never cause a blown-out solution because the value of E_c would increase. The conditions on the

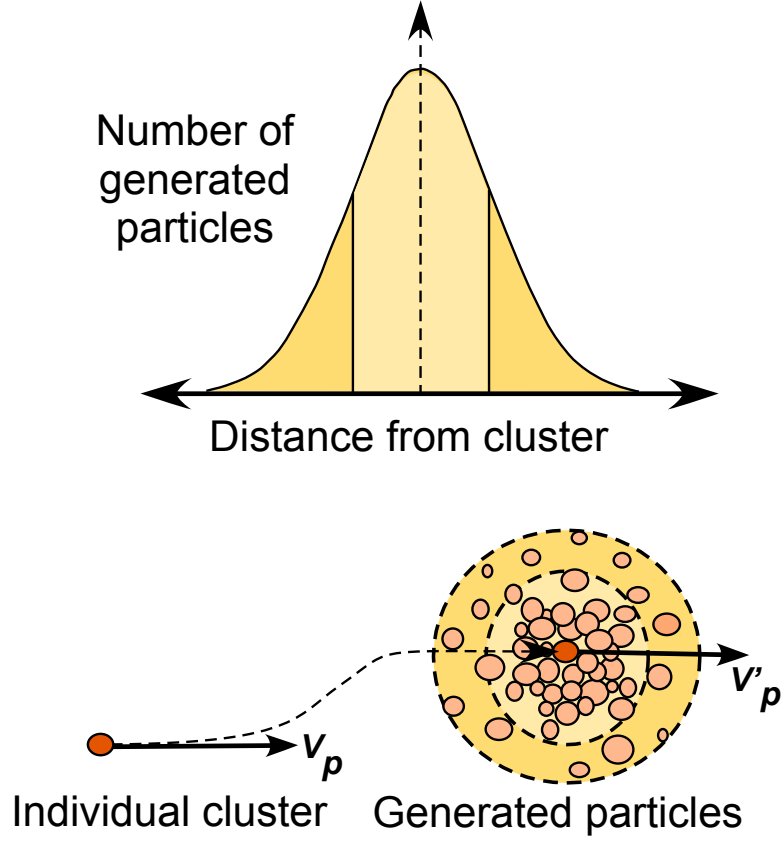


Figure 2.1: Schematic showing the Gaussian method of particle clustering.

above matrix to be positive definite are

$$k_1 > 0, \quad k_2 > 0, \quad (k_1 - k_2) > 0 \quad (2.3)$$

which is derived from Sylvester's Condition to have a positive definite matrix [40], and

where

$$\frac{k_2}{k_1} < \min \left[\frac{|V_{px}|}{|V_{py}| + |V_{pz}|}, \quad \frac{|V_{py}|}{|V_{px}| + |V_{pz}|}, \quad \frac{|V_{pz}|}{|V_{px}| + |V_{py}|} \right] \quad (2.4)$$

which is required for the matrix to be diagonally dominant.

The parameter k_1 is a scalar that dictates the spread in each of the principal directions, and k_2 can be viewed as a cross-coupling coefficient. Higher the value of k_1 , higher

will be the spread of the particles around the cluster centres in the principal directions. The positions of the n_2 particles distributed around each cluster are then calculated using the velocities obtained from the distribution matrix. The particles are distributed in a normalized Gaussian fashion irrespective of the values of the coefficients k_1 and k_2 .

2.3 k-means Method

The k-means clustering method is based on the principle that certain sets of individual particles can be decomposed into smaller groups of clusters, and that the resulting equations of motion are solved only for the clusters. The displacements of the cluster centers are then applied to all of the individual particles comprising the cluster. There are two aspects to this method. 1. Selecting the candidate particle groups to form a cluster; 2. The effects produced by clustering, declustering, and reclustering of the particle groups.

The general idea of the k-means clustering approach is depicted in Fig. 2.2. The k-means clustering algorithm is a category of cluster analysis that aims to partition n observations (i.e., particles) into k clusters in which each particle belongs to the cluster with the nearest mean. The most common algorithm uses an iterative refinement technique. Brief steps of the algorithms are as follows:

1. k initial “means” are randomly selected from the data set of particles.
2. k clusters are created by associating every particle with the nearest mean. The computational time of this step is $O(kN)$, where N is the total number of particles in the data set.
3. The centroid of each of the k clusters are computed and they become the new means.

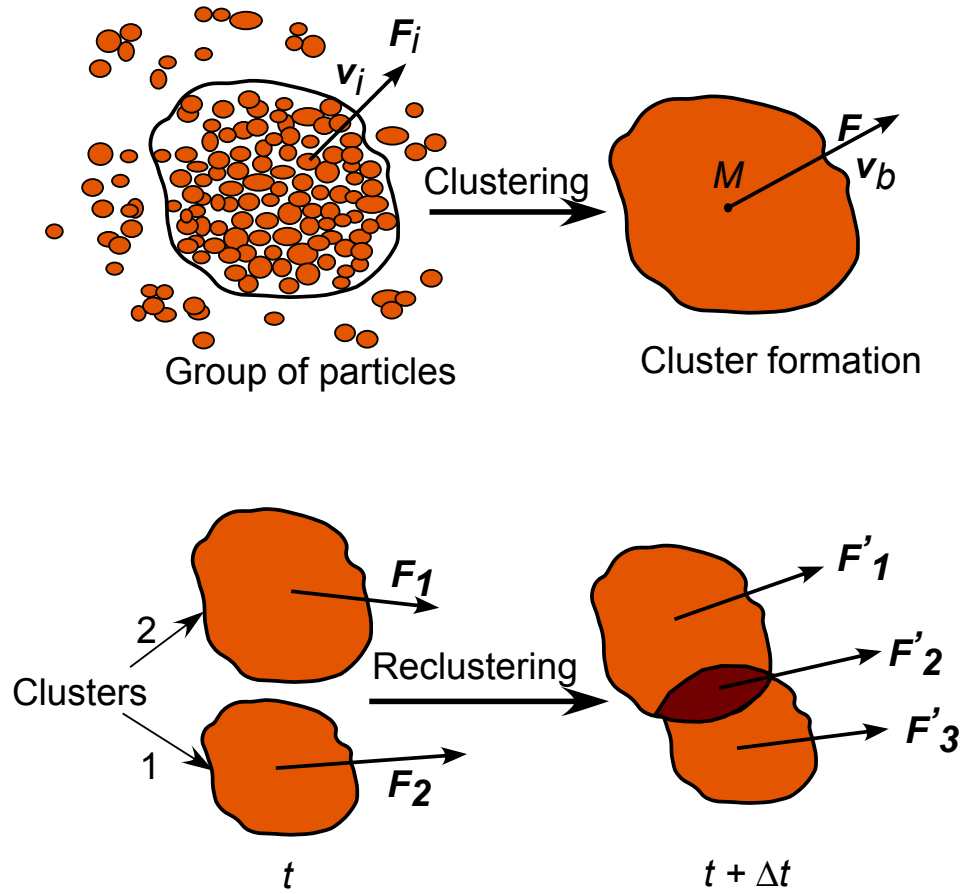


Figure 2.2: Schematic showing the k-means method of clustering and declustering.

4. Steps 2 and 3 are repeated until convergence has been reached.

Mathematically, given an initial set of k means $m_1^{(1)}, \dots, m_k^{(1)}$, the algorithm proceeds by alternating between the following two steps:

1. **Assignment step:** Each particle is assigned to the cluster with the closest mean,

i.e.,

$$S_i^{(t)} = \left[x_j : |x_j - m_i^{(t)}| \leq |x_j - m_{i^*}^{(t)}|, i^* = 1, \dots, k \right]$$

2. **Update step:** The new means are calculated and they are the centroid of the particles in the cluster, i.e.,

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Any two given clusters can be reclustered, if required, based on the criteria

$$\frac{S_{12}}{S_1 + S_2} < k_{12} \quad (2.5)$$

where S_{12} is the inter-cluster separation distance and S_1 and S_2 are the intra-cluster scatter, the latter being the average distance of all particles in a cluster from the cluster center. The threshold parameter, k_{12} , is typically less than unity and reflects the threshold distance under which two overlapping clusters become eligible for clustering. The foregoing is basically a variation of the Davies-Bouldin index [41].

The computational benefits of such a clustering method proves to be relatively attractive for brownout-type simulations, in some cases the method can be up to two-orders of magnitude faster compared to a direct simulation using the same total number of particles. Because the movement of their respective clusters approximates the motion of all of the individual particles in the cluster, the accuracy of the solution will obviously be compromised as the number of clusters is decreased. Therefore, some reasonably large number of clusters must always be chosen to avoid concerns about accuracy.

A significant disadvantage of this method is that the conditional clustering and declustering incurs a significant computational overhead, and this cost can become significant in problems having high velocity gradients in the carrier flow field where the process may need to be repeated frequently.

2.4 Osipsov's Method

Osipsov's method has its roots in Lagrangian theory and involves integrating equations for particle density along individual pathlines. The particle densities are obtained from the Lagrangian form of the conservation of mass by computing the change in volume of an element of "particle fluid" along its pathline. Although the exact origins of this method are not clear, a set of relevant equations first appeared in a work by Fernandez de la Mora and Rosner (FMR) [42]. A related line of development was carried out by Osipsov et al. [43].

Healy and Young [7] have compared the FMR and Osipsov's methods for simple flow problems, and have used the Osipsov's method to predict particle densities for the flow past a cylinder as well as for the more challenging problem of the particle-laden flow through a turbine cascade. A significant reduction in the computational costs was shown for the same accuracy.

Although both the FMR and Osipsov's method are based on the same principles, Osipsov's method is more suitable for handling flows with crossing pathlines, whereas the FMR method leads to a mathematical singularity [42]. Because crossing pathlines are typical of flows that are more vortical in nature, Osipsov's method becomes a potential candidate for helping to model brownout problems.

To illustrate Osipsov's method, a simplified two dimensional case can be considered. In this case, particles move only under the action of the Stokes drag and gravity forces, i.e.,

$$\frac{d\mathbf{V}_p}{dt} = -\frac{(\mathbf{V}_p - \mathbf{U})}{\tau_p} + \mathbf{g} \quad (2.6)$$

where \mathbf{V}_p and \mathbf{U} are the particle and flow velocities, respectively. The corresponding equation for the conservation of mass is

$$\rho_p = \frac{\rho_{p0}}{|J|} \quad (2.7)$$

where ρ_{p0} is the mass density of the particulate phase at $t = 0$. At this time, the element is rectangular of area $\delta x_{p0} \delta y_{p0}$ (i.e., a two-dimensional volume), as shown in Fig.

2.3. At a later time t , it has become trapezoidal-shaped and has area $|J| \delta x_{p0} \delta y_{p0}$ where $J = J_{xx}J_{yy} - J_{xy}J_{yx}$ and

$$\begin{aligned} J_{xx} &= \left(\frac{\partial x_p}{\partial x_{p0}} \right)_{y_{j,p0},t}, & J_{yy} &= \left(\frac{\partial y_p}{\partial y_{p0}} \right)_{x_{j,p0},t}, \\ J_{xy} &= \left(\frac{\partial x_p}{\partial y_{p0}} \right)_{x_{j,p0},t}, & J_{yx} &= \left(\frac{\partial y_p}{\partial x_{p0}} \right)_{y_{j,p0},t} \end{aligned} \quad (2.8)$$

The reasoning behind using the absolute value of the Jacobian is that when particle path-lines cross, the sign changes but the area enclosed remains positive.

For the purposes of concise presentation of the following derivation, the dimensions in space x and y are denoted as x_1 and x_2 respectively. Four new variables are now defined based on the relationship

$$\omega_{ij} = \frac{\partial J_{ij}}{\partial t} = \frac{\partial}{\partial t} \left(\frac{\partial x_{p,i}}{\partial x_{j,p0}} \right) = \frac{\partial V_i}{\partial x_{j,p0}}; \quad \forall (i, j) \in 1, 2 \quad (2.9)$$

Further differentiating the set of Eq. 2.9 with respect to t , gives

$$\begin{aligned} \frac{\partial \omega_{ij}}{\partial t} &= \frac{\partial}{\partial t} \left(\frac{\partial V_i}{\partial x_{j,p0}} \right) \\ &= \frac{\partial}{\partial x_{j,p0}} \left(\frac{\partial V_i}{\partial t} \right) \\ &= \frac{1}{\tau_p} \frac{\partial}{\partial x_{j,p0}} (U_i - V_{i,p}) \frac{1}{\tau_p} \frac{\partial}{\partial x_{j,p0}} (U_i - V_{i,p}) \\ &= \frac{1}{\tau_p} \left(\frac{\partial U_i}{\partial x_{j,p0}} - \omega_{ij} \right); \quad \forall (i, j) \in 1, 2 \end{aligned} \quad (2.10)$$

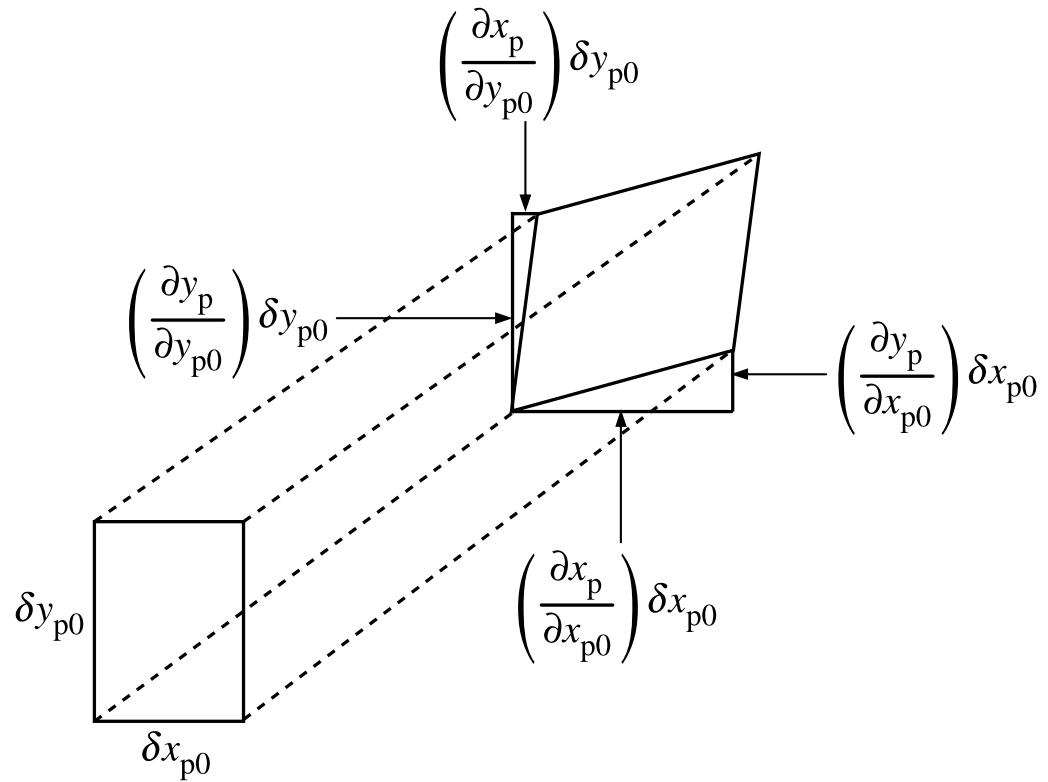


Figure 2.3: Schematic showing the change in shape of the particle fluid element. The dotted lines represent particle pathlines, which would normally be curved [7].

Using the chain rule for differentiation then

$$\frac{\partial \omega_{ij}}{\partial t} = \frac{1}{\tau_p} \left(\sum_{k=1}^2 J_{kj} \frac{\partial U_i}{\partial x_k} - \omega_{ij} \right); \forall (i, j) \in 1, 2 \quad (2.11)$$

Equations 2.6, 2.7, 2.9 and 2.11 can be integrated numerically along the pathlines. However, the derivative of the fluid velocity is also needed at the particle positions. These derivatives can be obtained using a second-order central differencing scheme. The particle concentration ρ_p is then obtained purely as a post-processing step because it is the Jacobian J that carries the information in the computations.

Equations 2.9 and 2.11 are numerically integrated using the same scheme used to integrate the momentum equations, i.e.,

$$\omega_{ij} = (\omega_{ij})_0 e^{-\Delta t / \tau_p} + (1 - e^{-\Delta t / \tau_p}) \left(\sum_{k=1}^2 \overline{J_{kj}} \frac{\partial U_i}{\partial x_k} \right); \forall (i, j) \in 1, 2 \quad (2.12)$$

A further integration gives an expression for the components of the Jacobian, i.e.,

$$\begin{aligned} J_{ij} = & (J_{ij})_0 + \tau_p (1 - e^{-\Delta t / \tau_p}) (\omega_{ij})_0 \\ & + (\Delta t - \tau_p (1 - e^{-\Delta t / \tau_p})) \left(\sum_{k=1}^2 \overline{J_{kj}} \frac{\partial U_i}{\partial x_k} \right); \\ & \forall (i, j) \in 1, 2 \end{aligned} \quad (2.13)$$

Notice that when the pathlines cross, the value of the Jacobian goes to zero, which implies that the value of ρ_p would go to infinity. This result is practically improbable as the effects of finite particle size and inter-particle collisions become more likely at high particle concentrations.

2.5 Error Estimations

The solutions obtained through clustering methods might save on computational time, but the accuracy obtained must also be considered. Ensuring that the errors stay within prescribed limits is critical because the errors can become compounded through the solution as it evolves. Computing the root-mean-square (rms) error to evaluate the accuracy of the solution may be appropriate, and this is a reasonable metric if the flow problems being considered are relatively simple (e.g., two-dimensional particle motion induced by single vortex in a uniform flow). However using the rms error as a method for error estimation was found to be less appropriate for more complex flow fields (e.g., a flow field with multiple vortices or a three-dimensional flow). For example, a single particle whose position maybe far away from the remainder of the cloud during the evolution of the solution will result in a high rms error. However, this single particle by itself is unimportant for all practical purposes, rendering the rms a less useful error metric.

A better metric of error estimation is to compare density plots between the clustered solution and the actual computed solution. For example, quantitative metrics to quantify how “good” or “bad” a brownout cloud have been developed by Tritschler et al. [44] using the Modulation Transfer Function (MTF) to predict the optical characteristics of the brownout cloud as seen by the pilot. In this case, the need for a high fidelity dust cloud solution is critical if the MTF is to be predicted with good accuracy.

Consider the case where two different solutions are compared using density plots, with the density being computed in terms of number of particles in each “bin.” If the size of the bin was the size of the entire space, then the two densities would be the same

(because the total number of particles are the same). As the bin size is reduced, the error increases between the two solutions. When extended to three-dimensions, this idea can be used to determine the applicability of clustering.

A solid angle of $d\theta$ is extended from the center of the sphere (e.g., the pilot's eye) as illustrated in Fig. 2.4. The distance $Rd\theta$ can be related to a mesh size in two-dimensions; now the mesh sizes $R_2d\theta > R_1d\theta$ as $R_2 > R_1$. This result implies that there will be some distance away from an observer location beyond which accuracy is maintained by clustering. It also follows from a physical reasoning that small errors in the position of particles will not be noticed if the observer is sufficiently far away.

The two density distributions can be quantified as intensity maps if the Pearson correlation coefficient is used. In this case, the correlation coefficient, r , between two density matrices, X and Y , is given by

$$r = \frac{\sum_i \sum_j [X(i, j) - \bar{X}][Y(i, j) - \bar{Y}]}{\sqrt{\sum_i \sum_j [X(i, j) - \bar{X}]^2 \sum_i \sum_j [Y(i, j) - \bar{Y}]^2}} \quad (2.14)$$

where $X(i, j)$ and $Y(i, j)$ indicate the i^{th} and j^{th} components of the density value, while \bar{X} and \bar{Y} are mean values of the intensity matrices X and Y , respectively. As the value of r approaches 1, the images are well correlated with $r = 1$ being an exact correlation. As the value nears $r = 0$, the images become poorly correlated. It should be notes that the Pearson coefficient indicates the overall correlation between two given density maps but is invariant to local differences. However, the Pearson coefficient is still a good metric to begin with for simple flow fields.

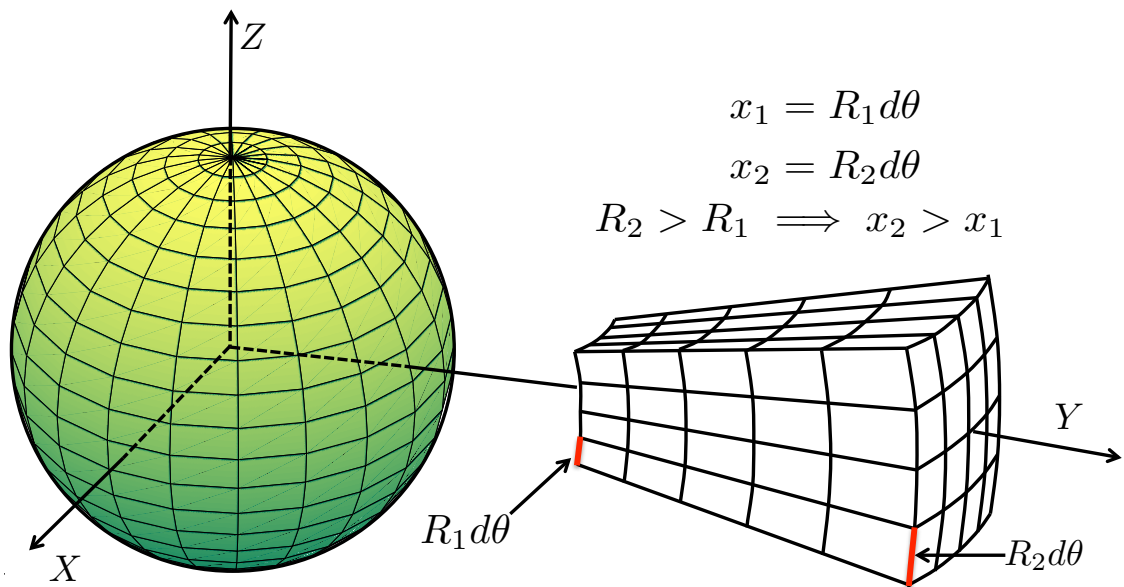


Figure 2.4: Spherical coordinate system centered around the pilot. Grid size of each “bin” increases when moving away from an observer.

