Brownout Modeling

by

Federico Rovere

A thesis submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
University of Glasgow
School of Engineering
July 2022

Declaration

I hereby declare that this dissertation is a record of work carried out in the University of Glasgow during the period from August 2018 to July 2022. The dissertation is original in content except where otherwise indicated.

....

.....
(Federico Rovere)

Abstract

The thesis compares different computational approaches for simulate brownout clouds. Brownout relates to blinding dust clouds stirred up by the helicopter rotor downwash during near-ground flight. This causes significant flight safety risks, including ground obstacle collisions and dynamic rollover due to sloped and uneven terrain. Operating helicopters in brownout conditions is very dangerous and has claimed many lives.

Due to the complexity of these phenomena, and the safety issues they may generate, brownout has been studied using different approaches, both numerical and experimental. In terms of the computational investigations: brownout works may be divided into the two approaches used to predict the dust cloud: Lagrangian and Eulerian.

A direct comparison of the two models is presented in the same framework, to define the best practice for brownout predictions. The present work involved the development of a Lagrangian particle tracking algorithm and an Eulerian model in the HMB3 (Helicopter Multi-Block) framework. Brownout clouds obtained with these models are compared with experimental results and between them. Results show that even if in proximity to a rotor, the accuracy of the two models is comparable, further from it the Lagrangian approach is more accurate than modelling based on the Eulerian approach.

Furthermore, the two models are compared in terms of computational efficiency, and results show how Eulerian is a better model to predict brownout, due to its intrinsic abilities not to being affected in its computational efficiency by the amount of particles.

In addition, brownout clouds are compared in terms of size at different rotor configurations. Results show how rotors with higher thrust coefficient are able to generate the more dangerous scenarios, generating bigger clouds than rotors operating at lower thrust coefficients.. The presence of the fuselage has also been studied, comparing cases with and without it.

The thesis includes more safety aspects, including risks posed to ground personnel and nearby structures due to the strong outflow generated by hovering and taxiing rotorcraft. In this work, an analysis of safety operations is performed in terms of forces that ground personnel may suffer due to strong rotor outflows. Force distribution over the human body and total force are computed from the resolved flowfield around a rotor. Different single rotor configurations have been taken into account, and in all cases considered, the whole area around the rotor can be considered safe according to military-based thresholds.

Publications

Journal

- Rovere, Federico, George Barakos, and Rene Steijl. "Safety analysis of rotors in ground effect." Aerospace Science and Technology (2022): 107655.

  Doi:https://doi.org/10.1016/j.ast.2022.107655

- Rovere, F., Barakos, G. N. and Steijl, R. (2023) Eulerian and Lagrangian Brownout Simulations Methods, In: Aerospace Science and Technology.

  Doi:https://doi.org/10.1016/j.ast.2023.108306

Conferences with Proceedings

- Rovere, F., Morelli, U., Steijl, R. , Barakos, G. N. and Vigevano, L. (2019) Evaluation of Rotor Blade Models for Rotor Outwash. In: 44th European Rotorcraft Forum, Delft, The Netherlands, 18-20 Sep 2018,

- Rovere, F., Barakos, G. N. and Steijl, R. (2019) Brownout Simulations of Model-Rotors In Ground Effect. In: 4th edition of the International Cross-industry Safety Conference (ICSC), Amsterdam, The Netherlands, 09-11 Oct 2019,

- Rovere, F., Barakos, G. N. and Steijl, R. (2019) CFD Analysis of a Micro-Rotor In Ground Effect. In: 45th European Rotorcraft Forum, Warsaw, Poland, 17-20 Sep 2019,

- Rovere, F., Barakos, G.N. and Steijl, R. (2020) CFD Analysis of a Micro-Rotor In Ground Effect. In: 2020 AIAA SciTech Forum, Orlando, FL, USA, 06-10 Jan 2020,

- Rovere, F., Barakos, G. N. and Steijl, R. (2020) Particle Tracking Analysis of a Micro-rotor In Ground Effect, In: 46th European Rotorcraft Forum, Moscow, Russia, 08-11 Sep 2020,

- Rovere, F., Barakos, G. N. and Steijl, R. (2021) Particle Tracking Analysis of a Micro-rotor In Ground Effect, In: AIAA Aviation Forum, online, 2-6 August 2021,

- Rovere, F., Barakos, G. N. and Steijl, R. (2021) Particle Tracking Analysis of a Rotor In Ground Effect. In: ERF2021, online, 7-9 September 2021.

Conferences without Proceedings

- Rovere, F., Barakos, G. N. and Steijl, R. (2019) "Brownout and White out modeling", 4th Vertical Lift Network Workshop, Cheshire, UK, 7-9 April 2019,

- Rovere, F., Barakos, G. N. and Steijl, R. (2021) "Brown-out simulations of model-rotors", NITROS International Workshop on Engineering for Rotorcraft Safety, Virtual Workshop 7-9 April 2021.

Technical Notes

- Rovere, F., "Lagrangian and Eulerian particle tracking in the HMB3 Framework," TN-22-002, CFD Laboratory, James Watt School of Engineering, University of Glasgow, 2022.

Acknowledgements

I would like to thank my supervisors Prof George Barakos and Dr René Steijl for their constant guidance, patience, encouragement, and technical knowledge throughout this work. Their endless support kept me motivated and made this thesis a reality. I also want to extend my gratitude to Prof Giuseppe Quaranta, and all NITROS group for their kind support.

I would also to thank the European Union that funded my work, through the European Union's H2020 research and innovation programme under the Marie Sktodowska-Curie grant agreement No 721920.

In addition, I would like to extend my gratitude to the whole CFD lab group.

Thanks to Ross, Thomas, Blaga, Kiril and Andrea for the kind environment and the family-like atmosphere in the group. A special thanks to Dr Tao Zhang, who has been a good friend to me during these years, sharing amazing and hard moments. Grazie vecchio!

Finally, I would thank my parents, Marcello e Raffaella, for their unconditional and endless love.

Cosa sarà mai portarvi dentro solo tutto il tempo?

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Thesis structure . . . 5

2 Literature survey **7**
- 2.1 Classification of published works . . . 7
- 2.2 Experimental analyses on full scale rotorcraft . . . 7
- 2.3 Experimental analyses on model scale rotorcraft . . . 12
- 2.4 Numerical computations for flight simulations . . . 25
- 2.5 Flow Field CFD simulations . . . 27
- 2.6 Phase coupling . . . 36
- 2.7 Dispersed phase modelling . . . 37
- 2.8 Full brownout simulations . . . 39
  - 2.8.1 Lagrangian Particle Tracking . . . 40
  - 2.8.2 Eulerian simulations . . . 42
- 2.9 Outflows over human body - PAXman model . . . 44
- 2.10 Outcome of the literature survey . . . 45
- 2.11 Objectives of this thesis . . . 48

3 Methodology **49**
- 3.1 Navier-Stokes equations . . . 50
  - 3.1.1 Non-dimensionalization . . . 52
  - 3.1.2 Governing equations in conservative vector form . . . 53
  - 3.1.3 Spatial discretization . . . 54
  - 3.1.4 Temporal discretization . . . 54
    - Steady state solver . . . 55
  - 3.1.5 Time marching solver . . . 56
- 3.2 Reynolds Averaged Navier Stokes Equations . . . 57
  - 3.2.1 Reynolds averaging . . . 57
  - 3.2.2 Turbulence modelling . . . 59

4 Particle tracking models **60**
- 4.1 Lagrangian particle tracking . . . 60
- 4.2 Particles uplift and bombardment . . . 63
  - 4.2.1 Uplift threshold model . . . 63
    - Splash entrainment model . . . 64

4.3 Eulerian modelling . . . 69
4.4 PAXman Model - Computing forces over human body . . . 73

5 Test Cases **75**
5.1 Isolated rotor OGE . . . 76
5.1.1 Experiments by Ramasamy et al. [36] . . . 76
5.1.2 Rotor performance results . . . 79
5.2 Isolated rotor hovering IGE . . . 86
5.2.1 Experiments by Lee et al. [26] . . . 86
5.2.2 Rotor performance results . . . 88
5.2.3 Flowfield visualization . . . 90
5.2.4 Taxiing rotor in IGE . . . 93
5.2.5 Non uniform actuator disk . . . 95

6 PAXman model - Forces on a human body **97**
6.1 Scaling factors . . . 97
6.2 Forces on human body - Hovering rotor IGE . . . 99
PAXman results . . . 99
6.3 Forces on human body - Forward flight IGE . . . 101
6.3.1 Full scale - Small scale Validation . . . 104

7 Lagrangian particle tracking **109**
7.1 Algorithm description . . . 109
7.2 Search algorithms . . . 111
7.2.1 Brute force search method . . . 111
7.2.2 Fast search method . . . 112
7.3 Validation test case - EH-60L Wong and Tanner experiments . . . 113
7.4 Isolated hovering rotor IGE . . . 116
7.4.1 Uplift criteria results . . . 116
7.4.2 Brownout clouds . . . 117
7.5 Forward flight rotor IGE . . . 122
7.5.1 Uplift results . . . 122
7.5.2 Brownout clouds . . . 124
7.6 Timing and processors scaling . . . 125

8 Eulerian simulations **128**
8.1 Algorithm . . . 128
8.2 Forward flying rotors IGE . . . 129
8.2.1 Validation Test Case - EH-60L Wong and Tanner Experiments . . . 129
8.2.2 Lagrangian and Eulerian results comparisons . . . 132
8.3 Timing and performance comparison of the methods . . . 140

9 Conclusions and future work **144**
9.1 Conclusions . . . 144
9.2 Future work . . . 147

References **148**

A Particle Tracking in Tecplot **160**
A.1 Test case description . . . 160
A.2 Particle positions . . . 160

B Particle tracking tool for HMB3 **169**
B.1 Lagrangian numerical model . . . 169
B.1.1 Massless particles . . . 169
B.1.2 Particles with mass . . . 170
B.2 FlowChart of the tracking methods . . . 171
B.3 Splash entraiment . . . 172
B.4 Search methods . . . 176
B.4.1 Brute force method . . . 176
B.4.2 New search method . . . 176
B.4.3 Chimera search . . . 177
B.4.4 Check search . . . 178
B.5 Input files . . . 178
B.5.1 st.expert.particles . . . 178
B.5.2 Flowfield files . . . 180
B.6 Usage of the particle tracking tool . . . 182
B.6.1 Output of the Particle tracking tool . . . 182
B.6.2 Tacplot and Lagrangian tool comparison . . . 183
Tecplot and Lagrangian tool comparison . . . 183
B.6.3 Particle tracking examples . . . 185
Steady flow - Rotor in taxiing . . . 185
Unsteady flow - Pitching NACA0012 Airfoil . . . 192

C Eulerian model of HMB3 **201**
C.1 Eulerian numerical model . . . 201
C.2 Input files . . . 205
C.2.1 st.expert.scalars . . . 205
C.2.2 Checkpoint files . . . 206
C.3 Eulerian modelling examples . . . 206
C.3.1 Steady flow - NACA0012 aerofoil . . . 206
Unsteady flow - Pitching NACA0012 aerofoil . . . 212

List of Figures

1.1 HH-60H helicopter landing in the vicinity of Baghdad [1]. 2
1.2 Tail rotor blade showing severe erosion and perforation of the leading edge [2]. . . . 2
1.3 A NH-90 in whiteout conditions [10]. 3

2.1 The diagram illustrates how the brownout has been investigated in the literature. . . 8
2.2 Outflow profile of a CH-53E hovering at AGL=20 ft with a GW of 45,000 lb. This profile is measured at a distance of 49.4 ft from the rotor axis (i.e. 1.25 x/R). In the figure the measured mean and peak velocity are used to validate the numerical computation described in Preston et al. [18]. 9
2.3 Top-down (left) and side view (right) of the dust cloud position of an EH-60L approaching touchdown, measured with the photogrammetry technique. The position of the aircraft is also measured by photogrammetry except at t=11.67 s and t=17.33 s when it was fully engulfed in the brownout cloud and on board data were used.It is interesting to notice that the cloud quickly passes ahead of the vehicle and, by t=4 s most of the aircraft is surrounded by the cloud [12]. 11
2.4 Brownout cloud developed by a EH-60L during approach to touchdown. It is evident the pick up of blue chalk which was spread on the ground in a line perpendicular to the flight path [12]. 12
2.5 The four blades analyzed by Milluzzo et al. [27] (a) and the correspondent measured history of the tip vortex characteristics as a function of wake age: (b) Peak swirl velocity; (c) Core radius. 16
2.6 Schematic representation of the flow surrounding an helicopter IGE according to Bourne et al. [31] (left) and brownout dust cloud developed during a flight test. From the figure it is possible to appreciate the toroidal shape of the cloud generated by this particular aircraft during hover IGE. 17
2.7 Stereo PIV mean velocity contour plots (left) and corresponding velocity flow fields (right) generated by a two-bladed rotor proposed by Bourne et al. [31]. . . . 18
2.8 Measurements of power (left) and trust (right) of a rotor hovering IGE at different height over a ground plane. The rotor was kept at constant blade pitch. On one hand, the power measurements show a steep reduction of the power below one radius of height and a dependence of the performance on the operating state of the rotor with higher blade loading having more significant effects on the IGE performance. On the other, the measurements show little effect on thrust for more than three rotor radii of height, while below one radius the thrust increased more rapidly. 19
2.9 Different advance ratios regimes. 21

2.10 Schematic showing the different particle motions and the fundamental uplift mechanisms responsible of the brownout cloud [47]. 21
2.11 Helicopter landing in brownout conditions [48]. In the figure are pointed out with black triangles the plume-like dust uplifting events in the cloud. The cloud is clearly three-dimensional and highly unsteady. 22
2.12 Schematic representation of the interaction between consecutive vortex filaments [48]. This interaction is one of the causes of locally higher near-ground velocity which, in turn, leads to the discrete uplift of sediment particles. 22
2.13 Consecutive PIV measured velocity fluctuations vector field (Reynolds decomposed) superimposed by the particle distribution on a background contour showing the instantaneous out-of-plane vorticity [40]. 24
2.14 Time-averaged particle concentration close to the ground below an hovering rotor measured with dual-phase PIV [39]. Evidently, the particle concentration is, on average, higher close to the ground between 1.5 and 3 radii outboard. Although, it is important to remember that, because of the three-dimensionality and unsteadiness of the phenomenon, the instantaneous particle concentration can have, locally, much higher values than shown in the present figure. 29
2.15 Schematic representation of the VFM [69] 30
2.16 Free-vortex wake solution at a ground plane using the method of images [65] . . . 31
2.17 Effect of the viscous ground on the VPM-predicted time-averaged rotor wash at a radial station of 2.2R for an isolated rotor hovering at 0.5R over the ground [72]. The VPM results are compared with the measurements by Lee et al. [26]. On the y-axis the height over the ground is normalized by the rotor radius while the outwash velocity on the x-axis is normalized by the nominal induced inflow in hover ($\sqrt{C_{T_{hov}}/2}$). 33
2.18 Wake-coupling methodology applied to a rotor hovering IGE in Thomas et al. work [89]. The red and green mesh represent the blade and ground mesh respectively while the tip vortex in between these two regions is tracked using VFM. 35
2.19 Different approaches for modelling particle and droplet clouds [94] 38
2.20 Comparison of dust clouds for two different climb rates [65]. 42
2.21 Schematic showing the Gaussian method of particle clustering. [100] 42
2.22 CH-47D PAXman outwash force contours during a 20 ft hover [19]. 45

3.1 Methodology workflow used in this work. 50

4.1 Schematic representation of particle-wall interaction. 65
4.2 PAXman model [18]. 74

5.1 Geometry of the rectangular blade used for computations. 77
5.2 Computational domain and boundary conditions, nodes for baseline mesh. 78
5.3 C_T and FoM variations with varying volume sizes h^3. The cell size $h = 1/N_{cell}^3$, where N_{cell} takes the sum of the near-field mesh cell numbers as presented in table 5.2 80
5.4 Schematic of circulation path. 81
5.5 Lift over the span. Experimental results by Ramasamy et al. [36]. The rotor was operating OGE, at collective $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.082$. The $k-\omega$ turbulence model was used for calculations. Baseline mesh. 82

5.6 Drag over the span. Experimental results by Ramasamy et al. [36]. The rotor was operating OGE, at collective $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.082$. The $k-\omega$ turbulence model was used for calculations. Baseline mesh. 84
5.7 Experimental and CFD wake visualization at OGE conditions The rotor corresponds to the design reported by Ramasamy et al. [36], at collective $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.082$. The $k-\omega$ turbulent model was used for calculations. Baseline mesh. 85
5.8 Geometry of the rectangular blade used in the present computation. 87
5.9 Computational domain and boundary conditions. 87
5.10 Experimental and CFD results for the trust ratios IGE at constant power. Experimental data by Lee et al. [26] 89
5.11 Experimental and CFD time-averaged outflow velocity profiles at different heights above the ground. The rotor was operating, $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.08$. Left column results are scaled with induced hover velocity, while the right column results are jet-scaled. Experimental data by Lee et al. [26]. 91
5.12 Experimental and CFD time-averaged outflow velocity profiles at different heights above the ground. The rotor was operating, $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.08$. Left column results are scaled with induced hover velocity, while the right column results are jet-scaled. Experimental data by Lee et al. [26]. 92
5.13 Wake Visualization, for the IGE case, proposed by Lee et al. [26]. The rotor operating at h/R=1, $\theta_{75} = 12deg$, $Re_{tip} = 35000$, $M_{tip} = 0.08$, $h/R = 1$ and $C_T = 0.03$. . 93
5.14 Forward flight mesh setup. 95

6.1 PAXman model forces calculated using the employed micro-rotor, scaled to full scale. The micro rotor rotor was operating at $\theta_{75} = 12deg$, $Re_{tip} = 35000$, $M_{tip} = 0.08$, $h/R = 0.5$ and $C_T = 0.035$. 101
6.2 PAXman model forces calculated using the employed micro-rotor, scaled to full scale. The micro rotor rotor was operating at $\theta_{75} = 12deg$, $Re_{tip} = 35000$, $M_{tip} = 0.08$, $h/R = 1$ and $C_T = 0.03$. 102
6.3 PAXman model forces calculated using the employed micro-rotor, scaled to full scale. The micro rotor rotor was operating at $\theta_{75} = 12deg$, $Re_{tip} = 35000$, $M_{tip} = 0.08$, $h/R = 1.5$ and $C_T = 0.028$. 103
6.4 PAXman model forces calculated using the forward flight rotor. The rotor was operating at $V_\infty = 10kts$ (μ=0.023) and $C_T = 0.0125$ 105
6.5 PAXman model forces calculated using the forward flight rotor. The rotor was operating at $V_\infty = 10kts$ (μ=0.023) and $C_T = 0.0175$ 105
6.6 PAXman model forces calculated using the forward flight rotor. The rotor was operating at $V_\infty = 40kts$ (μ=0.0932) and $C_T = 0.0125$ 106
6.7 Total PAXman model forces calculated for small to full scale, the rotor was operating in forward flight at μ=0.023, $C_T = 0.0125$. 107
6.8 PAXman model forces distribution calculated for small to full scale, the rotor was operating in forward flight at μ=0.023, $C_T = 0.0125$. 108

7.1 Particle tracking flow chart of the particle track tool. 110
7.2 Search algorithm for particle tracking. 112

7.3 Lagrangian particle tracking and Eulerian model validation. Experiments (black dots) [12] vs present study results (dots/isosurfaces with contours). The rotor was operating at $\mu = 0.058$, $C_T = 0.017$. The employed particles properties are listed in table 7.1. 114
7.4 Lagrangian model validation. Experiments (gray dots) [12] vs present study results. The rotor was operating at $\mu = 0.058$, $C_T = 0.017$. Particle properties are listed in table 7.1. 115
7.5 Uplift criteria results for rotors at different heights above the ground, and different scaling factors. The rotor was operating $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.08$. 118
7.6 Particle paths for rotors at different heights above the ground and different scaling factors. The rotors were operating at $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.08$. 121
7.7 Particle uplift criteria results. The rotor was operating at h/R=1.66. The employes particle properties are listed in table 7.1. 123
7.8 Particle tracking results for forward flying rotors, 10 kts (μ=0.023), top view. . . . 125
7.9 Total times and speedups for particle tracking. 126

8.1 Eulerian model validation. Experiments (black dots) [12] vs present study results (iso-surfaces with contours). The rotor was operating at $\mu = 0.058$, $C_T = 0.017$. The employed particle properties are listed in table 4.1. 130
8.2 Eulerian and Lagrangian models validation. Experiments (gray dots) [12] vs present study results. The rotor was operating at $\mu = 0.058$, $C_T = 0.017$. Particle properties are listed in table 4.1. 132
8.3 Lagrangian and Eulerian cloud outline, side view. The rotor operating in taxiing, $\mu = 0.022$, h/R=1.66, $C_T = 0.0125$. No fuselage setup. 134
8.4 Lagrangian particle tracking and cloud iso-surfaces, with the rotor operating in hover, h/R=1.66. Massless particles. 137
8.5 Lagrangian particle tracking and cloud iso-surfaces, with the rotor operating in forward flight $\mu = 0.011$, h/R=1.66. Massless particles. 138
8.6 Lagrangian particle tracking and cloud iso-surfaces, with the rotor operating in forward flight $\mu = 0.022$, h/R=1.66. Massless particles. 139
8.7 Lagrangian and Eulerian results, without fuselage. The rotor was operating in forward flight $\mu = 0.022$, h/R=1.66. Massless particles. 140
8.8 Speedups Lagrangian vs Eulerian methods. 141
8.9 Timing Lagrangian vs Eulerian for different numbers of processors. 142

A.1 Uploaded unsteady flowfield data 161
A.2 Setting Field Variables in Tecplot. 161
A.3 Numbering of zones per timestep 162
A.4 Unsteady time steps 163
A.5 Generate time steps for half revolution 163
A.6 Generate time step for the full solution 164
A.7 Velocity components 164
A.8 Streamtracer option window. 165
A.9 Markers for streamtracer. 165
A.10 Streamtracer option window. 166
A.11 Particles zone in the zones window. 167

A.12 Visualization. 167

B.1 Particle tracking tool flowchard 172
B.2 Schematic representation of particle-wall interaction 173
B.3 Search algorithm for particle tracking 177
B.4 Total time for particle tracking 184
B.5 Computing normalized coordinates. 190
B.6 Imposing normalized coordinates for plots 190
B.7 Lagrangian particle tracking results for steady forward flying rotor. The rotor is operating at 10 kts (μ =0.023), $C_T = 0.0125$, top view 191
B.8 To animate flowfield and particle tracking it is necessary to list properly the files. . 198
B.9 Tecplot edit time strands window. 199
B.10 Lagrangian particle tracking results for naca0012 unsteady. 200

C.1 Grid size and boundary conditions for steady NACA0012 aerofoil. 207
C.2 Eulerian results for steady aerofoil. The aerofoil was operating at $Re = 3.8\dot{1}0^6$, $M_\infty = 0.38$, $\alpha = 10\,\text{deg}$ 211
C.3 Eulerian results for unsteady aerofoil. The aerofoil was operating at $Re = 3.8 \cdot 10^6$ and M_∞=0.38. 217

List of Tables

2.1 One-phase experiments on scaled rotorcraft operating in ground effect. 13
2.2 Two-phase experiments on scaled rotorcraft operating in ground effect. 13
2.3 Studies for the simulation of the brownout cloud and its implementation in piloted flight simulators. 27
2.4 Numerical simulations in which a Lagrangian approach is used for the prediction of the flow field. 29
2.5 Numerical simulations in which an Eulerian approach is used for the prediction of the flow field. 33
2.6 Numerical simulations of full brownout phenomenon. 40

3.1 Examples of reference variables used for non dimensionalization in HMB3. 52

4.1 Properties of particles used in this work. 63
4.2 Polynomial coefficients used in figure 4.2. 74

5.1 Geometric properties of the Ramasamy et al. rotor [36]. 77
5.2 Meshes for Ramasamy et al. [36] test case. 79
5.3 Meshes for Ramasamy et al. [36] test case. 79
5.4 Geometric properties of the rotor by Lee et al. [26] 86
5.5 Chimera meshes for the Lee et al. [26] test case. M=Millions. 88
5.6 C_Q^{IGE}/C_Q^{OGE} as function of rotor height 89
5.7 Forward flight conditions and rotor geometric data. 94
5.8 Meshes for forward flight test case. M=Millions 94

6.1 Model scale thrust coefficients for different h/R. 98
6.2 Helicopter technical data [159] for three categories used in this work. 98
6.3 Normalized hover induced velocity with and without Cheeseman factor [160] 99
6.4 Full and Model Scale rotors conditions. 106

7.1 Properties of particles used in this work. 114

8.1 Simulations performed for Eulerian particle tracking. 133

B.1 Seeding entities input format 179
B.2 Variables contained in the particle tracking tool output files 183
B.3 Airfoil condition. 184
B.4 Forward flight conditions and rotor geometric data. 185
B.5 Aerofoil conditions 192

C.1 Aerofoil conditions 212

Nomenclature

Latin

A	Bangold model constant, A=0.0123
A_p	Particle frontal Area $[m^2]$
a	Speed of sound, $[m/s]$
$\boldsymbol{a}_p$	Particle acceleration, $[m/s^2]$
B	Balistic coefficient, $[kg/m^2]$
C	Particle concentration
c	Blade chord, $[m]$
c_p	Ratio of specific heat at constant pressure
C_Q	Rotor torque coefficient, $C_Q = \frac{Q}{\frac{1}{2}\rho_\infty V_{tip}^2 \pi R^3}$
C_T	Thrust coefficient, $C_T = \frac{T}{\frac{1}{2}\rho_\infty V_{tip}^2 A}$
C_D	Drag coefficient
D	Rotor diameter, $[m]$
DL	Disk loading, $[N/m^2]$
d_p	Particle diameter, $[m]$
d_i	Distance between particle and the i-th cell centre, $[m]$
e	Specific internal energy $[m^2/s^2$ or cofficient of resistution
E	Total energy per unit mass $[m^2/s^2]$, or ratio of erodible to total surface area
E_R	Total kinetic energy of rebounting particles $[kg/ms]$
V	Vertical dust mass flux, $[kg/ms]$
f	Body force vector $[N]$, or kinetic friction coefficient, or generic variable
F_1	Blending factor between models $k-\omega$ and $k-\varepsilon$
$\mathbf{F}$	Flux vector in x direction or force [N]
$\mathbf{F}_m$	Mass flux, $\mathbf{F}_m = [QQF]^T$ $[kg/ms]$
f_{PAXman}	$f_{PAXman} = \frac{1}{2}\rho V_{rad}^2 dA$, $[N/m]$
F_{PAXman}	$F_{PAXman} = \int_{A_{PAXman}} df_{PAXman}$, $[N]$
Fr	Froude number, $Fr^2 = U_{ref}^2/gLref$
g	Gravitional acceleration, $g = 9.81[m/s^2]$
$\mathbf{G}$	Flux vector in y direction,

h_{PAXman}	PAXman height scaled, m
$\mathbf{H}$	Flux vector in z direction
H	Enthalpy $[m^2/s^2]$
$\mathbf{J}_i$	Impulsive force acting at timestep i, $[kgm/s]$
$J_{X,Y,Z}$	Spatial component of Impulsive force acting on a particle, $[kgm/s]$
k	Turbulent kinetic energy, or Boltzmann coentant
K_i	Runge Kutta i-th intermediate step
k_h	Heat transfer coefficient, $[W/m^2K]$
L	Reference length,$[m]$
m_p	Particle mass, $[kg]$
m_R	Rebound particles mass, $[kg]$
M	Mach number, $M = V_{tip}/a_\infty$
$\mathbf{n}$	Unit normal vector
N_b	Number of blades
N_s	Number of particles emitted during impact
p_s	Plastic pressure of the soil exerted on particle
P	Pressure $[kg/ms^2]$
Pr	Prandtl number, $Pr = \mu_{ref} c_p/k_h$
q	Heat flux vector
Q	Rotor torque, $[N \cdot m]$ or horizontal dust mass flux, $[kg/ms]$
$\mathbf{r}$	Position vector,
$\mathbf{r}_0$	Initial position vector,
r	Radial coordinate along blade span, m
r_{max}	Maxium distance traveled by the particle during Δt, $[m]$
R	Rotor radius, $[m]$
$\mathbf{R}$	Flow residual vector
R_{air}	Specific ideal gases constant $[J/(kgK)]$,
Re	Reynolds number, $Re = V_{tip} c/\nu_\infty$
S	Surface m^2
Sc	Schmidt number, $Sc = \nu/\nu_p$
$\mathbf{S}_d$	Dust source term
$\mathbf{S}$	Source term
S_{ij}	Strain rate tensor $[1/s]$
S_{rotor}	Rotor disk area, $[m^2]$
S_p	Frontal particle area, $[m^2]$
St	Stokes number, $St = \tau_p/\tau_f$
t	Time $[s]$
T	Temperature K or Rotor thrust, $[N]$, or time $[s]$

u	Velocity x-component, $[m/s]$
$\boldsymbol{u} = [u,v,w]^T$	Flowfield velocity vector, $[m/s]$
$\boldsymbol{u}_p$	Particle velocity vector, $[m/s]$
$\boldsymbol{u}_R$	Rebound particle velocity vector, $[m/s]$
$\boldsymbol{u}_S$	Emitted particle velocity vector, $[m/s]$
u^*	Friction velocity, $[m/s]$
u_t^*	Wall friction velocity threshold, $[m/s]$
v	Velocity y-component, $[m/s]$
$\mathbf{V}(\mathbf{r}$	Local fluid velocity at point $\mathbf{r}$. $[m/s]$
V	Volume $[m^3]$
V_c	Crater volume $[m^3]$
V_{rad}	Radial velocity component, $V_{rad} = ucos(\Psi) + vsin(\Psi)$, $[m/s]$
v_i	Hover induced velocity, $[m/s]$
v_{max}	Highest value of radial velocity component, $[m/s]$
w	Velocity z-component, $[m/s]$
$\mathbf{w}_t$	Terminal velocity $[m/s]$
$\mathbf{W}$	Vector of conserved variables
x,y,z	Spacial coordinates in cartesian system $[m]$
$\mathbf{x} = [x,y,z]^T$	Fluid position vector $[m]$
$\boldsymbol{x}_p$	Particle position vector, $[m]$

Greek

α_0	Angle of impacting particle on the ground $[deg]$
β	Bangold model constant $\beta = 0.0003 kg/s^2$
β^*	Closure coefficient for turbulence models.
γ	Fluid specific ratio rate,
γ_{Srtr}	Vorticity source,
ε	Rate of dissipation of kinetic energy
$\varepsilon_X, \varepsilon_Y$	Velocity factors in wall plane component direction
$\Delta(\cdot)$	Variation of $(\cdot)$
δ_{ij}	Kronecker delta,
θ_{75}	Collective pitch at three-quarter radius, $[deg]$
Φ	Multivariate normal distribution
ϕ	Volume fraction occupied by particles
Ψ	Local azimuth angle, $[deg]$
λ_i	Dimensionless hover induced velocity, $\lambda_i = \frac{\sqrt{(C_T)}}{2}$
μ	Dynamic viscosity $[kg/(ms)]$, or advance ratio, $\frac{V_\infty}{V_{tip}}$
μ_B	Mean velocity of emitted particles $[m/s]$
ν	Kinematic viscosity, $[m^2/s]$
ν_p	Mass diffusion coefficient $[m^2/s]$
ρ	Density, $[kg/m^3]$
ρ_d	Number density, $[N/m^3]$
ρ_p	Particle density material, $[kg/m^3]$
σ	Rotor solidity, $\frac{Nb}{\pi R/c}$, or ratio beteween density of the fluid and soil material
$\sigma_k, \sigma_\omega, \sigma_{\omega 2}$	Closure coefficients for turbulence models.
σ_{ij}	Variance of velocity of particles emitted $[m^2/s^2]$
τ	Shear stress $[kg/(ms^2)]$, or pseudo timestep
τ_{ij}	Viscous shear stress tensor
τ_p	Particle relaxation time, $\tau_p = \frac{\rho_p d_p^2}{18\mu}$
τ_f	Characteristic fluid time scale, $\tau_f = \frac{U_{ref}}{L_{ref}}$
τ_{ij}^R	Reynolds shear stress tensor
τ_w	Shear stress at the wall $[kg/(ms)]$
Π_{mom}	Momentum coupling parameter, $\Pi_{mom} = \frac{C}{1+St}$
ω	Specific dissipation rate $[1/s]$, or vortex particles strenght
Ω	Multivariate normal distribution
μ_B	Mean of emitted particle velocity
Σ_B	Covariance matrix of emitted particle velocity
σ_{ij}	Covariance component of emitted particle velocity

Sub and Superscript

$*$	Dimensionlal value or unsteady
$''$	Disturbance values
$'$	Fluctuating part
m,n	Time steps
∞	Freestream value
i,j,k	Indices
i	Inviscid
ref	Reference Value
ms	Model scale
fs	Full scale
tip	Blade tip value
p	Particle
air	Air
v	Viscid
T	Traspose
t	Turbulent

Acronyms

AGL	Above Ground Level
BILU	Block Incomplete Lower-Upper
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
DDES	Delayed Detached-Eddy Simulation
DL	Disk Loading
DVE	Degraded Visual Environment
DEM	Discrete Element Method
DPM	Discrete Parcel Method
DNS	Direct Numerical Simulation
GCR	Generalizaed Coniugate Residual
GPU	Graphics Processing Unit
IGE	In Ground Effect
LIDAR	Laser Imaging Detection and Ranging
MUSCL	Monotone Upstream Centred Schemes for Conservation Laws
NTRS	NASA Technical Report Server
OGE	Out of Ground Effect
PIV	Particle Image Velocimetry
RANS	Reynolds Averaged Navier-Stokes
TF	Two-Fluid
URANS	Unsteady Reynolds Averaged Navier-Stokes
VFM	Vortex Filament Method
VPM	Vortex Particle Method
VTM	Vorticity Transport Method
FAA	Federal Aviation Administration

Chapter 1

Introduction

1.1 Motivation

Brownout may occur when a rotorcraft is operating in ground proximity. This phenomenon is due to the interaction of the rotorcraft wake with the particles of a loose sediment bed (figure 1.1). This interaction can eventually cause uplift of particles (sand and dust in the brownout or snow during the whiteout) from the ground and their entrainment into the air flow. When operating in desert areas or in snowy regions, the number of particles entrained can become extremely high creating a cloud around the rotorcraft. The main effects of this cloud of particles moving around the aircraft are on the pilot's visual environment and on the rotorcraft structure and equipment.

The uplifted particles vary with the geography and geology of the region where the rotorcraft is operating, as well, as with the environmental conditions. They range from dust to small rocks in brownout or from powdery snow to ice in whiteout. This variety, increases the complexity of the phenomenon, and creating problems when trying to draw general conclusions.

The entrained particles are advected by the flow field and, eventually, collide with the aircraft structure and blades. Impacting particles can cause mechanical damage [2] (see figures 1.2). To protect the aircraft structure, a range of techniques are used, varying from using special paints [3, 4] to tape protection on the blades [5, 6]. However, these protection systems require maintenance, reduce the aircraft payload and some methods may also affect the aerodynamic efficiency of the

Figure 1.1: HH-60H helicopter landing in the vicinity of Baghdad [1].

Figure 1.2: Tail rotor blade showing severe erosion and perforation of the leading edge [2].

aircraft [7]. These particles can also be ingested by the engines leading to blockage of the air filters and mechanical wear issues [8].

Despite the aforementioned effects of the particle entrainment, the most dangerous effect of brown and whiteout is the degradation of the pilot's visual environment (figure 1.3). In specific flight and environmental conditions, the cloud created by the rotorcraft wake can completely envelope the aircraft. In these conditions, generally termed to as Degraded Visual Environment (DVE), the particles cloud causes a reduction of the pilot visibility and, consequently, increases the risk of impact against objects or ground. Moreover, the cloud moves with respect to the pilot and, this

Figure 1.3: A NH-90 in whiteout conditions [10].

motion may give false cues to the pilot about the aircraft attitude. This last characteristic of the phenomenon has led to numerous accidents, contributing to make the brownout one of the most important causes of rotorcraft man induced mishaps [9].

Due to the complexity of these phenomena and the safety issues they may lead to, brownout has been studied using different approaches, from full scale aircraft tests [11, 12], to model-scale isolated rotors [13]. In the first case, during experiments, it is possible to replicate the real operational conditions the aircraft may encounter. In general, however, measurement techniques used in full-scale experiments, lack high resolution, and cannot provide a detailed view of the phenomena involved. On the other hand, model scaled studies can be performed in a laboratory, within a controlled environment using high resolution measuring techniques. However, due to the limited size of the rotors, the flow Reynolds number is lower with respect to full scale, leading to differences in the flowfield behaviour.

Furthermore, a range of existing numerical works focus on the possibility to predict the cloud generated by different configuration of aircraft (single rotor, tandem or tilt rotor) or flight configuration (different advance ratios and hover) [14, 15, 16]. Others tried to define how different pilot's manoeuvres may impact on brownout severity, [17] [14]. In computational investigations of brownout, a number of different approaches to modelling the dust cloud can be identified. The most common approach is called Lagrangian. In this case, every single particle is tracked in the flowfield during the whole simulation. This approach has the advantage to be accurate, especially

when particles interact with the ground, ejecting more particles, or when they interact with rotor or nearby structures, however it may be very expensive due to the huge amount of particles that a full developed brownout cloud contains.

On the other hand, the Eulerian approach considers the dispersed phase as a continuum, and solves a scalar equation, after the flowfield. In general, this approach has the advantage to be more affordable in terms of computational cost, but on the other hand, it is not possible to track the particles with enough accuracy to evaluate their interactions with surrounding objects and the ground. However, it may be still possible to evaluate the pilot visual environment and the DVE severity related to the clouds.

In the present work, computational fluid dynamics (CFD) and both numerical approaches have been used to predict brownout cloud generated by rotors at different configurations. A direct comparison for the two models is presented in the same framework, to define the best practice to predict brownout. Furthermore, comparisons in terms of different advance ratios, and different thrust coefficients are performed. The presence of the fuselage has also been studied, comparing cases with and without it.

Brownout may be a source of risks for aircraft crew and ground personnel, but also the IGE wake itself. Hovering and taxiing rotorcraft can be a source of problems for the nearby structures, ground personnel, and other aircraft operating nearby. The interaction between the rotor wake and the ground leads to the transition of the rotor induced flow from vertical (downwash) to radial flow (outwash). Outwash prediction is fundamental for the safety of IGE operations [18], [19]. Downwash typically affects activities directly under the aircraft (like search and rescue operations), while the outwash impacts the surrounding environment like people, equipment and structures during landing and take-off. Rotor wakes may affect also other aircraft operating nearby, like in the case of wake encounters [20]. Besides risks related to uplift of particles, in this work an analysis of safety operations is performed in terms of forces that ground personnel may experience in proximity of rotor outflows.

1.2 Thesis structure

The thesis structure is now presented:

- Chapter 1 Introduction. It contains motivation and a brief overview of the work.
- Chapter 2 Literature survey. A description of the state of art in brownout studies, including numerical and experimental works. At the end, gap in the literature highlighting need for further work, and the objectives of the thesis are clarified.
- Chapter 3 Methodology. In this chapter, the governing equations and mathematical models for fluid dynamics in this work are described. This include a description of the HMB3 solver.
- Chapter 4 Particle tracking models. In this chapter, governing equations and mathematical models used to solve uplift of particles, Lagrangian particle tracking and Eulerian model are described.
- Chapter 5 Test cases. It contains a description test cases taken into account in this work and numerical results are compared with experimental data.
- Chapter 6 PAXman results. Forces acting on human body are presented. Results are compared with safety threshold and other distance safety criteria.
- Chapter 7 Lagrangian particle tracking. Brownout clouds obtained using the Lagrangian particle tracking are presented. Results are compared with experimental results, and brownout clouds obtained from different rotor configurations are compared in terms of size and shape.
- Chapter 8 Eulerian model. Brownout clouds obtained using the Eulerian approach are presented. Results are compared with experimental results, and brownout clouds obtained using Lagrangian algorithm, furthermore, the two models are compared in terms of computational efficiency.

- Chapter 9 Conclusion and future work.

Chapter 2

Literature survey

To gain a better understanding of the research conducted in the past on brownout, an extensive literature survey has been conducted.

2.1 Classification of published works

The literature on brownout is extensive, and has to be sub-divided to be systematically analyzed. A possible classification is presented in figure 2.1. A first classification made is between experimental or numerical studies. Experiments can be further divided into model and full-scale. Simulations can be divided into flight and CFD studies.

In the following sections, the tree diagram of figure 2.1 will be used. The aim is to provide a detailed and critical overview of the literature available on the brownout which is the basis of the research work described in this thesis.

2.2 Experimental analyses on full scale rotorcraft

The experimental works considered relevant for the present investigation are those related to the flow visualization and measurements of rotor wakes in ground effect, and to the visualization and measurements of particles clouds developed by the interaction of the rotorcraft wake with a loose

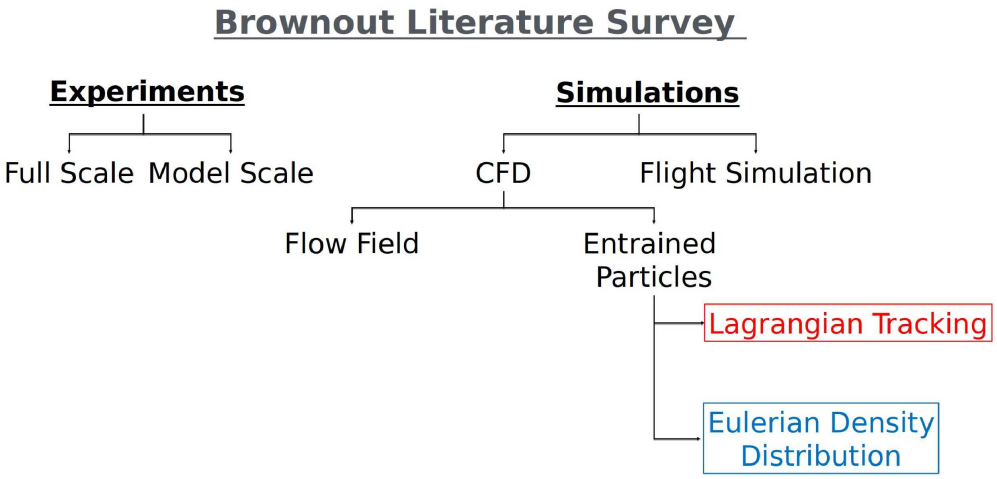


Figure 2.1: The diagram illustrates how the brownout has been investigated in the literature.

sediment bed. Unfortunately, all experiments, at full and model scale, conducted with dispersed particles used solid dust particles. No experiments have been found related to the uplift of snow flakes. This has an impact on the published numerical simulations.

Also, there is a significant lack of quantitative experimental data. In particular, few qualitative studies on the brownout have been conducted at full scale. As a consequence, the validation and development of simulations and models of brownout have been slow [12].

The aforementioned lack of data is mainly due to the following reasons. Firstly, the data of flight tests conducted mainly by the US army are rarely available and comprehensive. Secondly, these tests are expensive and also dangerous. The high cost is not only due to the operation of the aircraft in a highly wearing condition but also due to the necessity to perform a large number of tests in order to have acceptable repeatability in the data. Finally, a reliable and commonly adopted measurement technique for the flow field and the particle's trajectories and positions is still under development [12].

The majority of the available full scale tests were conducted without the presence of particles on the ground. The focus was on flow visualization measurement of the rotorcraft wake and outflow. Specifically, in Wadcock et al. [21], the downwash of a UH-60 Blackhawk, operating at a thrust coefficient of about 0.0065, was visualized with a tuft grid on the ground and on the fuselage.

The main finding drawn of the tuft study was the identification of a stagnation line on the ground at each side of the fuselage with zero lateral velocity. No quantitative data was provided.

Focusing on the outwash, an analysis on the CH-47D tandem rotorcraft were performed by Silva et al. [19]. This aircraft is well known for the development of extended brownout clouds. It was operated at a gross weight of 18,000 kg and a 20 m AGL (Above Ground Level) hover height during low ambient wind conditions. The measurements were done using a movable vertical array of anemometers. Similar studies were conducted on H-60, CH-53E, XV-15 and V-22 at different heights and weight. The outflow profiles of these aircraft are collected in Appendix K of Preston et al. [18] of which an example is given in figure 2.2. That work also focused on another aspect of safety operations nearby a lifting rotor IGE, including forces acting on ground personnel. These will be discussed in the following sections.

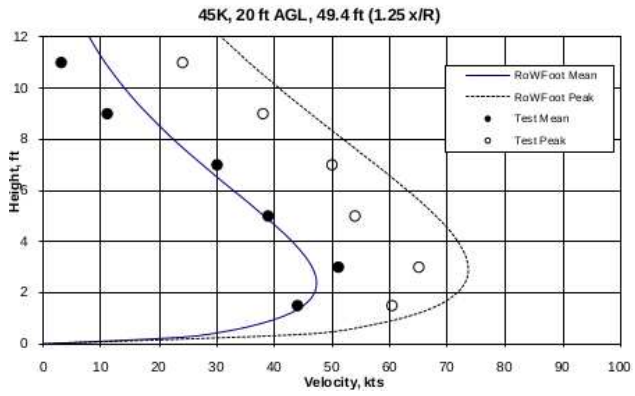


Figure 2.2: Outflow profile of a CH-53E hovering at AGL=20 ft with a GW of 45,000 lb. This profile is measured at a distance of 49.4 ft from the rotor axis (i.e. 1.25 x/R). In the figure the measured mean and peak velocity are used to validate the numerical computation described in Preston et al. [18].

More recent techniques expanded the measurements of rotorcraft flow fields. In particular, LIDAR and ultrasonic anemometry was extensively used in the H145 wake measurements by Sugiura [22]. The aircraft was operated between 0 and 40 kts taxiing speed, 1 m of skid altitude, and at a thrust coefficient of 0.012. The LIDAR data is useful to identify the position of the aircraft wake while the anemometers provide the time history of the velocity components at four locations.

The aforementioned lack of data is evident for measurements of brownout clouds. An affordable technique for the measurement of the position and velocity of dust particles during brownout

flight tests is currently missing. Up to now, the techniques used are: air sampling and photogrammetry. The former measurement technique was used during the Sandblaster testing program [23]. UH-I, CH-46, CH-53, V-22, HH-60, and MH-53 aircraft were operated in hover-taxi maneuver. An array of high-volume air samplers were deployed at fixed positions on the center-line of the flight path and downwind along the rotor-tip path and at distances of 18 m and 35 m downwind of the rotor-tip path. This technique was able to measure the concentration of particles and quantity their size distribution. The program concluded that higher cloud densities are associated with larger airframes and rotor disk loadings. Moreover, stronger downwash/outwash air currents are more effective in entraining large particles into the dust cloud. Even if the data provided by the aforementioned study is sufficient and accurate, it is difficult to use particle concentration measurements to validate numerical simulations as these were integrated over time and depend on the soil composition which is generally much more various in real world case, compared to numerical simulations.

Rodgers et al. [11] proposed a similar experiment, however samplers were mounted on the aircraft fuselage and results were in accordance with Sandblaster program in terms of high disk loading configuration, and size of particles found in the samplers, moreover higher particle concentration has been identified in the proximity of rotors overlap, and lower in close to hubs. Other single rotor aircraft have also been studied, concluding that the presence of additional operating rotors nearby increases the dust concentration near the tested aircraft.

Recently, photogrammetry has been proposed a technique to measure the location, size and average convective velocity of the dust cloud. This technique determines the 3D coordinates of an object taking multiple photographs from different perspectives. Only one application of this technique is documented for brownout cloud [12], but other works used photogrammetry to study weather clouds [24]. In the aforementioned study, two rounds of measurements have been conducted to test and improve the photogrammetry technique. Between the two rounds, the number of cameras was increased from 2 to 6 and their optical quality improved.

The improved technique was able to provide the data shown in figure 2.3. This data in particular is relative to the approach to touchdown of an EH-60L. The overall position of the

dust cloud was measured at different times as well as, the aircraft's position, when visible. This data could be useful for validation studies if all the parameters of the experiment were available. Unfortunately, the helicopter weight, wind conditions, and blade control angles are not provided. It is thus very difficult to replicate the experiment.

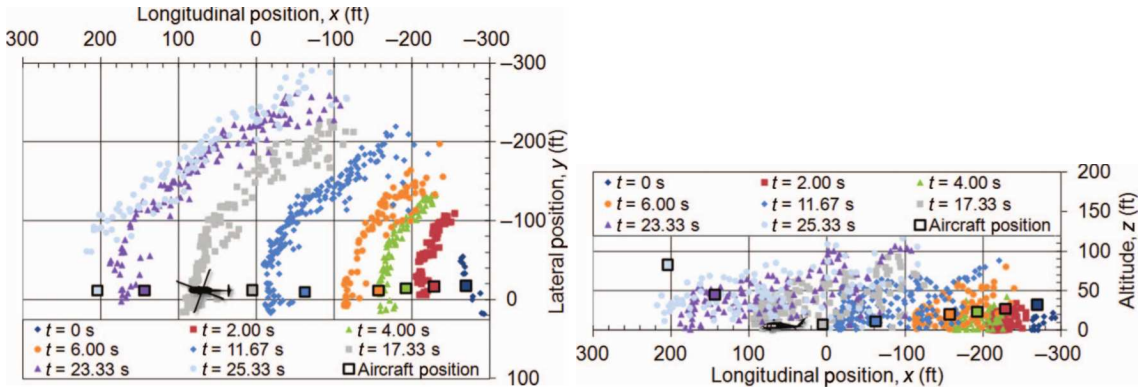


Figure 2.3: Top-down (left) and side view (right) of the dust cloud position of an EH-60L approaching touchdown, measured with the photogrammetry technique. The position of the aircraft is also measured by photogrammetry except at t=11.67 s and t=17.33 s when it was fully engulfed in the brownout cloud and on board data were used.It is interesting to notice that the cloud quickly passes ahead of the vehicle and, by t=4 s most of the aircraft is surrounded by the cloud [12].

In the aforementioned work, an interesting improvement of the technique was proposed and tested. It is the spreading of colored chalk lines on the ground to better visualize the particle pickup process. Interestingly, during the first test, when pure colored chalk was spread on the ground, little to no chalk was lifted by the helicopter wake even when it touched down in very close proximity of the chalk line. The authors reported that when the chalk was spread, it had a tendency to lump. This lumping tendency was thought to have prevented the chalk from becoming entrained into the flowfield.

During the following tests, the chalk was mixed with sand to have closer properties to the surrounding sand. The chalk entrainment was improved, obtaining the dust cloud shown in figure 2.4. Interestingly, and contrary to the author's expectation, less than 3 mm of chalk was removed after three landings and almost all of the chalk/sand mixture was in its original position on the ground while chalk was clearly visible in the cloud. These observations imply that the pickup forces for the cloud are very low.

Figure 2.4: Brownout cloud developed by a EH-60L during approach to touchdown. It is evident the pick up of blue chalk which was spread on the ground in a line perpendicular to the flight path [12].

2.3 Experimental analyses on model scale rotorcraft

Model scale rotor studies have been also conducted in laboratories. The objectives of these analyses are generally different from those of the full scale studies. On one hand, full scale experiments investigate the general development of the wake and/or brownout cloud for a given aircraft, eventually varying the performance and/or operating parameters (weight, attitude, flight path, etc.). On the other, laboratory studies are generally more oriented to the understanding of the physics of the rotorcraft wake in ground effect and the particle uplift from the ground as well as of the impact of the design and operating conditions on the phenomenon.

This divergence of the objectives is due to the differences between model scale and full scale experiments. First of all, model scale experiments are generally less expensive and, particularly, once the model is made and the measurements equipment installed, the cost of repeating the experiment is low compared to the initial investment. Secondly, in the laboratory each element of the rotorcraft can be isolated and studied separately. Thus, the majority of model experiments are related to the study of the wake of isolated rotors, as shown in table 2.1 and table 2.2 . This approach allowed the researchers to directly investigate the effects of the rotor design and operating conditions on the wake and dust cloud generation and growth. Finally, well established techniques

are available for accurate measurements of flow fields (namely PIV and hot wire anemometry) and dispersed particles trajectories. These advantages of the model scale compared to the full scale resulted in a large number of publications in this area.

Table 2.1: One-phase experiments on scaled rotorcraft operating in ground effect.

Authors (Year)	Rotorcraft	N. of Blades	Frequency [RPM]	Height/Radius	Scale/Radius	Method
Ramasamy and Yamauchi [25](2017)	Tandem helicopter CH-47D	3	3540	0.578	1/56th scale	PIV
Lee et al. [26] (2010)	Isolated rotor	2	2100-3600	2.5-0.25	86 mm	PIV
Milluzzo et al. [27] (2010)	Isolated rotor	1	1860	1	408 mm	FV and PIV
Milluzzo et al. [28] (2016)	Isolated rotor	2	2100	1	408 mm	FV and PIV
Ganesh et al. [29, 30] (2005)	Isolated rotor in forward flight	2	0-2100	0.72	457 mm	Hot wire
Bourne et al. [31] (2014)	Two isolated rotors with different chords and pitch angles	2	200-1000	1.5-2	70 mm	Stereo and Planar PIV
Curtiss et al. [32] (1984)	Isolated rotor in forward flight	4	2820	0.46-0.9	1.2 m	FV and six-component balance
Tanner et al. [33] (2015)	Rotor and NASA ROBIN-Mod7 fuselage	4	1150	0.87-2.09	1.69 m	FV, PIV, pressure sensitive paint, Pitot static probes
Rauleder et al. [34] (2014)	Isolated rotor	1	1860	1	408 mm	PIV

Table 2.2: Two-phase experiments on scaled rotorcraft operating in ground effect.

Authors (Year)	Rotorcraft	N. of Blades	Frequency [RPM]	Height/Radius	Scale/Radius	Method
Sydney et al. [35] (2011)	Isolated rotor [36]	1,2[36]	3600, 4500	2-0.25	85 mm	TR-PIV
Nathan and Green [13] (2009)	Isolated rotor in forward flight	2	5640	2-0.5	75 mm	FV and PIV
Johnson et al. [37] (2010)	Isolated rotor	2	3000	1	86 mm	FV and PIV
Glucksman-Glaser [38] (2013)	Isolated rotor in water	2	84-450	1	85 mm	FV and PIV
Rauleder et al. [34, 39, 40] (2014)	Isolated rotor	2	3600	1	85	PIV
Whitehouse et al. [41] (2010)	Isolated rotor in forward flight, rotor moving on a rail	4	2300		254 mm	Turbidity measurements

However, model scale experiments are not free of disadvantages. The first issue is the difficulty of replicating the relative movement between the ground and the aircraft. In fact, a common wind tunnel is difficult to use as the wind would entrain particles form the ground reducing the fidelity of the experiment [13] and a moving floor on the ground is difficult, if not impossible, to use with particles on it. However, the main issue related to the experiments on brownout at model scale is the scaling of the experiment. When scaling a rotor, two approaches are generally used: Mach scaling or Reynolds scaling. The former approach imposes the same Mach number of the full scale rotor at the tip of the rotor blade. While using the latter approach, the tip Reynolds number is kept equal to the full scale.

If particles are deposited on the ground to simulate the brownout conditions, a new scaling

issue arises. The particles size, shape and density have to be selected in to replicate the full scale phenomenon. The choice of particle characteristics is influenced by several factors. Their geometry and weight, influence their behaviour when entrained and uplifted. For example, too small particles have the tendency to clump together not replicating the real phenomenon accurately. Another issue is related to the measurement technique. If dual phase PIV flow measurement technique is used, the particles should be large enough to scatter more light than the smoke tracer particles and then be distinguishable [37]. The main problem is still related to the parameters to use for scaling the experiment.

Several different scaling parameters have been proposed and used [42]. The most frequent are the particle Reynolds number, the Stokes number $St = \tau_p/\tau_f$ (where τ_f is the characteristic fluid time scale, generally based on peak-to-peak velocity across the tip vortex, and the vortex core diameter and τ_p is the particle relaxation time [40]) and the Froude number. Tests in water tanks have also been conducted to try to better match the full and model scaled experiments. Also, using heavy metal particles the relative density between the fluid and the particles could not be close to the full scale [43]. Unfortunately, a common approach has not yet been adopted in the scientific community for the experiments [44]. Scaling issues are not affecting single-phase experiments. Thus, several researchers preferred to focus on the simplified one-phase flow assuming that the effects of the dispersed phase on the flow field is negligible. These experiments have been conducted, generally, to better understand the interaction between the rotorcraft wake and the ground surface. While table 2.1 summarizes the experiments, the conclusions drawn are here discussed and commented.

The influence of the ground plane on the time-averaged velocity field beneath an hovering rotor has been qualitatively characterized by a number of works. The wake produced by the rotor initially contracts radially causing an increase of the magnitude of the flow velocities within the boundary. Then the wake begins to interact with the ground. The ground plane forces the wake to turn radially outboard developing a wall-like jet, the outwash [26].

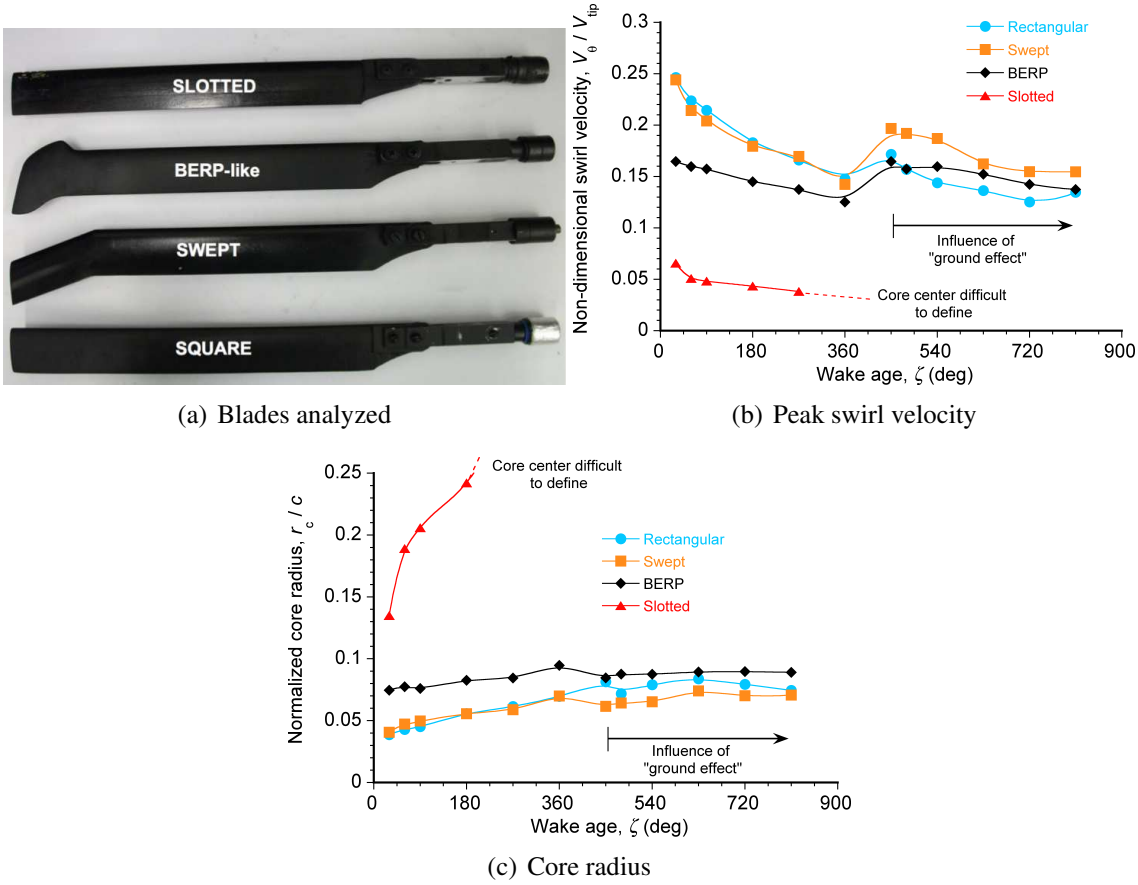
A small portion of model-scale experiments investigated the rotorcraft outwash. This branch of investigation is related to the corresponding full scale experiments described in the previous

section. In particular, Ramasamy et al. [25] replicated the full scale experiment on the CH-47D of Silva et al, experiments, [19] at model scale while Tanner et al. [33] investigated the shape of the outflow of a rotor operating over a fuselage. A linear relationship between rotor height and fuselage download was experienced. The outwash was found to be dependent on the rotor height but, interestingly, changes in thrust were only affecting the extent of the high velocity region inside the wall jet.

While the wall jet shape has been extensively studied in full scale analyses, the aforementioned experiment and several other scaled-rotor experiment studies are related to the impact of design and performance parameters on the rotorcraft wake in ground effect. These studies, as mentioned in [45], tried to investigate the effects of the blade design on the rotor wake and whether the magnitude of the effects caused by the presence of the ground plane is influenced by a number of rotor parameters.

Consequently, an important branch of research investigated the blade design and operational conditions impact on the IGE wake. In particular, in Milluzzo et al. [27] the effects of different blade tips, namely the four shown in figure 2.5, have been studied. On one hand, the flow of rectangular, swept, and BERP-like blades exhibited similar characteristics. Their tip vortices were convected downstream diffusing vorticity due to turbulence and viscosity. During the process, they interacted with their respective vortical sheets which were partially entrained into the vortex core region and relaminarized. Close to the ground, the vortices were stretched and, thus, their vorticity was reintensified and their swirl velocity increased. On the other hand, the slotted tip-blade produced a substantially different vortical wake. This blade enhanced the vorticity radial diffusion and substantially reduced the swirl velocity compared to the other three blades. At older wake ages, regions of concentration of vorticity were difficult to define resulting in a reduced overall intensity of the flow at the ground and lower velocity excursions. However, this blade may not be suitable for practical application, especially in brownout conditions where the slots could get closed by the-dust particles.

A similar approach was used by Bourne et al. [31]. Instead of using different tips here the wake of two two-blade rotors with different chords and collective was investigated. The main ob-



(a) Blades analyzed

(b) Peak swirl velocity

(c) Core radius

Figure 2.5: The four blades analyzed by Milluzzo et al. [27] (a) and the correspondent measured history of the tip vortex characteristics as a function of wake age: (b) Peak swirl velocity; (c) Core radius.

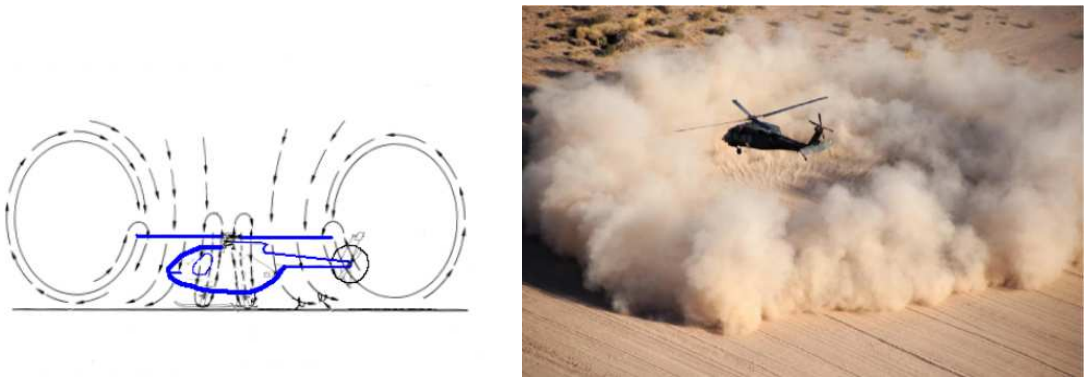


Figure 2.6: Schematic representation of the flow surrounding an helicopter IGE according to Bourne et al. [31] (left) and brownout dust cloud developed during a flight test. From the figure it is possible to appreciate the toroidal shape of the cloud generated by this particular aircraft during hover IGE.

jective of this study was to analyse whether a global recirculation flow structure is created by a rotorcraft hovering IGE and how it is influenced by the blade chord, thrust coefficient, height, and rotational speed. This structure would be responsible of the toroidal shape of the dust cloud shown in figure 2.6 often visible during flights. From the measurements made in this work which are shown in figure 2.7, it is possible to conclude that at specific rotational speeds and heights a global recirculation flow structure was generated beneath the IGE rotor. Moreover, when comparing the mean velocity field with the instantaneous velocity field it is evident that the recirculation structure was meandering over time. However, the dependence of the phenomenon on the operational parameters and blade design was not clear from this publication.

As in the aforementioned works, several researchers investigated the dependence of the wake on the rotor height. From all these studies, it is possible to draw the following general conclusions:

1. It is commonly accepted that there is a reduction of the rotor power required for a constant thrust and an increase in the rotor thrust for a constant power during IGE operations. The magnitude of this effect is clearly dependent on the rotor height over the ground plane as shown in figure 2.8 where the measurements by Lee et al. [26] and Fradenburgh et al. [46] are compared. The increase in performance is generally reported to improve with the disk loading. Interestingly, according to Lee et al. [26], these results are typical for any hovering rotor operating IGE independently from the

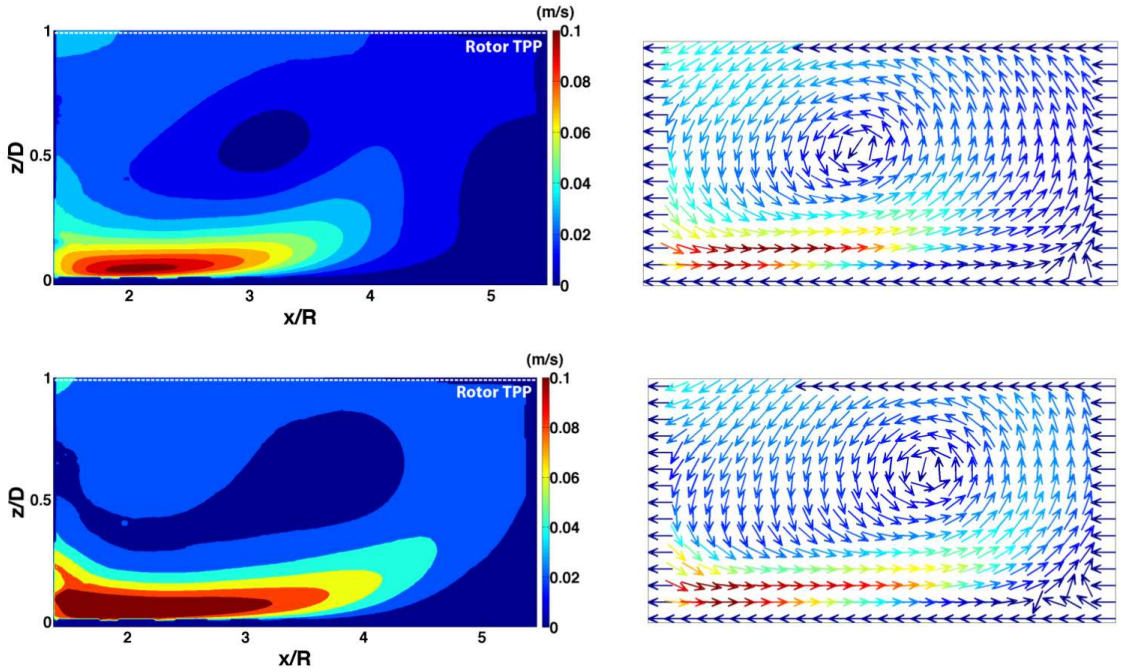


Figure 2.7: Stereo PIV mean velocity contour plots (left) and corresponding velocity flow fields (right) generated by a two-bladed rotor proposed by Bourne et al. [31].

scale and operational Reynolds number of the rotor;

2. As the rotor get closer to the ground the vorticity of the vortices at the moment of reaching the ground increases because less vorticity is diffused compared with higher rotor heights. Moreover, at low rotor heights the vortex filaments are subjected to axial stretching as they approach the ground. This vortex stretching partially counteract the vorticity diffusion in the region of the impingement of the tip vortex and vortex sheets with the ground. The ground was also found, to suppress the natural tendency of the vortical sheet to develop waves;

3. Local flow velocity peaks have been found when adjacent tip vortices pair near the ground and, eventually, merge. The vortex merging is countered by the shearing action of the developing wall jet. But, at low rotor heights, when the vortices retain significant vorticity, the vortex merging is likely to happen inducing large fluctuating shear forces on the wall;

4. Some authors identified a critical height at which the diffusion and shearing mechanism

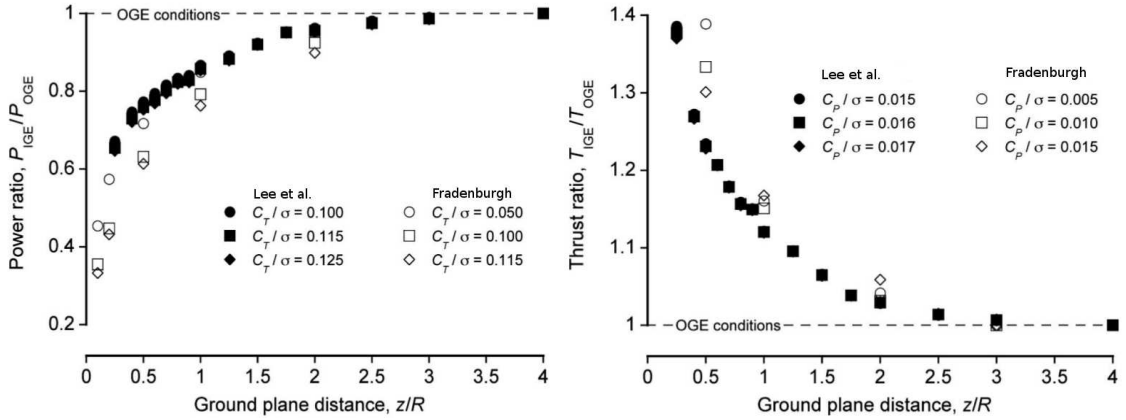


Figure 2.8: Measurements of power (left) and trust (right) of a rotor hovering IGE at different height over a ground plane. The rotor was kept at constant blade pitch. On one hand, the power measurements show a steep reduction of the power below one radius of height and a dependence of the performance on the operating state of the rotor with higher blade loading having more significant effects on the IGE performance. On the other, the measurements show little effect on thrust for more than three rotor radii of height, while below one radius the thrust increased more rapidly.

are in balance with the vortex stretching resulting in a maximum flow velocity at the ground plane. The value of this critical height is related to a specific rotor and operation condition.