2.6 Prototypical Flow

Although the actual dust flow surrounding a rotorcraft is very unsteady and three-dimensional, prototypical two-dimensional flows were chosen to evaluate the three clustering algorithms. The advantage of using more simplified flows is that a good understanding of the respective nuances of the clustering methods can be established, and the results can be used to expose quite easily the relative predictive advantages of each method as they would apply to more complex problems.

One prototypical flow field chosen was a series of convecting two-dimensional vortices that pass over a thin “saltation” region of randomly distributed dust particles in several layers, as shown in Fig. 2.5. This problem is representative of what happens in the two-phase flow generated below the rotor flow as observed in experiments [1].

The flow tangency condition at the ground plane was modeled using an image flow system. Particles are uplifted when the external forces acting on the particle exceed the gravitational and the drag force. Potential flow of the gas and small Reynolds numbers of the relative particle motion were assumed in the present study. To keep the model relatively simple, cohesive forces, electrostatic forces, etc. were neglected, and other mechanisms of particle uplift such as pressure effects [34] were not modeled. This representative flow was considered because the interest in this case lies more in the evaluation of the clustering algorithms and not on the quantitative prediction of the two-phase flow per se.

The saltation region consisted of particles that were treated as monodispersed spherical entities that behave as a dilute gas. Therefore, it was assumed that the particles are

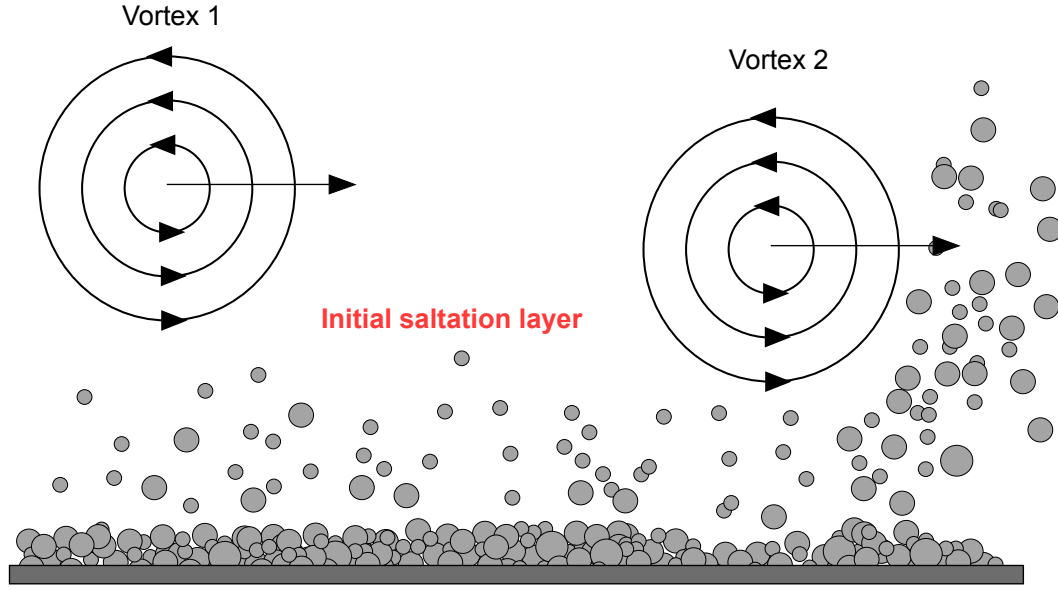


Figure 2.5: Schematic showing the passage of vortices over an initial saltation layer, and uplifting the dust [1].

driven by the carrier gas but not vice-versa, and that collisions between the particles can be ignored. The particles in the upper most part of the saltation layer are considered to be active and eligible for entrainment. As these particles are entrained into the flow, subsequent layers become active and so become eligible for entrainment.

2.6.1 Particle Convection

The equation of motion of the particles can be written as

$$\begin{aligned}
 m \frac{d\mathbf{V}_p}{dt} &= \sum \mathbf{F} \\
 &\simeq \mathbf{F}_d + \mathbf{F}_g \\
 &= -\frac{1}{2} \rho C_d A |\mathbf{V}_p - \mathbf{U}| (\mathbf{V}_p - \mathbf{U}) + m \mathbf{g}
 \end{aligned} \tag{2.15}$$

where the forces on the particles include a drag force, \mathbf{F}_d , gravity force, \mathbf{F}_g , as well as buoyancy, apparent mass, and Basset forces. For this simplified model, only the drag and gravitational forces were retained. The particle drag coefficient C_d is defined by

$$C_d(Re_p) = \frac{24}{Re_p} \left[1 + 0.15Re_p^{0.687} \right] \quad (2.16)$$

where the particle Reynolds number, Re_p , is

$$Re_p = \frac{|\mathbf{V}_p - \mathbf{U}| d_p}{\nu} \quad (2.17)$$

For the size of particles prominent in the brownout problem, then $Re_p \ll 1$, i.e., the particle experiences a Stokes flow. For these flows, $C_d = 24/Re_p$. The equation of motion is given by Eq. 2.6 where τ_p is the particle velocity relaxation time as given by

$$\tau_p = \frac{m}{\frac{1}{2}\rho C_d(Re_p) A |\mathbf{V}_p - \mathbf{U}|} \quad (2.18)$$

For the Stokes regime this turns into

$$\tau_p = \frac{\rho_p d_p^2}{18\mu} \quad (2.19)$$

Notice that the particle velocity relaxation time is a function of particle density and size.

For a dilute flow, the equation of motion can be decoupled in the two dimensions, and written as

$$\frac{dV_{px}}{dt} = -\frac{(V_{px} - U_x)}{\tau_p} \quad (2.20)$$

and

$$\frac{dV_{py}}{dt} = -\frac{(V_{py} - U_y)}{\tau_p} \quad (2.21)$$

Equations 2.20 and 2.21 can then be integrated along the particle pathlines using any algorithm for simultaneous Ordinary Differential Equations (ODEs). In the present study,

a predictor-corrector scheme was used. If an explicit integration method is used, the integration time-step must be less than the particle relaxation time τ_p to ensure numerical stability. This problem is not an issue at high Stokes numbers because the time step is limited for reasons of accuracy by the requirement that the carrier flow conditions do not change significantly over a single time step. However, at lower Stokes numbers, the equations become mathematically stiff and small time steps are required to ensure the stability of the time marching scheme.

One method of overcoming this numerical barrier is to use an approximate analytical integration method of the momentum equations over a time-interval long compared to the particle relaxation time τ_p , but are small compared to the time-scale changes of the carrier flow. This approach is simpler than traditional backward different numerical methods for stiff ODEs.

Consider the x -momentum equation (Eq. 2.20), which can be rewritten as

$$\frac{d(V_{px} - U_x)}{dt} + \frac{(V_{px} - U_x)}{\tau_p} = -\frac{dU_x}{dt} \quad (2.22)$$

Multiplying by e^{t/τ_p} and integrating from $t = 0$ to $t = \Delta t$ and assuming a constant dU_x/dt , then

$$\begin{aligned} V_{px}(t + \Delta t) &= U_x(t + \Delta t) + (V_{px}(t) - U_x(t))e^{-\Delta t/\tau_p} \\ &\quad - \tau_p \left(\frac{\overline{dU_x}}{dt} + g \right) (1 - e^{-\Delta t/\tau_p}) \end{aligned} \quad (2.23)$$

where $\overline{dU_x}/dt$ represents the average value over the time-step. Similarly the y -momentum

equation can be written as

$$V_{py}(t + \Delta t) = U_y(t + \Delta t) + (V_{py}(t) - U_y(t))e^{-\Delta t/\tau_p} - \tau_p \frac{dU_y}{dt}(1 - e^{-\Delta t/\tau_p}) \quad (2.24)$$

The advantage of this particular formulation is that there is no restriction on the time step from the consideration of numerical stability; the value of Δt is only limited by the accuracy of the assumption that dU_x/dt is constant over the chosen time-step.

The convection of particles is then given by the differential equation

$$\frac{d\mathbf{r}_p}{dt} = \mathbf{V}_p \quad (2.25)$$

Different numerical schemes were examined to determine the accuracy and stability of the integration, and it was concluded that the best choice is a three step backward Euler explicit scheme, which is second-order accurate. Therefore, Eq. 2.25 can be written in two dimensions as

$$x_p(t + \Delta t) = \frac{1}{11} (18x_p(t) - 9x_p(t - \Delta t) + 2x_p(t - 2\Delta t) + 6V_{px}(t)\Delta t) \quad (2.26)$$

$$y_p(t + \Delta t) = \frac{1}{11} (18y_p(t) - 9y_p(t - \Delta t) + 2y_p(t - 2\Delta t) + 6V_{py}(t)\Delta t) \quad (2.27)$$

2.6.2 Particle Bombardment on Bed

A relatively simple method has been used in the present case to represent bombardment effects when a particle impinges on the particle bed. The inelastic collision of the particle with the ground can be modeled using conservation of momentum applied along

the in-plane and wall-normal direction, which provide the rebound velocity of the particle after collision, i.e.,

$$V_{p,\text{par}} = e_{\text{par}} * V_{p,\text{par}} \quad (2.28)$$

$$V_{p,\text{norm}} = -e_{\text{norm}} * V_{p,\text{norm}} \quad (2.29)$$

where e_{par} and e_{norm} are the coefficients of restitution parallel and normal to the ground, respectively.

2.7 Summary

This chapter has discussed the different particle clustering strategies that were employed in the present study. The Gaussian method, k-means and the Osipov's method have been described in detail. The mathematical framework behind each of the three algorithms has also been explained with the differences between the methods. These particle clustering algorithms were applied to a two-dimensional prototypical flows that mimic brownout type flows and their relative advantages were studied. The prototypical flows have been explained, with the equations for particle convection and the assumptions made in the present study.

Chapter 3

Results and Discussion

3.1 Introduction

In this chapter, the three previously described methods of clustering (i.e., Gaussian, k-means and the Osipov's method) were evaluated from a numerical perspective when applied to the prototypical flows. The Gaussian distribution method retains the same particles throughout the calculation, although their positions change dynamically with respect to the cluster center, whereas the k-means method is based on repeated declustering and reclustering of particle groups. Osipov's method is based on integrating the particle fluid-element concentration along pathlines. The regions of applicability of each method have also been identified in detail. The relative advantages of the algorithms have been analyzed and from the studies conducted, and the best use of these methods to the problem of brownout is proposed. The computational benefits and the errors incurred from each of the methods has also been studied and quantified.

Metrics required to compare the “closeness” or accuracy between two brownout clouds are still in its initial stages, as discussed in Chapter 1. Some work has been done in this field by Tritschler et al. [44] through the use of the Modulation Transfer Function (MTF). However, to assess the validity of the clustering algorithms, both in terms of accuracy and computational time-gain, the solution obtained through the use of clustering algorithms must be compared against a known exact solution. In the current study, the

“exact solution” is referred to the solution that was obtained when all of the particles were tracked directly without the use of any clustering algorithms (i.e., a “brute-force” application of the particle tracking algorithm). The solutions obtained using the clustering algorithms were compared against the exact solution to assess its accuracy.

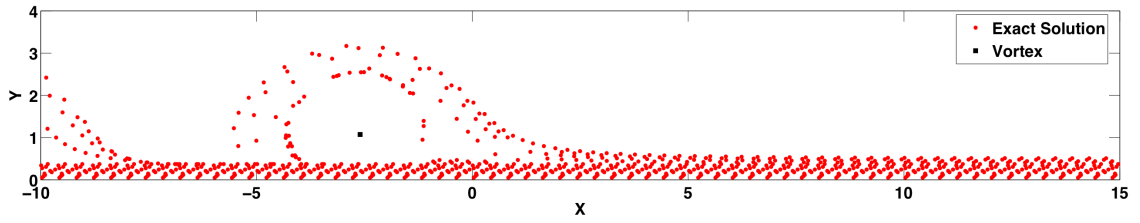
For the prototypical flow problem, the equations of motion of the particle were modified appropriately to include only the two relevant dimensions (i.e., parallel to the ground and normal to the ground). Hence, in the Gaussian formulation the mean vector μ_c is an array with two entries and the covariance matrix Σ_c being a 2×2 matrix written as

$$\mu_c = \begin{bmatrix} V_{px} & V_{py} \end{bmatrix} \quad (3.1)$$

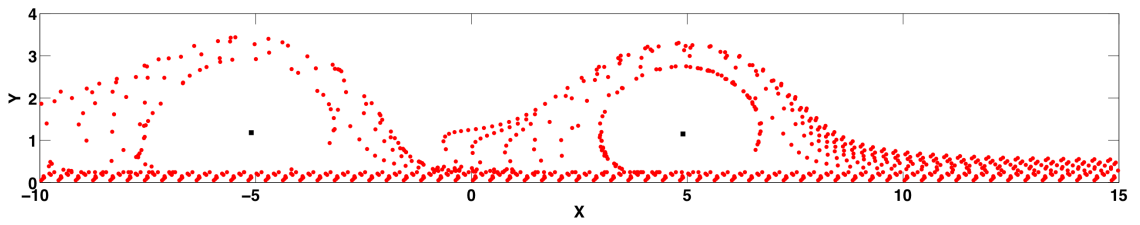
and

$$\Sigma_c = \frac{1}{E_c} \begin{bmatrix} k_1 |V_{px}|^2 & k_2 |V_{px}| |V_{py}| \\ k_2 |V_{py}| |V_{px}| & k_1 |V_{py}|^2 \end{bmatrix} \quad (3.2)$$

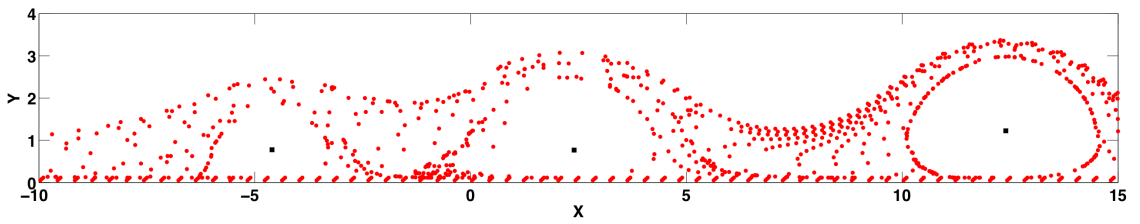
Figure 3.1 shows the positions of the particles and the vortices in the flow as they pass over the initial saltation layer at three different time steps. An initial saltation layer of 15 layers each consisting of 75 particles was chosen, i.e., a total of 1,125 particles. The particles in successive layers are made “active” as the particles directly above are mobilized and entrained into the flow. Although a simple two-dimensional flow field was chosen, it is important that this flow field has the same characteristic features of a rotor flow and its associated sediment uplift mechanisms. The results show the characteristic particle uplift mechanisms (discussed in the introductory chapter and seen in experiments) such as entrainment by the vortical flow, secondary vortex trapping, and particle settling under the action of gravity.



(a) Number of time steps = 75



(b) Number of time steps = 150



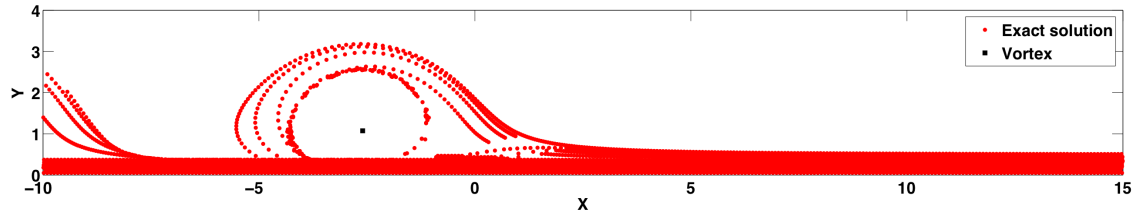
(c) Number of time steps = 225

Figure 3.1: Exact solution. Initial layer = 15, particles per layer = 75.

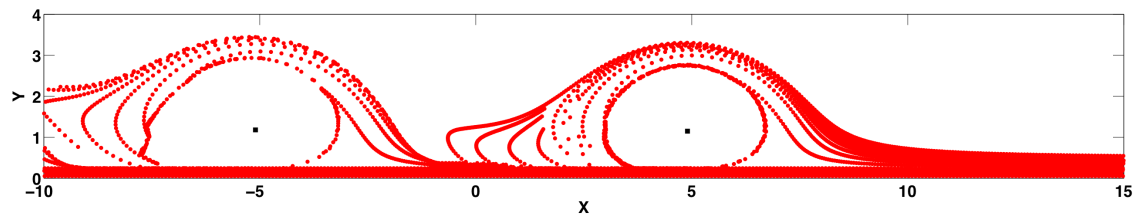
3.2 Gaussian Method

The Gaussian method is the first of the three clustering algorithms studied in the current work. This algorithm redistributes the clustered particles at every time-step based on currently existing information. This method was tested using two different approaches. First, this method was used to explore the capability of reconstruction a particle field. In this case, the Gaussian method was used to increase the number of particles in a “degraded” simulation (a case with fewer particles in the simulation) so as to match the total number of particles with a simulation performed using the exact method. The ability to accurately reconstruct the solution with fewer particles would result in computational time being gained using this method. Secondly, this method uses existing information to further bring out finer details in the particle field by increasing the total number of particles beyond the baseline solution.

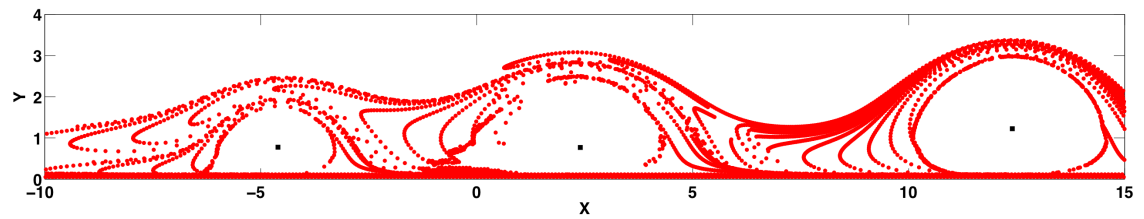
Figure 3.2 shows an exact solution with 15 layers of particles, each layer now consisting of $75 \times 5 = 375$ particles, such that the total number of particles have increased to 5,625. The particle trajectories can be seen to be very similar, except that the particle field is now more densely populated. It can be seen that the particle count is changed by altering the number of particles in each layer and not by altering the number of layers. The latter is not a valid comparison because of the nature of the problem. The particles in the layer below the active layer are made active when the active layer is depleted. By changing the number of layers rather than particles per layer, the mass injection rate into the flow is altered by sediment of the bed. However, by increasing the number of particles per layer, the particle density is altered keeping the mass injection rate constant.



(a) Number of time steps = 75



(b) Number of time steps = 150



(c) Number of time steps = 225

Figure 3.2: Exact solution. Initial layers = 15, particles per layer = 375.

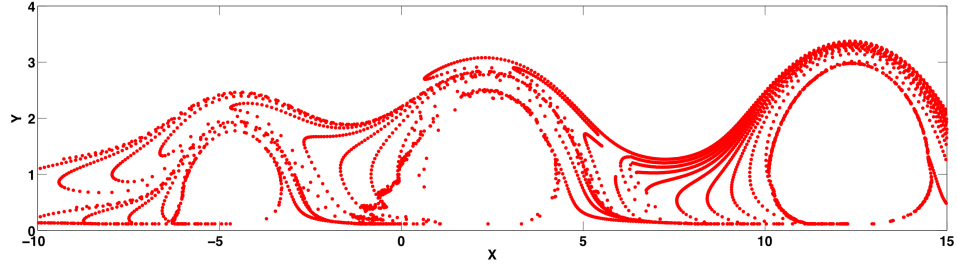
3.2.1 Effect of Number of Particles per Cluster

In this section, the effect of changing the number of particles per cluster are studied along with their effect on accuracy of the solution. This study helps to determine the optimum number of clusters that can be chosen for a given problem because a degraded solution is used to reconstruct the exact solution by using the Gaussian method. Results are presented for two cases: (a) 75 clusters per layer with each cluster representing 5 particles and, (b) 15 clusters per layer with each cluster representing 25 particles. The number of layers are kept constant at 15, the total number of particles in the simulation being kept constant at $15 \times 75 \times 5 = 15 \times 15 \times 25 = 5,625$.

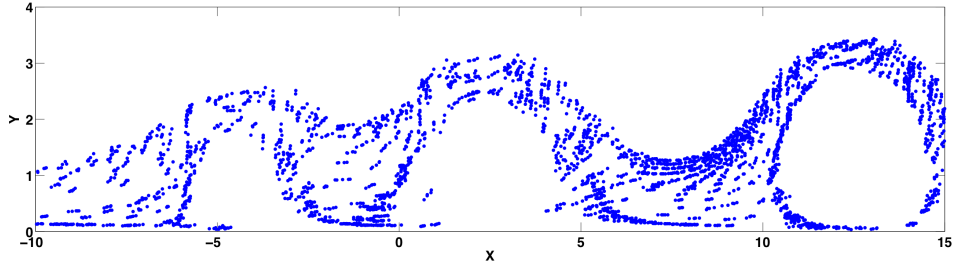
Figures 3.3(a) and 3.3(b) shows the particle positions for both the exact solution (red) and the clustered solution using the Gaussian method (blue). In all the results presented by this method, only the active and suspended particles are shown so as to not bias the information because a particle that is inactive will contribute nothing towards the solution. Consequently, the sediment bed is not shown.

The exact solution, as was shown in Fig. 3.2, contains 15 layers of 375 particles, whereas the Gaussian solution contains 15 layers of 75 particle clusters, with each cluster in this case representing 5 individual particles, i.e., 5,625 particles. This approach keeps the total number of particles between the exact solution and the Gaussian method the same, resulting in a valid comparison between the two solutions.

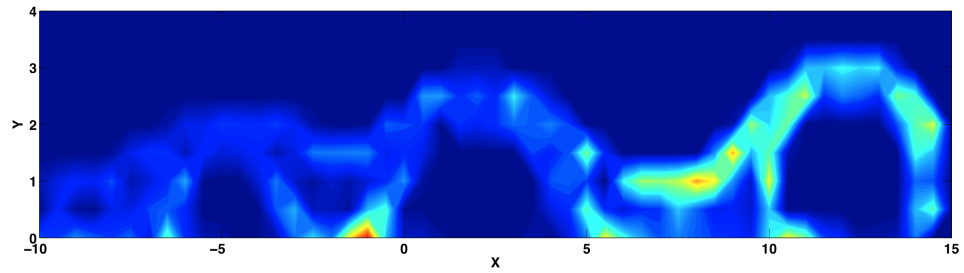
Figures 3.3(c) and 3.3(d) show the density plots corresponding to Figs. 3.3(a) and 3.3(b), respectively. Each “bin” in this case is 0.5×0.5 mesh units within which the particles are counted. Figures 3.3(c) and 3.3(d) show the comparison of the solution where



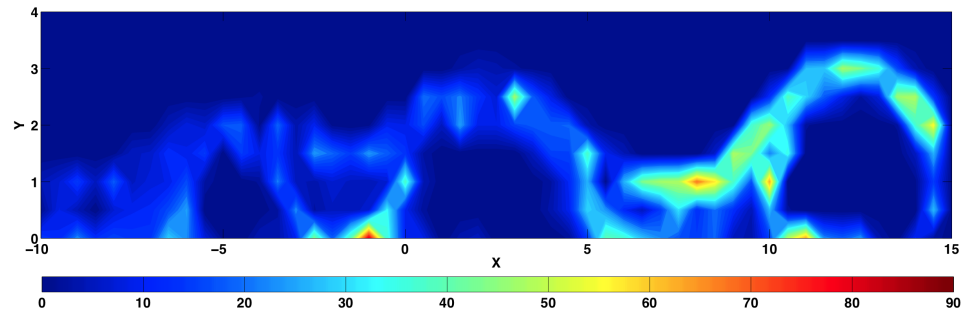
(a) Particle positions - Exact solution



(b) Particle positions - Clustered solution: Gaussian

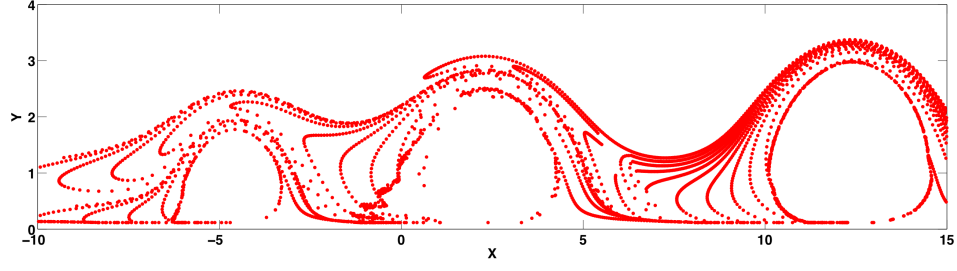


(c) Density contour plot - Exact solution

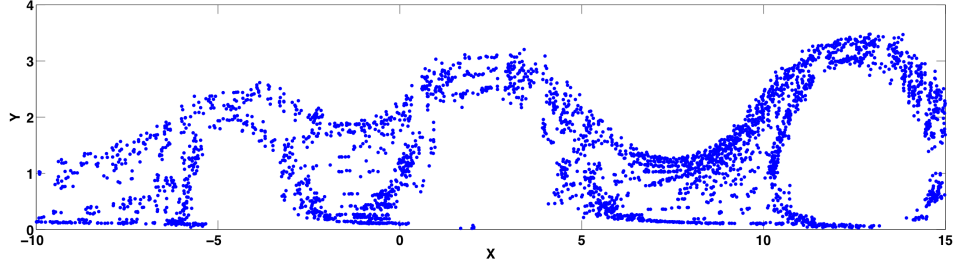


(d) Density contour plot - Clustered solution: Gaussian

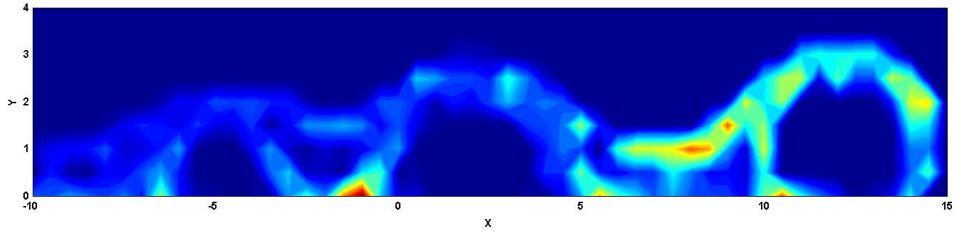
Figure 3.3: Gaussian solution. Initial layers = 15, clusters per layer = 75, particles per cluster = 5, $k_1 = 1$ and the corresponding density contour plots.



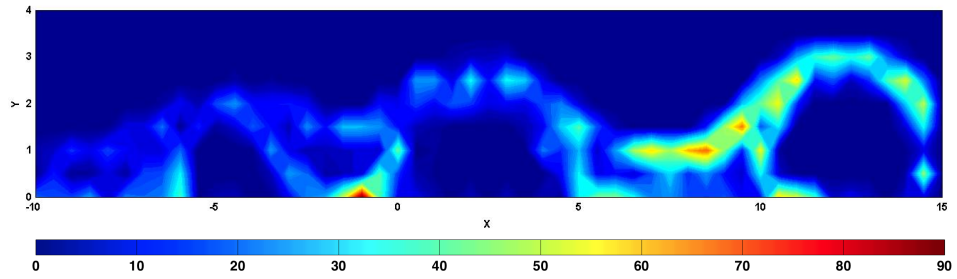
(a) Particle positions - Exact solution



(b) Particle positions - Clustered solution: Gaussian

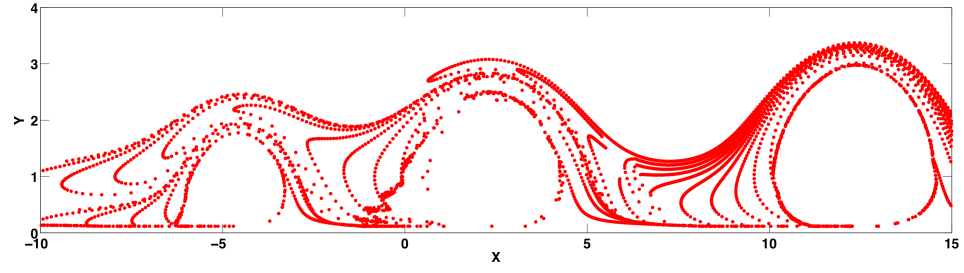


(c) Density contour plot - Exact solution

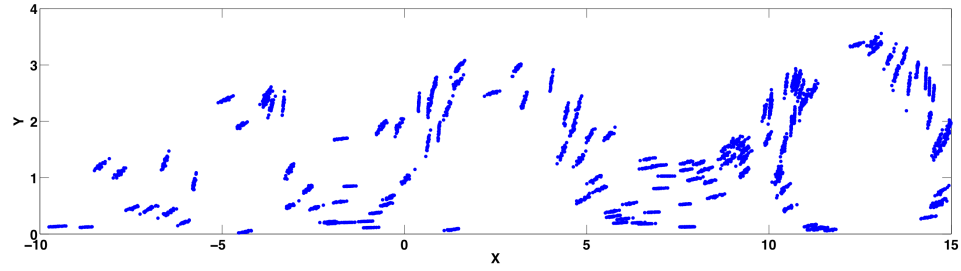


(d) Density contour plot - Clustered solution: Gaussian

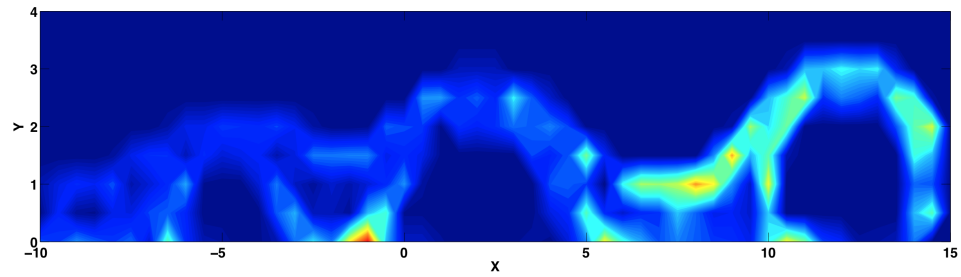
Figure 3.4: Gaussian solution. Initial layers = 15, clusters per layer = 75, particles per cluster = 5, $k_1 = 2$ and the corresponding density contour plots.



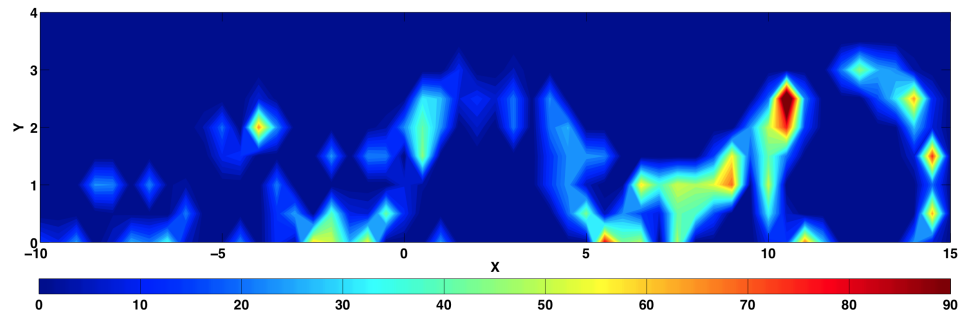
(a) Particle positions - Exact solution



(b) Particle positions - Clustered solution: Gaussian

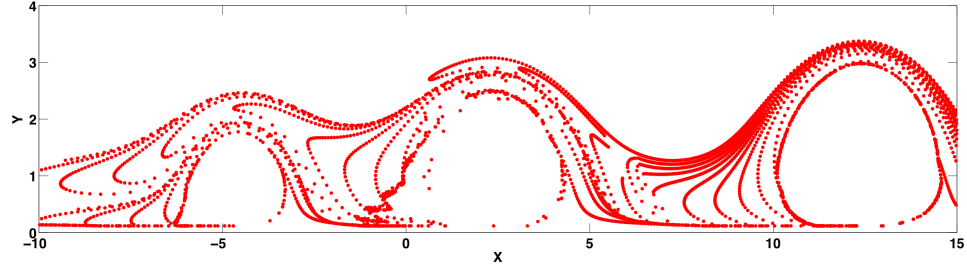


(c) Density contour plot - Exact solution

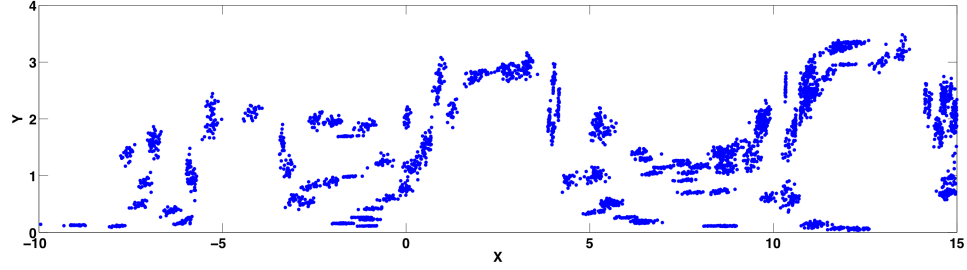


(d) Density contour plot - Clustered solution: Gaussian

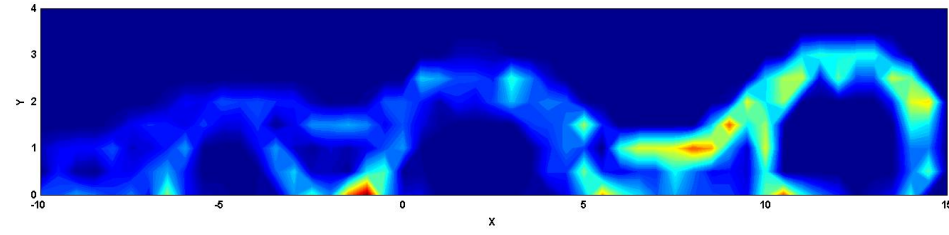
Figure 3.5: Gaussian solution. Initial layers = 15, clusters per layer = 15, particles per cluster = 25, $k_1 = 1$ and the corresponding density contour plots.



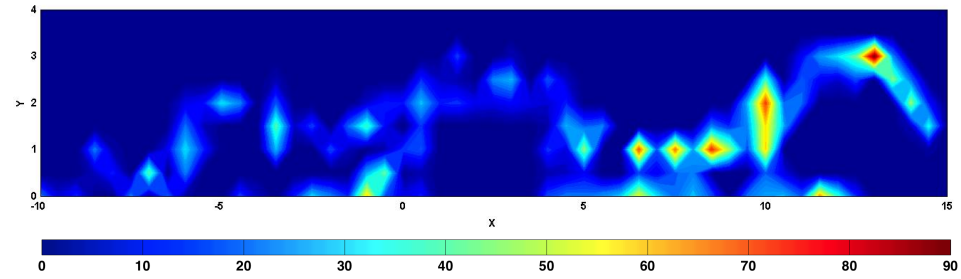
(a) Particle positions - Exact solution



(b) Particle positions - Clustered solution: Gaussian



(c) Density contour plot - Exact solution



(d) Density contour plot - Clustered solution: Gaussian

Figure 3.6: Gaussian solution. Initial layers = 15, clusters per layer = 15, particles per cluster = 25, $k_1 = 2$ and the corresponding density contour plots.

each cluster in the Gaussian solution represents 5 particles. Although the exact position of the particles may not match between the exact and the clustered solution, good overall agreement is obtained with the two density plots, indicating that the clustering solution is acceptable within a certain tolerance. The computational time gained with this method is five times that of the exact solution, with only a minimal overhead cost being required by the Gaussian distribution process.

Figures 3.4, 3.5 and 3.6 are arranged in the same manner as Fig. 3.3. Figure 3.4 differs from Fig. 3.3 in that the principal spread parameter $k_1 = 1$ in the previous and $k_1 = 2$ in the latter. As defined previously, k_1 controls the amount of spread from the mean of the distribution, i.e., the higher the value of k_1 the higher is the spread of the particles from the cluster center. It should be noted that k_2 is also dependent on k_1 because of the requirement that the matrix be positive semi-definite. The results are qualitatively similar for the particle positions and the density plots are in good agreement with each other, implying that the solution is within accepted tolerances.

Figures 3.5(a) and 3.5(b) show the exact solution and the Gaussian clustered solution as shown before, except that now the Gaussian clustered solution contains 15 layers of 15 particle clusters, with each cluster representing 25 particles; the total number of particles was the same (5,625). The particle positions computed by the clustering algorithms revealed concentrated areas of particle concentration with the density plot in this case showing significant deviations from the exact solution, seen in the difference between Figs. 3.5(c) and 3.5(d).

Figure 3.6 shows the result for 15 layers of 15 particle clusters, with each cluster representing 25 particles with $k_1 = 2$. Because the principal spread parameter is higher

in this case and the number of particles being represented by each cluster is higher, the resulting solution and the corresponding density plots show more deviations from the exact solution. This qualitative observation is supported in the next section with a rigorous mathematical study.

The efficiency of this method depends primarily upon the number of actual computed clusters because there is no additional overhead from reclustering as the solution proceeds. Clearly, if a sufficiently large enough number of clusters used, then the clustered particles generally remain a good approximation to the full solution when using an equivalent number of total particles.

3.2.2 Error Estimation in Two-Dimensions

The previous section dealt with assessing the accuracy of the obtained Gaussian clustered solutions by qualitatively comparing the density plots with the exact solution. In this section, Pearson's coefficient (discussed in the methodology section) is used as a metric to assess the accuracy of the Gaussian method applied to a two-dimensional prototypical flow. The density plots to be compared (i.e., the exact solution and the clustered solution) are treated as two matrices and Pearson's coefficient helps assess the correlation or "closeness" between the two matrices.

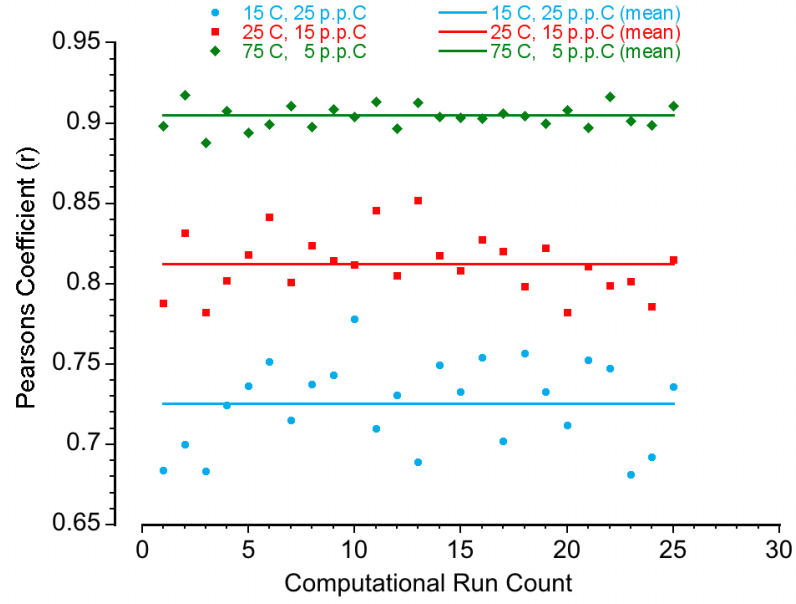
Because the Gaussian clustering algorithm is a statistical process, the results from the Gaussian method have to be compared with the exact solution over multiple runs so as to obtain a statistical average that is unbiased by the statistical nature of the algorithm. To obtain satisfactory results, each simulation was run 25 times and the result of each run

was used to obtain an averaged Pearson's coefficient.

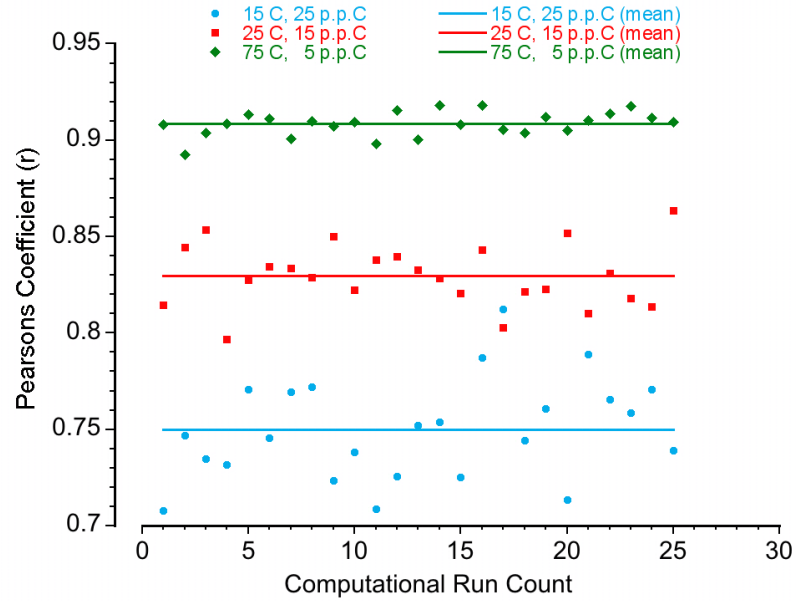
Figure 3.7(a) shows the variation of Pearson's coefficient for different computational runs for three different cases: (a) 75 clusters with 5 particles per cluster, (green) (b) 25 clusters with 15 particles per cluster (red) and, (c) 15 clusters with 25 particles per cluster (blue). In Figs. 3.7(a) and 3.7(b) 'C' denotes clusters and 'p.p.C' denotes particles per cluster. Figure 3.7(a) shows the results when the principal spread parameter $k_1 = 1$ and Fig. 3.7(b) for $k_1 = 2$. The scatter data points are the values of the Pearson's coefficient and the solid line represents the mean value.