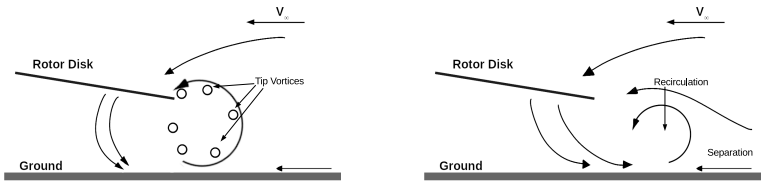
The aforementioned studies analyzed the effects on a rotorcraft wake of a ground plane and how the velocity field is influenced by design and operational parameters. These one-phase experiments are extremely important in understanding the flow field beneath a rotorcraft in ground effect. However, to go a deep understanding of the brownout phenomenon it is necessary to look at two-phase experiments. In these experiments, particles are dispersed on the ground below the model and the flow field and dust cloud development is investigated. An overview of the two-phase experiments available in the literature is given in table 2.2.

The majority of two-phase experiments are related to isolated rotors in hover. This is due to the mentioned difficulties in reproducing the relative speed between the sediment bed and the rotorcraft. Nevertheless, few forward flight experiments were conducted, namely by Nathan and Green [13] and Whitehouse et al. [41]. Particularly, the former tested a rotor in a closed, non-return, wind tunnel. A layer of talcum powder was spread on the floor of the wind tunnel upstream of

the rotor. The normalized advance ratio varied from 0.3 up to 1.3. The authors referred, without motivating the statement, that the particles were uplifted by the rotor induced flow alone. In the latter experiment, instead, the rotor was mounted on a rail and moved over a sediment bed. Because of the testing facility, limitations of their maximum advance ratio achieved was 0.05. However, the phenomenon of forward flight over a sediment bed was properly reproduced. Unfortunately, no detailed experimental results were presented.

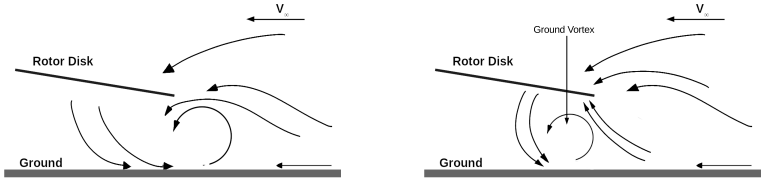
Thanks to the aforementioned studies, as well as other detailed experiments, namely by Curtiss et al. [32] and by Ganesh et al. [29], on rotors IGE at low advance ratios, the aerodynamics of this operating condition has been characterized. The general flow structure is highly dependent on the advance ratio as shown in figure 2.9. In particular, at low advance ratio, there is mean flow separation and a region of mean positive (anti-clockwise) vorticity ahead of the rotor. As the advance ratio increases, the region of positive vorticity increases in strength and reduces in size while getting closer to the rotor leading edge. Above a first critical advance ratio, the value of which depends on the rotor design and performance, the recirculation regime appears. In this flow regime the trailing tip vortices are re-ingested through the rotor. At higher advance ratios the onset of a different regime appears commonly known as the ground vortex regime. This regime is characterized by a well-defined elliptically-shaped horseshoe vortex under the rotor. Compared to the previous regime, the latter is more steady and maintained until the advance ratio gets close to one. As the advance ratio increase, the ground vortex disappears and the wake flows downstream.

While the described flow field of a rotor in forward flight is hardly analyzable with sediment particles, the analysis gets easier if the rotor is operated in hover. Thus, the fluid dynamics mechanism responsible for the movement of particles below a hovering rotor has been characterized; a schematic representation is shown in figure 2.10. The first mechanism involved is the uplift of steady particles from the sediment bed. These particles begin to move once the shear stress at the wall becomes sufficiently high. The heavier particles just roll along the bed, initially, in a mechanism known as creep. However, if the particle encounters an upward flow velocity, they can be rapidly entrained in the flow and, then, fall back on the ground as a consequence of the gravitational force acting on the particles. These particles bombard the bed causing the ejection of more



(a) Critical advance ratio, recirculation regime

(b) Advance ratio increases, a ground vortex ahead of the rotor appears.



(c) Advance ratio increases more, the ground vortex come closer to the rotor.

(d) Advance ratio increases and the ground vortex is under the rotor.

Figure 2.9: Different advance ratios regimes.

particles into the near wall flow. This is a cascade process which leads to the creation of a saltation layer in which the particles can be easily entrained by any passing vortical flow [47]. Moreover, the cloud development is also influenced by entrained particles which are reingested by the rotor and hit the ground at high speed causing the ejection of many other particles.

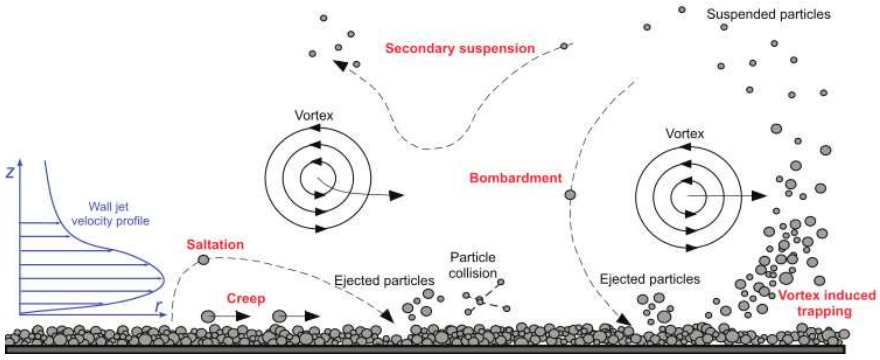


Figure 2.10: Schematic showing the different particle motions and the fundamental uplift mechanisms responsible of the brownout cloud [47].

As shown in figure 2.11, the uplift of particles is not uniform around the aircraft but it is a complex three-dimensional phenomenon. This is because, the previously described uplift process

Figure 2.11: Helicopter landing in brownout conditions [48]. In the figure are pointed out with black triangles the plume-like dust uplifting events in the cloud. The cloud is clearly three-dimensional and highly unsteady.

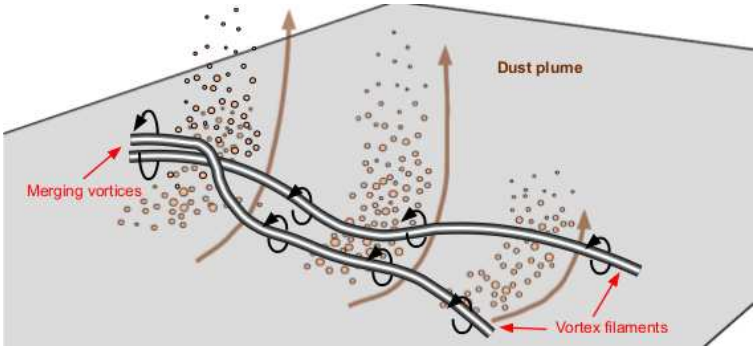


Figure 2.12: Schematic representation of the interaction between consecutive vortex filaments [48]. This interaction is one of the causes of locally higher near-ground velocity which, in turn, leads to the discrete uplift of sediment particles.

is dependent on the local flow field close to the ground, i.e. the wall shear forces and pressure gradients close to the sediment bed, which is highly three-dimensional. According to Syal and Leishman [48], this three-dimensionality is due to the behaviour of the tip vortices close to the ground. When the vortex filaments reaches the ground, they interact with the ground and each other, however this interaction seems not to be periodic. Only local parts of each filaments are merged with each other, generating strong three-dimensional phenomenon as figure 2.12. These variations can affect the wall boundary layer. In this scenario it may locally thicken and, separate from the wall generating secondary negative vorticity. This negative vorticity accelerates the distortion and diffusion for the vortex filaments when entrained and it increases the mobilization and uplift of sediment and particles [39].

While the structures dominating the flow are, undoubtedly, the tip vortices, of which the behaviour has been discussed in the present and previous sections, secondary vortical structures and turbulent motions also play a role in the dust cloud development [40]. In particular, a tip-vortex-induced, clockwise-rotating vortex structure can be identified in the flow. These structures convect downstream in between two consecutive blade tip vortices as shown in figure 2.13. It is clearly visible the secondary counter rotating vorticity structure in between the two tip vortices. Interestingly, the region of higher particles entrainment is correspondent to distances from the rotor axis higher than 1.6 r/R despite the vorticity of the vortices at this radius has significantly dissipated. It should be noted the entrainment by the tip vortex of the negative vorticity on the ground and their interaction. These secondary vortical structures are mainly responsible for a downstream ejection motion of the particles which significantly enhance the particles uplift and entrainment.

An important role is also played by turbulence. In particular, the turbulence structures close to the sediment bed are highly anisotropic because of the presence of the wall and the tip vortices. Noticeably, a coincidence of positive wall-normal velocity fluctuations and sediment particles suspension regions were measured [40]. The origin of these strong turbulent motions is mostly in the tip vortices and not in any naturally occurring flow structure near the wall. Because turbulent fluctuations were found to significantly affect the sediment uplift and entrainment.

As described in the present section, extensive work has been conducted to understand the effect of the flow field on the sediment particles. Few works though tried to quantify the effects of the entrained particles on the flow field. These investigations, albeit few in number, are necessary to decide whether the impact of the entrained particles on the carrier flow can be neglected. According to Rauleder et al. [39], the tip vortices and the secondary negative vorticity are the primary sources of anisotropy in the production of turbulence. This anisotropy in the turbulence field is enhanced by suspended particles after the tip vortex vorticity has sufficiently diffused. The uplifted particles are also responsible of attenuating the turbulence levels both in high and low concentration regions. Moreover, the coupling between the two phases produces significantly different distributions of Reynolds stresses when compared to the single-phase flow [34]. The results of the

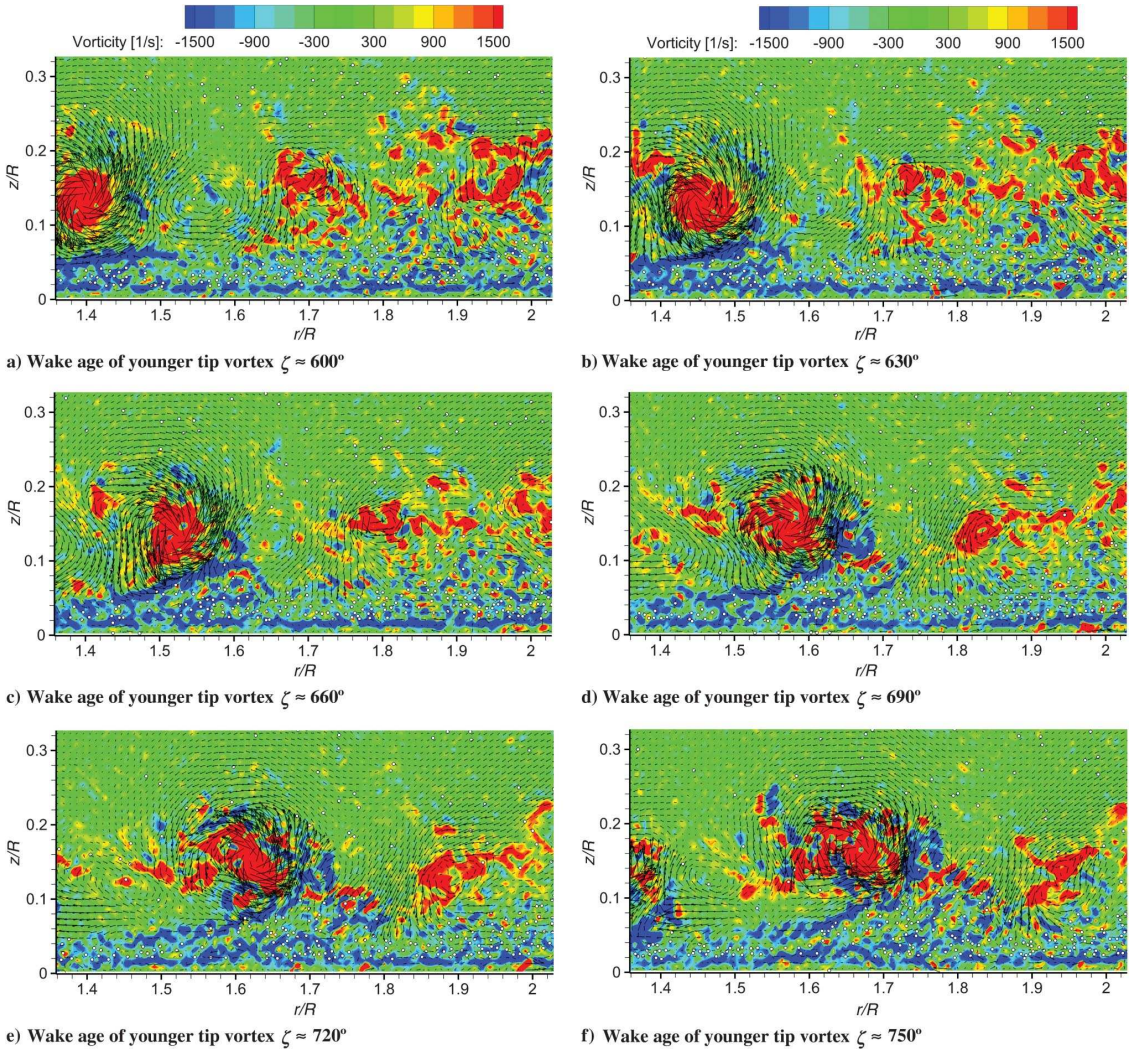


Figure 2.13: Consecutive PIV measured velocity fluctuations vector field (Reynolds decomposed) superimposed by the particle distribution on a background contour showing the instantaneous out-of-plane vorticity [40].

aforementioned studies have important implications for numerical simulations as explained in the following sections.

To conclude, the rotor/airframe interactions are discussed. Unfortunately, few works covered this in the context of brownout and, in particular, the investigation in Sydney et al. [49] is the most relevant. In the aforementioned work the influence of three different fuselage bodies on the rotor wake and sediment particles entrainment was investigated. A strong interaction was measured between the rotor wake and the bodies tail, while low interactions were found in the correspondence of the nose. The general effect of the bodies is to distort the rotor wake development causing lower velocities on the ground. Consequently, the particle was uplifted closer to the rotor. The rectangular body shape, which caused the lowest flow velocities at the ground, showed the highest concentration of suspended particles. The near-wall flow was highly three-dimensional with localized regions of higher flow velocities and more sediment uplift.

2.4 Numerical computations for flight simulations

A broad research field concerns the numerical simulation of brownout. The numerical simulations can generally be divided in two branches whether the objective is the development of a visual representation of the particles cloud for its implementation in flight simulators or a deeper understanding of the physical phenomenon. An overview of the former research branch is given in the present section, while the latter is described in the following one.

The development of a numerical simulation for the implementation in piloted flight simulators is one of the first numerical efforts made to reduce the hazard of flying in brownout conditions. In fact, since the first operations of the US army during the Gulf War, a lack of pilot training was recognized. It was due to the fact that the pilots had never experienced such flight conditions before [50]. The training of pilots with flight tests, is both dangerous and expensive because of the aircraft structure erosion and possible mishaps. Thus, a tool to safely train the pilots to avoid the hazards and costs of real flight tests was required. Table 2.3 summaries the literature on this subject.

The developers of these simulators were, then, mainly focused on a tool for the creation of

a realistic dust cloud rather than investigating the actual physics behind it. Thus, the first attempts at the task were made using empirical and semi-empirical brownout visualization models. These models were lacking the fidelity required to capture the complex interactions between the rotorcraft wake, the ambient winds, and the effect of vehicle manoeuvring, in combination with the entrainment of particles and visual obscuration effects for a wide range of possible aircraft configurations, flight conditions, sediment beds and ground topology [50].

In more recent years, then, increasing attention has been paid to the development of physics-based computational tools for the simulation of brownout. Of course, the developers of these simulators have been facing limitations on the fidelity of their tools due to the typical computational performance available and the necessity to provide a real time simulation to the pilot, simple and quick models were preferred leading to lack in details and physical coherence. The simulation model needs to finish all the computations in a cycle time under 20 milliseconds to create a realistic animation [51]. Consequently, the main challenge has been to develop a realistic dust cloud at the lowest computational cost.

Obviously, because the final goal of these works was the implementation in a flight simulator and, thus, a realistic rendering of the pilot's vision, an important part of these investigations is related to the dust cloud rendering. Several computer graphics techniques have then been employed to approximate the pilot's degraded visual environment from the particle's distribution, density and characteristics. This task was simplified by the extensive literature on the visual rendering of clouds which has been an area of significant research and development [52, 53, 54, 55, 20].

While some researchers approached the mitigation of the brown/whiteout hazard developing a tool for the pilots' training, few others [56, 17] investigated whether is it possible to optimize the rotorcraft flight path to reduce the risk of mishaps performing maneuvers over a loose sediment bed. Then, flight path optimization processes were performed for maneuvers which are recognized as risky in brownout conditions, namely, approach to landing and landing.

In these optimization problems, the rotor wake corresponding to the desired maneuver is calculated using a rotorcraft flight dynamic model. Then, the flow field surrounding the aircraft

and close to the ground is determined. Thus, the resulting dust particle's motion is computed. Unlike piloted flight simulations, these simulations do not have real-time constraints. Consequently, higher fidelity CFD technique and particles entrainment models are generally implemented. These are the same used in some CFD simulations discussed in the next section and will not be discussed here.

Based on the geometry of the dust cloud, the objective function of the optimization is computed at each iteration. To come out with a meaningful study, the objective function has to be a metric of the brownout severity. A commonly accepted metric of the brownout severity is yet to be defined and accepted though [57]. In the studies found in the literature, the objective function represents the cloud volume density in the best region of the FOV that the pilot could potentially use to perceive visual cues. It has to be noticed, though, that these studies were performed within the same research group and that the implementation of different objective functions has not been evaluated.

Although the literature on this subject is limited, researchers agree on two preferred approach profiles, namely, one in which the pilot would "outrun" the cloud at shallow approach angles, and another at steeper descent angles that delayed as much as possible the moment at which the rotor wake began to interact to the ground and cause the cloud to form [56, 17].

Authors (Year)	Rotor and Fuselage Modelling	Wake Modelling	Uplift Model	Entrainment Model	Visual Obscuration Model	Validation
Wachpress et al. [50, 58] (2009)	Curved vortex element for rotor(s) and fast panel for fuselage and ground	offline FVM with Hierarchical Fast Vortex method	Bagnold model	Lagrangian trajectory analysis [59]	Particle clusters scattering model, dynamic imposter generation [53]	Time averaged outwash data [60]
Wenren et al. [61] (2006)	Lifting line	Vorticity confinement technique	Simplified Bagnold model	Eulerian density approach	Particles concentration scattering, raymarching	
Gerlach [51] (2011)	None	Permanent ground vortex and turbulence	Stochastic functions	Particle system [62]	Perlin noise [63]	
de Reus et al. [64] (2010)	Non-viscous 3-state Peters/He inflow model	Off-line free vortex wake model		Particle tracking	Adjustable semi-transparent texture	

Table 2.3: Studies for the simulation of the brownout cloud and its implementation in piloted flight simulators.

2.5 Flow Field CFD simulations

As anticipated in the previous section, in the present and next sections the literature related to the investigation of the brownout phenomenon by numerical simulations is discussed. These works

are related to the use of high fidelity CFD techniques, but there is a whole spectrum of different goals. Some researchers focused on the understanding of the interaction of the rotor wake with the ground. Some others investigated the effects of the rotor wake on the sediment bed particles trying, eventually, to develop a model of the phenomenon. Finally, related to the experimental analyses on the subject, the effects of design and/or performance parameters on the development of the brownout cloud was also numerically studied.

In Rauleder et al. [34] an investigation into the importance of the coupling between the two phases of the brownout cloud has been outlined. All the numerical simulations conducted up to now assumed a one-way coupling between the two phases. In other words, it was assumed that the flow field influences the particles motion but the entrained particles do not affect the carrier phase flow. This assumption simplifies the computation, neglecting all the coupling effects. The induced error is reasonable in regions with a low particle concentration but should be used carefully close to the ground where the concentration of particles can become extremely high, as shown in figure 2.14. Another assumption commonly adopted is the dilute flow assumption for the dispersed phase, i.e. because of the low concentration of particles in the carrier phase it is possible to neglect particle collisions. As a consequence of the application of these two assumptions, the motion of any single particle is influenced only by the flow field while the carrier phase flow is independent from the dispersed phase. These simplifications allow to discuss the carrier phase and dispersed phase simulation separately. Specifically, the literature regarding the simulation of rotorcraft wake IGE is discussed in the present section while the following sections concern the particle uplift modelling and the dispersed phase modelling.

The main challenges in developing a numerical simulation of a rotorcraft wake IGE are two. Firstly, the tip vortices have to be preserved for a significantly long time to capture their interaction with the ground. Secondly, it is necessary to accurately resolve the boundary layer and turbulence at the ground. For the computation of the flow field, three general approaches can be used: grid free (Lagrangian), grid based (Eulerian) or a combination of the two, called hybrid method. The first approach used was the former in which the tip vortex is tracked in a particle-fixed Lagrangian frame of reference. The investigations conducted using this approach are summarized in table 2.4.

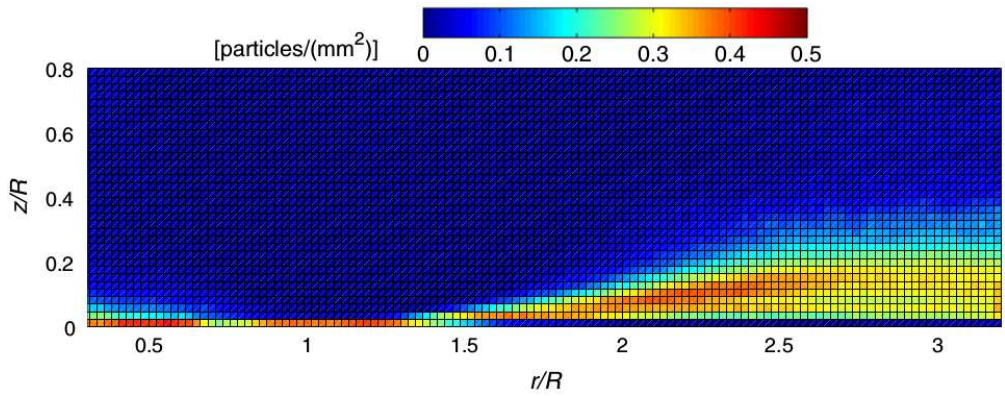


Figure 2.14: Time-averaged particle concentration close to the ground below an hovering rotor measured with dual-phase PIV [39]. Evidently, the particle concentration is, on average, higher close to the ground between 1.5 and 3 radii outboard. Although, it is important to remember that, because of the three-dimensionality and unsteadiness of the phenomenon, the instantaneous particle concentration can have, locally, much higher values than shown in the present figure.

Authors (Year)	**Method**	**Ground model**	**Blade model**
Syal et al. [65, 66] (2013)	Vortex Filament Method	Image system with logarithmic boundary layer	Weissinger-L type model
D'Andrea [67] (2011)	Vortex Filament Method	Full-unstructured panel method	Full-unstructured panel method
Wachpress et al. [50] (2009)	Vortex Filament Method	Image system or panel ground plane with vortex dissipation model [68]	Panel method
Govindarajan et al. [69] (2016)	Vortex Filament Method	Image system with logarithmic boundary layer	Lifting-line (main and tail rotor)
Tritschler et al. [56] (2014)	Vortex Filament Method	Image system	Weissinger-L type model
Ganesh et al. [29, 70] (2005)	Vortex Filament Method	Image system	Lifting-line
Keller et al. [58] (2006)	Vortex Filament Method	Image system or panel model	Vortex lattice lifting surface model
Xin et al. [71] (2015)	Vortex Filament Method	Panel method	Vortex panel method
Zhao et a. [72] (2015)	Vortex Particle Method	Image system and Viscous ground	Lifting-line

Table 2.4: Numerical simulations in which a Lagrangian approach is used for the prediction of the flow field.

These, so called, grid free or free-wake models can be further divided into potential methods and particle method.

In the potential methods, as described by Leishman in [73], it is assumed that viscous phenomena will be confined to much smaller length scale compared to potential flow phenomena. Consequently, the vorticity is all concentrated along the axis of each vortex filament, forming a vortex line singularity. Applying incompressible, inviscid and irrotational conditions, vortex lines move as material lines, and their motion is described by the motion of Lagrangian fluid markers. Under these assumptions, the convection of these markers is governed by the set of linear advection equations

$$\begin{aligned} \frac{d\mathbf{r}}{dt} &= \mathbf{V}, \\ \mathbf{r}(t=0) &= \mathbf{r}_0 \end{aligned} \tag{2.1}$$

where $\mathbf{r}$ is the position vector of a point in the vortex, $\mathbf{r}_0$ is the initial position vector of the point and $\mathbf{V}(\mathbf{r})$ is the local fluid velocity at the point $\mathbf{r}$. The velocity field results from the induced velocity of the vortex wake and any influence of surfaces and externally imposed velocity field through Biot-Savart law. The radial diffusion of vorticity and the vorticity intensification produced by filament stretching are generally incorporated by approximated models. Moreover, these methods rely on models, such as the lifting-line, for the representation of the blades and the determination of the tip vortex characteristics which are used for the generation of the vortex filaments. Figure 2.15 schematically illustrates the rotor wake modelling using VFM. Models are also necessary if it is required to simulate the interaction of the wake with surfaces (ground, airframe, etc.). Two methods are generally used: the method of images, illustrated in figure 2.16, which assures that the wake is mirrored at the plane, and the vortex panel method which applies flow calculations to finite panels and includes the induced velocity from these vortices in the Biot-Savart calculations. Both methods are not able to simulate the boundary layer, which requires a model, but while the method of images can only simulate planar surfaces, namely the ground, the vortex panel method can replicate complex surfaces.

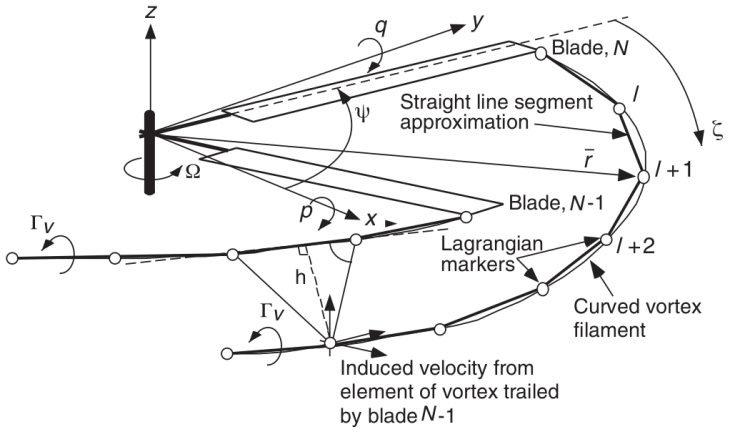


Figure 2.15: Schematic representation of the VFM [69]

Generally speaking, these methods can track the vorticity generated at the blade over long time at a relatively low computational cost. However, these methods rely on empiricism for the representation of the blade and surfaces as well as in determining the vortex-core radius and roll-up. Moreover, these methods neglect the vorticity within the vortex sheet and only the tip vortex

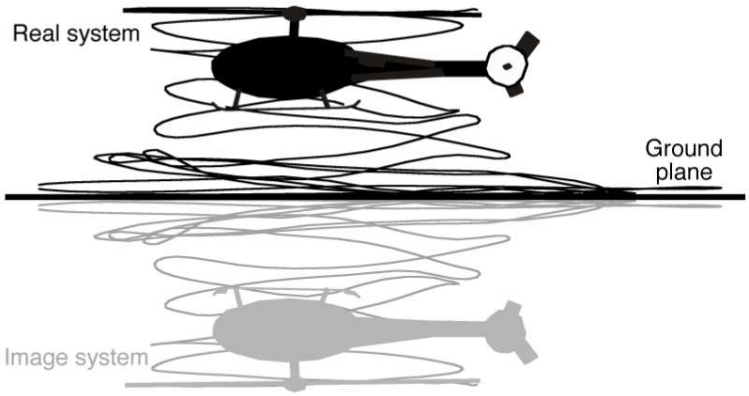


Figure 2.16: Free-vortex wake solution at a ground plane using the method of images [65]

is modelled. As mentioned, in contrast with the potential methods are the Vortex Particle Methods (VPM). In these methods the viscosity of the fluid is taken into account. Consequently, the decay and diffusion of vorticity is modelled without empiricism, as is the case of VFM. This means, however, that the complexity of the computations is increased and they are more computational expensive. VPM addresses the vorticity field solution directly from the vorticity-velocity form of the incompressible Navier-Stokes equations [72]. The governing set of equations for the vortex particle convection and dynamics is

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{u_{tot}} = \mathbf{u}_\infty + \mathbf{u}, \\ \frac{d\omega}{dt} &= \omega \cdot \nabla \mathbf{u} + \nu \nabla^2 \omega + \gamma_{S_{rtr}} \end{aligned} \tag{2.2}$$

where, $\mathbf{u}_\infty$ is the freestream velocity, $\mathbf{x}$ and $\mathbf{u}$ are the vortex particle position and self-induced velocity respectively, and ω the vortex particles strength. As the previously described VFM, VPM still require a model for the determination of the vorticity source $\gamma_{S_{rtr}}$ from the rotor. As anticipated, with both VFM or VPM, it is necessary to introduce a surface model whenever the wake interaction with a surface has to be considered. Two methods are generally used: the method of images and the vortex panel method. The former approach creates a mirror image of the above-ground rotor and its wake, and incorporates the induced velocity field of this image into the new flow field calculations [74]. The advantages of this method are its simplicity and that it does not require extra wake geometry data to be stored, even though its use is restricted to planar ground

surfaces. This method is more computationally efficient relative to other methods and implicitly satisfies the non-penetration boundary condition but cannot satisfy the no-slip boundary condition [66]. Therefore, it cannot distinguish between the scenarios with and without a ground in-plane motion [72]. Moreover, the ground boundary layer is not included in the model. On the other hand, the vortex panel method applies flow calculations to finite panels and includes the induced velocity from these vortices in the Biot-Savart calculations. It is more computationally expensive, but much more flexible in application; for example it can expand the application range of simulations to non planar surfaces like a building or an airframe. This is an inviscid treatment of the ground plane, and viscous effects (such as may be created when the tip vortices interact with the boundary layer on the ground plane) are not modeled, as with the method of images.

As described by He et al. [75], using VPM it is possible to model a viscous ground using the Brinkmann penalization technique. This technique allows to impose a no-slip boundary condition at the ground by creating vorticity from the ground surface. As shown in figure 2.17, the obtained results are in good accordance with experimental measurements. However, according to Zhao et al. [72], to capture the boundary layer above the ground requires too many vortex particles to be practical for applications.

From the description done about the Lagrangian methods it is possible to draw some general conclusions. These methods have a relatively low computational cost and do not have the problem of excessive vorticity diffusion, which is a characteristic of the Eulerian methods described later in this section. Furthermore, these methods are generally more versatile in introducing the aircraft airframe, as well as, multiple rotors in the simulation. Nevertheless, as outlined in the previous sections, in the development of the brownout cloud a fundamental role is played by turbulence near the ground, and the secondary vortex structures which cannot be simulated with the mentioned Lagrangian methods. Moreover, it is difficult to perform analyses of the impact of the blade design on particle entrainment, because most of the rotor details are not modelled with this approach.

Opposite to the Lagrangian methods are the Eulerian ones. The investigations conducted using such methods are summarized in table 2.5. From the table, it is possible to notice that in most of the simulations a RANS (Reynolds Averaged Navier Stokes) solver was used. With this

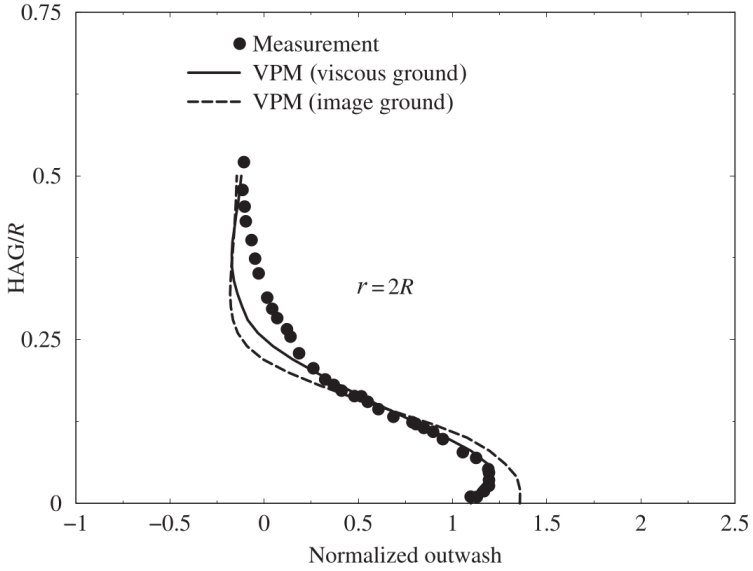


Figure 2.17: Effect of the viscous ground on the VPM-predicted time-averaged rotor wash at a radial station of 2.2R for an isolated rotor hovering at 0.5R over the ground [72]. The VPM results are compared with the measurements by Lee et al. [26]. On the y-axis the height over the ground is normalized by the rotor radius while the outwash velocity on the x-axis is normalized by the nominal induced inflow in hover ($\sqrt{C_{T_{hov}}/2}$).

method the blade and the ground plane do not have to be modelled while a turbulence model is needed. Thus, the number of empiricism is reduced. However, standard CFD methods are known to result in excessive numerical dissipation of vorticity. This is a strong limitation because, as already said, for brownout simulation it is necessary to conserve the vorticity generated at the rotor until it interacts with the ground, and even further. To conserve vorticity the generally adopted solution is to use a fine mesh along the path of the tip vortices and at the ground. As this solution increases the fidelity of the prediction, it increases also the computational cost.

Phillips and al. [16], used an inviscid Vorticity Transport Method (VTM) to model the wake of isolated rotors and full rotorcraft IGE. The VTM is somehow related to the VPM because it solves the unsteady and incompressible Navier-Stokes equations in vorticity-velocity form using structured meshes. An advantage of this method is that it captures small scale features resulting from the breakdown of larger vortical structures. The method of image is used to model the ground resulting in a finite slip velocity at the surface.

Wenren et al. [61], an unsteady and incompressible vorticity confinement method was used to

Authors (Year)	Method	Turbulence model	Rotor model
Wenren et al. [61] (2006)	Vorticity confinement	Surface confinement	Lifting-line
Wadcock et al. [21] (2008)	RANS solver	Spalart-Allmaras	Actuator disk
Haehnel et al. [76] (2008)	Vorticity confinement	Laminar flow	Momentum source
Phillips et al. [16, 77] (2010)	Inviscid VTM	None	Vorticity source region
Ghosh et al. [15] (2010)	RANS solver	$k-\varepsilon$	Momentum source [78]
McAlpine et al. [79] (2010)	RANS solver	$k-\varepsilon$	Actuator disk
Morales et al. [80] (2011)	DNS	None	Turbulent boundary layer with periodic vortex sheets
Thomas et al. [81] (2011)	Structured overset RANS solver	Spalart-Allmaras	None
Jasion et al. [82] (2012)	RANS solver	$k-\varepsilon$	Momentum sources in the cells
Lakshminarayan et al. [83] (2013)	Structured overset RANS solver	Spalart-Allmaras	None
Friedmann et al. [84] (2014)	Lattice-Boltzmann	None	None
Kutz et al. [85] (2014)	Structured overset RANS solver	$k-\omega$	None
Oruc et al. [86] (2016)	Unstructured RANS solver	Monotone Integrated LES (MILES)	Actuator disk
Kalra et al. [87] (2017)	Structured overset RANS solver	Spalart-Allmaras	None

Table 2.5: Numerical simulations in which an Eulerian approach is used for the prediction of the flow field.

model a UH-60A helicopter IGE using a lifting-line model for the rotor. This method solves the incompressible Navier-Stokes equations adding at the right hand side of the momentum conservation equation the vorticity confinement term. The equations were solved on coarse Cartesian grids. Tip vortices were shown to roll over the ground plane even if a coarse grid was used. Unfortunately, the results were not quantitatively verified.

As mentioned, a significant effort was applied to represent a rotor wake IGE using RANS solvers. The main challenge is to resolve the tip vortex evolution and its interaction with the ground. Consequently, Kalra at al. [88] used an overset grid technique around the blade and a refined background mesh along the tip vortex path to predict the wake of an isolated rotor simulation, comparing results with the Lee et al. experiments. [26]. The computed rotor performance was in good accordance with the experimental measurements but, even though the tip vortex evolution and ground interaction was resolved, the vortices were stronger than in the experiment, especially at older wake ages.

To improve the prediction of the wake outboard of the rotor, Thomas et al. [81] explored overset grids along the vortex paths close to the ground. In this work the aforementioned test case was used by Lee et al. [26]. The computed rotor performance did not change with the grid refinement but, the predicted strength of the vortices and their radial location at older wake ages showed better correlation with the experimental data.

To reduce the computational cost of RANS computations it is possible to determine regions

of the domain where a RANS computation is required to resolve viscous and turbulent phenomena and use a more efficient, less sophisticated, Lagrangian model in the rest of the domain. Applying this hybrid methodology, Thomas et al. [89] used a RANS solver in the regions near-blade and near-ground and VFM to track the tip vortex, as shown in figure 2.18, and run the same test case of Kalra et al. [88] on GPUs. The number of Eulerian grid points was reduced by 80% leading to a more efficient computations than the full RANS simulation [88]. The vortex trajectory computed with the hybrid method was in good agreement with the experimental measurements but the vortices at older wake ages were better captured by full RANS computations as well as the inboard radial velocity. The main reason of this differences is probably due to the absence the blade inboard vorticity in the free-wake model implemented.

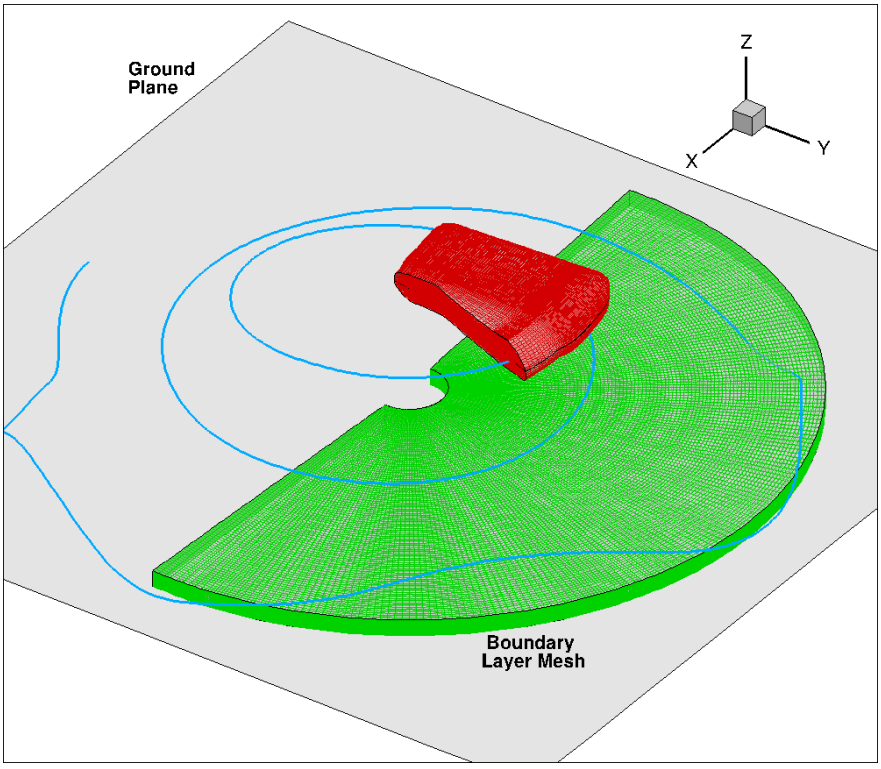


Figure 2.18: Wake-coupling methodology applied to a rotor hovering IGE in Thomas et al. work [89]. The red and green mesh represent the blade and ground mesh respectively while the tip vortex in between these two regions is tracked using VFM.

Kalra et al. [87] also used Delayed Detached-Eddy Simulation (DDES) to predict a rotor wake IGE. The decision to implement this method was taken based on the results obtained by the RANS simulation replicating the isolated rotor experiments by Milluzzo et al. [90]. The compu-

tation was conducted using the overset grid technique and the Spalart-Allmaras turbulence model. This is due to the high turbulence level at the center of the vortex cores, whereas theoretically, the vortex cores are expected to be laminar in nature [91]. After running a laminar simulation the authors identified the cause of the error in the turbulence model. Moreover, because of the important role played by turbulence in the brownout phenomenon, they wanted to accurately predict turbulence levels. To overcome the limitation of RANS simulations they explored a hybrid approach in which the near wall regions are modelled with RANS and the regions of massive separation are resolved using LES. Interestingly, the different choices of turbulence modelling did not change the thrust predictions.

2.6 Phase coupling

The present section is dedicated to answer the question, whether, in a two-phase particle-gas flow, it is necessary to take into account the effects of the particles on the carrier phase, and when these effects are negligible. To answer this question the physical model proposed in Giacinto et al. [92] for dilute gas-particle flow is presented and discussed. Dynamic and thermal equilibrium between the two phases is assumed, to consider the solid phase as a single homogeneous fluid with modified properties with respect to the gas flow. Moreover, the suspension is assumed to be dilute, i.e. composed of non-interacting particles, and the effective transport coefficients are calculated from the disturbance flow field generated by a single particle.

Let the solid particles be spherical with diameter d_p [m], constant particle material density ρ_p ($\frac{kg}{m^3}$), and hence with identical volume V_p and mass m_p. We can define the particle response time as: $\tau_p = \frac{\rho_p d_p^2}{18\mu}$ [s], where μ is the fluid viscosity. It is possible then to define the Stokes number as $St = \frac{\tau_p}{\tau_f}$ (dimensionless), where τ_f is the characteristic time of the flowfield, defined as $\tau_f = \frac{L_{ref}}{U_{ref}}$ [s]. In general, if $St << 1$, particle have enough time to respond to changes in the flowfield velocity, this leads to particle velocities close to velocity of the flowfield. On the other hand, when $St >> 1$, particles does not have time to respond to flowfield changes, and this mean that the particles are

poorly affected by flowfield [93]. To evaluate the applicability of this assumption an estimation of the volume fraction of the carried phase in a brownout cloud is conducted. The coupling can take place through mass, momentum and energy transfer between phases. In the brown/white-out there is no exchange of mass between the phases (for example evaporation) or heat transfer. Consequently, only momentum coupling is investigated.

According to Crowe et al. [94] and Noruozi et al. [93], the importance of the momentum coupling between the two phases can be assessed by comparing the drag force due to the dispersed phase with the momentum flux of the continuous phase, they obtained the momentum coupling parameter, defined as $\Pi_{mom} = \frac{C}{1+St}$. where C is the particle concentration, and St is the Stokes number. If $\Pi_{mom} << 1$, the momentum coupling is negligible and the one-way coupling assumption is valid. This means that the one way coupling can be used in case of very small particle concentration (like in the case of brownout/whiteout), or in case of large Stokes number. For example, the velocity of a high density particle (so with large St) is unaffected by the flowfield. On a similar way, the flowfield is unaffected by a low concentration dispersed phase, with small St number. It is important to point out that the obtained assumption may not be valid close to the ground, where the concentration is higher and is obviously dependent on the flow field and particle properties. Thus, it is necessary to analyse the results obtained from each simulation to check if the assumptions applied are valid for that specific case.

2.7 Dispersed phase modelling

An important decision when modelling a two phase flow is how to model the dispersed phase, whether sand, snow or water. There are basically three approaches for the numerical simulation of a cloud of particles as shown in figure 2.19. In the Discrete Element Method (DEM), figure 2.19 (A), the motion of the particle is analyzed by incorporating the fluid dynamic forces, the contact forces and the moments due to the neighboring particles. The properties of the particle cloud are then obtained by solving the equations of motion for all elements in the field. In the method shown in figure 2.19b, the Discrete Parcel Method (DPM), a parcel of particles which move through the

field is identified. Each particle of the parcel has the same dynamic properties (size, velocity, etc.) so the parcel is represented by one computational particle. Solving the properties of the computational particles as they move through the field gives, the local properties of the cloud. The third approach shown in figure 2.19c is the Two-Fluid (TF) model. In this case, the properties of the particles are assumed to be continuous within the field. Thus, differential conservation equations are written, discretised, and the solution of the resulting set of equations gives the properties of the cloud.

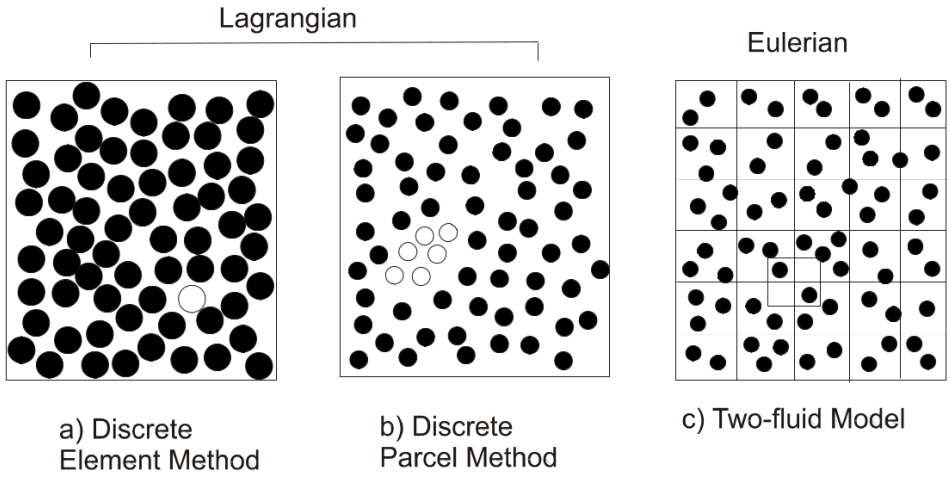


Figure 2.19: Different approaches for modelling particle and droplet clouds [94]

.

The mentioned methods can be further categorized as Lagrangian tracking or Eulerian. With the first two methods, DEM and DPM, the particles or parcels of particles are tracked through the field and the local cloud properties are defined by their properties as they pass the point in the field. This is the Lagrangian approach. Using the Eulerian approach, on the other hand, a set of conservation equations are solved simultaneously for each point in the field. In selecting the approach to be used in modelling the dispersed phase it is fundamental to identifying whether it has dense or dilute character [94]. By definition, in a dilute flow the time between collisions is much larger than the velocity time response. Consequently, the particle-fluid interaction is responsible for the particle motion while the particle-particle collisions are negligible. In other words, the information (particle velocity, temperature etc.) travels along particles trajectories and not in all directions as in the flow of a continuum substance. If the cloud of particles cannot be treated as a

continuum, the Eulerian modelling is not usable.

In past years, few investigations used the Eulerian approach [95, 96, 76, 15]. Their motivation was to reduce the computational cost of tracking all the particles of the brownout cloud. In a brownout cloud, the dispersed phase volume fraction α_d varies from zero far away from the rotor wake to relatively high values near the ground in the region of maximum uplift (up to 0.2 [76]). Consequently, there is not a general preferable choice between the two approaches and the choice is dependent on the total number of particles, the objectives of the simulation as well as the carrier phase flow field and particle properties.

It is interesting to notice that, while the brownout investigations have been mainly conducted using Lagrangian tracking, the ice accretion studies on aircraft used mainly Eulerian modelling [97, 98, 99]. In these computations, a cloud of water droplets is dispersed in the flow and the mass of the droplets that are collected on the surface is computed. In this case, the dispersed phase has a uniform concentration in the upstream flow field but, interestingly, the volume fraction is even lower than in the previous calculations [99] (around 10^{-6}) while the particle properties are comparable, resulting in a higher limiting volume size.

In the present study, according to the results achieved in previous brownout and ice accretion investigations, the implementation of both Lagrangian and Eulerian modelling of the dispersed phase is further attempted. This choice is made taking into account the large amount of particles necessary to replicate the dust cloud, which implies a high computational cost. In the past, computational simulations reached the amount of 10^9 particles to simulate brownout [66]. Moreover, even if clustering techniques are developed to reduce the computational costs, they do not have universal applicability and only under specific conditions lead to significant computational advantages [100].

2.8 Full brownout simulations

In the present section, the brownout simulations carried on in the literature are discussed. Because the methods used for the simulation of this two-phase flow are discussed in the previous sections, here the focus will be on the validation of the brownout computations and on the results obtained.

Authors (Year)	Carrier Phase Model	Dispersed Phase Model	Validation
Keller et al. [58] (2006)	VFM	Lagrangian tracking	None
Wenren et al. [61] (2006)	Vorticity Confinement	Eulerian modelling	None
Haehnel et al. [76] (2008)	Vorticity Confinement	Eulerian modelling	Rodgers [11]
Wachpress et al. [50] (2009)	VFM	Lagrangian tracking	None
Phillips and Brown [16] (2009)	VTM	Eulerian modelling	Nathan and Green [13]
Ghosh et al. [15] (2010)	RANS	Eulerian modelling	Nathan and Green [13]
D'Andrea [67] (2011)	VFM	Lagrangian tracking	None
Morales and Squires [80] (2011)	DNS	Lagrangian tracking	None
Thomas et al. [81] (2011)	RANS	Lagrangian tracking	Sydney et al.[35]
Thomas et al. [89] (2012)	Hybrid	Lagrangian tracking	Sydney et al. [35]
Jasion and Shrimpton [82] (2012)	RANS	Lagrangian tracking	Rodgers [11]
Syal and Leishman [66] (2013)	VFM	Lagrangian tracking	Wong and Tanner [12]
Tritschler et al. [56] (2014)	VFM	Lagrangian tracking	Wong and Tanner [12]
Kutz et al. [85] (2014)	RANS	Lagrangian tracking	Sydney et al. [35]
Govindarajan and Leishman [69] (2016)	VFM	Lagrangian tracking	Wong and Tanner [12]
Alfred et al. [17] (2017)	VFM	Lagrangian tracking	Wong and Tanner [12]
Hu et al. [101, 102] (2020)	RANS	Lagrangian tracking	None
Tan et al. [103] (2021)	VFM	Lagrangian tracking	Wong and Tanner [12]

Table 2.6: Numerical simulations of full brownout phenomenon.

Table 2.6 summarizes the brownout simulations available in the literature. Generally, the lack of experimental data of brownout dust clouds led to a difficult validation of brownout models. Consequently, most of the simulations are either not validated or not sufficiently validated because of the lack of data. Another general evaluation is related to the dispersed phased modelling, interestingly the great majority of the investigations implemented a Lagrangian tracking algorithm for this task. The main reason for this is that most of these works were carried on within the same research group at the University of Maryland.

2.8.1 Lagrangian Particle Tracking

In terms of particle tracking, there are basically two approaches for the numerical simulation of dispersed phases, and they can be categorized as Lagrangian and Eulerian. In the Lagrangian approach, the particles (or parcels of particles) are tracked through the flowfield and the local cloud properties are defined by their properties as they pass a point in the field. For methods that involve this approach, the motion of the particles is tracked using Newton's second law. This method is accurate in terms of particle velocities and positions, which is important for interactions with the ground, however, the computational cost may become prohibitive when the amount of

particles is large, like in the case of a brownout cloud. Previous works in the Lagrangian frame of reference are [17, 65, 50, 104]. In general, Lagrangian particle tracking methods are based on solving the particle motion equation:

$$m_p a_p = \sum F_i. \tag{2.3}$$

Here m_p is the mass of the particle, a_p its acceleration and $\sum F_i$ is the sum of the forces acting on it. The majority of works agreed to take into account only gravity and aerodynamic-drag effects [65, 85, 17, 67], however, some works take into account other types of forces acting on the particle, such as Saffmann and Magnus forces [101] or contact forces between particles [103]. As it is possible to see from table 2.6, several models for the carrier phase have been used in combination with a Lagrangian particle tracking, VFM and RANS (with different turbulence models) are the most popular. The Sydney at al. [35] and the Wong and Tanner [12] experiments have been intensively used for quantitative validations, while the work of Nathan and Green [13] is used for qualitative validations.

Because of the mentioned problem of brownout simulations, the computations have been extensively used to try and understand the full scale brownout phenomenon, and its relationship with rotorcraft design. In particular, D'Andrea [67] focused his work on the brownout cloud produced by different rotorcraft configurations. Other works focused on slotted tip blades [105, 102] and their impact on brownout severity. Their results confirmed that the reduced intensity of the wake generated by this type of blades, had an impact on brownout severity, reducing the cloud size and number of particles uplifted. Alfred et al. [17] tried to optimize the landing manoeuvres to reduce brownout severity. Furthermore, Syal et al. [105] studied high performance takeoff manoeuvres, see figure 2.20.

Over the past decades, several works focused on brownout physics and more specifically on sediment entrainment mechanisms. The Bagnold and the Shao models [106] [107] are popular in brownout works to define where particles are uplifted. However, some works successfully modeled bombardment and splash entrainment [85, 65, 81]. Some of these focused on the wake influence

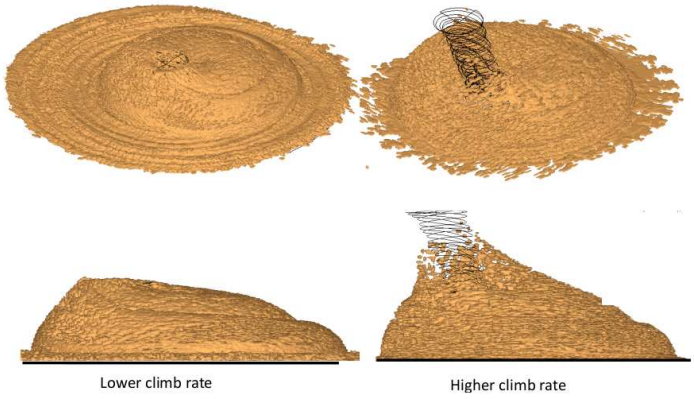


Figure 2.20: Comparison of dust clouds for two different climb rates [65].

[81], while others focused on modeling ot the interactions of different type of particles during the brownout evolution [81, 85]. The cost of brownout simulations can be prohibitive in the Lagrangian frame of reference, for this reason Govindarajan et al. [100], used to track clusters of particles and with different methods, it was possible to generate particles inside the tracked clusters, as shown in figure 2.21. In this way it was possible to track over 200 million of particles [65].

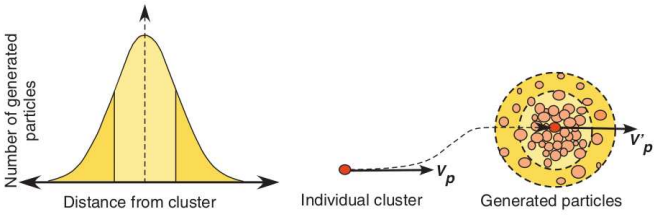


Figure 2.21: Schematic showing the Gaussian method of particle clustering. [100]

2.8.2 Eulerian simulations

Brownout is due to the presence of sand particles in the flowfield, however any kinds of particles can be involved such as snow, rain, ice and even small rocks. Several works used CFD and numerical models to predict brownout. In general, the main approach is to use CFD to predict the flowfield around an operating rotor, and solve another set of equations for particle motion. In the majority of works, the one-way coupling assumption has been adopted. It describes the dispersed

phase as dilute in the fluid, in this way it is possible to assume negligible the effect of the particles on the flowfield properties. In the past, various approaches have been used to predict the flowfield around the rotorcraft, including Vortex Transport Model (VTM) used in [77] or RANS (and URANS) with different turbulence models [15, 14, 108].

For Eulerian methods, the properties of the particles are assumed continuous within the field. Thus, differential conservation equations are written and discretised, and the solution of these defines the properties of the cloud.

Several works have been carried out at the Iowa University [15, 14, 109] for Eulerian particle tracking. All these works, used the same flowfield and dust model to predict brownout. The flowfield has been solved using incompressible RANS, with a momentum source term used to introduce the rotor effect in the flowfield. Rotor performance validations were included in [15, 14]. In Ghosh et al. work [15], validation with experimental results (by Rabbott [110]) in terms of rotor blade loading, while for the Garrick et al. work [14], the UH60 rotor performances (torque coefficient) at different advance ratios are compared with experiments. The dust model used in these works use a convection-diffusion equation, including terminal velocity (which allows for taking into account gravity) and a source term for particles uplift. A qualitative validation of the dust model used in these works, is described in the Ghosh et al. [15] who compared the cloud shape with experiment conducted by Nathan and Green at University of Glasgow [13]. Results presented in Garrick et al. [14] focused on comparing the cloud at the different advance ratios, including the difference in terms of fuselage and no-fuselage configurations. Furthermore, it includes also consideration about landing manoeuvres. On the other hand, Ghosh et al. [15] work presents results comparing different rotor layouts (single rotor, tandem, tilt-rotor and quad-rotor) at different heights above the ground. Results presented in these works define how these variables (height above the ground, rotor layout, taxiing speed) have a qualitative impact on the developed brownout cloud.

A similar Eulerian model is presented in [61, 76]. In these works, the flowfield has been computed as laminar and incompressible, adding the vorticity confinement method, to obtain the vortices and boundary layer. The Eulerian method is a convection equation with terminal velocity and source term based on the Bagnold model. However, the diffusion have been considered negli-

gible. Haehnel et al. [76] work presents a quantitative validation of the dust density around a H-21, compared with results presented by Rodgers et al. [11]. Results focused on the brownout cloud development with time.

Several works about brownout modelling in Eulerian frame of reference has been performed by Philips and Brown [16, 111, 77]. In these works, the dust model is a diffusion-convection model for dust, including source terms borrowed from the sedimentology literature. Philips and Brown [16, 111, 77] validated both flowfield and dust model with experimental results, the former quantitively with Lee et al. work [26] and Preston et al.[18], the latter qualitatively with the Nathan and Green experiments [13]. Part of the works focused on the different development and cloud shapes generated by different rotor configurations (mainly between tandem and single rotors), at different forward speed.

2.9 Outflows over human body - PAXman model

Few works have been carried out to define safety in term of outflow velocities nearby an operating IGE rotor, some of them have already been discussed in previous sections, because such works consider single-phase full-scale experiments. Silva et al. [19], studied the outflow around a CH-47D using anemometers all around the rotor. From the obtained outflow, they computed the PAXman forces, with the model that is used in the present work, and described in the next chapter. The aircraft has been tested at 40 ft, 60ft and 120 ft (OGE) above the ground. Results showed consistently higher forces on the starboard side of the rotorcraft (90deg azimuth), than on the port (270 deg azimuth), due to the rotational direction of the two rotors (figure 2.22). Where the forces are higher, they can be dangerous for ground personnel, reaching the caution limits proposed by the by PAXman model (around 330 N). A computational work, presented by Preston et al. [18], investigated the outflow and related PAXman forces for a tilt rotor hovering at different disk loading configurations and several heights above the ground. Results showed how high loaded rotor generate stronger outflows, and lead to stronger PAXman forces. Also Calver and Wenren [112] identified the rotor disk loading as a significant factor for human stability in rotorcraft outflows,

even if they specify that it is not the only important factor, that may effect human stability. Similar conclusions have been carried out by Ferguson [60].

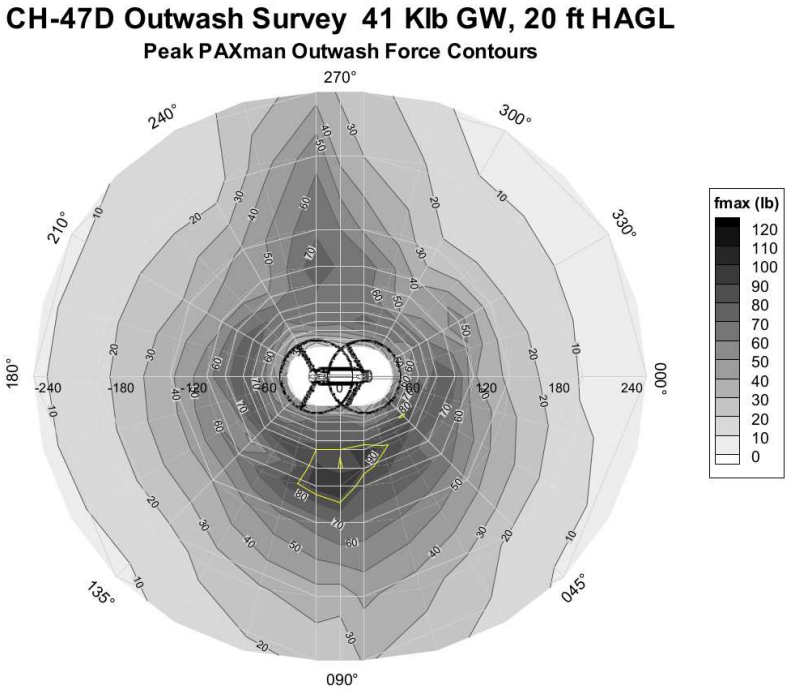


Figure 2.22: CH-47D PAXman outwash force contours during a 20 ft hover [19].

2.10 Outcome of the literature survey

It is clear from the literature that the model scale experiments provide more detailed data than full scale experiments. In particular, PIV measurements of the flow field of scaled rotorcraft IGE and brownout conditions are available and can be used for the validation of the carrier flow simulation [27, 25, 26]. Other useful data for the validation of particle entrainment simulations are the PTV measurements which provide the suspended particles position and velocity vector when the rotorcraft was tested with dispersed ground particles [48, 37, 40].

While these scaled measurements are generally accurate and detailed, they are limited to low tip Reynolds and Mach numbers and they are subject to the mentioned scaling problem of the solid phase particles. Therefore, full scale experiments are also needed to verify the accuracy of full scale simulations and validate them. Two sets of data are extremely useful in this task.

The first is the dataset containing the time averaged velocity outflow profiles of different rotorcraft (namely a H-60, a CH-53E, a XV-15 and, a V-22) in several flight conditions by Preston et al. experiments [18]. The second is the dataset which includes measurements of local time averaged particle concentration in the dust cloud generated by seven airframes (H-21, UH-I, CH-46, CH-53, V-22, HH-60, and MH-53) of the Sandblaster program [23] and Rodgers experiments [11].

As already mentioned, the experiments related to the uplift of sediments from the ground by a rotorcraft wake have been performed using exclusively particles replicating the sand which usually causes the brownout cloud. To validate simulations of snow or water entrainment with this data is not possible. On one hand, in the ice accretion computational studies [98], the water droplets trajectory computations are generally validated using the wind tunnel tests data of Papadakis et al. [113]. The aforementioned study provides a water droplet impingement database for a range of test geometries including airfoils, tail sections, and an S-Duct Engine Inlet. On the other hand, measurements of air-snow flow fields are available but mainly restricted to boundary layers over flat planes [114, 115, 116, 117, 118] and snowdrift around buildings [119]. The mentioned data can be used for the validation of the snow and water entrainment simulations.

With regard to the numerical computations in the literature, a great effort was made to properly compute a rotorcraft wake IGE. This is the first issue to address for the simulation of entrainment of particles from the ground. Because of the importance of the interaction of the rotor vortical structures (mainly by the tip vortex) with the sediment bed, it is essential to conserve these structures until they impinge the ground. Moreover, their interaction with the ground plane and its boundary layer has to be accurate.

In general, a choice has to be made between a grid free method which is computationally less expensive, preserves the tip vortex, but introduces empiricism because of the rotor and ground modelling; and a grid based method which particularly predicts the near ground and near blade flow fields but implies a higher computational cost.

In the present work, the latter approach is selected because of the importance of accurately predicting the ground boundary layer where the sediments are uplifted and entrained. Moreover,

it allows to resolve the rotor flow field to analyse the effects of the blade design and performance on the particles suspension. In particular, the flow field is studied with high-fidelity computational fluid dynamics using the HMB CFD solver [120]. As a first attempt a finer grid was used on the path of the tip vortex to preserve vorticity [87].

Once the rotorcraft wake IGE was accurately predicted, the following step was the actual simulation of the brownout phenomenon. This phenomenon consists of the uplift of particles from the ground and their entrainment into the flow. The next step can be divided in two different problems: the modelling of the particles uplift and the prediction of the behaviour of a two-phase flow. For the problem, the particles uplift, there are well established solutions in the sedimentology community where the uplift of particles due to water flows (i.e. rivers) or wind has been extensively investigated. Well established models are available and have already been implemented and validated for the uplift of sand particles by a rotorcraft wake. Moreover, dedicated models for the brownout uplift were also developed which take into account the characteristic features of the phenomenon, for example the previously described bombardment ejection of sediments due to particles reingested by the rotor. However, no attempts have been made for the implementation of snow or water droplets uplift models, even though these models have been extensively developed in other research communities [121]. In the present investigation these models are implemented in order to be able to simulate all the flows in which the uplift of sediments from the ground by the rotorcraft wake is involved.

Once the sediments are uplifted from the ground, they are entrained into the flow creating a two-phase flow. In the present work, accordingly to the survey conducted in the previous sections, one-way coupling between the phases is assumed as a first assumption, and a case by case evaluation is then conducted. In the literature, several works used the Lagrangian or the Eulerian model to predict brownout, but they have never be used in the same framework. A direct comparison between the two models is currently missing in the literature. In this work, both models are being used to compare directly these different approaches on the same test cases. For this reason, an Eulerian model and a Lagrangian algorithm have been developed for this work.