Because Pearson's coefficient is a global metric and not sensitive to local differences, care should be taken to limit the size of the density plots to the size of the solutions as the sparseness of the matrices outside the solution limits can bias the metric. As defined earlier, there is higher correlation between two density maps as the value of Pearson's coefficient (r) approaches 1.

Figures 3.7(a) and 3.7(b) confirm mathematically what was visually observed earlier. The case with 75 clusters representing 5 particles per cluster shows the highest correlation to the exact solution, with a mean value of r above 0.9 when both $k_1 = 1$ and $k_1 = 2$. These results correspond to Figs. 3.3(d) and 3.4(d), respectively. When 15 clusters per layer with 25 particle per cluster was chosen, a poor correlation was observed between the respective density plots (Figs. 3.5(d) and 3.6(d)) which correspond to low values of r (0.75 approx.). Although the ability of the Gaussian method to reconstruct a solution has been studied, the optimum number of clusters and the value of k_1 that will result in the lowest error is highly problem dependant and is tied both to the flow and to the particle physics. From the studies conducted, it is recommended that the optimum



(a) Variation in Pearson's coefficient for $k = 1$



(b) Variation in Pearson's coefficient for $k = 2$

Figure 3.7: Error estimation using Pearson's coefficient for Gaussian solution for both $k_1 = 1$ and $k_1 = 2$.

number of clusters be around 40%–50% of the total number of particles. However, this requirement results in an immediate reduction in computational time by about 60%.

3.2.3 Gaussian Method: Application to Three-Dimensional Problem

The Gaussian method was also applied to the three-dimensional helicopter brownout problem. The methodology and formulation of the particle tracking algorithm has been detailed in Ref. 6. The aerodynamic flow field below the rotor was modeled using a Lagrangian free-vortex wake method.

The results in Fig. 3.8 show a comparison of the solution obtained with 10^6 cluster centers (denoted by the black dots in Fig. 3.8(a)) and one using the distributed particle clusters in Fig. 3.8(b) with 100 particles per cluster, i.e., using a total of 100 million particles. The results clearly show that the clustered solution brings out the finer, structured details of the dust cloud, which is exactly the purpose of clustering.

Figure 3.8(c) shows the density of the dust cloud that is derived from the clustered solution. Figure 3.8(d) shows some details of the cloud near the ground, the concentrations which can be correlated to the positions of the tip vortices. Overall, these results show that the Gaussian clustering method can be used to improve the fidelity of the dust cloud by bringing out its details without adding much to the overall computational cost. Another example of the overall dust cloud computed using the Gaussian method is shown in Fig. 3.11, where the Gaussian clustered is again seen to bring out the details of the cloud.

Figure 3.10(a) shows the dust cloud during a landing maneuver. At this particular

time-step the rotor wake impinges on the ground causing an uplift of particles from the sediment bed. The black dots shown in Fig. 3.10(a) are the actual simulated solution. Figure 3.10(b) is a detailed view of the front end of the developing dust cloud in Fig. 3.10(a). This sudden uplift of particle are characteristic of brownout cloud during landings and it is important that the Gaussian method be able to capture this phenomena accurately. Figure 3.10(c) shows the Gaussian clustered solution of Fig. 3.10(b), with each cluster representing 100 particles. The clustered solution preserves the structure of the cloud and brings out the details of the dust field.

Figures 3.11(a) and 3.11(b) show applications of the Gaussian clustering technique to a landing simulation (shown is the rendered solution of the cloud), the details of which are explained in [6]. The predicted dust cloud successfully predicts the behavior of the particles in the flow, which indicates that the Gaussian method can be used to improve the fidelity of the brownout dust cloud predictions.

Figures 3.12 and 3.13 show a further comparison of the Gaussian clustered solution with an exact solution. The brownout cloud was obtained for simulated hovering flight conditions. Figure 3.12(a) shows the 10^5 cluster centers obtained with the direct or brute-force approach, which are then clustered using 100 particles per cluster to obtain the solution shown in Fig. 3.12(b). This result is then compared with a direct calculation using 10^7 particles, as shown in Fig. 3.12(c). Figure ?? presents an isometric view of the dust cloud that was subsequently obtained; Fig. 3.13(a) shows the clustered solution and Fig. 3.13(b) is the direct solution.

Notice that the solution obtained using the Gaussian clustering method matches the exact solution very closely. This outcome further strengthens the claim that the Gaussian

method can be used to improve a given lower resolution solution, but only if the initial solution contains sufficient information or particles. For example, it is unrealistic to expect that a solution with only 10^2 cluster centers and 10^5 particles per cluster would ever approach the fidelity of the direct solution made using 10^7 total particles. A snapshot of the dust cloud obtained using only 10^2 cluster centers is shown in Fig. 3.14. To this end, the use of clustering is not a panacea and requires good judgement as well as common sense.

Bear in mind that the metric to assess the accuracy of the resulting clustered solution, such as the Pearson's coefficient, are useful within the two-dimensional framework. However, such metrics cannot be extrapolated to the three-dimensional problem with the same fidelity. Further work will be required to develop appropriate metrics that can be used to gauge the error in the clustered solution of the dust clouds in three-dimensions, especially those that might be used to compute the optical characteristics of the cloud. To this end, some specific approaches are recommended in Chapter 4.

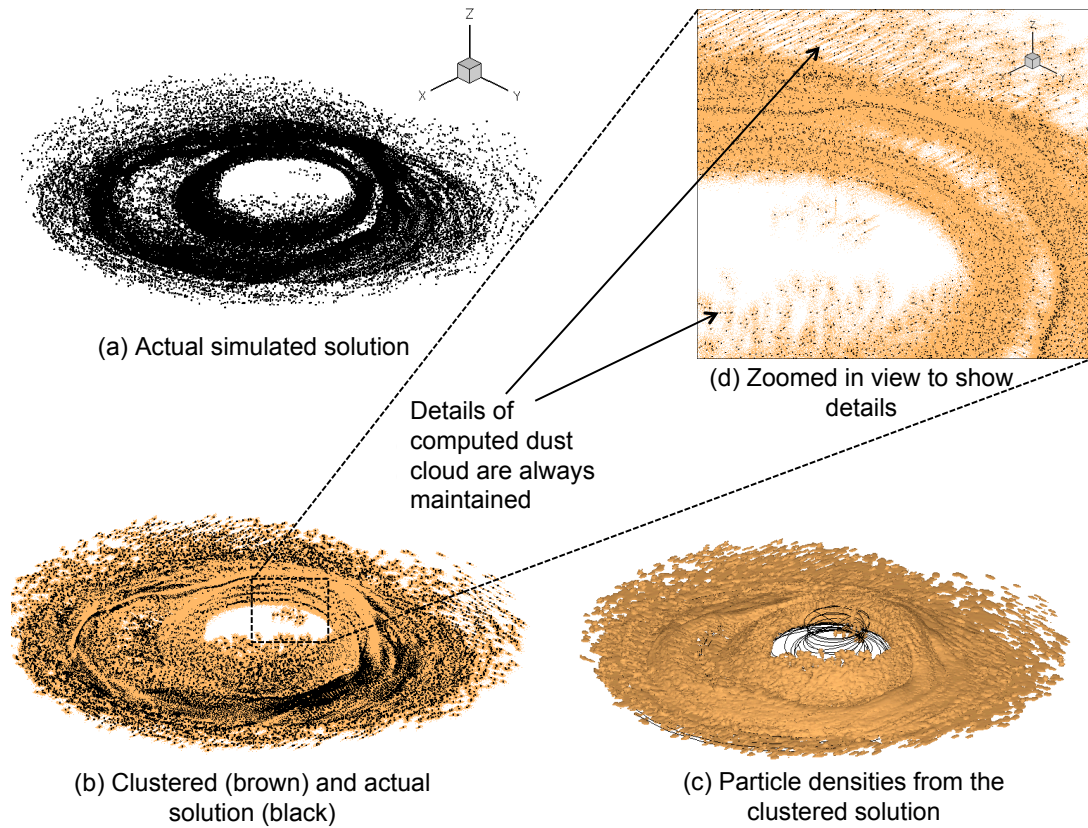
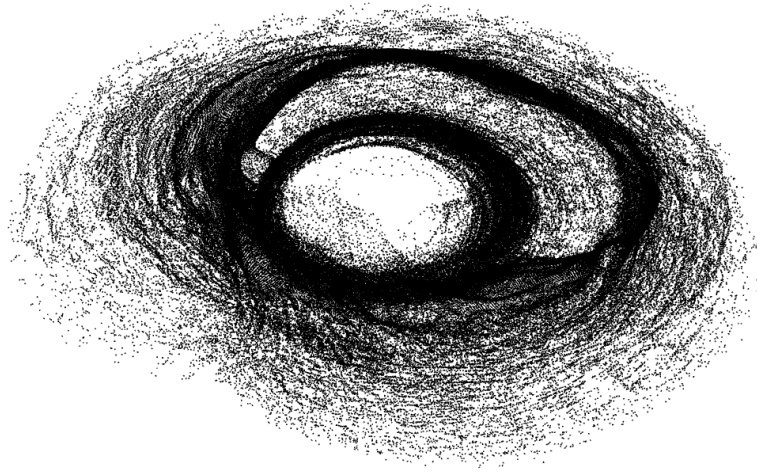
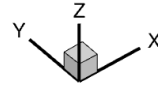
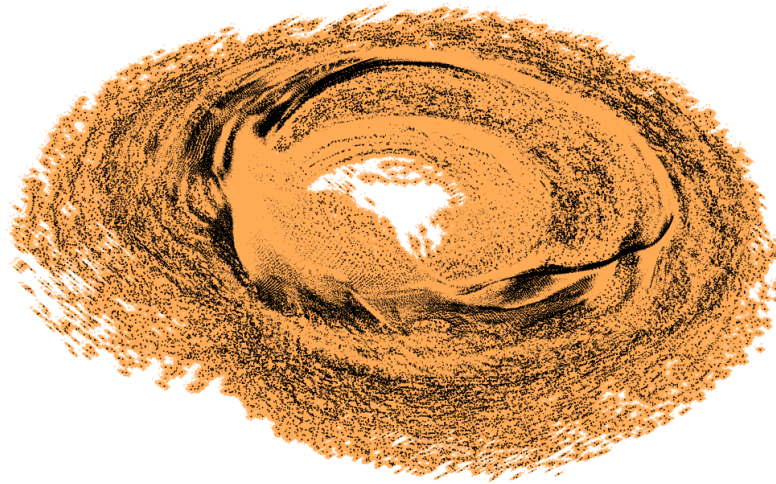
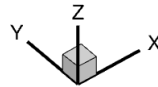


Figure 3.8: A solution obtained from brownout dust field computations when using the Gaussian clustering distribution method.

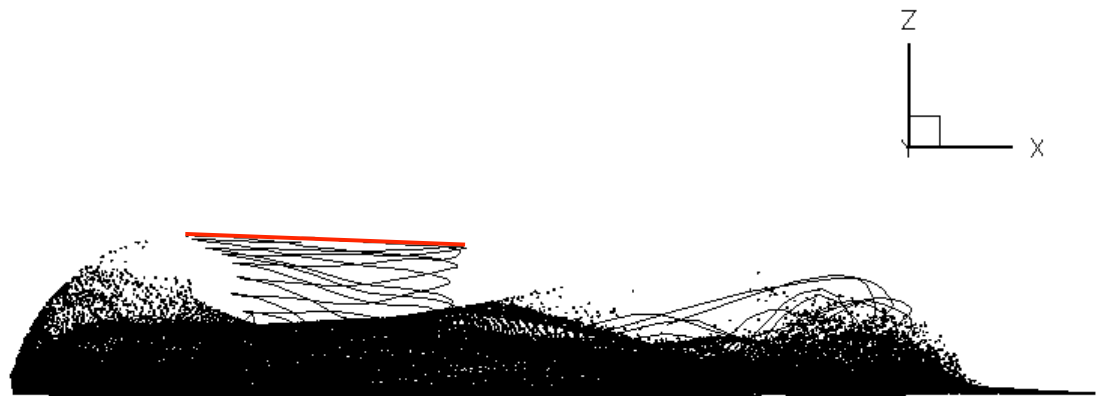


(a) Actual simulated solution

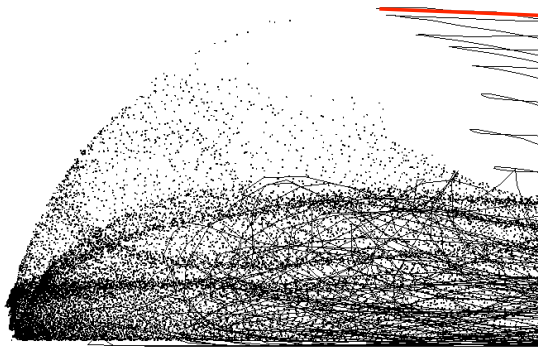


(b) Gaussian clustered solution

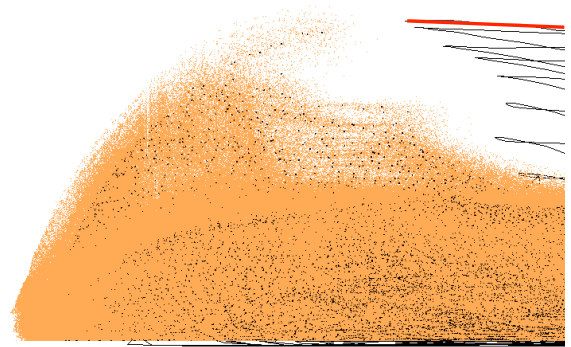
Figure 3.9: Another example of a brownout dust field computation using the Gaussian clustering distribution method.



(a) Actual simulated solution



(b) Zoomed in view of the actual solution



(c) Zoomed in view of the Gaussian clustered solution

Figure 3.10: Actual and clustered solution during a landing maneuver using the Gaussian clustering distribution method.

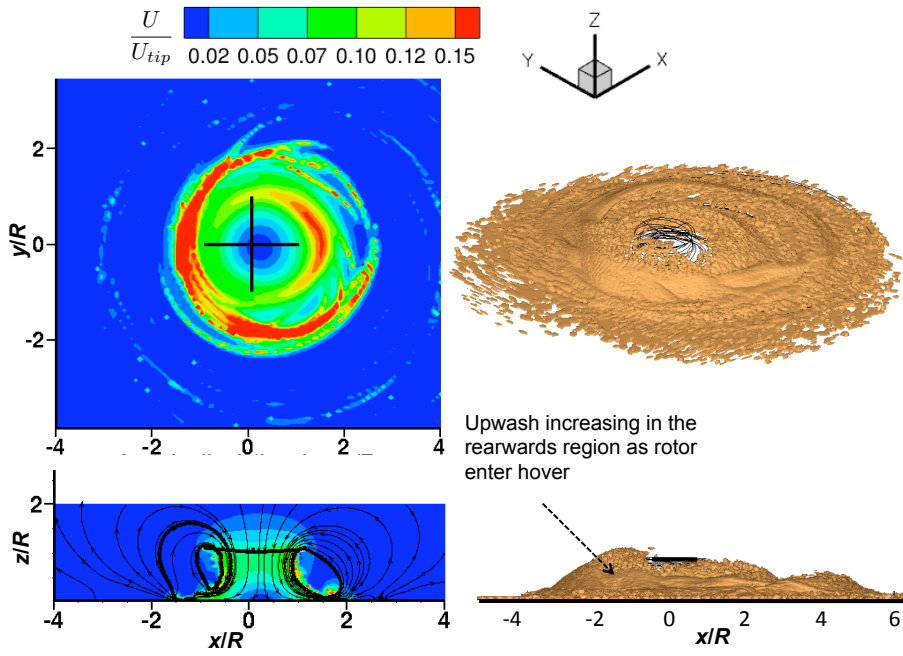
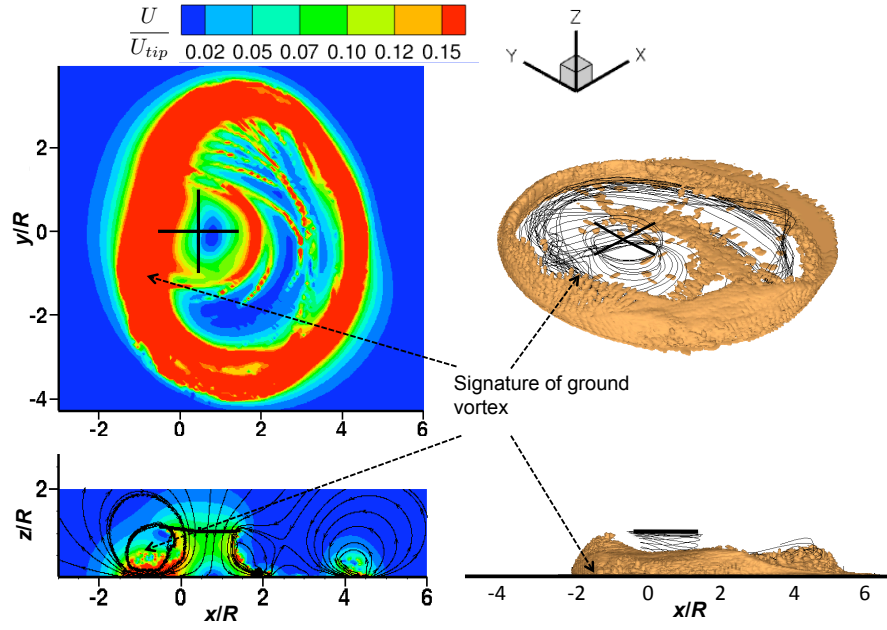


Figure 3.11: Development of the dust cloud during the landing approach at time $t = 3$ and 6 seconds.



(a) Actual simulated solution with 10^5 particles

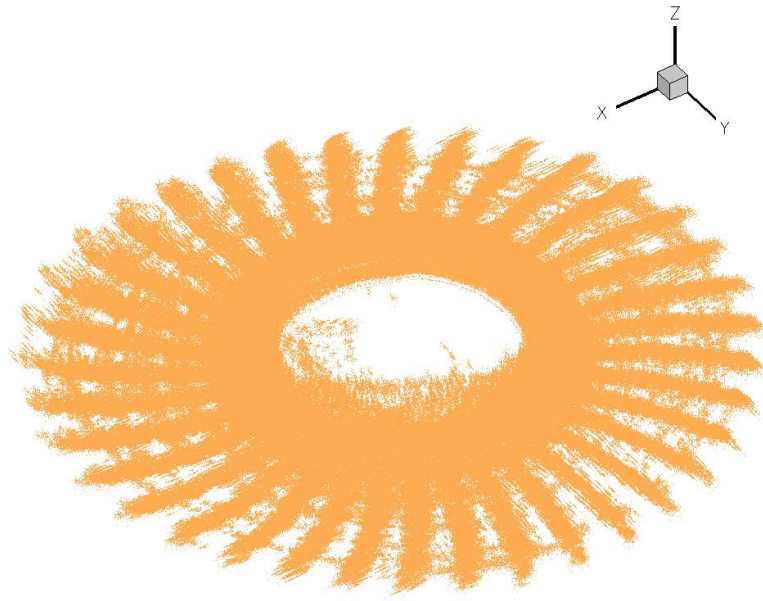


(b) Clustered solution with 10^7 particles

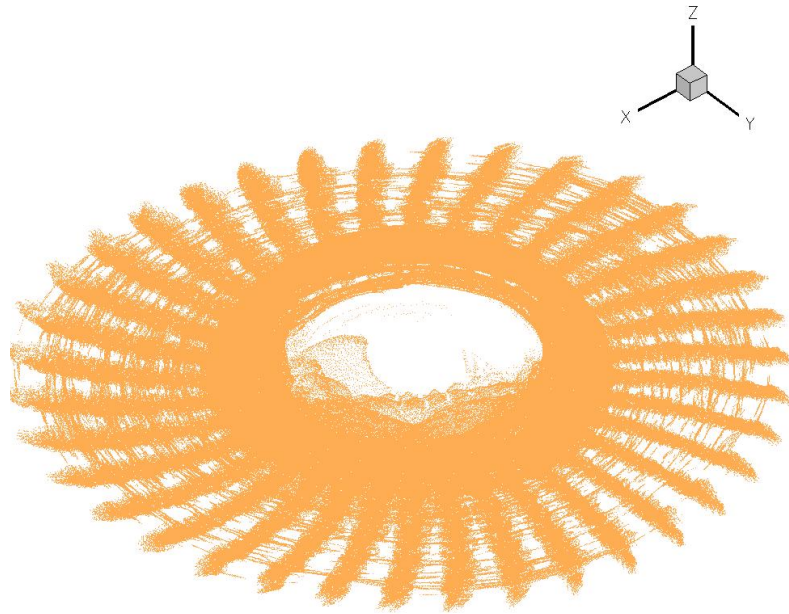


(c) Actual simulated solution with 10^7 particles

Figure 3.12: Comparison of the dust cloud between a brute-force calculation and Gaussian clustered solution. Lateral view.