2.11 Objectives of this thesis

As said in the previous section, a direct comparison between the two models is currently missing in the literature. In this work, both models are taken into account to properly simulate brownout clouds. A Lagrangian particle tracking method and an Eulerian method have been implemented in the framework of HMB3, and the obtained results have been compared directly. The two models are validated against experimental results, and then they are compared in terms of numerical and numerical efficiency. Furthermore, the safety of IGE operations is analysed, defining safe zones around a rotor operating IGE in terms of forces on ground personnel and the presence of particles. The safe zones, are then compared with existing safety distance criteria for rotorcraft operation, like the 3 rotor diameter for wake encounters of FAA.

To summarize, the objectives of the thesis are:

1. To implement a Lagrangian particle tracking method in the framework of HMB3 CFD solver and verify the implementation.

2. To implement an Eulerian method in the framework of HMB3 CFD solver and verify the implementation.

3. To evaluate the safety of operational scenario for rotors operating IGE, in terms of PAXman forces on ground personnel and brownout cloud sizes.

4. To define which are the geometrical and operational conditions that affect the safety of ground personnel operating nearby the rotor.

5. To compare Lagrangian and Eulerian methods, evaluating results in terms of brownout cloud size on same test cases.

6. To evaluate the parallel performance of the two methods, comparing code timing and speedup for the parallelization.

7. To document the best practice for simulations according to the conditions of each test case.

Chapter 3

Methodology

In the present work, the Helicopter Multi-Block (HMB3) was used for all numerical fluid simulations. HMB3 is a 3D flow solver for compressible Navier-Stokes equations of the University of Glasgow. HMB3 is parallel and initially developed to analyse rotorcraft flows with structured multi-block grids. [122, 123]. In addition, the solver offers a variety of turbulence models and hybrid RANS/LES methods. It has been successfully used for a variety of flows including rotors and wind turbines [124, 125, 126, 127], transonic cavity flows, shock wave boundary layer interactions, and supersonic flows around missiles.

Initially, HMB3 is used to solve flowfield around operating rotors, then a Lagrangian particle tracking and Eulerian model have been developed in parallel. Both numerical models involved the one-way coupling assumption, which defines that the flowfield is not affected by the presence of the particle. The workflow is described by figure 3.1. This chapter describes HMB3, the particle tracking models are described in the next chapter.

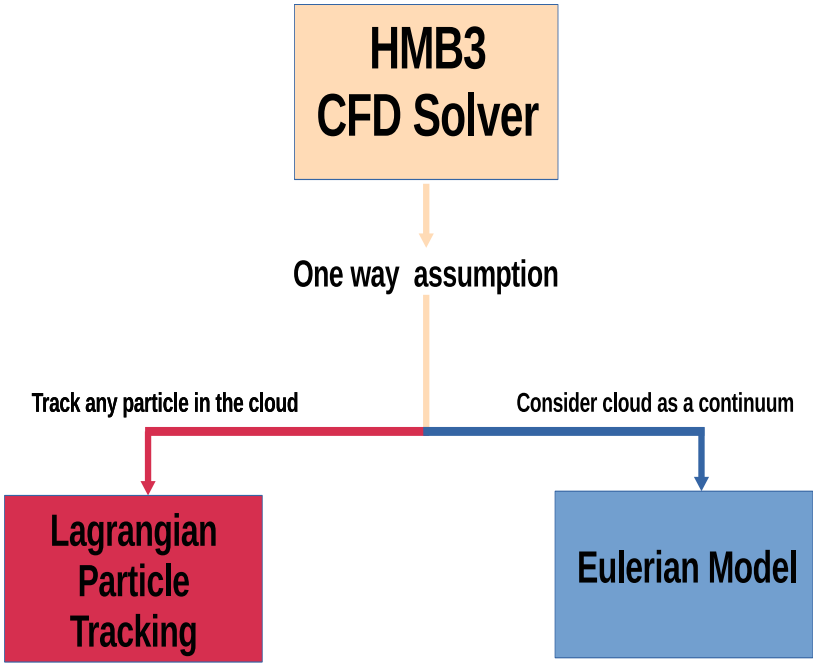


Figure 3.1: Methodology workflow used in this work.

3.1 Navier-Stokes equations

The Navier-Stokes equations system includes the conservation laws of the fluid in addition to two thermodynamic equations of the state for pressure P and temperature T [128, 129]. The mass conservation (i.e continuity equation) is a scalar partial differential equation. It describes that mass is conserved. In Cartesian coordinates it can be written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0, \tag{3.1}$$

where ρ is the fluid density and u_i is the i-th velocity component. The conservation of momentum can be expressed as:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = \rho f_i - \frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}, \tag{3.2}$$

where f_i is an acting body force, while τ_{ij} is the stress tensor of the fluid. Considering a Newtonian fluid [1] and the Stokes's hypothesis [2] , τ_{ij} can be written as function of the strain rate tensor S_{ij} as:

$$\tau_{ij} = 2\mu \left(S_{ij} - \frac{1}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k} \right), \tag{3.3}$$

u_j and u_k are the j-th and k-th velocity component.

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{3.4}$$

with μ as molecular viscosity, and δ_{ij} the Kronecker delta. The energy conservation equation (1st thermodynamic law) states that the total energy of an isolated system is constant and it can be defined, in Cartesian coordinates, as:

$$\frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_j}\left[u_j(\rho E + P) \right] - \frac{\partial}{\partial x_j}\left(u_i \tau_{ij} - q_j \right) = 0, \tag{3.5}$$

E is the total energy per unit of mass: $E = (e + \frac{1}{2}u_i u_i)$, notice that the $\frac{1}{2}u_i u_i$ component defines the kinetic energy. q_j is the heat flux component, and it is expressed by the Fourier's law:

$$q_j = -k_h \frac{\partial T}{\partial x_j}, \tag{3.6}$$

where k_h is the heat transfer coefficient. Assuming an ideal gas, the state equation which gives the relation between pressure and density is:

$$P = \rho R_{air} T, \tag{3.7}$$

where R_{air} is the specific ideal gases constant for dry air $R_{air} = 287.058 \frac{J}{KgK}$. Finally, Sutherland's law is used to relate the molecular viscosity of air and its temperature:

$$\mu = \left(\frac{T}{T_0} \right)^{\frac{3}{2}} \left(\frac{T + 110}{T_0 + 110} \right), \tag{3.8}$$

typically for air the constant values of T_0 and μ_0 are fixed to: $T_0 = 273.15K$, $\mu_0 = 18.5110^{-6} \frac{kg}{ms}$.

[1] In a Newtonian fluid the viscous stresses are proportional to the rates of change of the fluid velocity vector

[2] The Stokes's hypothesis implies that the bulk viscosity is zero

3.1.1 Non-dimensionalization

Reference variables needed are, U_{ref}, ρ_{ref}, L_{ref} and T_{ref}. These reference values are chosen depending on the nature of the problem. In table 3.1 are shown the examples used in HMB3 for non-dimensionalization. Using the reference values it is possible to obtain the dimensionless

	Not rotary wing system	Rotary wing system in axial flight
L_{ref}	Characteristic length of the problem	Rotor aerodynamic chord c
ρ_{ref}	Free-stream density ρ_∞	Free-stream density ρ_∞
U_{ref}	Free-stream velocity U_∞	Rotor tip velocity V_{tip}
T_{ref}	Free-stream temperature T_∞	Free-stream temperature T_∞

Table 3.1: Examples of reference variables used for non dimensionalization in HMB3.

variables:

$$x_i = \frac{x_i^*}{L_{ref}}, \quad \rho = \frac{\rho^*}{\rho_{ref}}, \quad T = \frac{T^*}{T_{ref}}, \quad U = \frac{U^*}{U_{ref}}. \tag{3.9}$$

It is also possible obtain other variables that are useful for the non-dimensionalization of the system of equations:

$$t = \frac{U_{ref}}{L_{ref}} t^*, \quad P = \frac{P^*}{\rho_{ref} U_{ref}^2}, \quad \mu = \frac{\mu^*}{\mu(T_{ref})}, \quad e = \frac{e^*}{U_{ref}^2}, \quad . \tag{3.10}$$

Variables with superscript *, denote dimensional values.

The dimensionless system of equations is composed by the mass conservation, the momentum conservation and the energy conservation. Furthermore, to close the system, the relations for ideal gases are needed. They correlate the static temperature T, gas density ρ, pressure P, total enthalpy H, total energy E and internal energy e.

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \dfrac{\partial (\rho u_i)}{\partial x_i} = 0, \\ \dfrac{\partial \rho u_i}{\partial t} + \dfrac{\partial (\rho u_i u_j)}{\partial x_j} = -\dfrac{\partial P}{\partial x_j} \dfrac{1}{\gamma M_{ref}^2} + \dfrac{1}{Re} \dfrac{\partial \tau_{ij}}{\partial x_j}, \\ \dfrac{\partial \rho E}{\partial t} + \dfrac{\partial}{\partial x_j} \left[u_j (\rho E + \dfrac{p}{\gamma M_{ref}^2}) \right] - \dfrac{\partial}{\partial x_j} \left(\dfrac{1}{Re} u_i \tau_{ij} - \dfrac{1}{PrRe} \dfrac{\partial T}{\partial x_j} \right) = 0, \end{cases} \tag{3.11}$$

Here, $Re = \frac{U_{ref}}{L_{ref}} \nu_{ref}$ is the Reynolds number, defined as the ratio between the inertial and viscous forces. While $Pr = \frac{\mu_{ref} c_p}{k_h}$ is the Prandtl number, which is the ratio between the momentum and thermal diffusivity. In the above, c_p is the specific heat at constant pressure and k_h the gas thermal conductivity, while the dynamic viscosity μ_{ref} has been already defined. Finally, γ, is the specific heats ratio. In this work it has a value of 1.4. $M_{ref} = \frac{U_{ref}}{a_{ref}}$ is the Mach number, obtained by scaling the reference velocity with the reference sound speed. $a_{ref} = \sqrt{\gamma P_{ref}/\rho_{ref}}$. The relations for ideal gases are summarized in the following:

$$H = E + \frac{P}{\rho}, E = e + \frac{1}{2}(u_i u_i), P = (\gamma - 1) e\rho, T = \gamma M_{ref}^2 \frac{P}{\rho}. \tag{3.12}$$

3.1.2 Governing equations in conservative vector form

The Navier stokes equations are written in a conservative and dimensionless vector form. To do this, the equations will be expressed by the same generic form.

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial (\mathbf{F}^i + \mathbf{F}^v)}{\partial x} + \frac{\partial (\mathbf{G}^i + \mathbf{G}^v)}{\partial y} + \frac{\partial (\mathbf{H}^i + \mathbf{H}^v)}{\partial z} = \mathbf{S}, \tag{3.13}$$

where $\mathbf{W}$ is the vector of conserved variables:

$$\mathbf{W} = (\rho, \rho u, \rho v, \rho w, \rho E)^T, \tag{3.14}$$

are $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ are the flux vectors in the x-,y- and z-direction. The superscripts i and v indicate the inviscid and viscid components of the flux vectors. They are expressed as:

$$\begin{aligned}
\mathbf{F}^i &= [\rho u, \rho u^2 + p, \rho uv, \rho uw, u(\rho E + p)]^T, \\
\mathbf{G}^i &= [\rho v, \rho uv, \rho v^2 + p, \rho vw, v(\rho E + p)]^T, \\
\mathbf{H}^i &= [\rho w, \rho uw, \rho vw, \rho w^2 + p, w(\rho E + p)]^T, \\
\mathbf{F}^v &= \frac{1}{Re}[0, \tau_{xx}, \tau_{xy}, \tau_{xz}, u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + q_x]^T, \\
\mathbf{G}^v &= \frac{1}{Re}[0, \tau_{xy}, \tau_{yy}, \tau_{yz}, u\tau_{xy} + v\tau_{yy} + w\tau_{yz} + q_y]^T, \\
\mathbf{H}^v &= \frac{1}{Re}[0, \tau_{xz}, \tau_{yz}, \tau_{zz}, u\tau_{xz} + v\tau_{yz} + w\tau_{zz} + q_z]^T,
\end{aligned} \tag{3.15}$$

3.1.3 Spatial discretization

The Navier-Stokes equations in differential form, can be written in the Arbitrary Lagrangian Eulerian (ALE) formulation for time-dependent domains, defining a system of conservation laws for time-dependent control volume as:

$$\frac{d}{dt}\iiint_{V(t)} \mathbf{W}dV + \iint_{\partial V(t)} (\mathbf{F}^i(\mathbf{W}) - \mathbf{F}^v(\mathbf{W}), \mathbf{G}^i(\mathbf{W}) - \mathbf{G}^v(\mathbf{W}), \mathbf{H}^i(\mathbf{W}) - \mathbf{H}^v(\mathbf{W}))\mathbf{n}dS = S, \quad (3.16)$$

where V(t) is the time-dependent domain, with boundary $\partial V(t)$ and outward unit normal $\mathbf{n}$. $\mathbf{F}^i$,$\mathbf{G}^i$ and $\mathbf{H}^i$ are the inviscid fluxes while $\mathbf{F}^v$,$\mathbf{G}^v$,$\mathbf{H}^v$ are the viscous fluxes, already defined in 3.15. In absence of volume forces, and in an inertial frame of reference S=0. The Navier-Stokes equations are discretised using a cell-centered finite volume method on structured multi-block grids leading to the following system of ODEs.

$$\frac{d}{dt}(\mathbf{W}_{i,j,k}V_{i,j,k}) = -\mathbf{R}_{i,j,k}(\mathbf{W}), \quad (3.17)$$

where $\mathbf{W}$ are the cell conserved variables defined in equations 3.14, $\mathbf{R}$ represents the residuals, i,j and k are the indices that define a single cell, and $V_{i,j,k}$ the cell volumes. Osher's upwind scheme [130] has been used to discretised the convective terms [130], while MUSCL variable extrapolation scheme, with Van Albadai's limiter [131], has been used to provide second-order accuracy. The central differencing spatial discretization method is used for the viscous terms. Boundary conditions are set using ghost cells in the exterior of the computational domain.

3.1.4 Temporal discretization

Using forward or backward finite difference schemes it is possible to solve, in time, ordinary differential systems, like equation 3.16. The explicit schemes are based on the known variables values, and are easy to implement, however they may suffer for numerical instability if the time-step is not small enough. On the other hand, the implicit schemes are more stable and they can use larger

time steps. However, implicit schemes have non-linear formulations due to the inclusion of unknown variables values in future time steps. The non-linear formulations are usually solved with a linearisation approach, approximating the non-linear system with linear systems. In general, when simulations are focusing on flow features with very small time scales, like DNS (Direct Numerical Simulation), or simulation of turbulent flow structures, the explicit schemes are more popular. One the other hand, when flows are dominated by large time scales, like flowfield surrounding rotors, implicit schemes are more common. In particular, for steady state solutions evolving from initial conditions, the implicit scheme is especially suitable. In the current work, implicit schemes are used for steady and unsteady simulations.

Steady state solver

An implicit time-marching scheme is used to integrate equation 3.13 in time and to reach the steady state solution. It is defined by:

$$\frac{\mathbf{W}_{i,j,k}^{n+1} - \mathbf{W}_{i,j,k}^{n}}{\Delta t} = -\frac{1}{V_{i,j,k}} \mathbf{R}_{i,j,k}(\mathbf{W}_{i,j,k}^{n+1}). \tag{3.18}$$

This is a non-linear algebraic system of equations, and it can be simplified by linearizing the flux residual:

$$\begin{aligned} \mathbf{R}_{i,j,k}(\mathbf{W}_{i,j,k}^{n+1}) &= \mathbf{R}_{i,j,k}(\mathbf{W}_{i,j,k}^{n}) + \frac{\partial \mathbf{R}_{i,j,k}}{\partial t}\Delta t + O(\Delta t^2), \\ &\approx \mathbf{R}_{i,j,k}(\mathbf{W}_{i,j,k}^{n}) + \frac{\partial \mathbf{R}_{i,j,k}}{\partial \mathbf{W}_{i,j,k}} \frac{\partial \mathbf{W}_{i,j,k}}{\partial t}\Delta t, \\ &\approx \mathbf{R}_{i,j,k}(\mathbf{W}_{i,j,k}^{n}) + \frac{\partial \mathbf{R}_{i,j,k}}{\partial \mathbf{W}_{i,j,k}}\Delta \mathbf{W}_{i,j,k}. \end{aligned} \tag{3.19}$$

The non-linear system described in equation 3.18, is then written as:

$$\left[\frac{V_{i,j,k}}{\Delta t}\mathbf{I} + \frac{\partial \mathbf{R}_{i,j,k}}{\partial \mathbf{W}_{i,j,k}}(\mathbf{W}_{i,j,k}^{n})\right]\left(\mathbf{W}_{i,j,k}^{n+1} - \mathbf{W}_{i,j,k}^{n}\right) = \mathbf{R}_{i,j,k}(\mathbf{W}_{i,j,k}^{n}). \tag{3.20}$$

Solving the linear system 3.20 may be prohibitive due to the high number of equations and its stiffness. To avoid this problem, an iterative Generalized Conjugate Residual (GCR) [132] method

is used. It is capable of solving large systems of equations efficiently in term of time and memory requirements. Furthermore, a Block Incomplete Lower-Upper (BILU) factorization method is used as a pre-conditioner for the system. To smooth out the initial flow, at the very beginning of the implicit scheme, a small amount of explicit iterations are performed. An approximate flux Jacobian is used (first order discretization) leading to reduced CPU time and memory requirements [133], through a reduced size of the linear system and a more diagonally dominant Jacobian. Furthermore, the BILU pre-conditioner is applied in a decoupled manner between grid blocks to reduce the parallel communication and improve the parallel efficiency. The steady state solver for the turbulent case is added like the mean flow described previously. For the turbulence model equations the flux residual also contains the dissipation source term, however, the production term is solved explicitly. The eddy viscosity is calculated from the turbulent quantities as specified by the model and is used to advance both mean flow and turbulence solutions.

3.1.5 Time marching solver

The implicit dual-time stepping Jameson's method [134] is used for time-accurate computations. This implicit scheme has a larger stability region, and it is possible to solve the implicit equations at each real time step using inner iterations in pseudo time. This way, the same steady solver is used again. The ordinary system described in 3.18 is written as:

$$\frac{\mathbf{R}^*_{i,j,k}}{V_{i,j,k}} = \frac{3\mathbf{W}^{n+1}_{i,j,k} - 4\mathbf{W}^{n}_{i,j,k} + \mathbf{W}^{n-1}_{i,j,k}}{2\Delta t} + \mathbf{R}_{i,j,k}(\mathbf{W}^{k_m}_{i,j,k}) = 0, \tag{3.21}$$

where $\mathbf{R}^*_{i,j,k}$ is the unsteady residual. A pseudo time term τ can be used to convert the system of equations into a format very close to the steady formulation in equation 3.21.

$$\frac{\mathbf{W}^{n+1,m+1}_{i,j,k} - \mathbf{W}^{n+1,m}_{i,j,k}}{\Delta\tau} + \frac{1}{V_{i,j,k}}\mathbf{R}^*_{i,j,k}(\mathbf{W}^{n+1,m+1}_{i,j,k}) = 0 \tag{3.22}$$

Where m refers to the m-th time step in terms of the pseudo time τ. The non-linear system of equation can hence follow the same solution approach as the steady state problem, and the CFD code can re-use exactly the same subroutines built for the steady discretization.

3.2 Reynolds Averaged Navier Stokes Equations

The Navier-Stokes equations describe laminar or turbulent flows. However, they present few exact solutions mainly due to their non-linearity, and variety of boundary conditions. Their complexity has motivated on to the introduction of simplification and assumptions. There are two main physical phenomena associated with turbulence. The first is its diffusion, which include the transport of mass, momentum and heat due to the chaotic time-depended motion of fluid. The second is the dissipation of the turbulent kinetic energy, which involve the viscous forces at the Kolmogorov scale. The work by Reynolds in 1894 [135] presented the decomposition of the flow variables in mean and fluctuating parts. His model is the base concepts of the most of the current turbulence models.

3.2.1 Reynolds averaging

The Reynolds averaged Navier-Stokes equations are obtained from the decomposition in mean and fluctuating parts of the flow variables. After the decomposition the whole system of equations are averaged in time. The entire procedure is called Reynolds averaging. The decomposition can be written as follows:

$$f = \overline{f} + f', \tag{3.23}$$

where $\overline{f}$ indicates the mean values, while f' is the fluctuating part. The mean values is obtained by the time averaging, that can be defined as:

$$\overline{f}(x) = \lim_{T \to 0} \frac{1}{T} \int_T f(x,t)dt. \tag{3.24}$$

The average can not be a function of the time, and it is necessary that T is long enough relative to the maximum period of the assumed fluctuations. The time step should be chosen large enough, for the average of turbulent fluctuations to be zero, and small enough to resolve unsteadiness of the mean flow.

The compressible Reynolds Averaged Navier-Stokes equations (called Favre equations) need the definition of the density-weight average (or Favre average): $\hat{f} = \dfrac{\overline{\rho f}}{\overline{\rho}}$, and f'' is the corresponding disturbance.

The Navier-Stokes equations can then be rewritten as:

$$\begin{cases} \dfrac{\partial \overline{\rho}}{\partial t} + \dfrac{\partial \overline{\rho}\hat{u}_i}{\partial x_i} = 0, \\ \dfrac{\partial \overline{\rho}\hat{u}_i}{\partial t} + \dfrac{\partial \overline{\rho}\hat{u}_j\hat{u}_i}{\partial x_j} = -\dfrac{\partial \overline{P}}{\partial x_j} + \dfrac{\partial \overline{\tau}_{ij} + \tau_{ij}^R}{\partial x_j}, \\ \dfrac{\partial \overline{\rho}\hat{E}}{\partial t} + \dfrac{\partial \left[u_j\left(\overline{\rho}\hat{E}\right)\right]}{\partial x_j} = -\dfrac{\partial}{\partial x_j}\left(\overline{\tau}_{ij}\hat{u}_i + \overline{\tau_{ij}u_i''}\right) - \dfrac{\partial}{\partial x_j}\left(\overline{q}_j + c_p\overline{\rho u_j'' T''} - \hat{u}_j\tau_{ij}^R + \dfrac{1}{2}\overline{\rho u_i'' u_i'' u_j''}\right). \end{cases} \tag{3.25}$$

Notice that mass conservation equation does not change, due to the linearity with respect to velocity, on the other hand for the momentum and energy conservation equations additional terms appear into equations, due to the non-linearity of the convection terms. It is possible to notice that the unknowns variables are more than the equations, this is called the turbulence closure problem. To close the system of equations it is necessary model some elements:

$$\tau_{ij}^R \quad , c_p\overline{\rho u_j'' T''}, \quad \overline{\tau_{ij}u_i''}, \quad , \frac{1}{2}\overline{\rho u_i'' u_i'' u_j''}, \tag{3.26}$$

The first is the Reynolds stress tensor and it is modelled following the Boussinesq eddy viscosity hypothesis. which introduces the concept of eddy viscosity μ_t which relates the Reynolds stress tensor at the mean strain rate tensor $\hat{S}_{ij} = \dfrac{1}{2}\left(\dfrac{\partial \hat{u}_i}{\partial x_j} + \dfrac{\partial \hat{u}_j}{\partial x_i}\right)$:

$$\tau_{ij}^R = 2\hat{\mu}_t\left(\hat{S}_{ij} - \frac{1}{3}\frac{\partial \hat{u}_k}{\partial x_k}\Delta_{ij}\right) - \frac{2}{3}\overline{\rho}k\delta_{ij}. \tag{3.27}$$

The second term is modeled following the Reynolds analogy:

$$c_p\overline{\rho u_j'' T''} \approx -\frac{c_p\hat{\mu}_t}{Pr_t}\frac{\partial \tilde{T}}{\partial x_j}. \tag{3.28}$$

where Pr_t is the turbulent Prandtl number, and it is constant. For air is used 0.9. μ_t is the turbulence viscosity, The present research has chosen the two-equation $k-\omega$ SST model [136] for most of the

simulations. Finally, the last two terms are modelled together as:

$$\overline{\tau_{ij}u_i''} - \frac{1}{2}\overline{\rho u_i'' u_i'' u_j''} \approx \left(\hat{\mu} + \frac{\hat{\mu}_t}{\sigma_k}\frac{\partial k}{\partial x_j}\right) \tag{3.29}$$

σ_k is a coefficient associated with the turbulence model.

3.2.2 Turbulence modelling

To close the RANS equations, Wilcox developed the original $k-\omega$ model [137], using a transport equations for the turbulent kinetic energy k and another for the specific dissipation rate ω. The $k-\omega$ model follows the eddy viscosity assumption of the Reynolds stresses and defines the eddy viscosity $\mu_t = \rho\dfrac{k}{\omega}$. The $k-\omega$ model can handle well adverse pressure gradients near the wall, however its performance worsens when it deals with free shear layers. Therefore, Menter [136] proposed a blending between the $k-\omega$ model and the $k-\varepsilon$ models (where ε is rate of dissipation of turbulent kinetic energy), which beyond better in free streams flows. The final model uses two transport equations:

$$\begin{cases} \dfrac{D}{Dt}(\rho k) = \tau_{ij}\dfrac{\partial u_i}{\partial x_j} - \beta^*\rho\omega k + \dfrac{\partial}{\partial x_j}\left[(\mu + \sigma_k\mu_t)\dfrac{\partial k}{\partial x_j}\right]. \\ \dfrac{D}{Dt}(\rho\omega) = \dfrac{\gamma}{\nu_t}\tau_{ij}\dfrac{\partial u_i}{\partial x_j} - \beta^*\rho\omega^2 + \dfrac{\partial}{\partial x_j}\left[(\mu + \sigma_k\mu_t)\dfrac{\partial \omega}{\partial x_j}\right] + 2\rho(1-F_1)\sigma_{\omega 2}\dfrac{1}{\omega}\dfrac{\partial k}{\partial x_j}\dfrac{\omega}{\partial x_j}, \end{cases} \tag{3.30}$$

$\nu_t = \mu_t/\rho$ is the turbulent kinematic viscosity. β^* , σ_k, σ_ω and $\sigma_{\omega 2}$ are closure coefficients [136]. F_1 is a blending function combining the two models $k-\omega$ and $k-\varepsilon$. Furthermore, the $k-\omega$ SST model adds the Bradshaw's assumption. It states that the shear stress is proportional to the turbulent kinetic energy in the boundary layer, with the eddy viscosity assumption to account for the shear stress transport effect to improve the handling of adverse pressure gradients. These models have been successfully used within the HMB3 environment in past works [124, 125, 126, 127].

Chapter 4

Particle tracking models

Lagrangian and Eulerian models used in this work to predict brownout are described in this chapter. Furthermore, to extend the work to safety considerations more, the PAXman model is presented. Used mainly in military environment, the PAXman model is used to estimate forces on ground personnel due to strong rotor outflows.

4.1 Lagrangian particle tracking

In the Lagrangian frame of reference, the motion of a single particle is solved by Newton's second law,

$$\frac{\partial \mathbf{u}_p}{\partial t} = \sum \mathbf{F}. \tag{4.1}$$

Where $\boldsymbol{u}_p$ is the particle velocity, t the time and $\Sigma\mathbf{F}$ is the sum of external forces acting on the particle. For the brownout particle tracking the forces taken into account are the aerodynamic force and the gravity. The particles are driven by the flowfield velocity and their positions in time are obtained by integrating their equations of motion. The following ODE system can be obtained:

$$\begin{cases} \dfrac{\partial \mathbf{u}_p}{\partial t} = \dfrac{0.5\rho_{air}(\mathbf{u}-\mathbf{u}_p)||(\mathbf{u}-\mathbf{u}_p)||}{B} - \mathbf{g}, \\ \dfrac{\partial \mathbf{x}_p}{\partial t} = \mathbf{u}_p. \end{cases} \tag{4.2}$$

where $\boldsymbol{u}_p$ is the particle velocity, $\boldsymbol{u}$ is the velocity of the flowfield in the position of the particle, and B the particle ballistic coefficient, defined as $B = \dfrac{m_p}{S_p C_D}$. Here m_p is the particle mass, $S_p = \pi d_p^2/4$ is the particle frontal area (particles are assumed spherical), C_D is the particle drag coefficient, and finally $\mathbf{g}$ is the acceleration of gravity. The particle motion equation has been non-dimensionalized using the flowfield reference values for length, velocity and density, defined in Chapter 3. To solve the particle acceleration equation, a fourth-order Runge-Kutta method was used. To obtain the velocity and position of particle at any timestep, it is necessary to integrate two equations, one for particle's velocity, one for the particles position:

The Runge-Kutta method can be expressed by:

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^n + \frac{\Delta t}{6}(K_1^v + 2K_2^v + 2K_3^v + K_4^v), \tag{4.3}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \frac{\Delta t}{6}(K_1^p + 2K_2^p + 2K_3^p + K_4^p). \tag{4.4a}$$

Where the K_i is the i-th intermediate step. Initially the velocity and the acceleration of the particles are evaluated at timestep t^n. K_1^p and K_1^v are computed as:

$$K_1^p = \mathbf{u}_p(t_n, \mathbf{x}_p^n), \tag{4.5a}$$

$$K_1^v = \frac{0.5\rho_{air}(\mathbf{u}^n - \mathbf{u}_p^n)||(\mathbf{u}^n - \mathbf{u}_p^n)||}{B} - \mathbf{g}. \tag{4.5b}$$

Then, it is necessary to evaluate the particles position and velocity at timestep $t^{n+\frac{1}{2}}$:

$$\mathbf{u}_p^{n+\frac{1}{2}} = \mathbf{u}_p^n + \frac{\Delta t}{2}K_1^v, \tag{4.5c}$$

$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2} K_1^p. \tag{4.5d}$$

Knowing the half time step particle status it is possible to go further with the Runge-Kutta steps:

$$K_2^p = \mathbf{u}_p(t_{n+\frac{1}{2}}, \mathbf{x}_p^{n+\frac{1}{2}}), \tag{4.5e}$$

$$K_2^v = \frac{0.5\rho_{air}(\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}_p^{n+\frac{1}{2}})||(\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}_p^{n+\frac{1}{2}})||}{B} - \mathbf{g}. \tag{4.5f}$$

The half step position is then updated, with the new values obtained, using

$$\mathbf{u}_p^{n+\frac{1}{2}} = \mathbf{u}_p^n + \frac{\Delta t}{2} K_2^v, \tag{4.5g}$$

$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2} K_2^p. \tag{4.5h}$$

$$K_3^p = \mathbf{u}_p(t_{n+\frac{1}{2}}, \mathbf{x}_p^{n+\frac{1}{2}}), \tag{4.5i}$$

$$K_3^v = \frac{0.5\rho_{air}(\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}_p^{n+\frac{1}{2}})||(\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}_p^{n+\frac{1}{2}})||}{B} - \mathbf{g}, \tag{4.5j}$$

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^n + \Delta t K_3^v, \tag{4.5k}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t K_3^p. \tag{4.5l}$$

$$K_4^p = \mathbf{u}_p(t_{n+1}) \tag{4.5m}$$

$$K_4^v = \frac{0.5\rho_{air}(\mathbf{u}^{n+1} - \mathbf{u}_p^{n+1})||(\mathbf{u}^{n+1} - \mathbf{u}_p^{n+1})||}{B} - \mathbf{g}. \tag{4.5n}$$

4.2 Particles uplift and bombardment

4.2.1 Uplift threshold model

When the wake reaches the ground and interacts with the loose sediment, particles can be uplifted. To simulate this phenomenon, the Bagnold model (see [106, 138]) is a popular approach. The model has been developed within the sediment research community to simulate the pick up of particles in river flows. In 2000 Shao et al. [107] proposed a formulation, based on the Bagnold model, that has been adopted in this work. It is a threshold model, based on the wall friction velocity $u_* = \sqrt{\frac{\tau_w}{\rho}}$. The threshold value depends on particle and fluid properties, and on gravity. It is computed as:

$$u_t^* = \sqrt{A\left(\frac{\rho_p}{\rho_{air}} g d_p + \frac{\beta}{\rho_{air} d_p}\right)}, \tag{4.6}$$

where u_t^* is the threshold velocity, while A and β are coefficients: A=0.0123, β=0.0003 $\frac{kg}{s^2}$. The particle properties, used in this work, are listed in table 4.1, while for air and gravity the following values have been used: $\rho_{air} = 1.225 kg/m^3$ and $g = 9.81 m/s^2$. When $u^* > u_t^*$ particles are uplifted and entrained in the flowfield.

In general, sand particles involved in brownout have similar values of density [12, 65, 35], and the main changing parameter is the particle diameter, which vary from 100 μm, to 1 μm. [12, 65]. In past works, the effect of different types of particles has been analysed, and their different behaviours described [35, 139]. It has been found that the majority of long term suspended particles in the cloud, are in the range of 1-10 μm, due to their reduced size, [65]. It has been assumed that this range of dimensions can be considered narrow enough to represented by a single particle diameter within the range. In table 4.1 the ballistic coefficient and the Bangold threshold have been computed starting from the values listed in earlier works [12, 65, 35, 106, 138].

ρ_p (kg/m^3)	d_p (μm)	C_D	B (kg/m^2)	u_t^* (m/s)
2650	9	1.048	0.03	0.58

Table 4.1: Properties of particles used in this work.

Splash entrainment model

When an entrained particle hits the ground, it may have sufficiently high energy to launch more particles. The kinetic energy gained by the hit particles can overcome their cohesive forces and lead to uplift. Once the particle hits the ground, it rebounds and enters again the flowfield. This phenomenon is called splash entrainment, and in this work, it has been modelled using a probabilistic approach, proposed in [65] and [85]. Before describing how the splash Entrainment works, it is necessary to model the particle wall interaction using the hard sphere model, which is described in [140]. It is based on the integrated form of the equations of motion, but instantaneous deformations of the particles do not appear in the formulation explicitly. The momentum difference between two time-steps (a and b) is equal to the impulsive force acting on the particle during that time period.

$$\mathbf{J}^{(b)} = m(\mathbf{u}_p^{(b)} - \mathbf{u}_p^{(a)}). \tag{4.7}$$

Computing the impulsive force acting before and after the collision, it is possible to compute the post-collisional translation velocities, solving the momentum equations. The hard sphere model [140] introduces the following assumptions: 1) Particle deformations are neglected. During the collision process, the particle radii are constant. 2) Coulomb's friction law applies to particles that are sliding along a wall. Considering three different timesteps: t=0 pre-collision time, t=1 collision time and t=2 post collision time, as described in figure 4.1, we can define two different impulses acting on the particles during impact:

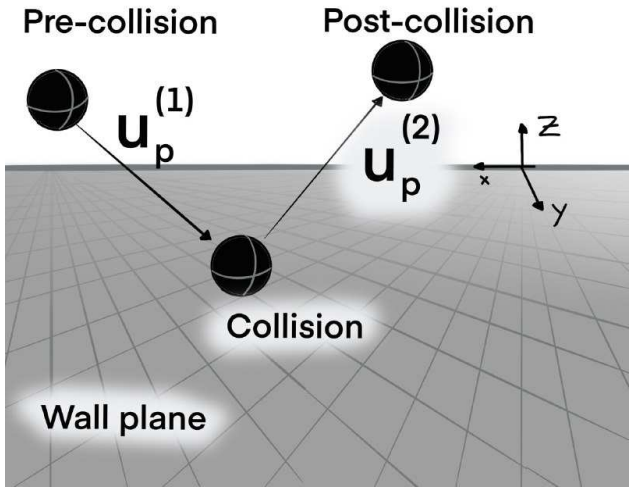


Figure 4.1: Schematic representation of particle-wall interaction.

$$\mathbf{J}^{(1)} = m(\mathbf{u}_p^{(1)} - \mathbf{u}_p^{(0)}), \tag{4.8a}$$

$$\mathbf{J}^{(2)} = m(\mathbf{u}_p^{(2)} - \mathbf{u}_p^{(1)}). \tag{4.8b}$$

Considering a 3D case, with mass-point particles, and the wall perpendicular to the Z axis. It is possible to define a boundary condition at the wall as:

$$u_{pZ}^{(1)} = 0. \tag{4.9}$$

In [140] the definition of the coefficient of restitution e is given by:

$$J_Z^{(2)} = eJ_Z^{(1)}, \tag{4.10}$$

in this work $e = 0.6$ as in Syal et al. [65]. This coefficient represents the loss of momentum due to inelastic effects. From the Coulomb's friction law, we have:

$$J_X^{(1)}\mathbf{i} + J_Y^{(1)}\mathbf{j} = -\varepsilon_X f J_Z^{(1)}\mathbf{i} - \varepsilon_Y f J_Z^{(1)}\mathbf{j}, \tag{4.11a}$$

$$J_X^{(2)}\mathbf{i} + J_Y^{(2)}\mathbf{j} = -\varepsilon_X f J_Z^{(2)}\mathbf{i} - \varepsilon_Y f J_Z^{(2)}\mathbf{j}. \tag{4.11b}$$

Here, f is the coefficient of kinetic friction ($f = 0.2$ in this work as in Syal et al. [65]), ε_X and ε_Y are factors indicating the proportion of the velocity in each wall plane component direction, defined as

$$\varepsilon_X = \frac{u_{pX}}{\sqrt{u_{pX}^2 + u_{pY}^2}}, \varepsilon_Y = \frac{u_{pY}}{\sqrt{u_{pX}^2 + u_{pY}^2}}. \tag{4.12}$$

From definition, it is possible to see that $\varepsilon_X^2 + \varepsilon_Y^2 = 1$. In this work, these values are constant during the collision processing, in other words: $\varepsilon_X^{(0)} = \varepsilon_X^{(1)} = \varepsilon_X^{(2)}$, and the same for ε_Y. Considering the velocity of the impacting particle, the friction and restitution coefficients it is possible to solve the system for the 12 unknown (three components of velocity and three components of the impulse force acting on a particle at times (1) and (2)). The rebound velocity components of the particles are:

$$u_{Rx} = u_{px}^{(0)} + \varepsilon_X f(1+e)u_{pz}^{(0)}, \tag{4.13a}$$

$$u_{Ry} = u_{py}^{(0)} + \varepsilon_Y f(1+e)u_{pz}^{(0)}, \tag{4.13b}$$

$$u_{Rz} = -eu_{pz}^{(0)}. \tag{4.13c}$$

In the case of no emitted particles during the impact ($N_s = 0$), the rebound velocity is the velocity of the particle rebound, in other words $\mathbf{u}_R(N_s = 0) = \mathbf{u}_p$. However, if other particles are emitted, the kinetic energy, due to the rebound velocity, is shared among all emitted particles. After computing N_s, using the momentum and energy conservation law it is possible to compute the average and standard deviation of particles velocity. Hu and Shao [107] described the model to compute the number of particles emitted by splash entrainment. The main assumption of their model involves the volume of the void-shaped crater created on the ground by the impacting particle. Following this approach, the volume of the crater is equal to the volume of the sand emitted from the bed. The crater volume excavated by an impacting particles is a function of the impacting particle mass, diameter, velocity and finally, soil properties,expressed as:

$$V_c = \frac{\pi \rho_p d_p^3 |\mathbf{u}_p^2|}{12 p_s} \left(sin(2\alpha_0) - 4sin^2(\alpha_0) + \frac{7.5\pi |\mathbf{u}_p| sin^3(\alpha_0)}{\beta_v d_p} \right). \tag{4.14}$$

where V_c is the crater volume, α_0 is the angle of the impacting particle velocity and the ground plane, $\beta_v = \sqrt{\frac{2p_s d_p}{m_p}}$ and p_s is the plastic pressure of the soil exerted on the particle and in general, it depends on the mechanical properties of the sediment bed. In this work, it is $10^3 Nm^{-2}$, following the approach in [141] and [142],

$$N_s = \frac{V_c}{\pi d_p^3/6}. \tag{4.15}$$

In this work, only a single type of particles are considered, with properties listed in table 4.1. However, the number of particles ejected in this way, may still change considerably with the velocity and angle of the impacting particle. In other works, different types of particles have been taken into account, and results show that N_s may reach high values ($10^5 - 10^6$), when particles of larger diameter (around $d_p = 10\mu m$), eject small diameter particles ($d_p = 1\mu m$) [141]. Using the rebound velocity defined previously, it is possible to compute the velocity of the emitted particles following a probabilistic approach. The splash-entrained particles are released with a velocity which follows a normal distribution. To compute the average and variance velocities of the emitted particle, it is necessary to impose momentum and energy conservation among emitted particles. The starting velocity of the emitted particle is defined with $\mathbf{u}_S$, and $\mathbf{u}_{Si}$ is the velocity vector of the i-th emitted particle. The relationship between $\mathbf{u}_S$ and $\mathbf{u}_R$ can be obtained using the momentum conservation law,

$$m_R \mathbf{u}_R = \sum_{i=1}^{N_{s+1}} m_i \mathbf{u}_{Si}. \tag{4.16}$$

The multivariate normal distribution for the emitted particle velocity is $\Phi(\mu_B, \Sigma_B)$:

$$\Phi = \frac{1}{\pi^{3/2} det(\Sigma_B)^{3/2}} exp^{-\frac{1}{2}(\mathbf{u}_s - \mu_B)^T \Sigma_B^{-1} (\mathbf{u}_s - \mu_B)}, \tag{4.17}$$

the mean and the covariance matrices of the normal distribution are:

$$\mu_B = \frac{1}{N_s + 1} \sum_{i=1}^{N_s+1} \mathbf{u}_{Si}, \tag{4.18a}$$

$$\Sigma_B = \begin{bmatrix} \sigma_{xx}^2 & 0 & 0 \\ 0 & \sigma_{yy}^2 & 0 \\ 0 & 0 & \sigma_{zz}^2 \end{bmatrix}, \tag{4.18b}$$

where

$$\sigma_{ij}^2 = \frac{1}{N_s+1} \sum_{i=1}^{N_s+1} (\mathbf{u}_{Si} - \mu_B)(\mathbf{u}_{Si} - \mu_B)^T, \tag{4.18c}$$

is the variance. The mean and variance can be estimated using the momentum and energy conservation laws. In general, impacting particles may eject different types of particles with different densities and diameters with respect to the impacting particle. However, in this work the particles are assumed equal in mass: $m_R = m_i$. Following this assumption, it is possible to obtain:

$$\mu_B = \frac{1}{N_s+1} \sum_{i=1}^{N_s+1} \mathbf{u}_{Si} = \frac{\mathbf{u}_R}{N_s+1}. \tag{4.19}$$

Furthermore, it is necessary to assume that the total kinetic energy of the rebounding particle is conserved, and it is computed as:

$$E_R = \frac{1}{2} m_R |\mathbf{u}_R|^2 = \frac{1}{2} \sum_{i=1}^{N_s+1} m_i |\mathbf{u}_{Si}|^2. \tag{4.20}$$

Assuming the covariance matrix diagonal, that only σ_{xx}^2,σ_{yy}^2 and σ_{zz}^2 are non-zero, and using the equation, it is possible to obtain the value of the variance from the kinetic energy and the rebound velocity as:

$$\begin{aligned} \sigma_{xx}^2 &= \frac{E_R}{m_R(N_s+1)} \frac{u_{Rx}^2}{(u_{Rx}^2+u_{Ry}^2+u_{Rz}^2)} - \frac{u_{Rx}^2}{(N_s+1)^2}, \\ \sigma_{yy}^2 &= \frac{E_R}{m_R(N_s+1)} \frac{u_{Ry}^2}{(u_{Rx}^2+u_{Ry}^2+u_{Rz}^2)} - \frac{u_{Ry}^2}{(N_s+1)^2}, \\ \sigma_{zz}^2 &= \frac{E_R}{m_R(N_s+1)} \frac{u_{Rz}^2}{(u_{Rx}^2+u_{Ry}^2+u_{Rz}^2)} - \frac{u_{Rz}^2}{(N_s+1)^2}. \end{aligned} \tag{4.21}$$

The splash entrainment model is now closed, using the impacting particle velocity and mass, it is possible to define the number of particles emitted and their initial velocity. This will allow to properly simulate the interaction between the ground and the impacting particle.

4.3 Eulerian modelling

The particle dust transport model used in this work is based on the convection-diffusion equation described in [140, 16, 15]. In its integral form, it can be defined as:

$$\int_V \frac{\partial \rho_d}{\partial t} dV + \int_V (\mathbf{u} + \mathbf{w_t}) \cdot \nabla \rho_d dV = \int_S \mathbf{S_d} \cdot \mathbf{n} dS + \int_V \nu_p \nabla^2 \rho_d dV \tag{4.22}$$

and using a number density: $\rho_d = \frac{N_p}{V}$ $[N_p/m^3]$ where w_t is the fall out velocity, which takes into account the effect of gravity with, ν_p the particle diffusion coefficient, and S_d a the source term, computed starting from the uplift Bagnold method. The fall out velocity is a vector, and considering the gravity direction as z, it can be written as $w_t = [0\ 0\ \hat{w}_t]^T$.

Starting from the 2nd Newton law, for a falling particle in still air where the gravitational force and the aerodynamic drag are equal:

$$(\rho_p - \rho) g \frac{\pi d_p^3}{6} = C_D \frac{\pi d_p^2}{4} \frac{\rho w_t^2}{2} \tag{4.23}$$

For $Re_p < 1$, Reynolds define $C_D = 24/Re_p$. Considering $Re_p = \frac{d_p w_t}{\nu_{ref}}$ it is possible to obtain:

$$w_t = \frac{g d_p^2}{18 \nu_{ref}} \frac{(\rho_p - \rho)}{\rho} \tag{4.24}$$

In this situation, we assume that $Re_p < 1$ everywhere, which may not be strictly true, however for the of this work it is acceptable. Furthermore, we consider that particles reach their terminal velocity instantaneously. The terminal velocity as used so far is dimensional, and expressed in terms of $[m/s]$, and it is necessary to be non-dimensionalized with U_{ref}. It is necessary to define $St = \frac{\tau_p}{\tau_f}$, where τ_p is the particle response time, defined as $\tau_p = \frac{\rho_p d_p^2}{18 \mu_{ref}}$, while $\tau_f = L_{ref}/U_{ref}$ is the fluid response time. Finally, we define the Froude number. $Fr^2 = \frac{U_{ref}^2}{g L_{ref}}$. With some algebra it is possible to reach:

$$\frac{w_t}{U_{ref}} = \frac{g d_p^2}{18 U_{ref} \nu_{ref}} \frac{\rho_p}{\rho} \left(1 - \frac{\rho}{\rho_p}\right), \tag{4.25}$$

Introducing the particle response time:

$$\hat{w_t} = \frac{g\tau_p}{U_{ref}}\left(1 - \frac{\rho}{\rho_p}\right), \tag{4.26}$$

and introducing the Stokes number,

$$\hat{w_t} = \frac{StL_{ref}g}{U_{ref}^2}\left(1 - \frac{\rho}{\rho_p}\right), \tag{4.27}$$

Finally, with the Froude number:

$$\hat{w_t} = \frac{St}{Fr^2}\left(1 - \frac{\rho}{\rho_p}\right). \tag{4.28}$$

Once that the terminal velocity has been modelled, it is necessary define the source terms. The source term can be modelled using the mass fluxes defined by Marticorena [143], in the horizontal (Q) and vertical (F) direction. In this case, the x and y axis define the ground plane, while z is normal to it. We define $\mathbf{F_m} = [Q\,Q\,F]^T$. It is necessary to start with the horizontal flux, computed starting from the threshold friction velocity.

$$Q = Ecu_*^3\frac{\rho}{g}\left(1 - \frac{u_t^*}{u_*}\right)\left(1 + \frac{u_t^{*^2}}{u_*^2}\right). \tag{4.29}$$

Where u_t^* is the threshold friction velocity (Bagnold model [106], [107]), while $u_* = \sqrt{\frac{\tau_w}{\rho}}$ is the friction velocity. Furthermore, $c = 0.261$ is a coefficient of the model [143], ρ is the flowfield density, and E is the ratio of erodible to total surface area, taken as 1 for simplicity. From the horizontal flux, it is possible to compute the vertical flux [143] as:

$$F = Qe^{13.4f-6.0} \tag{4.30}$$

In this work the parameter was set to f=0.1, following [143]. The vertical flux F is defined in Manticorena et al. work [143] as the mass of fine particles passing thought a horizontal unit area per unit time. The equation derived by Manticorena et al. [143], is defined as a mass flux, with units of $[\frac{kg}{ms}]$. To use it for the source term, it must be modified to obtain a flux as number of particles

per unit volume. The flux mass has been modified as: $\mathbf{S_d} = [Q\ Q\ F]^T \frac{d_p}{m_p A_p}$. Where m_p is the particle mass, d_p is the particle diameter and A_p the particle frontal area, which can be considered the smallest area through a particle can enter the flowfield from the ground.Then,

$$Q\frac{d_p}{m_p A_{ref}} = \frac{Ecd_p}{m_p A_{ref}} u_*^3 \frac{\rho}{g}\left(1-\frac{u_t^*}{u_*}\right)\left(1+\frac{u_t^{*2}}{u_*^2}\right) = \frac{pu_*^3}{m_p A_{ref}}\frac{\rho d_p}{g}. \tag{4.31}$$

where $p = Ec\left(1-\frac{u_t^*}{u_*}\right)\left(1+\frac{u_t^{*^2}}{u_*^2}\right)$, and contains only dimensionless values. It is important to notice that if $u_{*t} > u_*$, the threshold model is not satisfied, in other words, particles cannot be uplifted, and this may lead to a zero flux, leading to $S_d = 0$, in case of $u_t^* > u_*$. We now define ϕ, the volume fraction occupied by the particles as $\phi = \rho_d V_p$. To obtain this, we can multiply both sides of equation with V_p, volume of the particle. It also necessary to apply the divergence theorem to the source term in equation 4.22, to obtain:

$$\int_V \frac{\partial \phi}{\partial t} dV + \int_V (\mathbf{u}+\mathbf{w}_t)\cdot\nabla\phi dV = V_p \int_V \nabla\cdot\mathbf{S_d} dV + \int_V \nu_p \nabla^2\phi dV. \tag{4.32}$$

Considering an arbitrary volume, we can obtain:

$$\frac{\partial\phi}{\partial t} + (\mathbf{u}+\mathbf{w}_t)\cdot\nabla\phi = V_p\nabla\cdot\mathbf{S_d} + \nu_p\nabla^2\phi \tag{4.33}$$

It is now necessary to non-dimenzionalize equation 4.33, using:

$$t = \frac{t^*}{L_{ref}/U_{ref}} \quad x = \frac{x^*}{L_{ref}} \quad u = \frac{u^*}{U_{ref}} \tag{4.34}$$

As in the previous section, variables with superscript * have dimensions. It is necessary to define $\sigma = \frac{\rho}{\rho_p}$, which is the ratio between the density of the fluid and the density of the soil material. We also define the ratio between the particle diameter and the reference length: $\hat{d_p} = \frac{d_p}{L_{ref}}$. Furthermore, the Schmidt number is introduced: $Sc = \frac{\nu}{\nu_p}$, where ν is the kinematic viscosity of the fluid, and ν_p is the mass diffusion coefficient.

Rewriting the expression, with the dimensionless variables, we obtain:

$$\frac{\partial\phi}{\partial t}\frac{U_{ref}}{L_{ref}} + (\mathbf{u}+\mathbf{w}_t)\cdot\nabla\phi\frac{U_{ref}}{L_{ref}} = \frac{V_p}{L_{ref}}\nabla\cdot\mathbf{S_d} + \nabla^2\phi\frac{1}{L_{ref}^2}\frac{\nu}{Sc} \tag{4.35}$$

and with some algebra, it is possible to obtain:

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} + \mathbf{w_t}) \cdot \nabla \phi = \frac{V_p}{U_{ref}} \nabla \cdot \mathbf{S_d} + \nabla^2 \phi \frac{1}{Sc} \frac{1}{Re} \tag{4.36}$$

where $\mathbf{n}$ can be defined as $\mathbf{n} = [0\ 0\ 1]^T$. Considering the relationship between horizontal and vertical fluxes defined in 4.30, we can write:

$$\nabla \cdot \mathbf{S_d} = \left(\frac{\partial}{\partial x} \left(p u_*^3 \right) + \frac{\partial}{\partial y} \left(p u_*^3 \right) + \frac{\partial}{\partial z} \left(p u_*^3 \right) e^{13.4f - 6.0} \right) \frac{U_{ref}^3 \rho V_p d_p}{g m_p A_p} \tag{4.37}$$

Here, the friction velocity u_* is dimensionless.

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} + \mathbf{w}_t) \cdot \nabla \phi = \left(\frac{\partial p u_*^3}{\partial x} + \frac{\partial p u_*^3}{\partial y} + \frac{\partial p u_*^3}{\partial z} e^{13.4 - 6.0} \right) U_{ref}^3 \frac{V_p \rho}{g} \frac{1}{\rho_p V_p \frac{\pi \hat{d}_p L_{ref}}{4}} \frac{1}{U_{ref}} + \nabla^2 \phi \frac{1}{ScRe} \tag{4.38}$$

Defining $S = \left(\frac{\partial p u_*^3}{\partial x} + \frac{\partial p u_*^3}{\partial y} + \frac{\partial p u_*^3}{\partial z} e^{13.4 - 6.0} \right)$, for simplicity, and using the first term on the right of equation we obtain:

$$S U_{ref}^3 \frac{\rho}{g} \frac{V_p}{\rho_p V_p \frac{\pi \hat{d}_p^2 L_{ref}^2}{4}} \frac{L_{ref}}{U_{ref}} = S U_{ref}^2 \frac{4 \sigma V_p}{g \pi \hat{d}_p} \tag{4.39}$$

Introducing $\sigma = \frac{\rho}{\rho_p}$ and the Froude number, we have:

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} + \mathbf{w}_t) \cdot \nabla \phi = S \frac{4 Fr^2 \sigma}{\pi \hat{d}_p} + \nabla^2 \phi \frac{1}{ScRe} \tag{4.40}$$

The Sc number can be modelled using the Einstein-Stokes equation for Brownian motion. [140]. Then,

$$\nu_p = \frac{kT}{3 \pi \mu d_p}, \tag{4.41}$$

where k is the Boltzmann constant, T is the temperature, μ is the fluid viscosity, and as usual d_p particle diameter. Once that ν_p is defined, it is possible to obtain $Sc = \frac{\nu}{\nu_p}$.

4.4 PAXman Model - Computing forces over human body

The PAXman model was originally developed for military personnel, and it is used to compute wind forces on people operating in proximity to the rotor. It is based on the projection of a crouching 6ft tall person immersed in the outwash. The details of the geometry of the PAXman model are shown in figure 4.2, as reported in [19] and [18]. The red line shown in figure 4.2 represents the outline of the human body from a frontal view. It can be considered as the flow blockage area of a person facing the outwash of a rotor. At low height, the width is small representing the legs area. However, at middle-upper height, the width is larger, representing the chest and shoulder part. The coefficients described in table 4.2 are used to define the polynomial representation defined in figure 4.2. They are dimensionless and defined in [19, 18]. According to Silva et al. [19] and Preston et al. [18] the caution zone starts when the force acting on the PAXman is more than 80 lbf (335 N), and the hazard zone is defined after 115 lbf (510 N). The first threshold defines a limit where ground personnel may experience problems when working and performance limitations. On the other hand, the danger limit express the situation when ground personnel risk injury. The distribution of the force over the body is calculated as: $f_{paxman} = \frac{1}{2}\rho_{air}V_{rad}^2 x$ where ρ_{air} is density of the air, V_{rad} is the radial velocity and x is the horizontal coordinate of the PAXman model. The total force is the integral of the distribution of the force over the height of the PAXman model.

$$F_{paxman} = \int_{h_{PAXman}} f_{paxman} dz \tag{4.42}$$

Using this model it is possible to calculate the force distribution acting on a human body at a specific radial station.

To be comparable with the small-scale rotor, the PAXman height is scaled with the rotor radius for three different full-scale rotors, chosen to represent three different aircraft disk loading categories: low, medium and high. Their technical data are listed in next Chapters. For small-scaled rotor, the PAXman height has been scaled with the rotor radius, giving $h_{PAXman} = 0.3R$. Using the reference area expressed by the polynomial representation of figure 4.2, the distribution of the force produced by the outwash on the ground personnel is computed as proposed in [19].

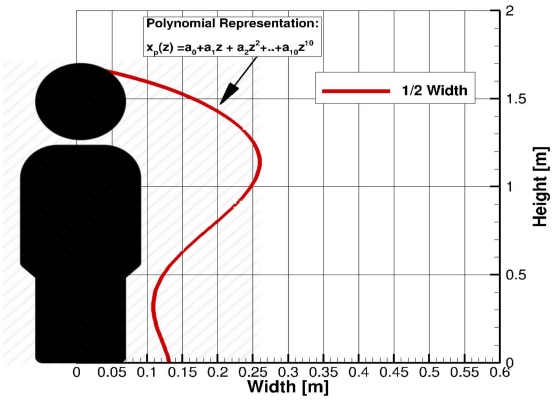


Figure 4.2: PAXman model [18].

a0	4.30939e-01
a1	-4.63972e-02
a2	-1.39649e-01
a3	1.37545e-01
a4	1.37545e-01
a5	-5.49253e-04
a6	2.21653e-04
a7	-4.18444e-05
a8	1.45194e-05
a9	-7.80009e-08
a10	-1.89822e-07

Table 4.2: Polynomial coefficients used in figure 4.2.

Chapter 5

Test cases[1]

In this chapter, the test cases used in this work are described, and numerical results are analyzed. CFD results are obtained with HMB3, that was described described previously in chapter 3. Numerical results involve rotor performance and flowfield analyses. Where possible, CFD results are compared with experimental data.

The first test case can be considered as a starting point for the further investigations. It has been originally presented by Ramasamy et al. [36], and is an experimental investigation of a small isolated rotor hovering OGE. Ramasamy et al. [36] presented results in terms of rotor performance and blade loads. A few years later, the same small rotor has been used by Lee et al. [26] for IGE experiments. In this case, experimental results involved rotor performance and outflow measurements. Finally, test cases involving a full-scale rotor in forward flight are presented. These are not compared with experimental data, however in previous works [144], HMB3 results for this specific test case have been compared with experimental and numerical results.

The final aim of this chapter is not a full validation of HMB3, this has been done extensively in previous works [124, 125, 126, 127]. However, comparing CFD results with experimental data, especially in terms of flowfield for rotor IGE, is important before starting the investigation involving particle tracking, with high accuracy.

[1] Part of this chapter has been published in Rovere Federico, Barakos George, and Steijl Rene. "Safety analysis of rotors in ground effect." Aerospace Science and Technology (2022): 107655

5.1 Isolated rotor OGE

HMB3 is well validated tor rotors in hover and forward flight, some additional work is carried out here on rotors tested for small scale rotors. [124, 125, 145, 146, 127].

5.1.1 Experiments by Ramasamy et al. [36]

The first computed cases were meant as a starting point for following investigations. In this simulation, the experiments by Ramasamy et al. [36] were analyzed. This experiment was selected because of its simplicity and the quality of the available data available. The test case subject of this first study was the hovering of an isolated micro-rotor OGE. The rotor had a radius of 86 mm and used two rectangular untwisted blades. The blades had cambered, circular-arc airfoil sections with a constant radius of curvature of 55 mm. The baseline blade had uniform chord of 19 mm, with a maximum camberline displacement of 1.25 mm and a constant thickness of 3.7%. The baseline section had blunt leading and trailing edges. Geometric properties are listed in table 5.1.

In the experiment, the flow field and performance of the blade were measured at a rotational frequency of 50 Hz. The nominal operating tip Mach number and chord Reynolds number were 0.082 and 35,000 respectively. Thrust and torque coefficients have been measured with different techniques (balance and bound circulation calculation) and PIV flow visualization has been performed. The test data will be compared with CFD results obtained during this work in the next sections. Figure 5.1 presents the blade geometry.

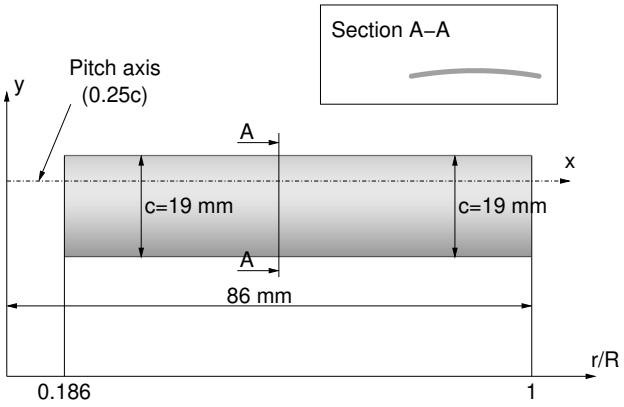


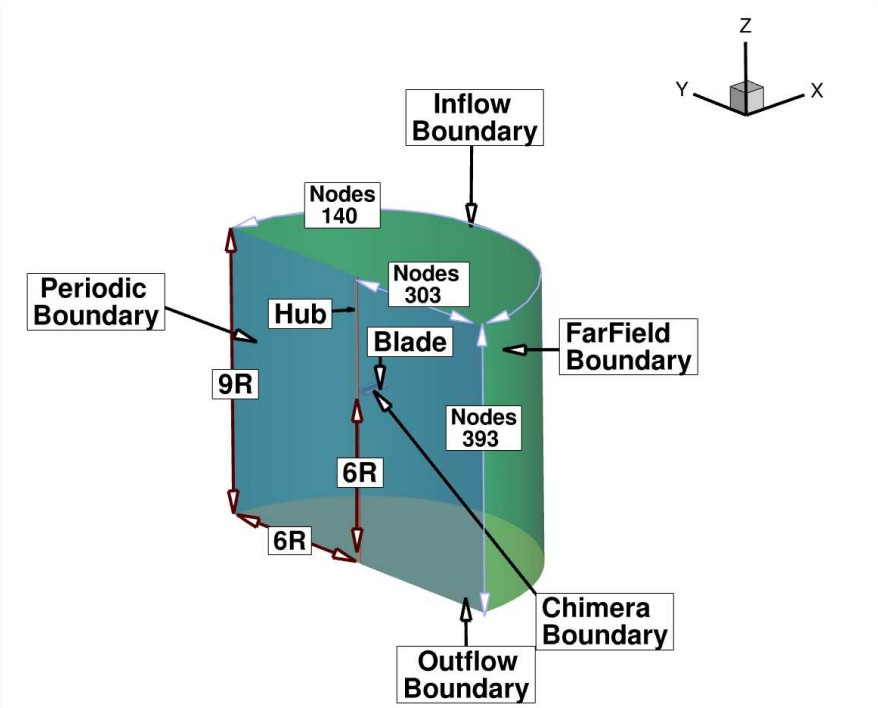
Figure 5.1: Geometry of the rectangular blade used for computations.

Parameter	Value
Number of blades, N_b	2
Rotor radius, R	86 mm
Blade chord (constant), c	19 mm
Aspect ratio, R/c	4.526
Camber-to-chord ratio	0.0645
Linear twist angle, Θ	0^o

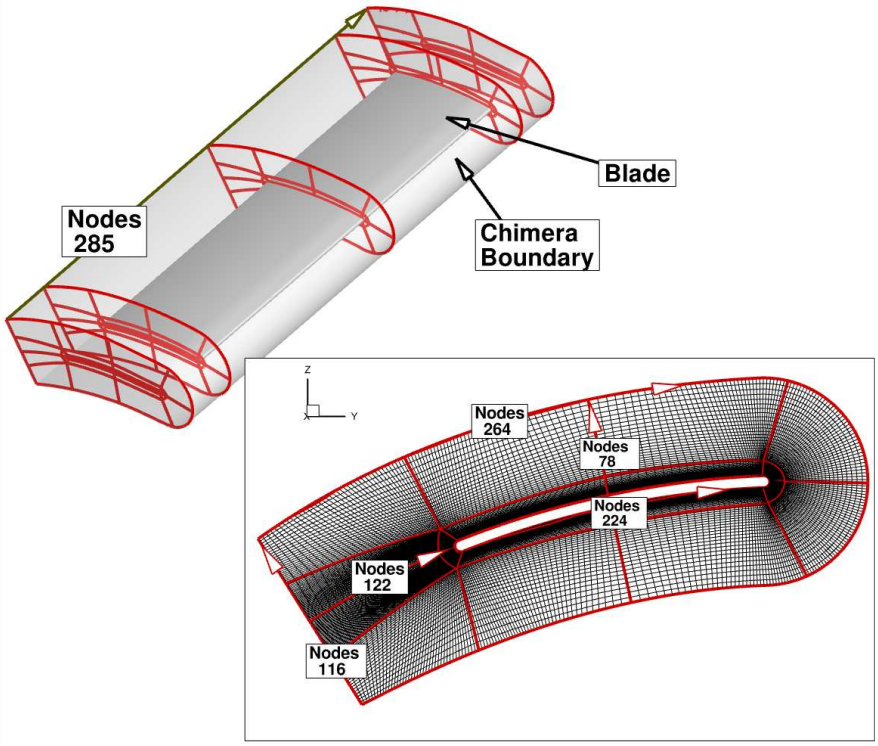
Table 5.1: Geometric properties of the Ramasamy et al. rotor [36].

The OGE test case has been analyzed, and the characteristics of the employed meshes are presented in table 5.2. Results in terms of rotor performances and forces along the rotor span are presented. The wake generated by the rotor is assumed to be steady. The hover configuration OGE can be seen as a steady problem, at least before blade stall. The periodicity of the flow in the azimuthal direction allows modelling of only half of the domain, as shown in figure 5.2 (a). The blade geometry of [36] was modelled using ICEM hexa [147]. The blunt leading and trailing edges were rounded, but this is not expected to have any significant influence on the solutions. Three meshes were generated to check for mesh influence on the solution. Froude inflow and outflow conditions have been imposed as reported in figure 5.2. The distances between the rotor blade and the boundary are based on experience from previous studies using the HMB3 solver [127, 148]. The "potential sink/Froude" boundary condition is used to minimize far-field influence.

A C-H multi-block topology was used around the blade, combined with a background mesh using the chimera method [149]. For all cases, the position of the farfield boundary were extended to 3R (above) and 6R (below and radial) from the rotor plane, which assures an independent solution with the boundary conditions employed. This set up has been used in previous works with HMB [127, 148]. In the radial and vertical directions, a non-uniform spacing was used to have a finer mesh



(a) Background mesh setup



(b) Chimera mesh setup

Figure 5.2: Computational domain and boundary conditions, nodes for baseline mesh.

close to the wake region.

Mesh Size	**Coarse**	**Baseline**	**Finer**
Number of cells (Millions)	11.7	19.6	57

Table 5.2: Meshes for Ramasamy et al. [36] test case.

5.1.2 Rotor performance results

The results obtained in terms of thrust and torque coefficients are shown in table 5.3.

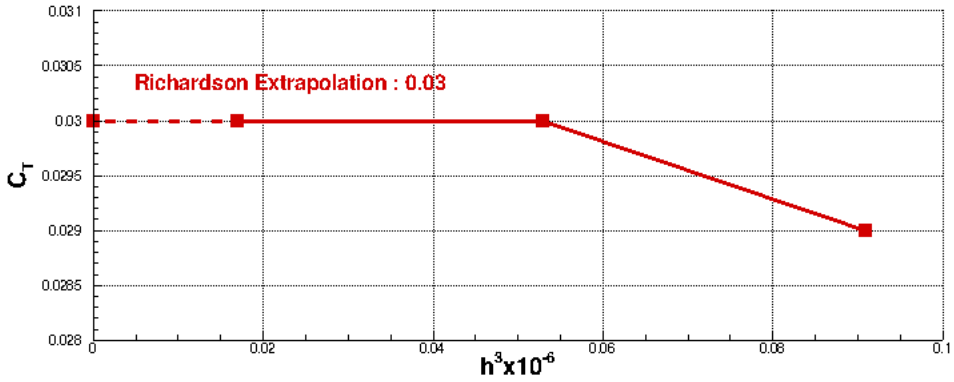
	C_T	C_Q
CFD (Current work, baseline mesh)	0.0302	0.0053
Ramasamy et al. Experiments [36]	0.0318	0.0061
Error percentage	5.6%	13.7%

Table 5.3: Meshes for Ramasamy et al. [36] test case.