(a) Clustered solution with 10^7 particles - 10^5 cluster centers

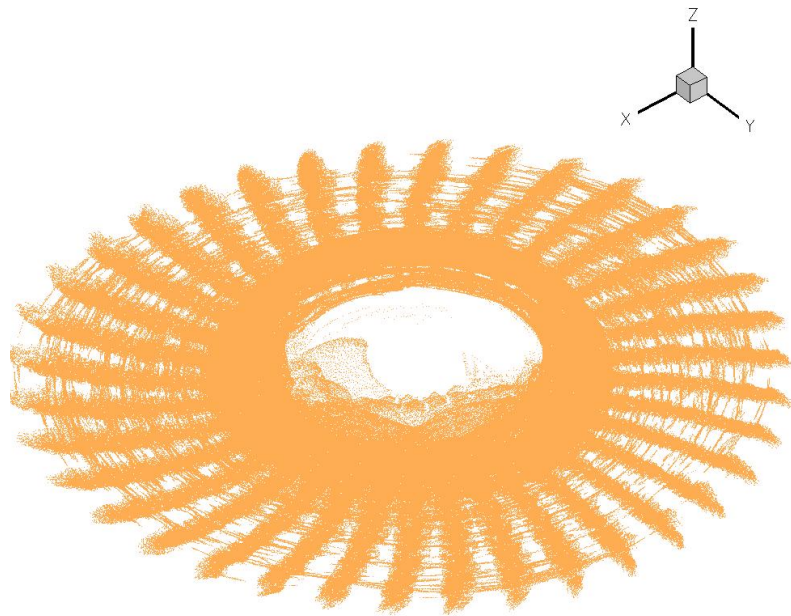


(b) Actual simulated solution with 10^7 particles

Figure 3.13: Comparison of the dust cloud between a direct or brute-force calculation and Gaussian clustered solution. Isometric view.



(a) Clustered solution with 10^7 particles - 10^2 cluster centers



(b) Actual simulated solution with 10^7 particles

Figure 3.14: Comparison of the dust cloud between a direct or brute-force calculation and Gaussian clustered solution. Isometric view.

3.3 k-means Method

3.3.1 Introduction

The k-means method is the second algorithm studied in this work as a potential candidate for the brownout particle tracking problem as a means of reducing computational costs. As was discussed in Chapter 2, the k-means method is based on grouping a certain number of particles into a cluster and then convecting this cluster as a group, thereby avoiding the computational cost of tracking each and every individual particle. In this case, if two clusters are in proximity to each other, then there is an option to decluster and recluster. In this section, results for the process of reclustering are shown and the implications on the accuracy of the particle locations is discussed. Both theoretical and computational arguments are used to support the various claims and ideas in this section. Figure 2.2, shown previously, outlines the approach used for the k-means approach.

3.3.2 Declustering and Reclustering

Let there exist two particle clusters at some intermediate time t whose centers are C_1 and C_2 , respectively. Assuming that the conditions for reclustering are satisfied, let three new clusters be formed whose centers are C_3 , C_4 and C_5 . Therefore, in the subsequent time step the clusters are convected under the forces F'_1 , F'_2 , and F'_3 acting on cluster C_3 , C_4 and C_5 , respectively. A schematic of this process is shown in Fig. 3.15.

Although intuitively reclustering appears to be a logical step when two or more clusters approach each other, the approach actually results in a degeneration of the so-

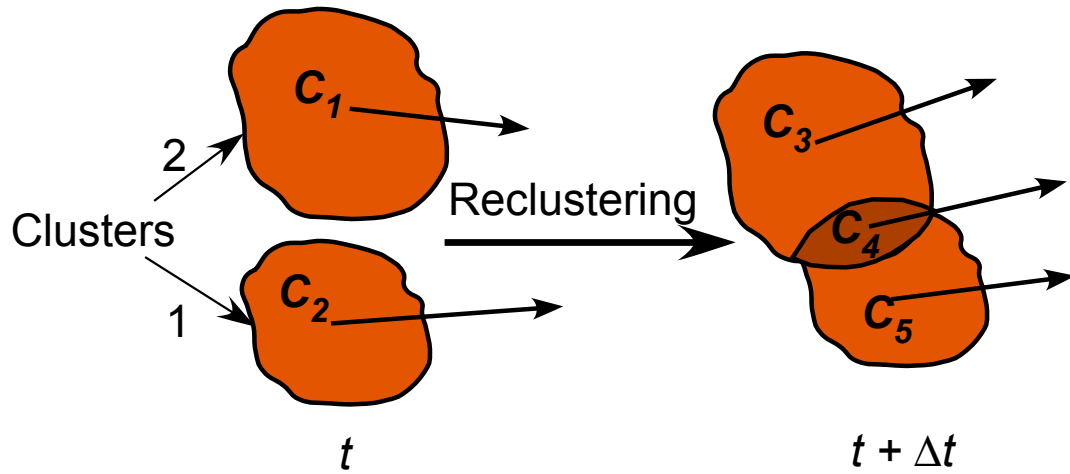


Figure 3.15: Schematic showing the process of reclustering and formation of new clusters.

lution, especially if there are high velocity gradients. The newly formed cluster centers do not possess any time-history from the previous time steps, which is an artifact of the k-means method; information about the velocities and positions at the previous time steps rest only with the original cluster centers C_1 and C_2 . If reclustering were to be carried out at every time-step, then the solution should use a self starting scheme such as Runge-Kutta or TRBDF2 (Trapezoidal Backward Difference schemes) to maintain a near second-order accurate solution. The use of a simple first-order Euler explicit scheme would result in a poor first-order accurate solution.

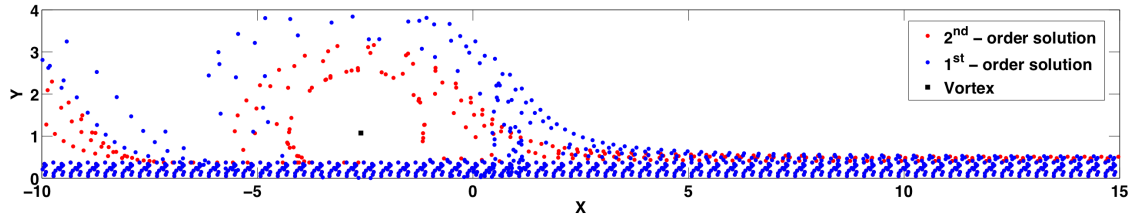
Figure 3.16 shows the difference in the solution in the prototypical flow chosen between a first-order and second-order accurate solution in time. The particles in blue represent the solution obtained using a first-order Euler explicit scheme, while the particles in red represent the second-order solution using the three step backward difference scheme. Repeated reclustering would also require the time step to be small to overcome

both the accuracy issue of a first-order scheme and the stability issues of an explicit formulation.

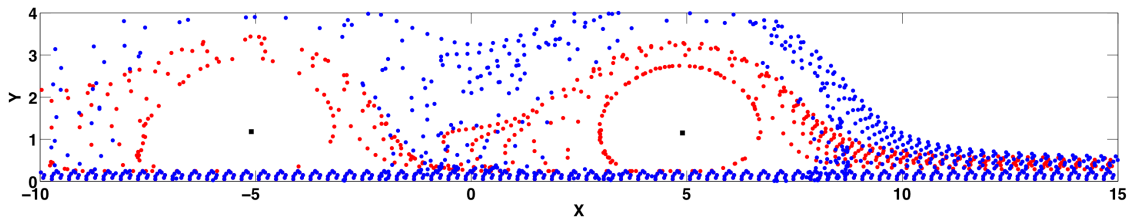
Reclustering poses another disadvantage from the accuracy point of view of the solution. Referring back to Fig. 3.15, the cluster centers C_1 and C_2 have become C_3 , C_4 and C_5 after reclustering. The newly formed cluster centers are formed as an artifact of the reclustering method and the clusters are tracked after reclustering based on the cluster centers. This approach however, does not guarantee the accuracy of the newly formed cluster centers. In other words, the scheme cannot always satisfy the condition that C_3 , C_4 and C_5 would possess the identical position and velocities of the particles if reclustering was not performed.

In the case of the prototypical flow, the top layer of dust particles is initially active and as a particular layer becomes depleted the subsequent layers become active. Because of the non-uniformity of the flow, particles in the subsequent layer are not activated in an orderly fashion. If declustering and reclustering were performed at each step, then all of the active particles can be grouped into clusters and convected. It has been shown that it is not particularly advantageous to perform the reclustering operation, which implies that the clusters that are formed should not be declustered to maintain the accuracy of the solution. Any new particles that become active from the depletion of the saltation layer or otherwise must now be grouped into new clusters. To preserve accuracy every particle must be its own cluster; in other words, to track them as if they were individual particles, which obviously defeats the purpose of clustering.

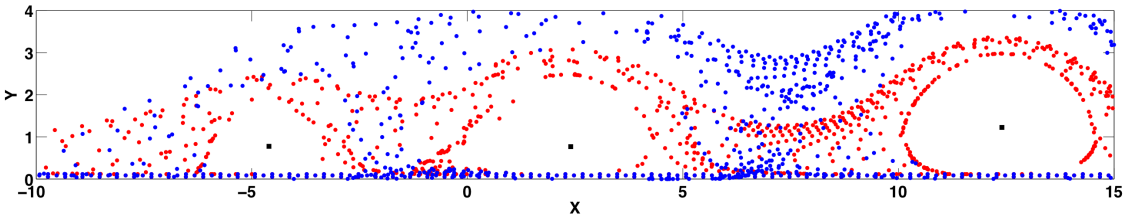
Because of the questions about the efficacy of the k-means method and its application to particles that become progressively active over time, a simpler flow field has been



(a) Number of time steps = 75



(b) Number of time steps = 150



(c) Number of time steps = 225

Figure 3.16: Difference in the solution between a first-order (blue) and second-order (red) accurate scheme.

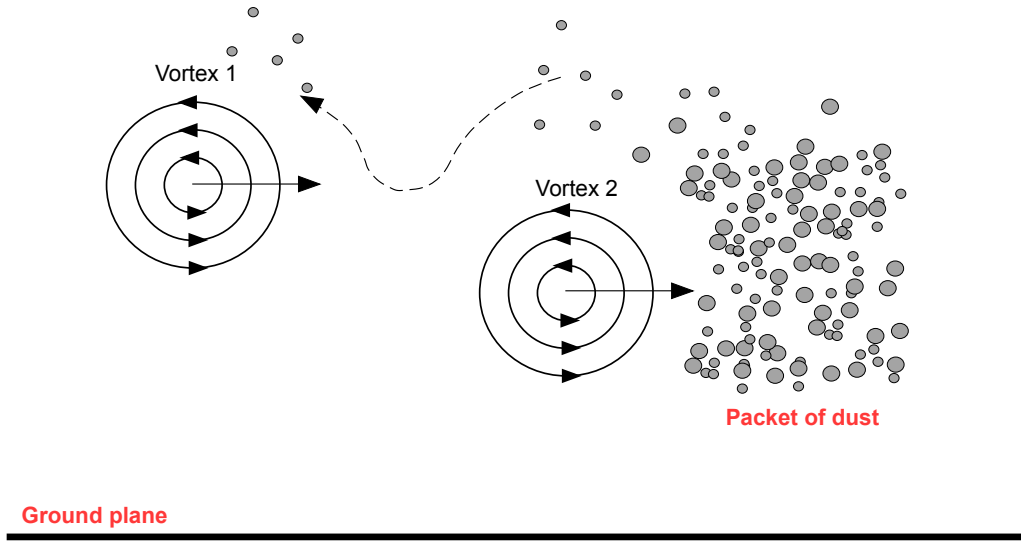


Figure 3.17: Schematic showing the simplified vortical flow field chosen to better illustrate the k-means clustering method.

chosen to understand the intricacies and the benefits of the k-means clustering algorithm. To this end, a pair of vortices that pass through a packet of dust was chosen as an example to further study the characteristics of k-means method, which is still a characteristic of the vortical flow nature of the brownout phenomenon. A schematic of this problem is shown in Fig. 3.17. Although the comparison of density plots has been chosen as the primary metric for error estimation, the rms error proves to be adequate for this simpler problem.

3.3.3 Computational Time Gain and Accuracy

Figure 3.18 shows the comparison between the costs to compute the clustered and exact dust cloud solution when using the k-means method. The abscissa shows the ratio of number of clusters, N_c , used to simulate the total number particles, N . The ordinate

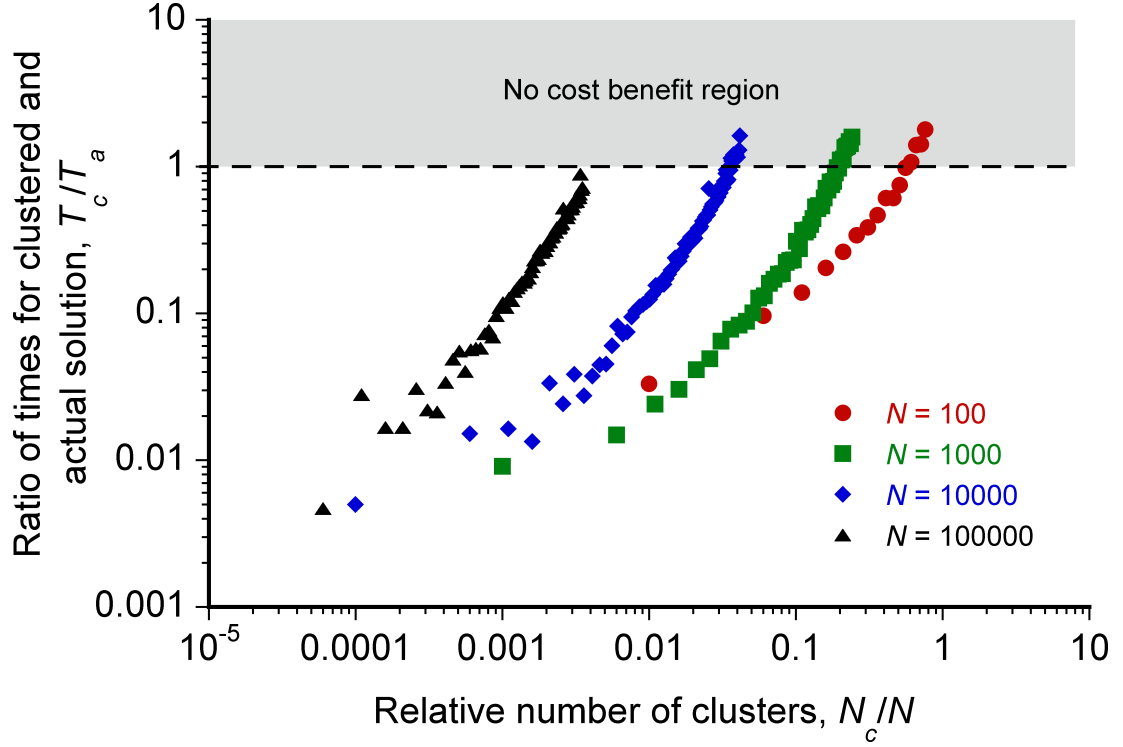


Figure 3.18: Comparison of CPU time for clustered and actual simulations for different number of particles using the k-means method.

shows the cost of the clustered solution, T_c , and the cost for the exact solution, T_a . The four lines show the benefits of clustering as a function of the number of particles in the simulation. The no cost benefit region is where $T_c/T_a > 1$, i.e., the clustered solution takes more time than the full particle solution, along with corresponding losses in accuracy.

Figure 3.19 shows the comparison between rms error of the clustered and the actual solution. For a given number of particles the computational time may be much smaller for the clustered solution over the exact solution, but the rms error is high. As the number of clusters are increased, the rms error decreases, which is to be expected. However, beyond a certain number of clusters for a given number of particles, the computational time enters the no cost benefit region, which happens when the overhead cost is comparable to the

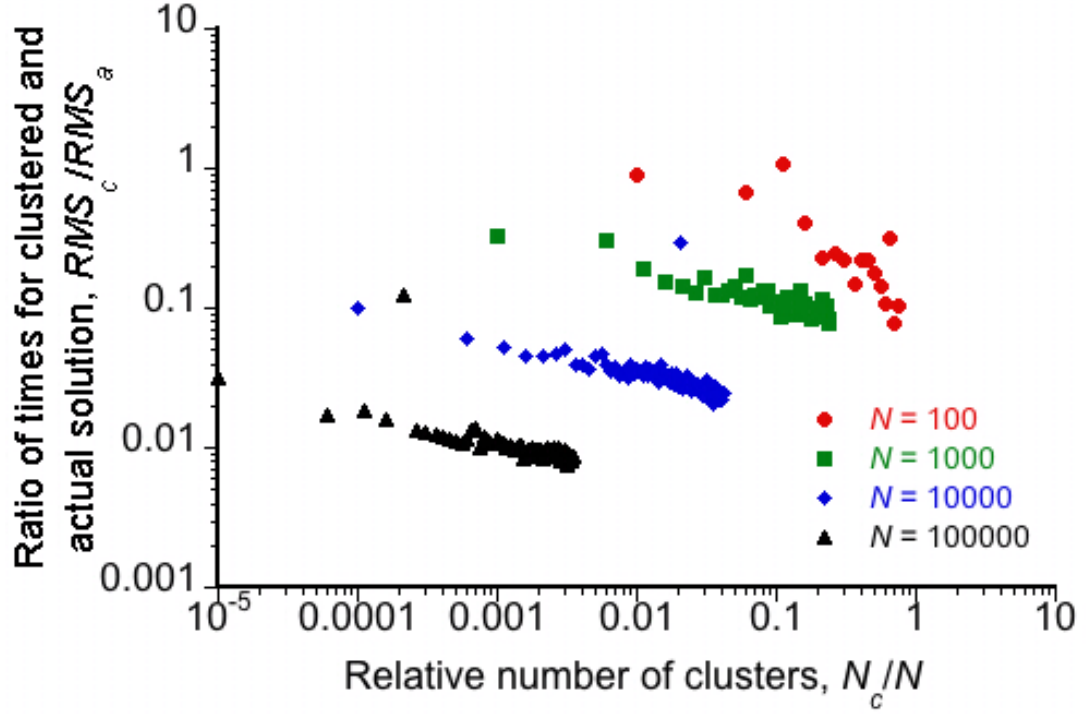


Figure 3.19: Comparison of RMS error for clustered and actual simulations for different number of particles using the k-means method.

actual simulation time. The exact values of the time gain and the accuracy are problem dependent, but the results shown in Figs. 3.18 and 3.19 reflect the general trend of the computational time and rms error. It can be deduced from Figs. 3.18 and 3.19 that if the method is applied to a region where fewer clusters would result in low error, the method would be highly beneficial in reducing computational time while still keeping the error to within tolerable limits.

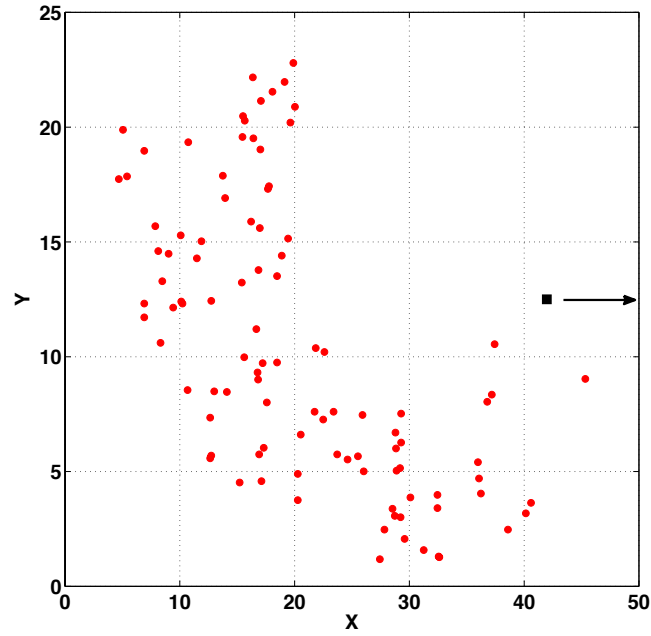
3.3.4 Effect of Velocity Gradients

The previous section discussed the general trends of computational time and the accuracy of the k-means method. It was observed that if fewer clusters would result

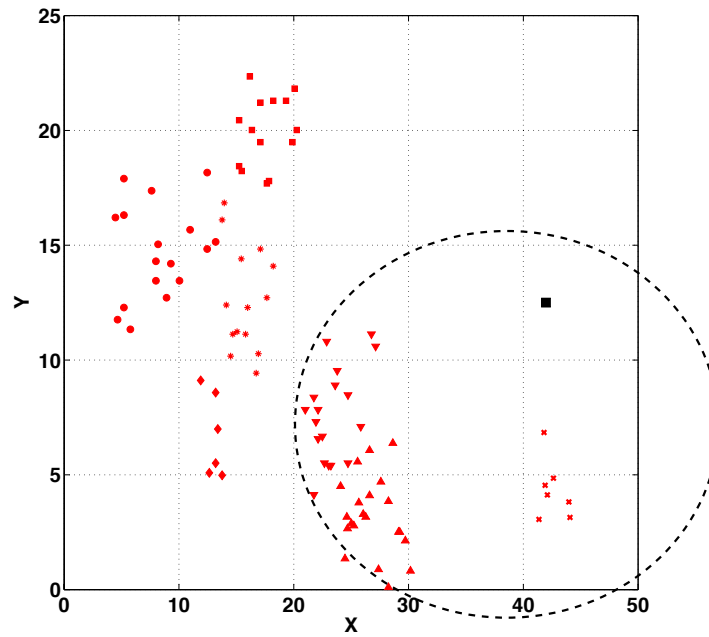
in lower error, this method is beneficial. In this section, the effect of the flow velocity gradients are discussed to identify potential regions in the flow where the k-means method could be applied.

Figure 3.20 shows the comparison in results between the exact and clustered solution using the k-means algorithm. For the purposes of illustration, the 7 clusters chosen were chosen to represent the 100 particles in this simulation. Clearly, the particle positions close to the vortex (i.e., inside the dotted circle) are grossly different from those of the exact solution, which is depicted in Fig. 3.20(a). This outcome suggests that a large number of particle clusters are required in regions of high velocity gradients (because of the presence of the vortex) to resolve the effects of the gradients accurately. However, in other regions this overhead can be avoided, which is where the usefulness of this method resides. Therefore, it may be beneficial to adopt this method of clustering in the far-field regions where the velocity gradients are generally more modest, and so the overheads associated with reclustering can be minimized there.

Figures 3.21, 3.23, 3.25 and 3.27 show the comparison of particle displacement between the exact and clustered solution for 10, 50, 100 and 200 clusters and corresponding density plots are shown in Figs. 3.22, 3.24, 3.26 and 3.28. The initial packet of dust particles contained 250 particles, and each figure shows the exact solution (red) at three different time steps through the solution along with the k-means clustered solution (blue) at the same time steps. In these cases, reclustering of the solution was not performed. Consequently, both the exact solution and the k-means clustered solution are second-order accurate in time. As was found with the previous results, a large number of clusters are required to resolve the velocity gradients present in the vicinity of a vor-



(a) Exact solution



(b) Clustered solution: k-means

Figure 3.20: Comparison of solution for actual and clustered solution using k-means method. The different symbols in (b) represent different clusters and the solid black square represents the position of the vortex.

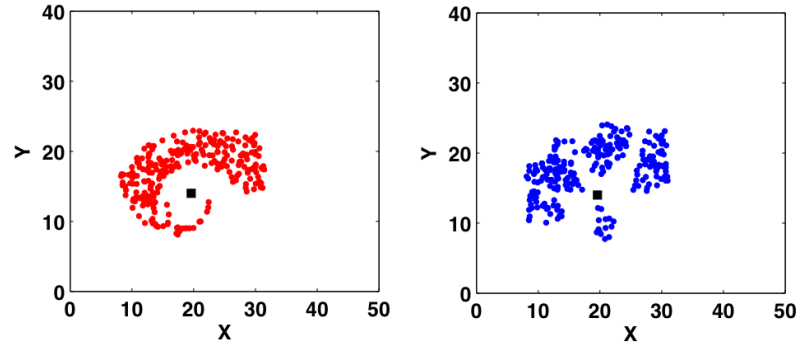
N_c/N	$T_{\text{clus}}/T_{\text{exact}}$
0.04	0.056570
0.20	0.246134
0.40	0.509545
0.60	0.780580
0.80	1.024650
1.00	1.282777

Table 3.1: Ratio of clustered time to exact time for the current problem as a fraction of total number of clusters to the number of particles.

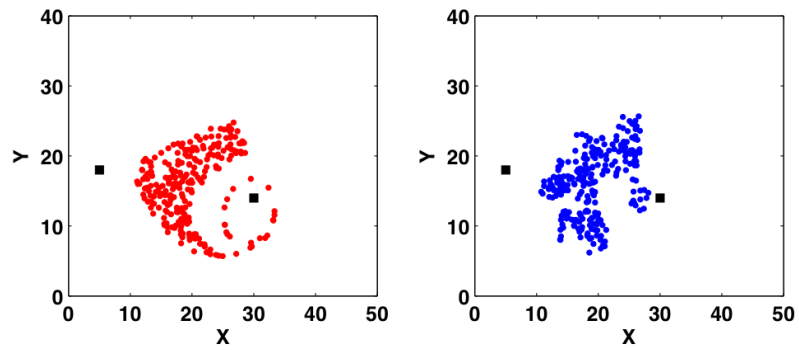
tex flow, which can be seen in the difference between Fig. 3.21(c) and Fig. 3.27(c). In regions of low velocity gradients, the solutions obtained by using only a few clusters are sufficiently accurate compared to the exact solution.

Table 3.1 gives the ratio of clustered time to exact time for this as a fraction of total number of clusters to number of particles. As the number of clusters is increased, the clustered solution becomes more accurate but the solution takes longer than for the exact solution. For $N_c/N > 0.80$ the solution becomes infeasible because of the overhead costs while also sacrificing some amount of accuracy.

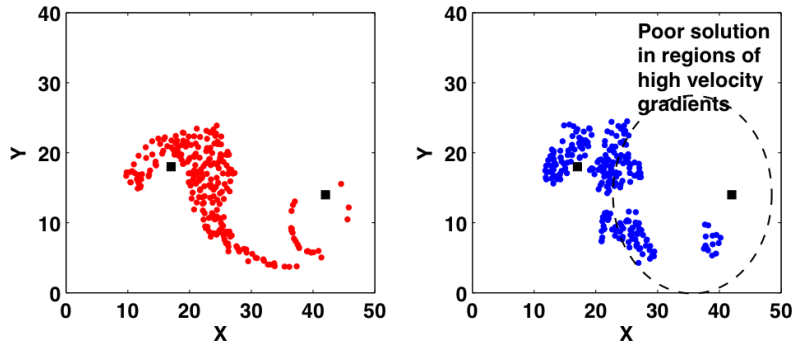
Based on the outcomes of the present two-dimensional study, the k-means method has the potential to significantly reduce the computational cost of the particle tracking simulation. The primary factor governing the time savings and the fidelity of the clustered solution is the number of clusters chosen for the k-means method. Studies conducted on the prototypical examples elucidate the need for a larger number of clusters in regions of



(a) Number of time steps = 60

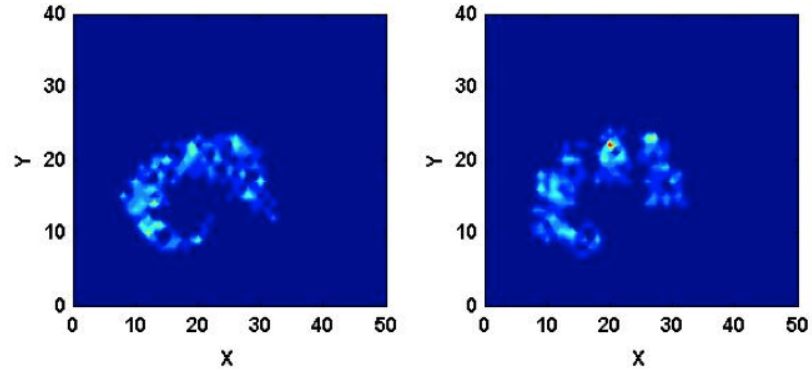


(b) Number of time steps = 125

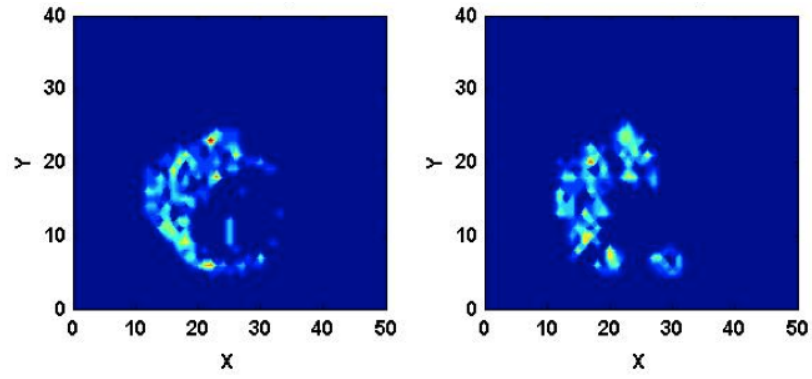


(c) Number of time steps = 200

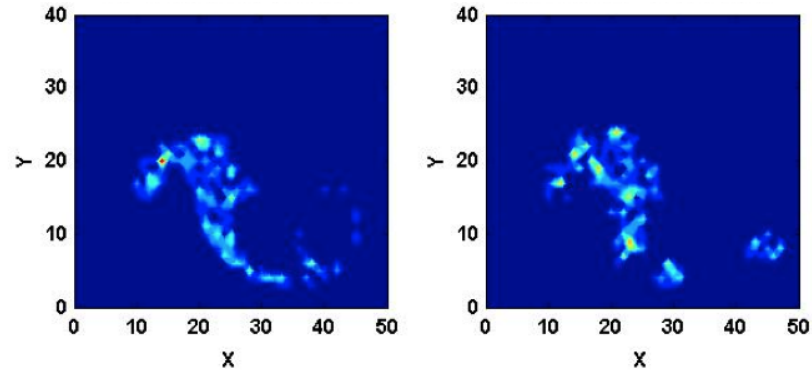
Figure 3.21: Comparison of particle displacement for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 10. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.



(a) Number of time steps = 60

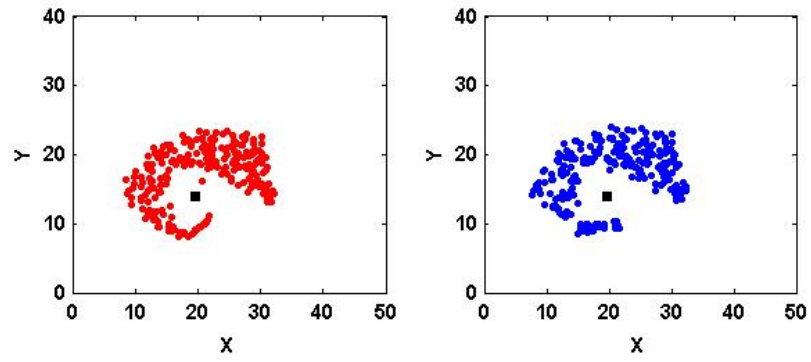


(b) Number of time steps = 125

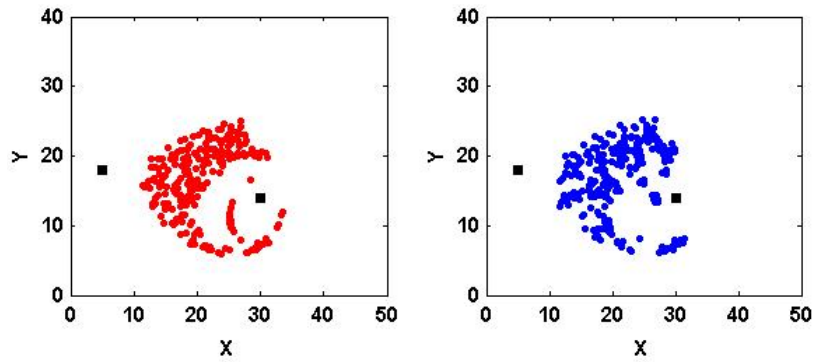


(c) Number of time steps = 200

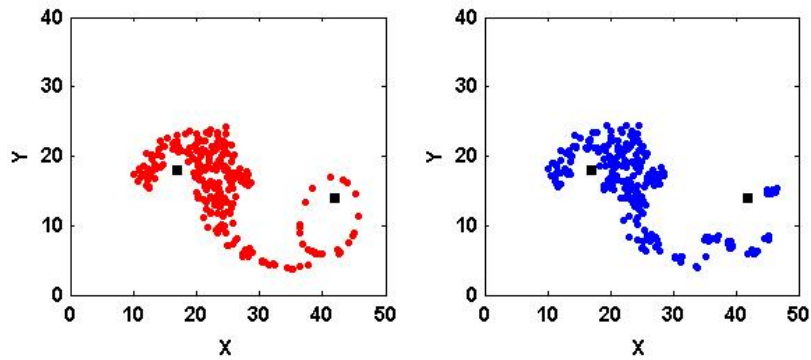
Figure 3.22: Comparison of density plots for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 10. Number of clusters = 10. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.



(a) Number of time steps = 60

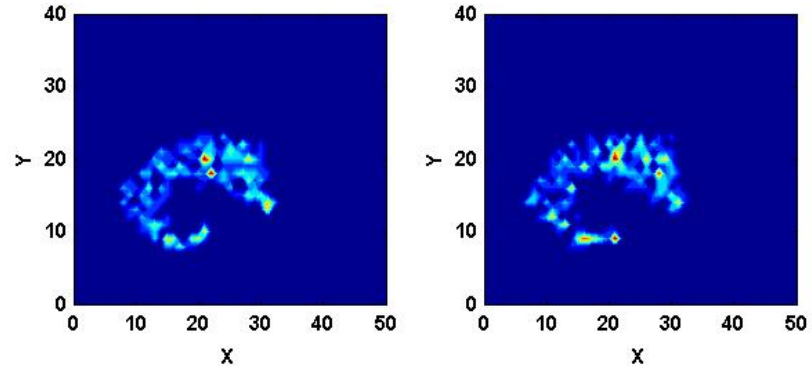


(b) Number of time steps = 125

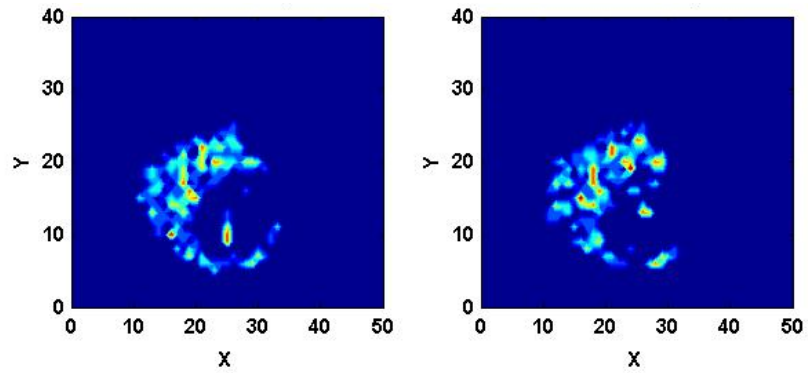


(c) Number of time steps = 200

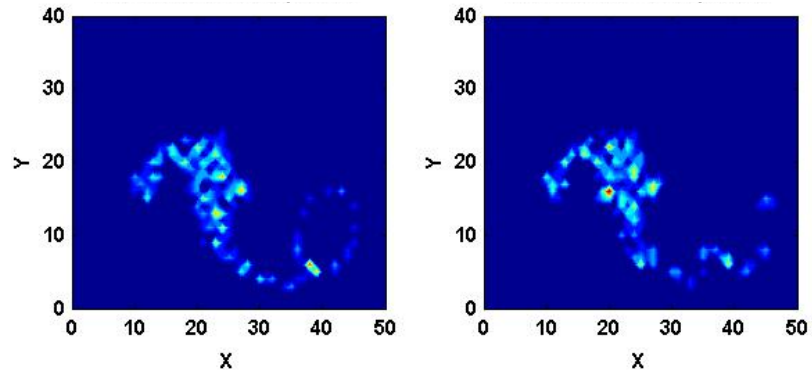
Figure 3.23: Comparison of particle displacement for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.



(a) Number of time steps = 60

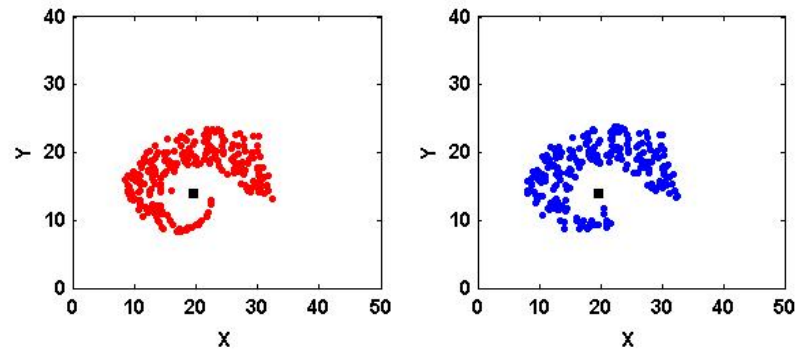


(b) Number of time steps = 125

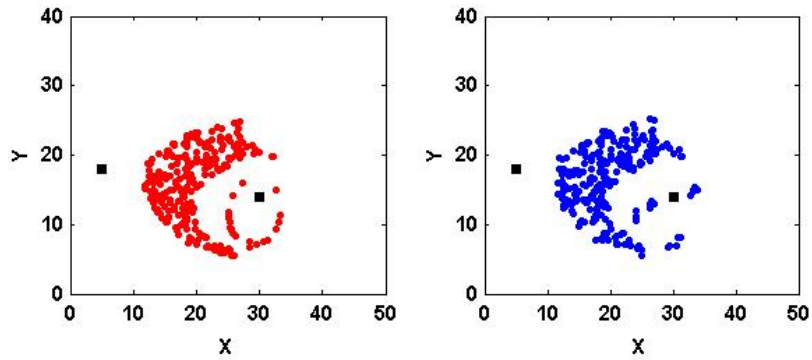


(c) Number of time steps = 200

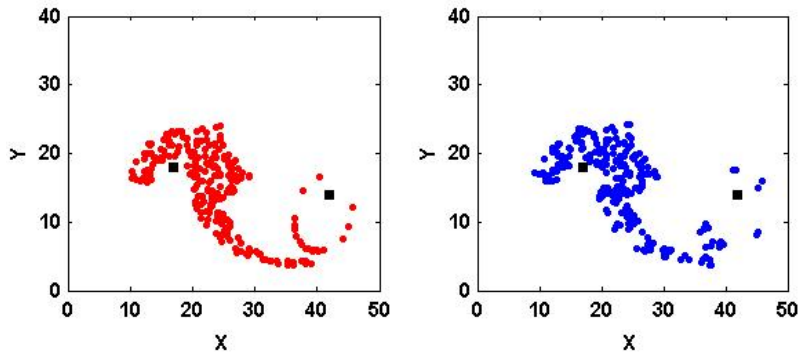
Figure 3.24: Comparison of density plots for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.



(a) Number of time steps = 60

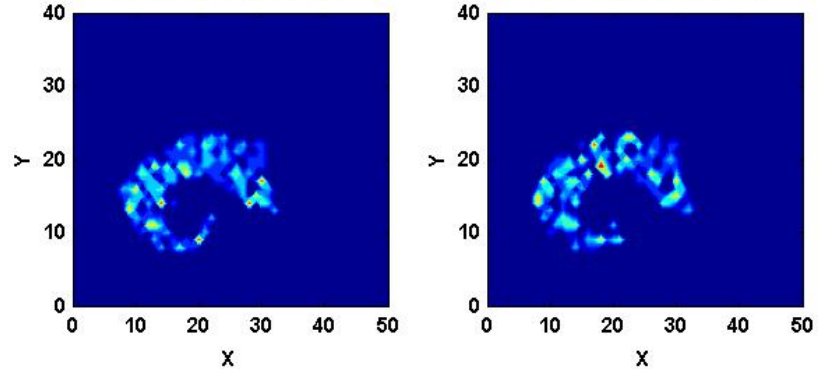


(b) Number of time steps = 125

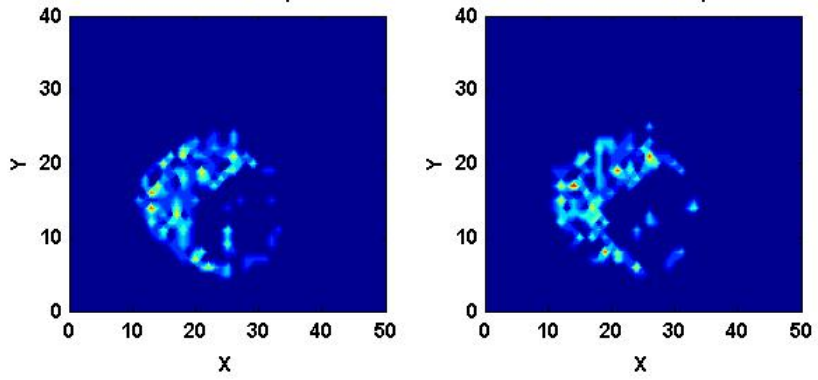


(c) Number of time steps = 200

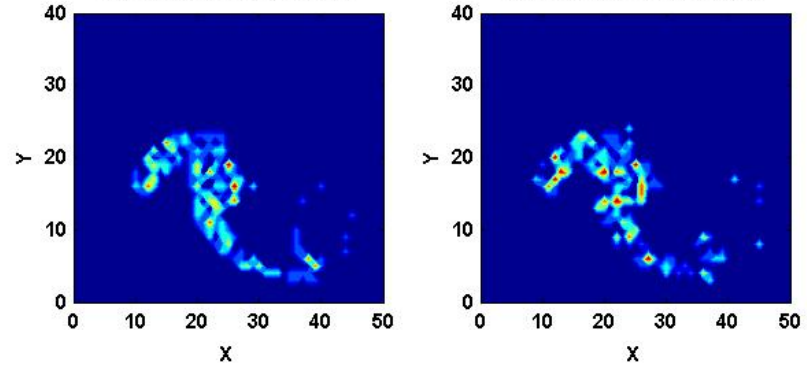
Figure 3.25: Comparison of particle displacement for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.