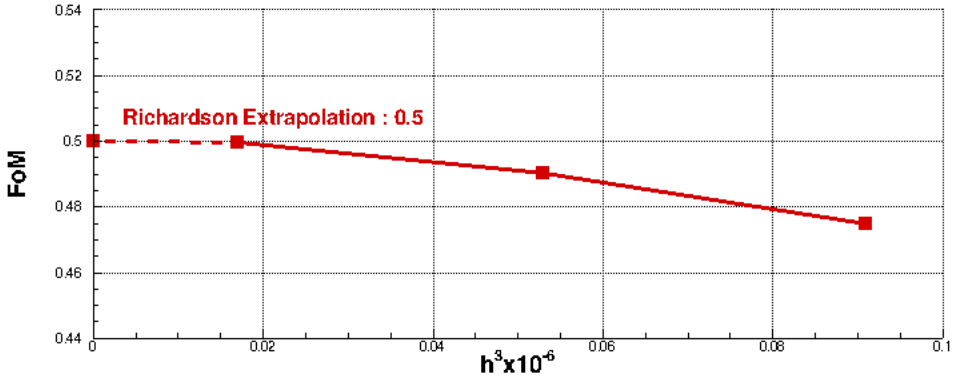
The CFD simulations predicted lower values than the experimental results for the torque and thrust at the collective and Reynolds numbers employed. The differences between the CFD and the experiment results are due to the low Reynolds numbers used for simulations and experiments. At experimental configuration, extended separation zones are present on the blade, and the shape of the blade (in terms of airfoil and plane shape) loses part of its influence on performance. For this reason, no further blade configurations were taken into account for CFD validation other than the baseline. Similar results were obtained by Karla et al. [?]

Mesh convergence is carried out, and in figure 5.3 results are presented in terms of C_T and FoM variation with the cell size. The cell size h is represented by $\frac{1}{N_{cells}^{1/3}}$, where N_{cells} is the the number of cells of the considered mesh. This is due to the complex mesh topologies and geometries which make the overall cell size hard to determine. As said, the mesh convergence is obtained with three levels of mesh refinement. Furthermore, for the thrust coefficient there is not substantial change between the baseline and finer grids. Richardson extrapolation for the current results for zero cell size has been obtained for both values [150, 151].

Due to the small dimensions of the rotor, it was not possible to measure the pressure along the span, so Ramasamy et al. [36] computed the lift as a function of r/R, starting from the bound



(a) Richardson extrapolation for C_T



(b) Richardson extrapolation for *FoM*

Figure 5.3: C_T and FoM variations with varying volume sizes h^3. The cell size $h = 1/N_{cell}^3$, where N_{cell} takes the sum of the near-field mesh cell numbers as presented in table 5.2

circulation at different span locations. This is difficult, but this technique has already been used in other works [152]. Using the velocity components obtained by PIV, Ramasamy et al. [36] obtained the bound circulation at different span locations. Velocity components are integrated around a closed path around the airfoil, as shown in figure 5.4. In this work, lift has been computed in an identical way, so it was possible to compare lift along the span starting from the bound circulation.

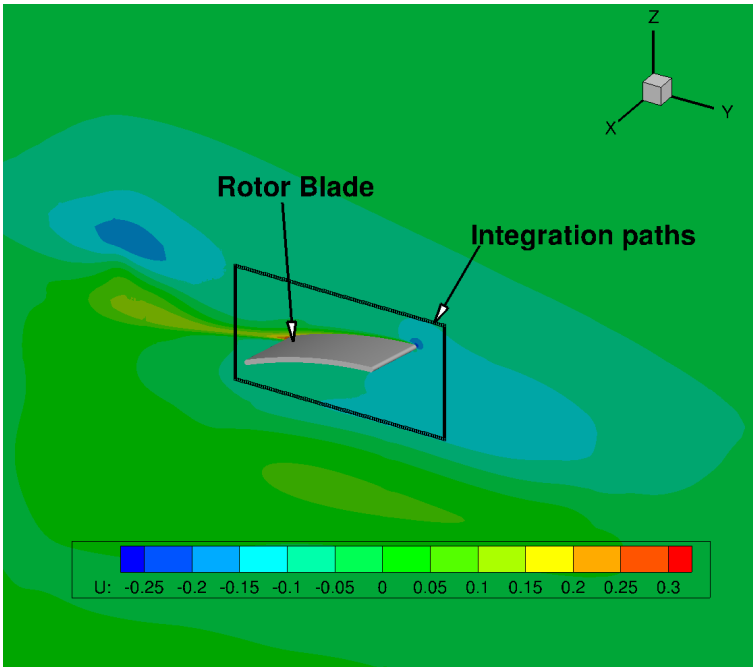


Figure 5.4: Schematic of circulation path.

The bound circulation is computed from velocity components as:

$$\Gamma_b = \oint_c \mathbf{uds}, \tag{5.1}$$

where Γ_b is the bound circulation, c is the closed path and **ds** is the infinitesimal part of the loop perimeter. Once the bound circulation is obtained, it is possible to obtain the lift along the span using Kutta–Joukowski's circulation theorem:

$$L = \rho \Gamma_b r \Omega \tag{5.2}$$

Where ρ is the air-density, Γ_b is the circulation along a closed path around the airfoil, r is the radial station, and Ω the rotational speed. The lift has also been computed with a surface pressure

integration. Both CFD results are compared in figure 5.5.

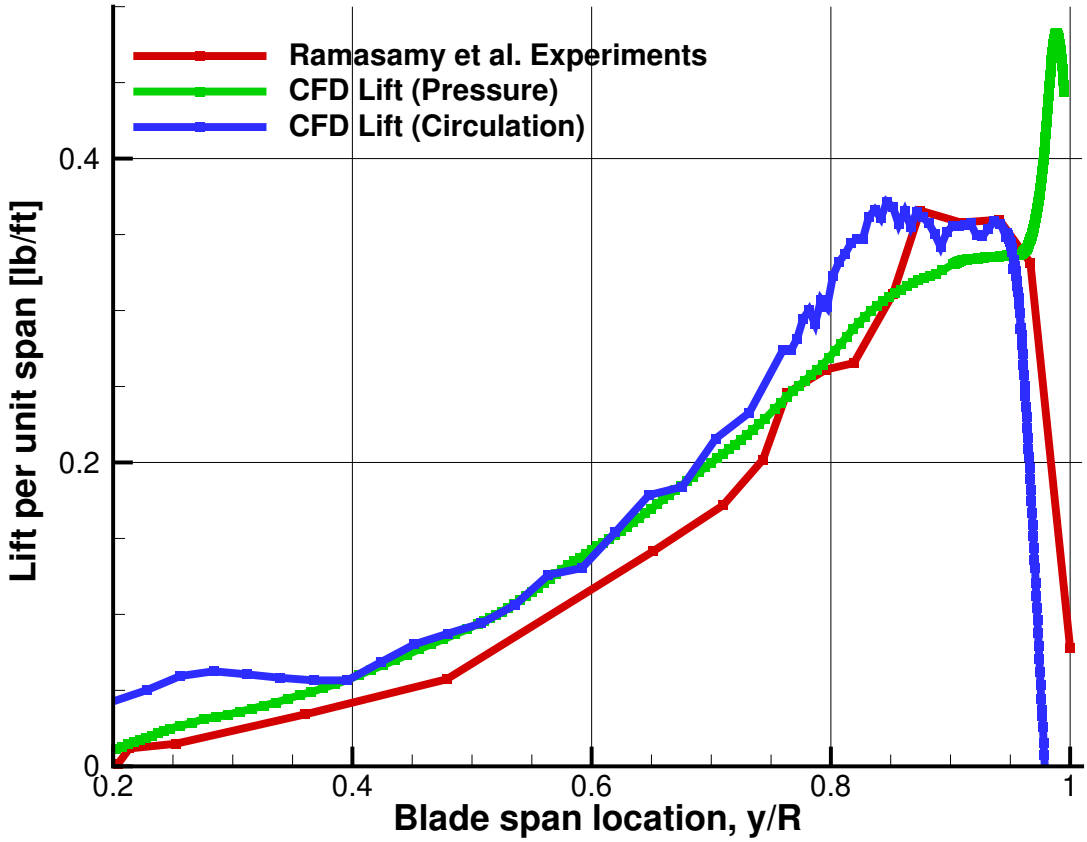


Figure 5.5: Lift over the span. Experimental results by Ramasamy et al. [36]. The rotor was operating OGE, at collective $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.082$. The $k-\omega$ turbulence model was used for calculations. Baseline mesh.

In general, the lift results along the span, are consistent with experiments. However, near the tip, the pressure and circulation based results appear different. At r/R=0.9, where the circulation lift data reach their peaks, the error between CFD and experimental circulation based lift is around 1.5%, while between CFD pressure based and experimental is about 8.3%. The reason for this is 3D effects due to the tip vortex in the proximity of the tip, and this violates the two-dimensionality assumed in the circulation calculation. Furthermore, in a two bladed rotor, the vorticity generated by the second blade affects the bound circulation computed [152]. The pressure integration results are more accurate, and capture the tip influence. The circulation comparison shows that CFD and experiments agreed when the processed in the same way.

For the drag, Ramasamy et al. [36] computed the loses of momentum stations ahead and behind the rotor, using:

$$D = \int_{-\infty}^{\infty} (p_1 + \rho ||\mathbf{u}_1||^2) d\mathbf{s} - \int_{-\infty}^{\infty} (p_2 + \rho ||\mathbf{u}_2||^2) d\mathbf{s}. \tag{5.3}$$

Where D is the drag, p is the flow pressure, ρ is air density and $\mathbf{u}$ is the flow velocity. Finally, subscripts 1 and 2 define the stations ahead and behind the blade, respectively. Infinite bounds are related to freestream conditions.

In figure 5.6 CFD pressure based drag and experimental results are compared. As seen for the lift, the CFD and experimental results are in agreement, even if in the proximity of the tip, with the CFD overestimating the drag. At r/R=0.9, where the CFD and experimental results reach their peaks, the error is about 45%. As described for the lift case, this may be due to 3D effects due to the tip vortex, which cannot be fully taken into account fully with the applied drag deficit approach.

In figures 5.7 CFD rotor wake and experimental results are compared. CFD results are shown in terms of vorticity magnitude. CFD rotor wake shows similar wake features with respect to the experiment. It is possible to distinguish three separated tip vortices in the wake. The first, closer to the blade, has a higher and denser vorticity concentration, which start to diffuse when the vortices move inside the wake. Experimental results show clearly the contraction of the wake, which is not so evident in CFD result, due to a different visualization.

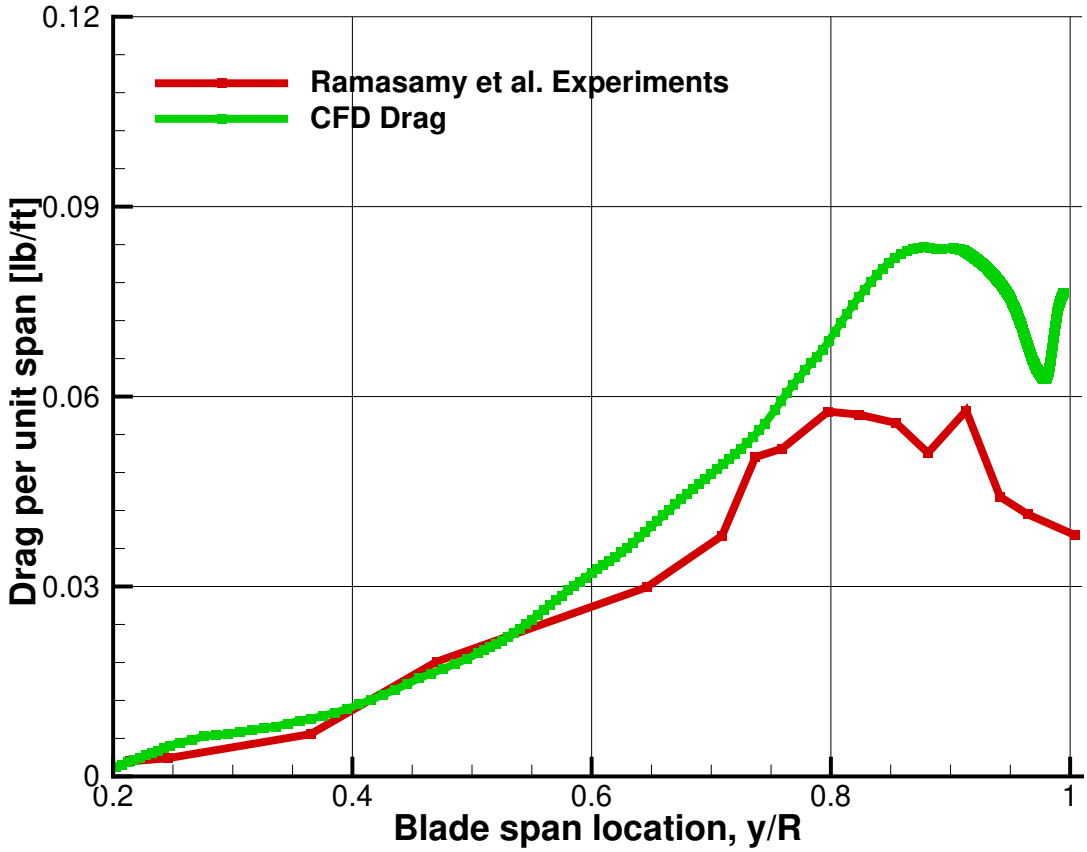
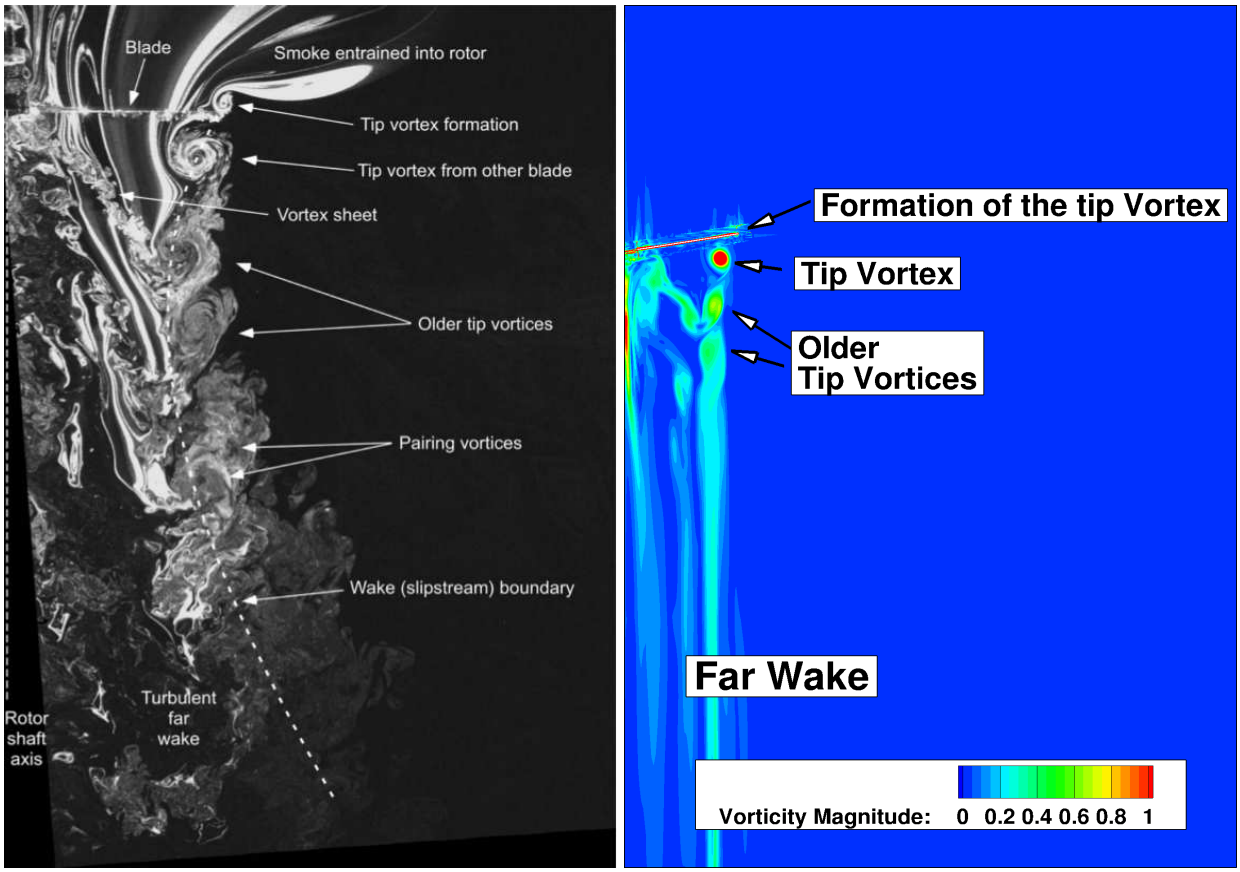


Figure 5.6: Drag over the span. Experimental results by Ramasamy et al. [36]. The rotor was operating OGE, at collective $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.082$. The $k-\omega$ turbulence model was used for calculations. Baseline mesh.



(a) Experimental by Ramasamy et al. [36]

(b) CFD

Figure 5.7: Experimental and CFD wake visualization at OGE conditions The rotor corresponds to the design reported by Ramasamy et al. [36], at collective $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.082$. The $k-\omega$ turbulent model was used for calculations. Baseline mesh.

5.2 Isolated rotor hovering IGE

HMB3 is well-validated for rotor in hover and forward flight, some additional work is carried out here on rotors tested near ground. [108, 124, 125, 145, 146, 127].

5.2.1 Experiments by Lee et al. [26]

Experimental data of the rotor described in the previous section are also available for hovering IGE. The IGE experiment was performed at the University of Maryland by Lee et al. [26]. The same two-bladed rotor of the previous test case was positioned over a flat ground plane at different heights. The rotor was set parallel to the ground plane and operated at a rotational frequency of 50 Hz, which corresponded to a tip speed of 27.02 ms^{-1}. The experimental data of this work involve loads measurements (obtained by a micro mass balance) and flow PIV visualization. Particular effort had been spent on the visualization and measurement of the outwash. Geometrical data and operational conditions are resumed in table 5.4.

Table 5.4: Geometric properties of the rotor by Lee et al. [26]

Parameter	Value
Number of blades, N_b	2
Rotor radius, R	0.086 m
Blade chord, c	0.019 m
Solidity, σ	0.14
Tip speed, V_{tip}	27 m/s
Rotor height to diameter, h/R	0.25 to OGE
Collective pitch at 0.75 radius θ_c	12deg

To obtain better results, a central body (hub) has been added to the configuration. Figure 5.8 illustrates the blade geometry.

The rotor was modelled using two overset meshes, and the ground was modelled by imposing no-slip conditions. The full rotor domain was computed as unsteady (figures 5.2.1). The unsteady time step was changed during the simulation. The first two revolutions were computed at 5deg. Then, the timestep was gradually reduced reaching 0.5deg/timestep for the last revolutions performed. In total, 5 full revolutions were simulated. The mesh has been refined near the ground

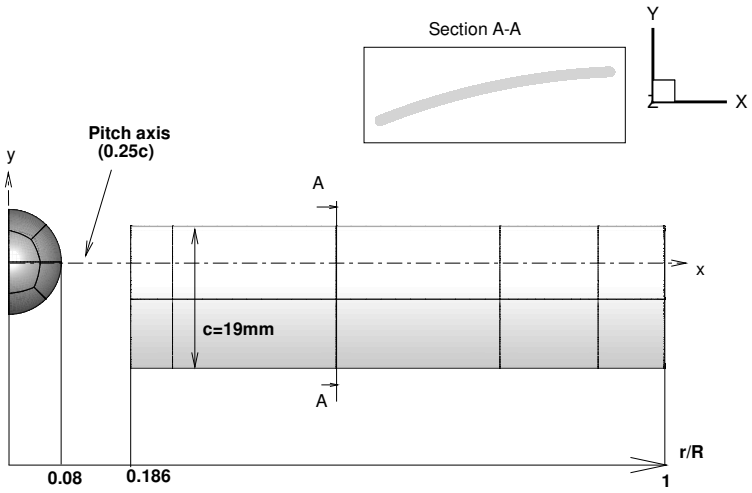


Figure 5.8: Geometry of the rectangular blade used in the present computation.

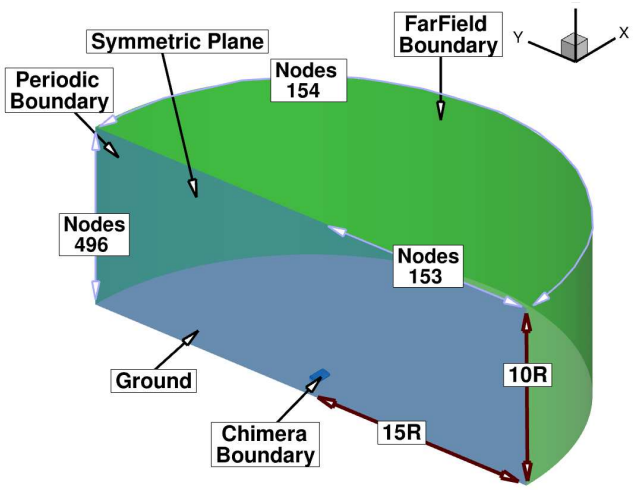


Figure 5.9: Computational domain and boundary conditions.

and near the rotor to accurately resolve the wake features. Different rotor heights above the ground configurations were tested, with the rotor collective $\theta = 12deg$.

A single composite mesh as been used for these calculations, where the distance between the rotor and the ground has been changed to obtain the different cases. Mesh sizes are listed in table 5.5. The mesh is composed of a cylindrical background, a central hub and two blades. The reference length used in mesh scaling is the blade chord.

Background Mesh	20 M
Foreground Mesh (one Blade)	19.5 M
Foreground Mesh (Hub)	0.01M
Final Mesh	40M

Table 5.5: Chimera meshes for the Lee et al. [26] test case. M=Millions.

5.2.2 Rotor performance results

Comparisons in terms of C_T^{IGE}/C_T^{OGE} are given in table 5.10,where CFD results are compared to the experiments by Lee et al. [26]. In the present CFD work, the increment of the thrust coefficient IGE is about 4% for 1.5 radius height (h/R=1.5), 10% for h/R=1, 27% for h/R=0.5 and 43% for h/R=0.25 with respect to the OGE case. To obtain the experimental results presented in figure 5.10, the Lee et al. [26] data are interpolated to find the thrust produced at constant power. On the other hand, CFD results show that the power coefficient is almost constant between the OGE and IGE cases for all distances from the ground tested, as shown in table 5.6. For this reason, a direct comparison between Lee et al. [26] interpolated data and CFD results is possible. CFD results show the same trend as experiments, increasing the thrust coefficient when the distance between the rotor and the ground is decreased. The error between experiment and CFD results in terms of C_T^{IGE}/C_T^{OGE} is about 3% with h/R=0.5, 2% at h/R=1, and at h/R=1.5 the error is 3%.

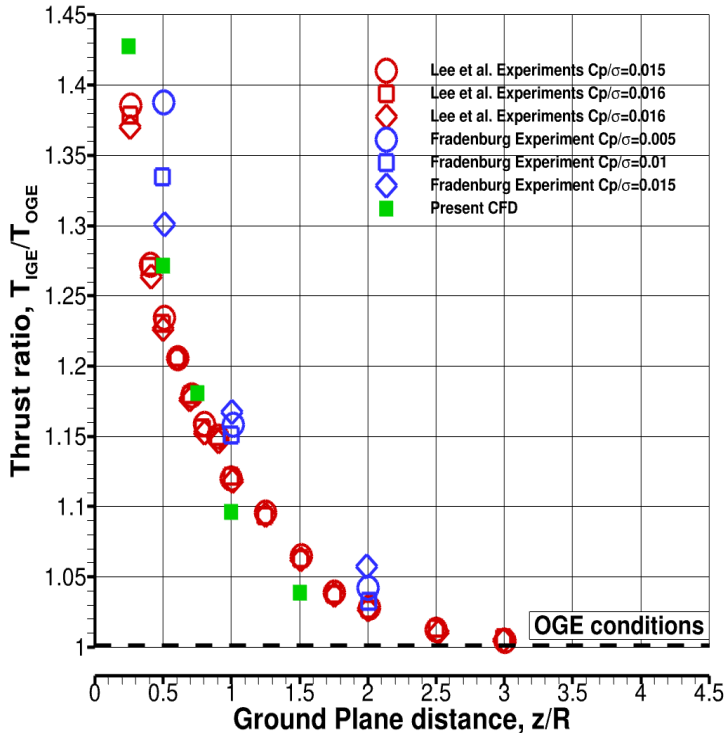


Figure 5.10: Experimental and CFD results for the trust ratios IGE at constant power. Experimental data by Lee et al. [26]

The results in terms of the power coefficient ratios between IGE and OGE are close to 1.0 for all cases. It can be considered that the power between OGE and IGE is fairly constant for all rotor heights, with a small increment in the thrust coefficient, as expected. Similar results in terms of C_Q^{IGE}/C_Q^{OGE} have been obtained by Karla et al. [88] for the same test case. These CFD results are compared in table 5.6, with present simulations.

Table 5.6: C_Q^{IGE}/C_Q^{OGE} as function of rotor height

Performance	Exp	CFD Karla2010	CDF Present Work
h/R=0.25	1.070	1.18	1.072
h/R=0.5	1.042	1.07	1.044
h/R=1.0	1.049	1.06	1.034
h/R=1.5	1.038	1.06	1.031

5.2.3 Flowfield visualization

In figures 5.11 and 5.12 the time-averaged radial velocities are shown and compared with experiments at three heights above ground, and at two radial stations. The time-averaged outflow has been normalized using two reference velocities. The left columns of figures 5.11 and 5.12 show results scaled with the reference hover induced velocity $v_i = \frac{\sqrt{C_T}}{2}$, as proposed by Lee et al. [26]. The right column shows the same results scaled like a jet, using the peak radial velocity v_{max}.

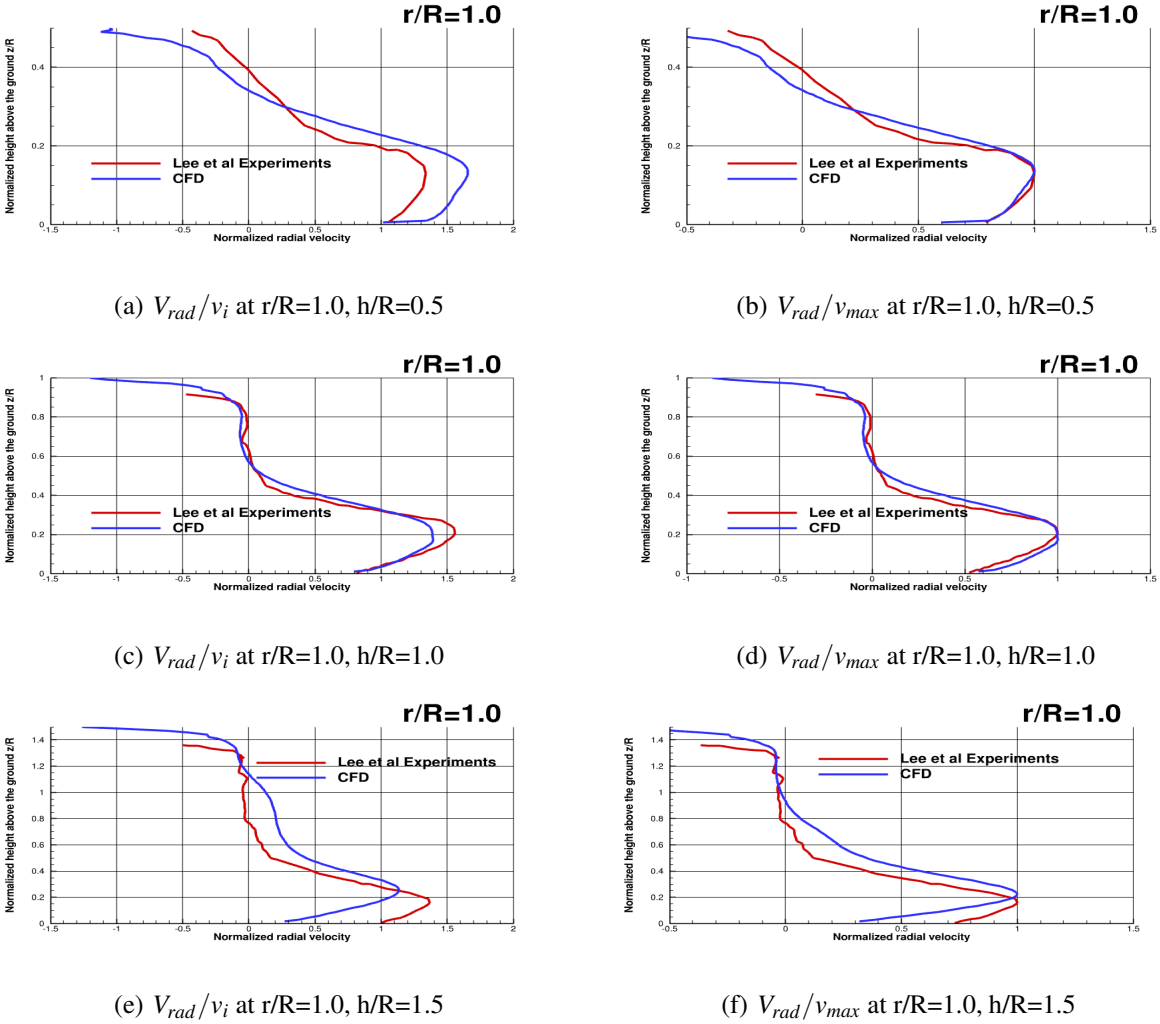
At a radial distance of r/R=1, the radial velocity peak is reached around 0.2R, independently of the rotor distance from the ground, with a scaled maximum velocity of around 1.5 for all three rotor configurations. At this radial station, the momentum is well distributed along the distance between the rotor and the ground, for rotors at h/R=0.5 and h/R=1.0. For h/R=1.5 the momentum is more localized near of the ground.

For the radial position at r/R=1, with h/R=0.5, the error at the peak is around 23% (overestimated), 10% for h/R=1 (underestimated) and 16% (underestimated) for h/R=1.5. Furthermore the CFD predicts well the peak height when the rotor is operating closer to the ground. The error in peak position is about 5% for h/R=0.5 and 4% for h/R=1, while it is 60% for h/R=1.5, this may appear significant, however considering $\frac{h_{EXP}^{peak} - h_{CFD}^{peak}}{1.5}$, the delta between the CFD and experimental peak height position, normalized with the distance above the ground, is only 7%.

On the other hand, for r/R=1.5 the momentum is concentrated closer to the ground for all three rotor configurations, with a lower peak position (under 0.2R). At this radial distance, the flow oscillations are almost dissipated, and the outflow is well established, defining a jet like flow, typical of rotors IGE as also described by Ramasamy et al. [25]. At this radial distance, the CFD overestimates the peak velocity for all configurations. However, the errors are in the immediate vicinity of area where peak values occur. This can be seen in fig 5.11 (b) and (d), where the outflow is scaled with v_{max}. In these cases the momentum distribution is closer to experiments, furthermore the peak position is well predicted by CFD.

In figures 5.13, the rotor wake is presented as (a) iso-surfaces of Q-criterion and (b) contours of vorticity magnitude, for the h/R=1 configuration. The Q criterion is related to the velocity

gradient tensor, and its iso-surfaces are a good indicator of how vortical flow may be [153]. It is possible to notice how the wake initially contracts inside the rotor area, and then expands as the tip vortices reach the ground. Contour levels in figures 5.13 (b) also show how the vorticity generation contained in the vortices diffuse. The higher vortex is smaller and with high vorticity values with respect to the vortices in the proximity of the ground. Finally, it is also possible to detect the presence of the starting vortex in proximity of 1.2R. Its strong influence on the seeded ground will be discussed later, in the chapter 7.



(a) V_{rad}/v_i at r/R=1.0, h/R=0.5

(b) V_{rad}/v_{max} at r/R=1.0, h/R=0.5

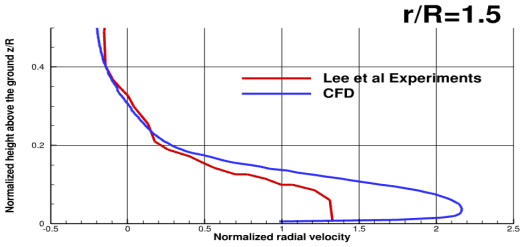
(c) V_{rad}/v_i at r/R=1.0, h/R=1.0

(d) V_{rad}/v_{max} at r/R=1.0, h/R=1.0

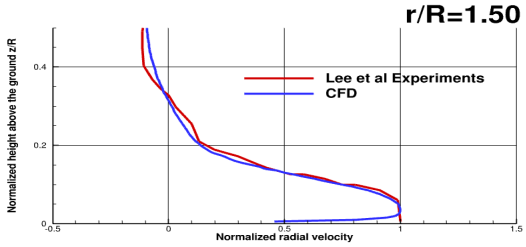
(e) V_{rad}/v_i at r/R=1.0, h/R=1.5

(f) V_{rad}/v_{max} at r/R=1.0, h/R=1.5

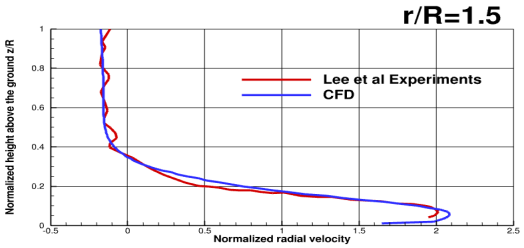
Figure 5.11: Experimental and CFD time-averaged outflow velocity profiles at different heights above the ground. The rotor was operating, $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.08$. Left column results are scaled with induced hover velocity, while the right column results are jet-scaled. Experimental data by Lee et al. [26].