(a) Number of time steps = 60

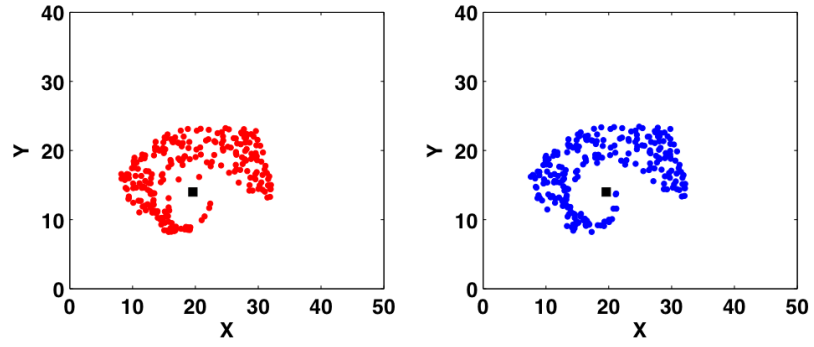


(b) Number of time steps = 125

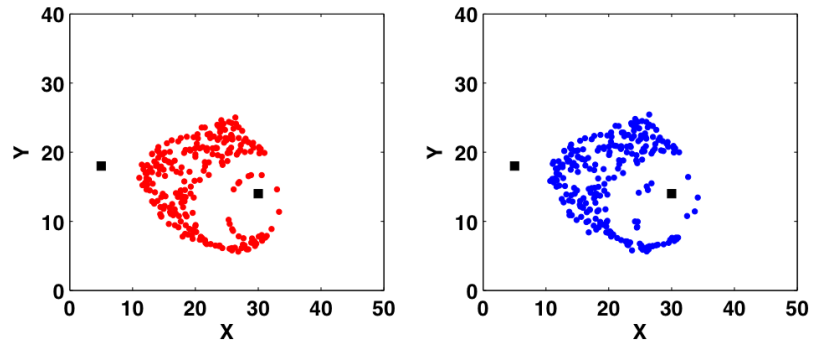


(c) Number of time steps = 200

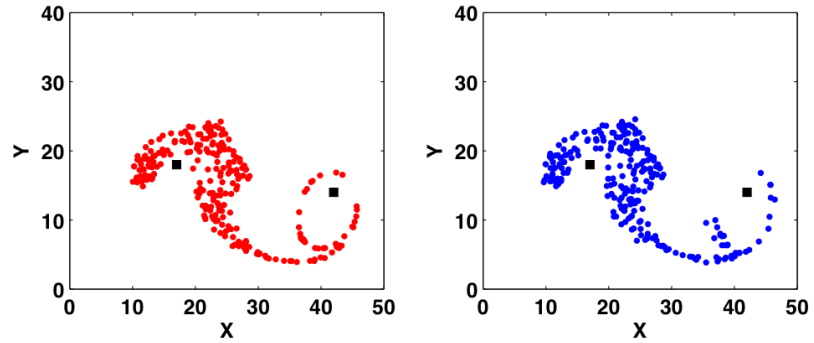
Figure 3.26: Comparison of density plots for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.



(a) Number of time steps = 60

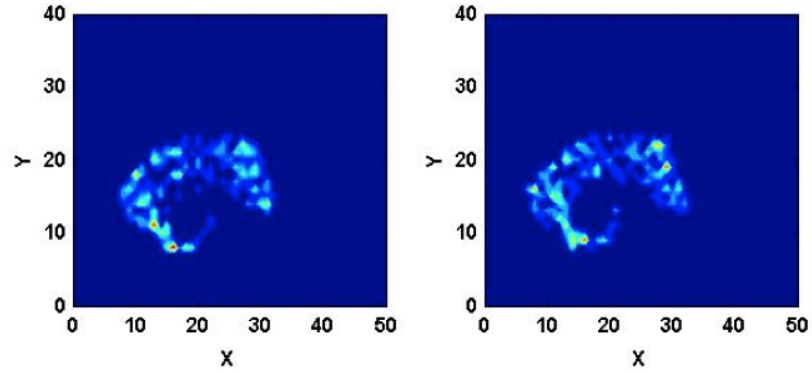


(b) Number of time steps = 125

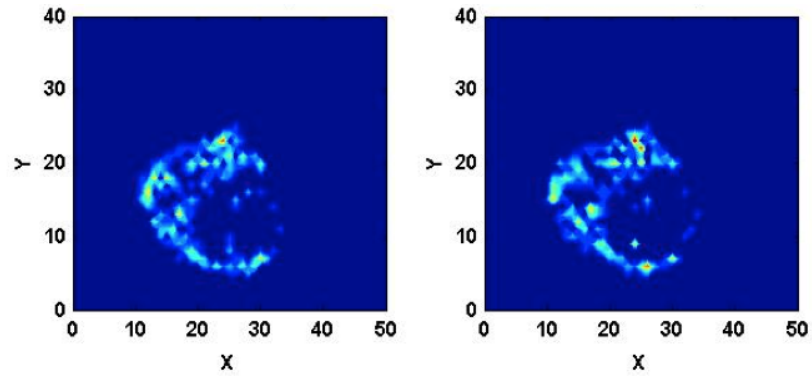


(c) Number of time steps = 200

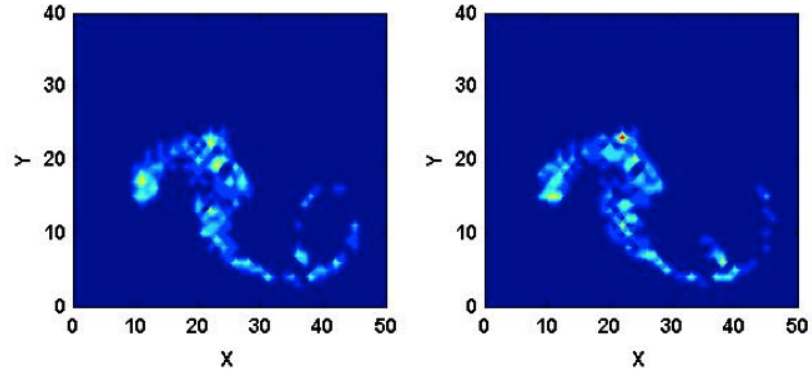
Figure 3.27: Comparison of particle displacement for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.



(a) Number of time steps = 60



(b) Number of time steps = 125



(c) Number of time steps = 200

Figure 3.28: Comparison of density plots for exact and clustered solution using k-means method. Red: Exact solution, Blue: Clustered solution. Number of clusters = 200. (a) Number of time steps = 60; (b) Number of time steps = 125; (c) Number of time steps = 200.

high velocity gradients to resolve the flow-gradients accurately.

Implementation of the k-means method into the three-dimensional problem requires understanding of how best to select the groups for clustering. As mentioned before, the depletion of the saltation layer, resulting in the injection of new particles, will require reclustering of the particles, which may lead to a loss in accuracy. A feasible solution in such a scenario would be to cluster only the particles present in the far-field where the velocity gradients are low, and track the particles in the near-field individually. This strategy will preserve the fidelity of the solution, while avoiding declustering and reclustering dilemmas. Other techniques may exist and could be explored as part of the future work. The k-means method is highly problem dependant and if applied with finesse, can result in significant computational benefits while maintaining the accuracy of the solution within tolerable limits.

3.4 Osiptsov's Method

3.4.1 Introduction

Osiptsov's method is the third method that was implemented to try to alleviate the computational costs on tracking particles in the brownout problem. In this method, particle concentrations along pathlines is computed using a Jacobian transform in the Lagrangian field. The calculation of these concentrations requires the computation of eight extra equations (i.e., for ω_{ij} and $J_{ij} \forall i, j \in 1, 2$) in two-dimensional space, along with the derivatives of the velocities at the particles current position. Although the execution of these calculations can be an expensive investment, the method directly computes the par-

ticle density field, which is the main benefit of the method. However, Osipsov's method results in a non-uniform distribution of data points, which then needs to be interpolated to obtain the overall density field.

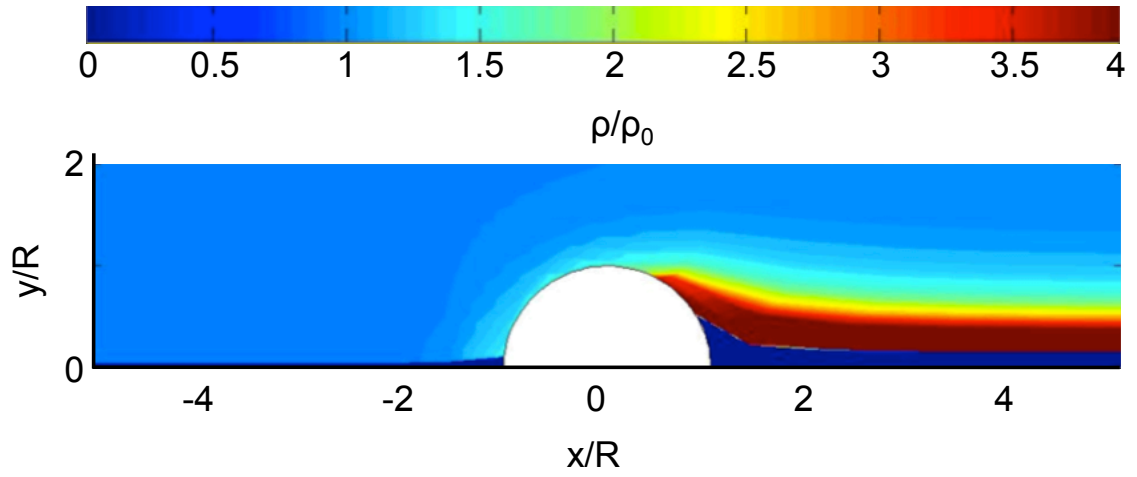
3.4.2 Results for a Potential Flow

Before this method is applied to the prototypical flow used in this study, the algorithm was tested on a simpler two-dimensional potential flow around a circular cylinder. This test was performed to validate the algorithm with the work done by Healy et al. [7] and to test the interpolating algorithm.

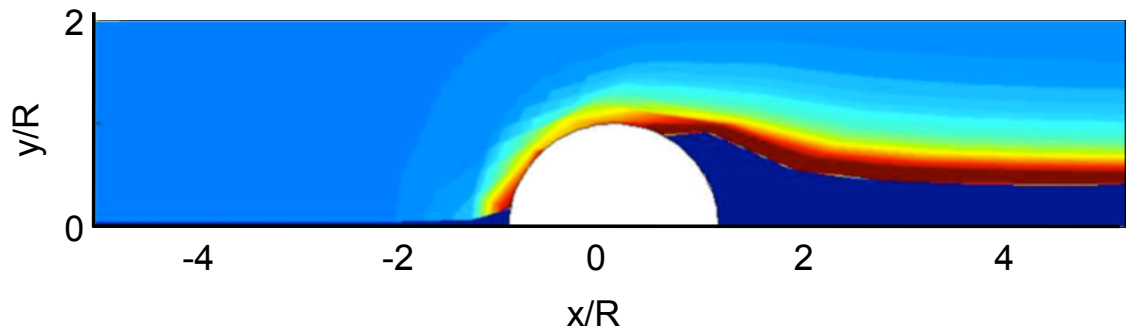
Shown in Fig. 3.29 is the density contour plot for the particle density over the cylinder. A steady flow of dust particles were simulated from left to right in this study. Figure 3.29(a) shows the result for a flow velocity of 1ms^{-1} and Fig. 3.29(b) for 2ms^{-1} . The results indicate that the density plots follow the physics of the flow accurately and the method captures the stagnation region at the front of the cylinder and behind the cylinder, indicated by $\rho/\rho_0 = 0$. A simple linear interpolating algorithm was used to obtain the density values at all points on the map.

3.4.3 Results for the Prototypical Flow

The potential flow around a cylinder follows a steady and uniform pattern, unlike the vortical nature of the flow seen with rotorcraft wakes. In this section, Osipsov's method is applied to the prototypical flow discussed previously. Figure 3.30(a) shows the particle displacements at a given time step with the solution containing $15 \text{ layers} \times 75$



(a) Density contour flow for $U_\infty = 1 \text{ ms}^{-1}$



(b) Density contour flow for $U_\infty = 2 \text{ ms}^{-1}$

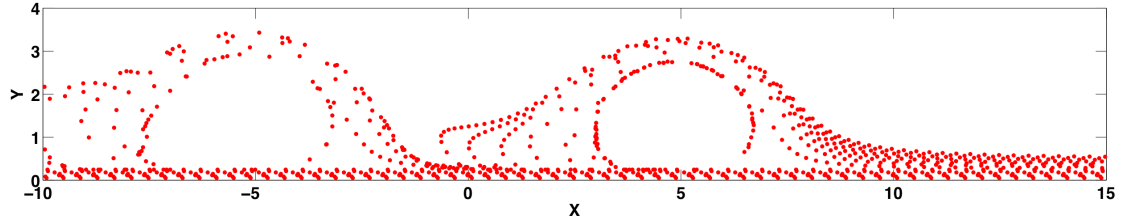
Figure 3.29: Density contour plot using Osipsov's method around a cylinder in potential flow. (a) Flow velocity of 1 ms^{-1} ; (b) Flow velocity of 2 ms^{-1} .

particles per layer. Figure 3.30(b) shows the corresponding density field obtained when using Osipsov’s method, and Fig. 3.30(c) shows the density field obtained by counting the number of particles in each “bin.”

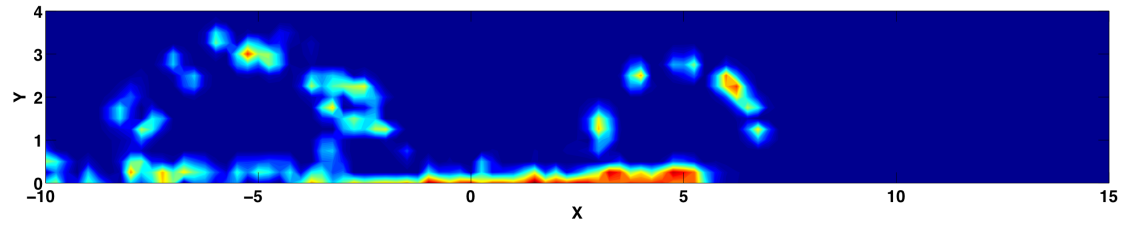
The results in Fig. 3.31 are similar to those in Fig. 3.30, the solution in this case initially containing $15 \text{ layers} \times 375 \text{ particles per layer}$. Unfortunately, the correlation between the density fields is not very satisfactory, even though the remnants of the density field are still seen when using Osipsov’s method. This outcome does not mean that the method does not work. However, further work will be needed to establish the full capabilities of Osipsov’s method for modeling the brownout problem.

Though Osipsov’s method has a fundamentally strong mathematical background, the reason it does not predict accurately the density contour is because of the interpolating algorithm. Osipsov’s method results in scattered density data that needs to be interpolated onto a regular grid. In the previous case of the potential flow, because of the time-invariant nature of the solution, a simple linear interpolating algorithm was sufficient. In the case of a vortical flow with crossing pathlines, such an algorithm or any method that relies on interpolating or extrapolating values between any two given scattered might not yield the best solution.

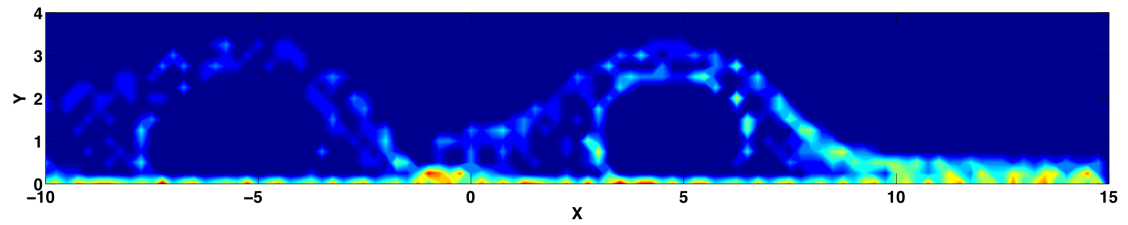
The interpolation algorithm used for the current work was the Radial Basis Function (RBF) with a Gaussian kernel. The mathematics for the RBF method is outlined in Appendix A. From the derivation, it is evident that the application of the RBF requires solution of a $N \times N$ linear system, where N is the number of particles. If performed directly, this requires $O(N^3)$ operations and is obviously impractical for large N . The use of iterative methods with efficient preconditioning, such as that studied in [45], ef-



(a) Particle displacements of exact solution

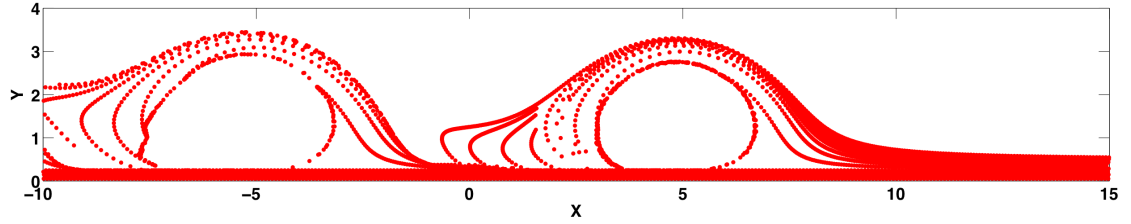


(b) Density field obtained using Osipov's method

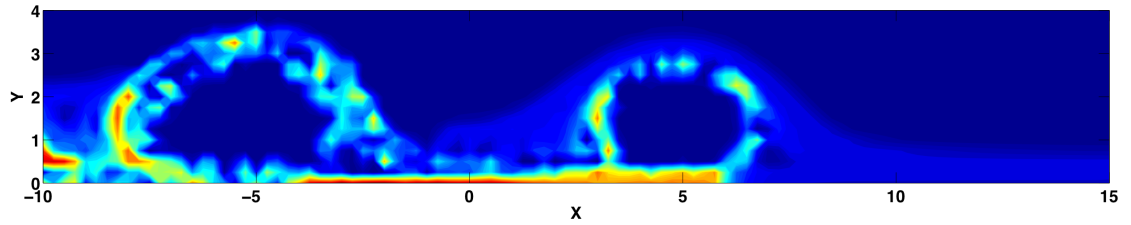


(c) Density field obtained using direct counting

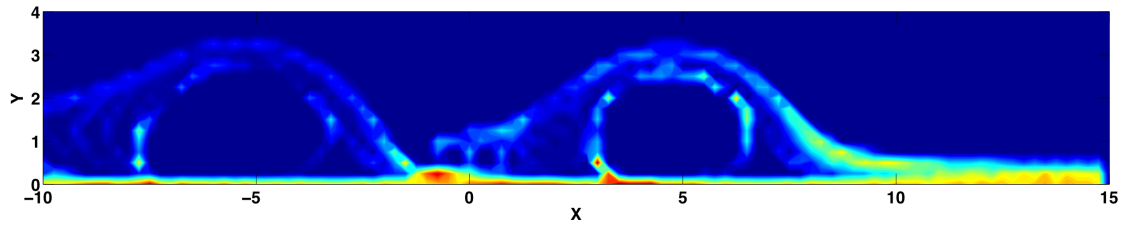
Figure 3.30: Comparison of Osipov's method with the exact solution. Number of particles = 15 layers, 75 particles per layer. (a) Density field obtained using Osipov's method; (b) Density field obtained using direct counting.



(a) Particle displacements of exact solution



(b) Density field obtained using Osipov's method



(c) Density field obtained using direct counting

Figure 3.31: Comparison of Osipov's method with the exact solution. Number of particles = 15 layers, 375 particles per layer. (a) Density field obtained using Osipov's method; (b) Density field obtained using direct counting.

fectively reduces the cost to $O(N^2)$, which is the cost to build the preconditioner and to perform matrix-vector products (the number of iterations in this case is substantially smaller). However, these approaches have dealt with the biharmonic and multiquadric kernels, and future work will be required to test or develop new preconditioners for the Gaussian kernel as an efficient Fast Gauss Transform is now available [46].

Further steps toward the reduction of the cost of the RBF was explored in [47], where both costs were reduced to $O(N \log N)$ by the use of special data structures and the Fast Multipole Method [48] for matrix-vector multiplication. The main idea in the FMM is to split the matrix-vector product into a near and a far field solution, and factored approximate representations of the functions in the far-field are utilized. This decomposition of the matrix-vector kernel into a particular form of data structure is what expedites the calculations to $O(N \log N)$. In the present study, a fast Kriegering method [49] was used for the interpolation, which has complexity $O(N^2)$. The development of a $O(N \log N)$ RBF applicable to the present problem, should make Osipsov's method more attractive.

Based on outcomes from the present study, Osipsov's method may also prove to be useful in other aspects of the rotorcraft brownout simulation, such as the dust-laden flow through engines. Problems such as predictions of the effectiveness of air-particle separators, as well as of the erosion of the turbine blades and degradations in engine performance from the dust, could be modeled more computationally efficiently by using Osipsov's method. To study the applicability of the method to these types of internal flows, a RBF interpolating algorithm would not be necessary because a simpler interpolating algorithm lends itself more readily to these flows as they have fewer crossing pathlines.

Chapter 4

Conclusions

Three particle clustering algorithms, namely the Gaussian method, k-means method, and Osipov's method, were developed with the objective of reducing the computational cost of the tracking of very large numbers of particles in brownout dust cloud simulations. The details of the algorithms were studied using a simplified two-dimensional flow field that was representative of the unsteady vortical flows found near the ground below a rotor system. For each method, a detailed numerical study was carried out on a prototypical flow to understand where the benefits of each method lies, where caution in its use has to be exercised, and where it falls short or fails. Direct or "brute-force" solutions, where each and every particle was tracked throughout the flow, were used as a reference. Although it has been shown that clustering algorithms do not have universal applicability, the present work has provided considerable insight into the potential use of such clustering methods and their bounds of applicability for modeling brownout problems.

The following conclusions have been drawn from this study:

1. Each of the clustering methods discussed in this thesis potentially possess the capability to significantly reduce the computational effort in finding a solution for a brownout dust cloud. These methods, however, must be applied carefully and judiciously because there can be an inverse relationship between the time gained to the accuracy lost. The present study has identified the limits of applicability of

each method with respect to both accuracy and computational time, and how certain flow features such as steep velocity gradients in the carrier flow and crossing particle pathlines can affect the solution for the dust field.

2. The Gaussian distribution method was explored to test its capability of reconstructing a given solution and also to increase the number of particles in the simulation without compromising on computational cost. This method was designed to preserve the actual computed solution, even in regions of high velocity gradients, where the method shrinks the cluster to a single particle. It was shown that the clustered solution could represent the actual solution if a sufficiently large number of particle clusters relative to the total number of particles is used. In the case of the Gaussian method, the algorithm was extended to the full three-dimensional problem of modeling brownout, with encouraging results.
3. The k-means clustering method was devised to convect multiple particles in a single group or cluster through the flow, in which case the computational cost could be limited by computing the solution only for the clusters. It has been shown that the strength of the k-means method lies in its applicability to regions of low velocity gradients, such as the far-field regions of the dust cloud around the rotor system. In such regions, the computational savings in finding the positions of the particles in the dust cloud can be over an order of magnitude over the use of a direct particle tracking method. There also exists a ceiling on the number of clusters that can be used, which is a consequence of the computational overheads incurred by the process of declustering and reclustered, beyond which it is more advantageous both

from the aspect of accuracy and cost to perform a direct or brute-force solution.

4. Osipsov's method contains the potential to greatly reduce computationally time, and works well for particle simulations where the particle density fields are more uniform. Simple linear interpolation algorithms are effective for such problems, which expedites the computational process significantly. For a three-dimensional problem, however, the interpolation process used to extract the density information is $O(N^3)$, which can prove to be very expensive if the results are required at each and every time step. The availability of fast $O(N \log N)$ interpolation algorithms on multidimensional unstructured data sets would greatly improve the attractiveness of this method.

4.1 Suggestions for Future Work

The results obtained during the present work have provided a new understanding into the use of particle clustering algorithms and their application to the rotorcraft brownout problem. The work has contributed to the understanding of different clustering strategies and their relative advantages. From a computational perspective, however, there are still several technical areas that could be addressed to better understand these algorithms and their potential use for reducing the computational costs of modeling brownout dust clouds.

4.1.1 Improvements in Estimating Accuracy

There is a need for an exact solution for the dust cloud so as to gauge the accuracy of the clustered solution, which is the only way to calculate proper metrics that can be used to estimate the errors between the two solutions (i.e., between the clustered solution and the brute-force approach). In the current study, comparisons of density contour maps between the exact and clustered solutions were employed to estimate the accuracy of the clustered solutions. The Pearson coefficient has also been used to augment the results that were obtained using the density contour maps, which provide a better mathematical basis for comparing accuracy.

Another possible technique for estimating the accuracy of the clustered solution is to view the obtained solution as a problem in multi-body dynamics. The dust cloud obtained, either through the exact or the clustered solution, can be treated as multiple bodies relative to a reference coordinate system. All of the particles have to satisfy the primary equations (Newton's second law, linear and angular momentum conservation) and the values obtained using the clustered solution can be compared against those obtained in the exact solution. The clustering schemes could then be modified or adapted based on the accuracy of the solution that is obtained.

4.1.2 Improvements in the Gaussian Method

The Gaussian method was shown to be effective for both the prototypical two-dimensional problem and the three-dimensional brownout problem. The solution to the prototypical flow using the Gaussian method was compared against the exact solution

using density plots to assess the accuracy of the solution. As previously mentioned, sophisticated methods to quantify the accuracy of the clouds in three dimensions through optical means are required, and to this end some initial work has already been done by Tritschler et al. [44]. Although such approaches are still in their early stages, they have shown success in quantifying the characteristics of a brownout cloud from the pilot's perspective, i.e., for assessing the relative severity of brownout.

4.1.3 Improvements in the k-means Method

The relative advantages of the k-means algorithm has been studied and discussed in the current work, which provides a basis for further studying its potential application to Lagrangian particle tracking methods. However, there are still aspects of the k-means algorithm that require further investigation. For example, the issues with reclustering and declustering could be explored in regard to accuracy and practical application. However, more realistic scenarios would require an understanding of how particles that are entrained into the flow are to be treated as clusters. The k-means algorithm is a heuristic algorithm and works efficiently when the particles to be divided into clusters are well specified in space as distinct groups. In other words, the k-means algorithm works best when there is an understanding of the optimum number of clusters required. Because the brownout problem results in an almost continuous distribution of particles throughout the spatial field of interest, further research is required to study complimentary algorithms that possess the ability to determine an optimal number of clusters.

Additional work may be able to adapt the algorithm to alter the total number of

clusters to give a higher number of clusters in regions of high velocity gradients. The disadvantage of an algorithm that can detect regions of high velocity gradients in the flow is that they are inherently computationally expensive. Such overheads can potentially degrade the practical applicability of the k-means method. However, a well devised search algorithm, for instance, one that uses octree data structures, could prove to be computationally less expensive. Finally the k-means algorithm could be applied to the full three-dimensional brownout problem to study its performance both in terms of accuracy of the solution and the reductions in computational time that could be gained. A possible solution could be to apply the k-means algorithm just in the far-field regions of the brownout cloud, where the velocity gradients are relatively low and losses in cloud fidelity are minimized.

4.1.4 Improvements in Osipov's Method

Osipov's method has been shown to have benefits for simple two-dimensional flow fields, and has the potential to be applied to more complex multi-dimensional flow fields. The primary limiting factor of this method is the need for an interpolating algorithm, which is of computational cost $O(N^2)$. Osipov's method also requires the velocity gradient information at each particle location every time step in each direction. In the three-dimensional space, calculation of velocity gradients requires computing the velocities in adjacent spatial locations, following which finite difference schemes can be used to compute the gradients. In a vortex method, the repeated use of the Biot-Savart law makes such a calculation expensive. Consequently, it is important that the interpolating

algorithm used to obtain the density contour map from the scattered data points be highly computationally efficient. The FMM can be implemented to reduce the time from cost $O(N^2)$ to $O(N \log N)$. Further work to investigate different interpolating algorithms may result in methods that are more efficient, which may make the Osipov's method more attractive for modeling the large scale dust clouds of brownout. Although Osipov's method may not be as of yet applicable to the external flow field, it can assist in predicting the dust-laden internal flow, say through turbine cascades, to simulate phenomenon such as flow choking from these dust particles and the abrasion on the turbine blades.

Appendix A

Derivation of Radial Basis Function

The interpolating algorithm used for obtaining the contour density plot from the scattered density data in Osipov's method was the Radial Basis Function (RBF). Let there be n sources and m receivers (treated here as the number of particles and grid points respectively) such the density at any receiver point ρ_j , is given by

$$\rho_j = \sum_{i=1}^n \lambda_i \phi(|\mathbf{r}|), \quad j = 1, \dots, m \quad (\text{A.1})$$

where $\mathbf{r} = \mathbf{r}_i - \mathbf{r}_j$ and the matrix $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$, are the weighting parameters to be found. The values of λ are obtained by solving Eqn. A.1 such that the receivers are replaced with the sources

$$\rho_j = \sum_{i=1}^n \lambda_i \phi(|\mathbf{r}|), \quad j = 1, \dots, n \quad (\text{A.2})$$

In the above equation, the values of ρ and ϕ matrices are all known. The quantity ρ is an artifact of the Osipov's method. The ϕ matrix, or more commonly known as the kernel, can take many forms based on the problem, a few examples being the biharmonic ($\phi(\mathbf{r}) = r$), polyharmonic ($\phi(\mathbf{r}) = r^{2n+1}$) and the multiquadratic kernels ($\phi(\mathbf{r}) = (r^2 + c^2)^{1/2}$). However, for the current problem, the kernel that worked best was the Gaussian kernel, as given by

$$\phi(\mathbf{r}) = \exp(-r^2/h^2) \quad (\text{A.3})$$

where h is a user defined parameter. It is useful to note that the function reduces to around 1% of its peak value at a radial distance of $2h$ from the center of the Gaussian. For the

current work, the value of h chosen was 0.25.

Rewriting in the matrix form, then Eqn. A.2 can be written as

$$\rho = \lambda \phi \tag{A.4}$$

The matrix λ can be obtained through matrix inversion, resulting in

$$\lambda = \phi^{-1} \rho \tag{A.5}$$

The inversion of ϕ is computationally expensive and leads to the overall $O(N^3)$ cost. This cost can be reduced to $O(N^2)$, the algorithm for which is outlined in [45,47] which was used in the present study.

Bibliography

- [1] Sydney, A., Baharani, A., and Leishman, J. G., “Understanding Brownout using Near-Wall Dual-Phase Flow Measurements,” 67th Annual Forum Proceedings of the American Helicopter Society, Virginia Beach, VA, May 3–5, 2011.
- [2] “Sandblaster 2 Support of See-Through Technologies for Particulate Brownout,” Task 1 Technical Report, Sponsored by Defense Advanced Research Projects Agency (DOD) Strategic Technology Office, Issued by U.S. Army Aviation and Missile Command under Contract No. W31P4Q-07-C-0215, MRI Project No. 110565, October 31, 2007.
- [3] “Sandblaster 2 Support of See-Through Technologies for Particulate Brownout,” Task 5 Final Technical Report, Sponsored by Defense Advanced Research Projects Agency (DOD) Strategic Technology Office, Issued by U.S. Army Aviation and Missile Command under Contract No. W31P4Q-07-C-0215, MRI Project No. 110565, October 31, 2007.
- [4] Phillips, C., and Brown, R.E., “Eulerian Simulation of the Fluid Dynamics of Helicopter Brownout,” *AIAA Journal of Aircraft*, Vol. 46, No. 4, July 2009, pp. 1416–1429.
- [5] Morales, F. Naqvi, I., Squires, K. D., and Piomelli, U., “Euler-Lagrange Simulations of Particle Interactions with Coherent Vortices in Turbulent Boundary Layers,”

62nd Annual Meeting of the Division of Fluid Dynamics, Minneapolis, Minnesota, November 2009.

- [6] Syal, M., Govindarajan, B., and Leishman, J. G., “Mesoscale Sediment Tracking Methodology to Analyze Brownout Cloud Developments,” 66th Annual Forum of the American Helicopter Society, Phoenix, AZ, May 10–13, 2010.
- [7] Healy, D. P., and Young, J. B., “Full Lagrangian Methods for Calculating Particle Concentration Fields in Dilute Gas-Particle Flows,” *Proceedings of the Royal Society*, Vol. 461, No. 2059, 2005, pp. 2197–2225.
- [8] Mapes, P., Kent, R., and Wood, R., “DoD Helicopter Mishaps FY85-05: Findings and Recommendations,” U.S. Air Force, 2008.
- [9] National Transportation Safety Board, NTSB Accident Briefs: LAX01LA283, LAX01LA304, LAX04LA285, SEA05CA173.
- [10] Jansen, C., Wennemers, A., and Groen, E., “FlyTact: A Tactile Display Improves a Helicopter Pilots Landing Performance in Degraded Visual Environments,” *Haptics: Perception, Devices and Scenarios*, Vol. 502, No. 4, 2008, pp. 867–875.
- [11] Milluzzo, J., Sydney, A., Rauleder, J., and Leishman, J. G., “In-Ground-Effect Aerodynamics of Rotors with Different Blade Tips,” 66th Annual Forum Proceedings of the American Helicopter Society, Phoenix, AZ, May 10–13, 2010.
- [12] Cheeseman, I. C., and Bennett, W. E., “The Effect of the Ground on a Helicopter Rotor in Forward Flight,” ARC RM 3021, 1955.

- [13] Fradenburgh, E. A., “The Helicopter and the Ground Effect Machine,” *Journal of the American Helicopter Society*, Vol. 5, No. 4, 1960, pp. 26–28.
- [14] Knight, M., and Hefner, R. A., “Analysis of Ground Effect on the Lifting Airscrew,” NACA TN 835, 1941.
- [15] Hayden, J.S., “The Effect of the Ground on Helicopter Hovering Power Required,” American Helicopter Society 32th Annual National V/STOL Forum Proceedings, Washington D.C., May 10–12, 1976.
- [16] Curtiss, H. C., Sun, M., Putman, W. F., and Hanker, E. J., “Rotor Aerodynamics in Ground Effect at Low Advance Ratios,” *Journal of the American Helicopter Society*, Vol. 29, No. 1, 1984, pp. 48–55.
- [17] Light, J. S., and Norman, T., “Tip Vortex Geometry of a Hovering Helicopter Rotor in Ground Effect,” American Helicopter Society 45th Annual Forum Proceedings, Boston, MA, May 22–24, 1989.
- [18] Prouty, R. W., “Ground Effect and the Helicopter,” AIAA Paper 85-4034, AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting, Colorado Springs, CO, October 14–16, 1985.
- [19] Lee, T. E., Leishman, J. G., and Ramasamy, M., “Fluid Dynamics of Interacting Blade Tip Vortices With a Ground Plane,” *Journal of the American Helicopter Society*, Vol. 55, No. 2, April 2010, pp. 1–16.

- [20] Ramasamy, M., Johnson, B., and Leishman, J. G., “Understanding the Aerodynamic Efficiency of a Hovering Micro-Rotor,” *Journal of American Helicopter Society*, Vol. 53, No. 4, October, 2008, pp. 412–428.
- [21] Bagnold, R. A., *The Physics of Blown Sand and Desert Dunes*, Dover Publications, Inc., Mineola, NY, 1941.
- [22] Greeley, R., and J. D. Iversen, *Wind as a Geological Process on Earth, Mars, Venus and Titan*, Cambridge University Press, New York, NY, 1985.
- [23] Kalra, T., Lakshminarayan, V., and Baeder, J., “CFD Validation of Micro Hovering Rotor in Ground Effect,” AHS International Specialists Conference Proceedings on Aeromechanics, San Francisco, CA, January, 2010.
- [24] Lakshminarayan, V. K., “Computational Investigation of Micro-Scale Coaxial Rotor Aerodynamics in Hover,” Ph.D. dissertation, Department of Aerospace Engineering, University of Maryland at College Park, USA, 2009.
- [25] Bhagwat, M. J., and Leishman, J. G., “Time-Accurate Modeling of Rotor Wakes Using A Free-Vortex Wake Method,” 18th AIAA Applied Aerodynamics Conference Proceedings, Denver, CO, August 2000.
- [26] Bhagwat, M. J., and Leishman, J. G., “Stability, Consistency and Convergence of Time-Marching Free-Vortex Rotor Wake Algorithms,” *Journal of the American Helicopter Society*, Vol. 46, No. 1, January 2001, pp. 59–71.

- [27] Bhagwat, M. J., and Leishman, J. G., “Accuracy of Straight- Line Segmentation Applied to Curvilinear Vortex Filaments,” *Journal of the American Helicopter Society*, Vol. 46, No. 2, April 2001, pp. 166–169.
- [28] Bhagwat, M. J., and Leishman, J. G., “Generalized Viscous Vortex Core Models for Application to Free-Vortex Wake and Aeroacoustic Calculations,” American Helicopter Society 58th Annual Forum Proceedings, Montréal, Canada, June 11–13, 2002.
- [29] Leishman, J. G., Bhagwat, M. J., and Bagai, A., “Free-Vortex Filament Methods for the Analysis of Helicopter Rotor Wakes,” *Journal of Aircraft*, Vol. 39, No. 5, September–October, 2002, pp. 759–775.
- [30] Leishman, J.G., *Principles of Helicopter Aerodynamics* Cambridge University Press, New York, NY, 2006.
- [31] Phillips, C., Kim, H. W., and Brown, R.E., “The Flow Physics of Helicopter Brownout,” 66th Annual Forum of the American Helicopter Society, Phoenix, AZ, May 10–13, 2010.
- [32] Morales, F. Naqvi, I., Squires, K. D., and Piomelli, U., “Euler-Lagrange Simulations of Particle Interactions with Coherent Vortices in Turbulent Boundary Layers,” *Bulletin of the American Physical Society*, Vol. 54, No. 19, 2009.
- [33] Wong, O. D., and Tanner, P. E., “Photogrammetric Measurements of an EH-60L Brownout Cloud,” 66th Annual Forum of the American Helicopter Society, Phoenix, AZ, May 10–13, 2010.

- [34] Syal, M., and Leishman, J. G., “Comparisons of Predicted Brownout Dust Clouds with Photogrammetry Measurements,” 67th Annual Forum Proceedings of the American Helicopter Society, Virginia Beach, VA, May 3–5, 2011.
- [35] Thomas, S., Lakshminarayan, V. K., Kalra, T. S., and Baeder, J. D., “Eulerian-Lagrangian Analysis of Cloud Evolution using CFD Coupled with a Sediment Tracking Algorithm,” 67th Annual Forum Proceedings of the American Helicopter Society, Virginia Beach, VA, May 3–5, 2011.
- [36] Hu, Q., Gumerov, N. A., Duraiswami, R., Syal, M., and Leishman, J. G., “Toward Improved Aeromechanics Simulations Using Recent Advancements in Scientific Computing,” 67th Annual Forum Proceedings of the American Helicopter Society, Virginia Beach, VA, May 3–5, 2011.
- [37] Wachspress, D., A., Whitehouse, G., R., Keller, J., D., and Yu, H., “A High Fidelity Brownout Model for Real-Time Flight Simulations and Trainers,” 65th Annual Forum Proceedings of the American Helicopter Society, Grapevine, TX, May 27–29, 2009.
- [38] Teske, M. E., Bird, S. L., Esterly, D. M., Curbishley, T. B., Tay, S. L., and Perry, S.G., “AgDRIFT: A Model for Estimating Near-Field Spray Drift from Aerial Applications,” Atmospheric Sciences Modeling Division, Air Resources Laboratory/NOAA (MD-20), Research Triangle Park, North Carolina 27711, USA, August 2001.

- [39] MacQueen, J. B., “Some Methods for classification and Analysis of Multivariate Observations,” *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California, 1967.
- [40] Wei Li, “Practical Criteria for Positive-Definite Matrix, M-Matrix and Hurwitz Matrix,” *Applied Mathematics and Computation*, Vol. 185, 2007, pp. 397–401.
- [41] Davies, D. L., and Bouldin, D. W., “A Cluster Separation Measure,” *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 1, No. 4, 1979, pp. 224–227.
- [42] Fernandez de la Mora, J., and Rosner, D. E., “Inertial Deposition of Particles Revisited and Extended: Eulerian Approach to a Traditionally Lagrangian Problem,” *Physicochemical Hydrodynamics*, Vol. 2, 1981, pp. 1–21.
- [43] Osipov, A. N., “Modified Lagrangian Method for Calculating the Particle Concentration in Dusty-Gas Flows with Intersecting Particle Trajectories,” *Proceedings of Third International Conference on Multiphase Flow*, Lyon, France, 1998.
- [44] Tritschler, J., Syal, M., Celi, R., and Leishman, J. G., “A Methodology for Rotor Design Optimization for Rotorcraft Brownout Mitigation,” 66th Annual Forum Proceedings of the American Helicopter Society, Phoenix, AZ, May 10–13, 2010.
- [45] Faul, A. C., Goodsell, G., and Powell, M. J. D., “A Krylov Subspace Algorithm for Multiquadric Interpolation in Many Dimensions,” *IMA Journal of Numerical Analysis*, Vol. 25, 2005, pp. 1–24.

- [46] Yang, C., Duraiswami, R., Gumerov, N. A., and Davis, L. S., “Improved Fast Gauss Transform,” *Proceedings of the International Conference on Computer Vision*, France, October 2003.
- [47] Gumerov, N. A., and Duraiswami, R., “Fast Radial Basis Function Interpolation Via Preconditioned Krylov Iteration,” *SIAM Journal of Scientific Computing*, Vol. 29, No. 5, 2007, pp. 1876–1899.
- [48] Greengard, L., and Rokhlin, V., “A Fast Algorithm for Particle Simulations,” *Journal of Computational Physics*, Vol. 73, No. 2, December 1987, pp. 325–348.
- [49] Sakata, S., Ashidaa, F., and Zakob, M., “An Efficient Algorithm for Kriging Approximation and Optimization with Large-Scale Sampling Data,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 93, No. 3–5, January 2004, pp. 385–404.