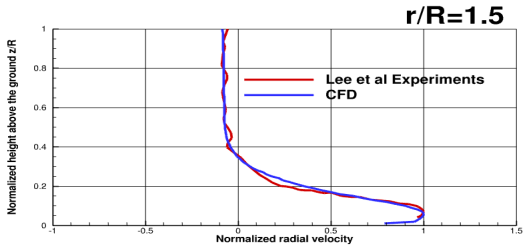
(a) V_{rad}/v_i at r/R=1.5, h/R=0.5



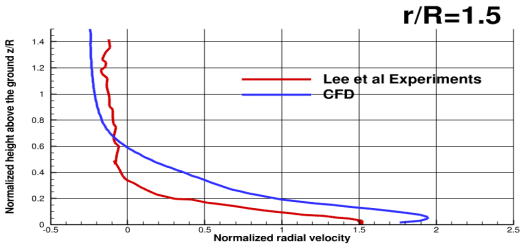
(b) V_{rad}/v_{max} at r/R=1.5, h/R=0.5



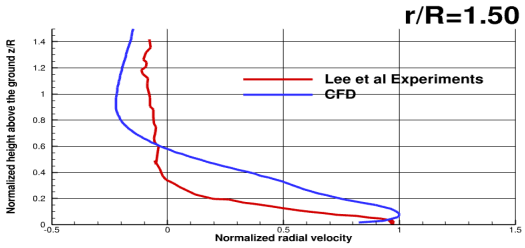
(c) V_{rad}/v_i at r/R=1.5, h/R=1.0



(d) V_{rad}/v_{max} at r/R=1.5, h/R=1.0

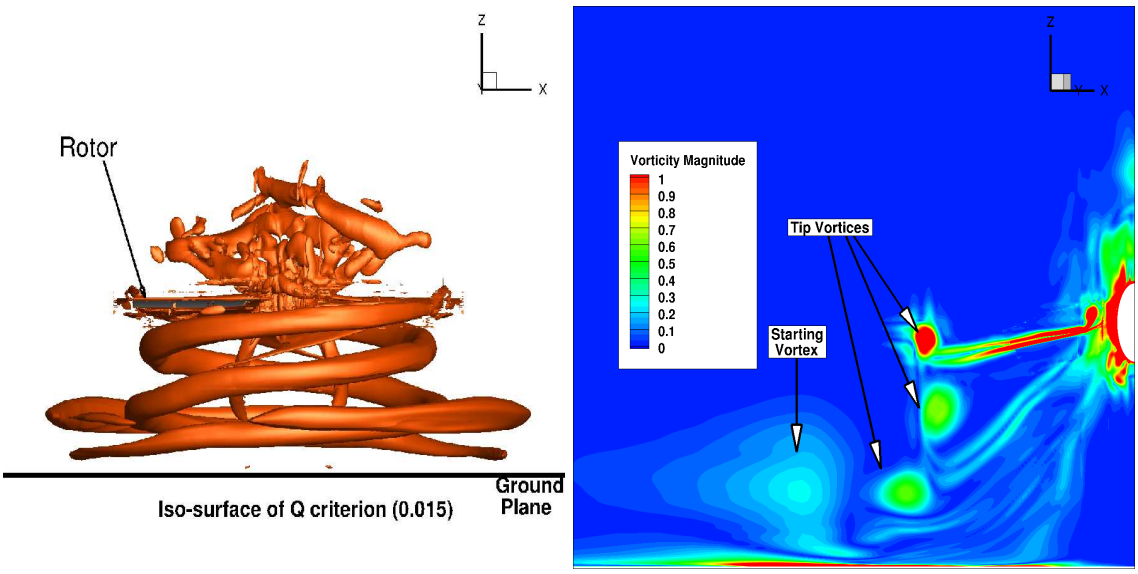


(e) V_{rad}/v_i at r/R=1.5, h/R=1.5



(f) V_{rad}/v_{max} at r/R=1.5, h/R=1.5

Figure 5.12: Experimental and CFD time-averaged outflow velocity profiles at different heights above the ground. The rotor was operating, $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.08$. Left column results are scaled with induced hover velocity, while the right column results are jet-scaled. Experimental data by Lee et al. [26].



(a) Q criterion iso-surface (b) Vorticity

Figure 5.13: Wake Visualization, for the IGE case, proposed by Lee et al. [26]. The rotor operating at h/R=1, $\theta_{75} = 12deg$, $Re_{tip} = 35000$, $M_{tip} = 0.08$, $h/R = 1$ and $C_T = 0.03$.

5.2.4 Taxiing rotor in IGE

The forward flight cases have been directly computed at full-scale. The flow conditions and geometric data are listed in table 5.7. Two advance ratios have been considered. Furthermore, for the test case at 10 kts two disk loadings been tested. The wake structures of a forward flight rotor strongly depend on its taxiing speed [144, 22]. As widely reported in past numerical and experimental works [144, 22], at lower taxiing speed, a recirculation region is formed in front of the aircraft, while in the case of higher taxiing speeds the flowfield resembles a horseshoe vortex. The safety analyses performed in this thesis were strongly influenced by the obtained wakes. The two different wake patterns may generate different scenarios in terms of safety for outflow forces and for particle uplift. Furthermore, using two thrust coefficients for the same helicopter in forward flight, allows understanding of how the disk loading influences safety near operating rotors.

CFD simulations were performed using a chimera mesh, composed by a background and a foreground component for both cases. All mesh sizes are listed in table 5.8. The mesh setup for the 40kts case is presented in figure 5.14. The two meshes show different sizes, due to the different expected wake topologies for 10kts and 40kts, respectively. The inner mesh contains the

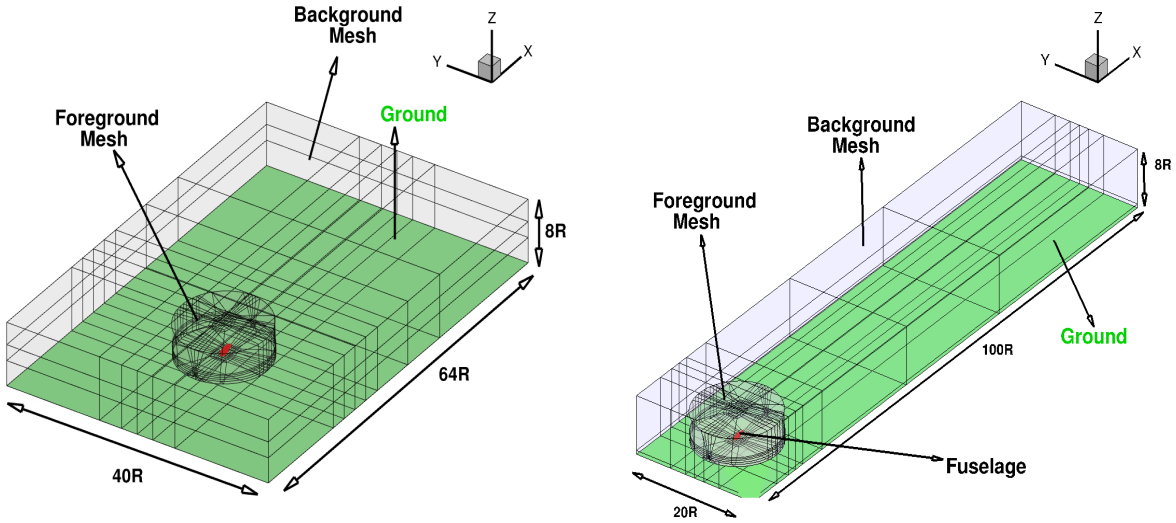
V_∞ $[kts]$	10	10	40
V_∞ $[m/s]$	5.14	5.14	20.58
M_∞	0.0151	0.0151	0.0605
Re	144500	144500	578000
μ	0.0233	0.0233	0.0932
N blades	4	4	4
R [m]	5.5	5.5	5.5
c [m]	0.41	0.41	0.41
DL $[kg/m^2]$	370	510	370
C_T	0.0125	0.0175	0.0125
σ	0.095	0.095	0.095
$\frac{C_T}{\sigma}$	0.13	0.18	0.13

Table 5.7: Forward flight conditions and rotor geometric data.

fuselage, and it is of cylindrical shape. It contains approximately 15 million cells. The fuselage is an approximate shape of a utility helicopter [144]. On the other hand, the background meshes were Cartesian. The first mesh was designed for the high advance ratio case (40 kts), it has a width of 20R in the spanwise direction, and extends 75R downstream. It has been refined near the ground and near its center, with a total of 23 million cells. A different background mesh has been designed for the 10kts case, due to the expected flowfield characteristics that include a horseshoe vortex. For this taxiing speed the wake is expected to extend in the spanwise direction. In this case, the width of the domain was 40R, while the stream wise length was about 64R as shown in figure 5.14 (b). The background mesh for the 10 kts case contains 30 million cells.

Forward speed case	10 kts	10 kts (no fuselage)	40 kts
Background Mesh	30 M	28 M	23 M
Foreground Mesh (Fuselage)	15M	-	15M
Final Mesh	45M	28M	38M

Table 5.8: Meshes for forward flight test case. M=Millions



(a) CFD domain for 10 kts

(b) CFD domain for 40 kts

Figure 5.14: Forward flight mesh setup.

The ground has been computed as a moving wall, imposing the freestream velocity on its, to simulate the relative velocity between the aircraft and the ground. This approach is computationally cheaper than translating a foreground mesh inside a background mesh, and it is frequently used in wind tunnel experiments, usually in the automotive field, but also for rotor cases [13].

5.2.5 Non uniform actuator disk

The rotor has been modelled with a steady, non-uniform actuator disk as proposed by Shaidakov [154, 120]. In an uniform actuator disk, the pressure jump is constant across the rotor disk, and it depends only on the thrust coefficient C_T and the advance ratio μ [45]. However, in forward flight, the load distribution is not uniform and a more accurate model is needed. In HMB3, the Shaidakov actuator disk model is implemented. It expresses the pressure jump as a function also of rotor blade radius, azimuth angle Ψ, rotor solidity, rotor attitude, lift coefficient and freestream velocity. Furthermore, in the model takes into account the rotor hub and the blade tip offload, as well as the rotor reverse flow region. This model has been successfully used in several works [155, 156, 120].

The predictive capabilities of the actuator disk model have been proved in past works with

HMB3 [120, 144, 157]. Furthermore, the work of Tanabe et al. [144] a comparison between the rotor wake computed by an actuator disk and full resolved blades approach has been carried out. Results suggested that the non-uniform actuator disk is able to capture the main features of the wake generated by a rotor in forward flight operating IGE.

Chapter 6

PAXman model - Forces on a human body[1]

6.1 Scaling factors

Analyzing the flowfield velocities generated by model scale rotors is not adequate for safety purposes, because their air velocities are too low. For this reason, scaling factor has been applied to the flowfield velocity components for the hovering rotor. Model scale refers to the micro-rotor operating conditions presented in the experimental paper by Lee et al. [26], described previously in Chapter 5. The thrust coefficients of the micro-rotor are listed in table 6.1. These values have been obtained by the simulations described in Chapter 5.

First, a scaling is necessary to obtain values comparable to full scale rotor wake velocities. For this initial purpose, the blade tip velocity can been used, as listed in table 6.2 for the full-scale rotor. This first step it is necessary to go from a small-scale rotor to a full-scale rotor, however is not enough is different helicopter categories have to be considered.

A common reference for outflow velocities is the normalized hover induced velocity, λ_i, expressed as $\lambda_i = \sqrt{C_T}/2$. For full-scale rotors, the C_T^{fs} can be obtained by the data, considering the aircraft in hover flight, and different disk loading (DL) categories. DL, V_{tip}, and R are specified in table 6.2, while $\rho_{air} = 1.225 kg/m^3$ was assumed. The thrust coefficient is presented following

[1] Part of this chapter has been published in Rovere Federico, Barakos George, and Steijl Rene. "Safety analysis of rotors in ground effect." Aerospace Science and Technology (2022): 107655

the UK convention, as described in the nomenclature. It is possible, taking into account the effect of the different thrust coefficients, to scale the velocities using the ratio of hover induced velocity between full-scale and model-scale cases. In this way, it is possible to consider different categories of rotorcraft related to disk loading. The helicopter categories used in this work are listed in 6.2. It is important to say that the category itself does not define solely the weight of the helicopter. Operational characteristic such as rotor diameter, max take off weight (MTOW) disk loading and other parameters define the helicopter category [158].

The scaling factor obtained is $\sqrt{\frac{C_T^{fs}}{C_T^{ms}}}\frac{V_{tip}^{fs}}{V_{tip}^{ss}}$. This scaling factor takes into account the different sizes of the model and full-scale rotors, and the different disk loading configurations. Multiplying the three velocity components by the scaling factor, approximates the outflow velocities generated by a full-scale rotor.

h/R	C_T^{ss}
1.5	0.028
1.0	0.03
0.5	0.035

Table 6.1: Model scale thrust coefficients for different h/R.

Category	Disk Loading (N/m^2)	V_{tip} (m/s)	R [m]	C_T^{fs}
Low	280	220	5.5	0.0094
Medium	420	220	8.1	0.0141
High	560	220	8.1	0.0189

Table 6.2: Helicopter technical data [159] for three categories used in this work.

It is important to say that even this scaling does not fully represent the complexity of the phenomena involved. This approach does not take into account the different Reynolds numbers of the full and model-scale rotors, which may change the outflow and its development in time. Furthermore, using this approach, the full scale thrust coefficients do not take into account an IGE effect, because the full scale thrust coefficient is the same for all three rotor heights above the ground. Considering Cheeseman's work [160], it is possible to obtain C_T IGE from C_T OGE and the distance from the ground. As described in Chapter 5, the thrust coefficient may change dramatically with the rotor distance from the ground. Table 6.3 shows the comparisons of the

normalized hover induced velocity ($\lambda_i = \sqrt{C_T}/2$) with and without Cheeseman's factor [160] for the different rotor heights, and the rotor configuration with higher disk loading.

h/R	λ_i without Cheeseman's factor	λ_i with Cheeseman's factor
1.5	0.068	0.070
1.0	0.068	0.071
0.5	0.068	0.073

Table 6.3: Normalized hover induced velocity with and without Cheeseman factor [160]

For the h/R taken into account, the Cheeseman's factor is stronger at h/R=0.5. In this case, the ratio between normalized, hover induced velocity with and without Cheeseman's factor is around 1.068. Due to the limited influence ratio has on the scaling factor, its effect is not taken into account in this work.

6.2 Forces on human body - Hovering rotor IGE

PAXman results

Forces over PAXman bodies have been initially computed for the hovering cases. Figures 6.1 (a), 6.2 (a) and 6.3 (a) show the total forces acting on a scaled human body by the outflow at different distances from rotor for the three different rotor configurations. As mentioned before, the forces have been computed by scaling the velocities to obtain more realistic values, and compare them with the safety thresholds suggested for civilian and military operations.

The PAXman model uses two threshold limits. After the caution limit, ground personnel may experience performance limitations in their work. When forces are over the danger level, the ground personnel risks injury. People may fall, or may have difficulty walking or may even be hit by objects uplifted by the outflow. [18]

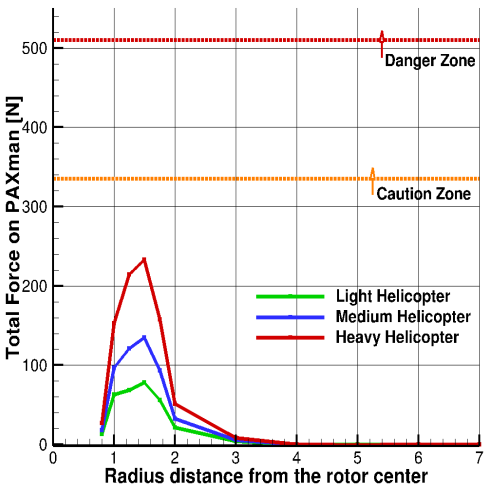
The forces for all three rotor configurations reach their peaks at a radial distance between 1.25R and 1.5R, but they weaken at further radial stations. This result is due to the outflow distribution at different radial stations, with the higher velocities occurring around 1.5R. At all three rotor distances from the ground, the strongest force is produced by the aircraft with the highest C_T.

In general, all results are lower with respect to the caution limit of the PAXman model. When the rotor is operating at h/R=1.5 and h/R=1, the forces reach similar peaks, around 270 N, for the helicopter with highest disk loading, slightly stronger than the configuration at h/R=0.5, for the same aircraft category. In all cases, further than 2R, the forces drop quickly. This is due to the outflow velocities that drop away from the rotor. After 3R the forces are almost negligible, based on the PAXman criteria, for all three scaling factors, and for all three rotor distances from the ground.

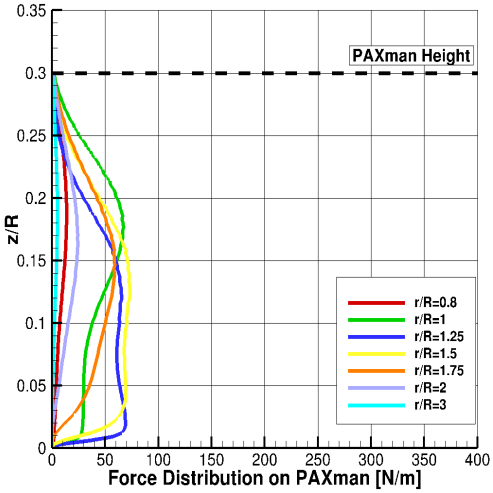
Figures 6.1 (b), (c) and (d) show the distributions of the forces over the PAXman height at different radial stations for the three helicopters interested. For all cases, near the rotor, the force is the strongest at a height corresponding to the PAXman chest. This result is due to the shape of the outwash distribution, and the larger blocking area at chest level. This result is in accordance with other experiments. In the work of Silva et al. [19] the force peaks were around the middle of the human body, but these experiments were full-scale.

Figure 6.1 (b) shows the results for the low DL category aircraft. Here, the PAXman height is higher with respect to the other two cases. This is due to the smaller size of the rotor blade of the lowest DL helicopter. This leads to a difference in the force distribution for the low DL category helicopter with respect to the other DL cases. For all three rotor distances from the ground, the results for the lowest DL helicopter category show forces distribution over the PAXman body, with minor forces over the chest area. On the other hand, the medium and high DL cases show stronger forces on this part of the body.

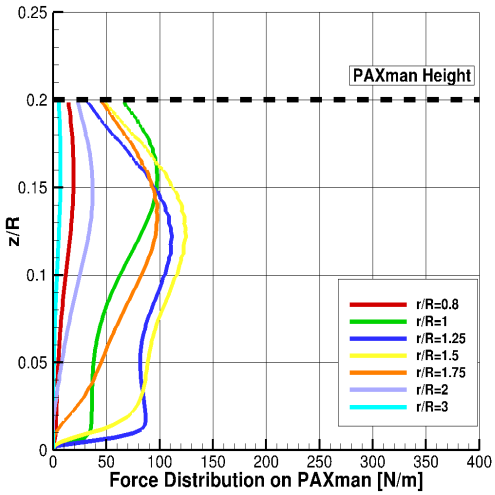
In general, the forces for the lower DL category helicopter, at every radial distance, are smaller then for the higher DL cases. This is in agreement with what is shown in figure 6.1 (a), where the lowest DL aircraft produces weaker forces in comparison to the medium and the high categories. PAXman force distributions for h/R=1 and h/R=1.5 are shown in figures 6.2 (b), (c), (d) and 6.3 (b), (c), (d). Similar to h/R=0.5, the most exposed part of PAXman body is the chest, with the strongest forces seen at a radial distance of r/R=1.5. In general, higher DL category helicopter outflows are stronger with respect to the low DL cases.



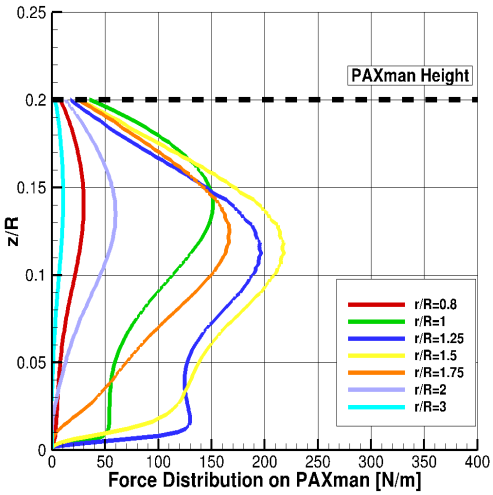
(a) Total radial force F_{PAXman} for the three helicopter



(b) Radial force distribution f_{PAXman} for low DL helicopter



(c) Radial force distribution f_{PAXman} for medium DL helicopter

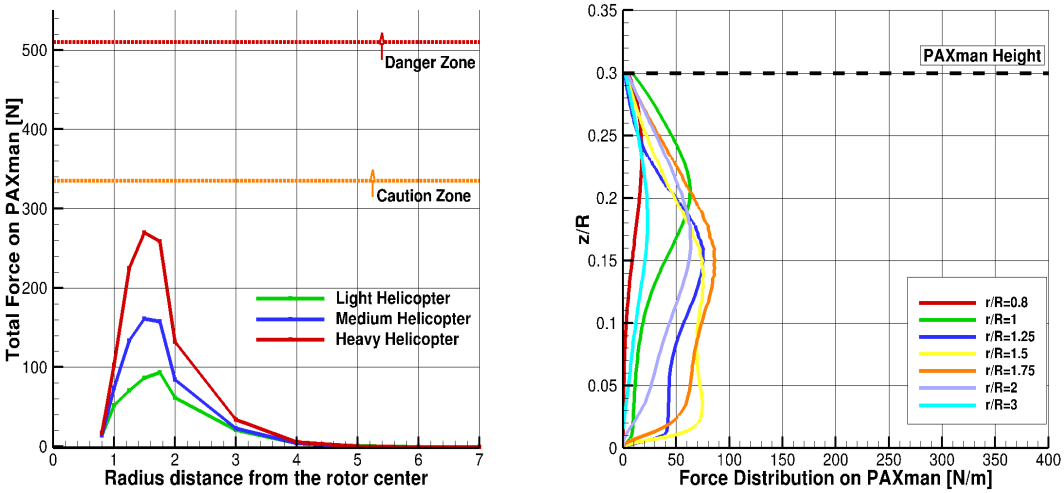


(d) Radial force distribution f_{PAXman} for high DL helicopter

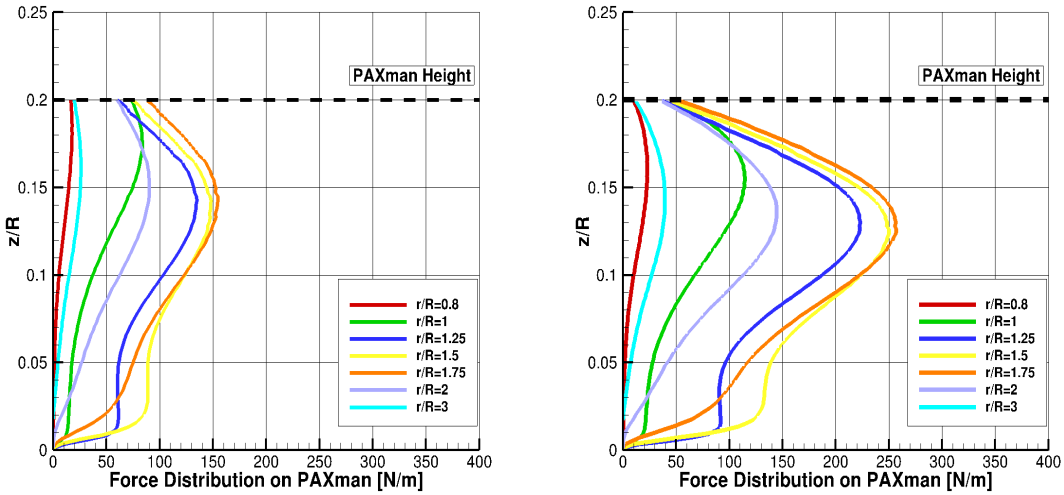
Figure 6.1: PAXman model forces calculated using the employed micro-rotor, scaled to full scale. The micro rotor rotor was operating at $\theta_{75} = 12deg$, $Re_{tip} = 35000$, $M_{tip} = 0.08$, $h/R = 0.5$ and $C_T = 0.035$.

6.3 Forces on human body - Forward flight IGE

In the previous section, for the hovering rotor, the forces have been computed by scaling the velocities to full scale aircraft, to obtain more realistic values and compare them with the safety thresholds suggested for civilian and military operations.



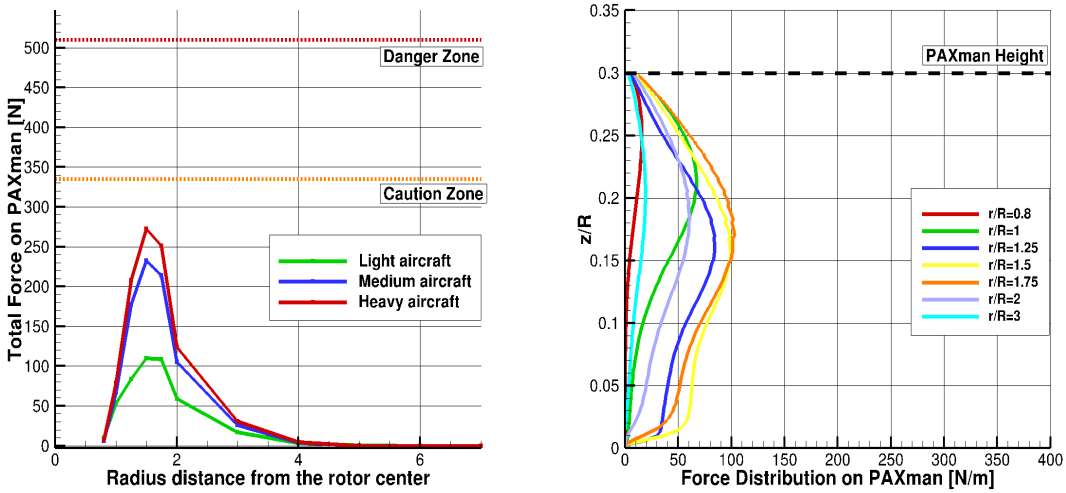
(a) Total radial force F_{PAXman} for the three helicopter (b) Radial force distribution f_{PAXman} for low DL helicopter



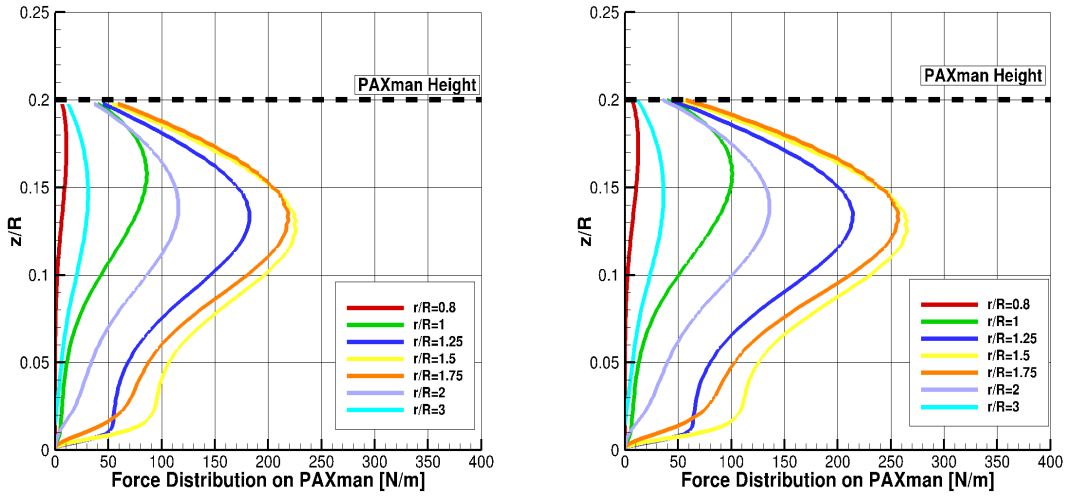
(c) Radial force distribution f_{PAXman} for medium DL helicopter (d) Radial force distribution f_{PAXman} for high DL helicopter

Figure 6.2: PAXman model forces calculated using the employed micro-rotor, scaled to full scale. The micro rotor rotor was operating at $\theta_{75} = 12deg$, $Re_{tip} = 35000$, $M_{tip} = 0.08$, $h/R = 1$ and $C_T = 0.03$.

For forward flying rotors, this was not necessary, due to the fact that simulations were for full-scale rotors. Figure 6.4 shows results for a forward flying rotor at 10kts and $C_T = 0.0125$. In this case, y/R is the lateral distance from the center line of the taxiing path, while x/R is in the downstream direction. A station is in front of the rotor if $x/R < 0$ and behind the rotor if $x/R > 0$.



(a) Total radial force F_{PAXman} for the three helicopter (b) Radial force distribution f_{PAXman} for low DL helicopter



(c) Radial force distribution f_{PAXman} for medium DL helicopter (d) Radial force distribution f_{PAXman} for high DL helicopter

Figure 6.3: PAXman model forces calculated using the employed micro-rotor, scaled to full scale. The micro rotor rotor was operating at $\theta_{75} = 12deg$, $Re_{tip} = 35000$, $M_{tip} = 0.08$, $h/R = 1.5$ and $C_T = 0.028$.

Results show lower peak values (less than 80 N) with respect to the hovering rotors. This may be due to the different heights above the ground of the two rotors (the hovering rotor was operating below h/R=1.5, while the forward flight rotor is about h/R=1.6). However, both cases, suggest that the stronger forces are between of 1R and 2R, and drop after 3 R. Considering a higher

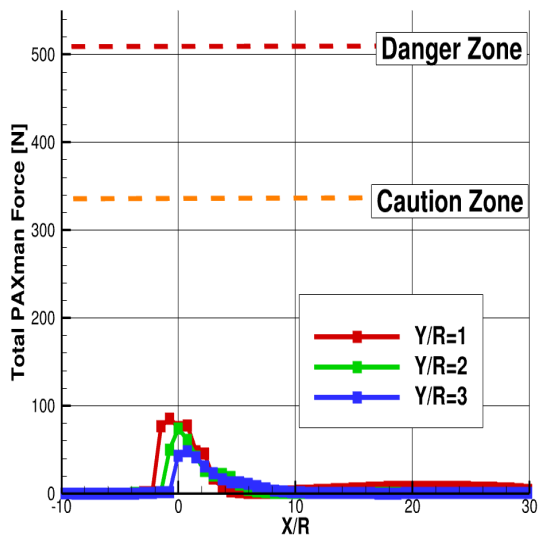
thrust coefficient ($C_T = 0.0175$), the results show higher peaks. In this case, the forces reach peaks of 100N for a span wise direction of 1R. As already seen in previous works [108], helicopters with high disk loading generate more dangerous scenarios.

Increasing the taxiing speed to 40 kts, the situation changes. The force peaks drop to 10N, reaching harmless levels, as shown in figure 6.6 (a). Results suggest that, in terms of total PAXman forces, when the rotor is hovering, the forces are higher and more dangerous with respect to the forward flight cases, and if the taxiing speed increases the situation becomes safer. This is due to the presence of the hovering rotor wake close to the ground. As described in [144], when the rotor is flying at 40 kts, the wake is concentrated behind the fuselage, and it is above the ground. However, lowering the taxiing speed, the wake shape changes, spreading all around the fuselage and moving closer to the ground [144]. In general, for forward flight cases, the most dangerous downstream station is behind the rotor, at 1 or 2 R in the x/R direction.

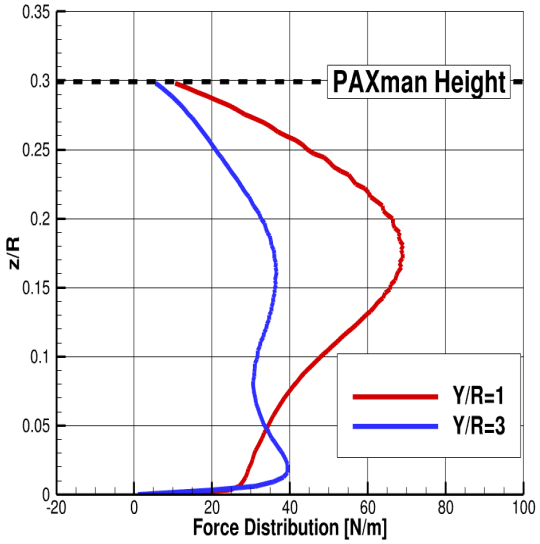
In previous works [108], similar analyses were conducted, and results showed that outflows were stronger when the rotor was operating closer to the ground. Figure 6.4 (b) shows the distributions of the forces over the PAXman height at different radial stations for the rotor in forward flight. The forces are stronger at a specific PAXman height (corresponding to the PAXman chest). This result is due to the shape of the outwash distribution and the larger blocking area at the chest level. In general, the force distributions weaken, with increasing the radial distance from the aircraft. This is in agreement with what is shown in figures 6.4 (a), 6.5 (a) and 6.6 (a). Similar results, are shown in figures 6.5 (b) and 6.6 (b) where the force distribution for the forward flying rotors is presented for the taxiing speeds. Once more, the chest area is the most stressed part of the PAXman model, however, the force values are low.

6.3.1 Full scale - Small scale Validation

In this subsection, PAXman results are compared to validate the scaling method between full scale and small scale. Test case employed is the forward flight test case. The grid employed is without the fuselage, already described in Chapter 3. The full scale operational conditions have been listed in table 6.4 for small and full scale operational conditions. Small scale operational conditions

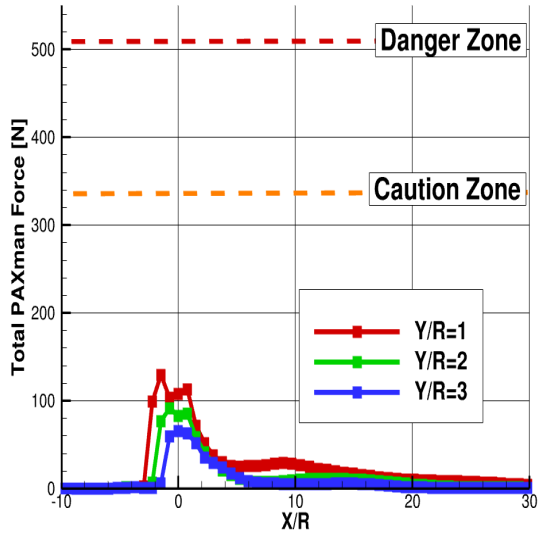


(a) Total PAXman forced flight

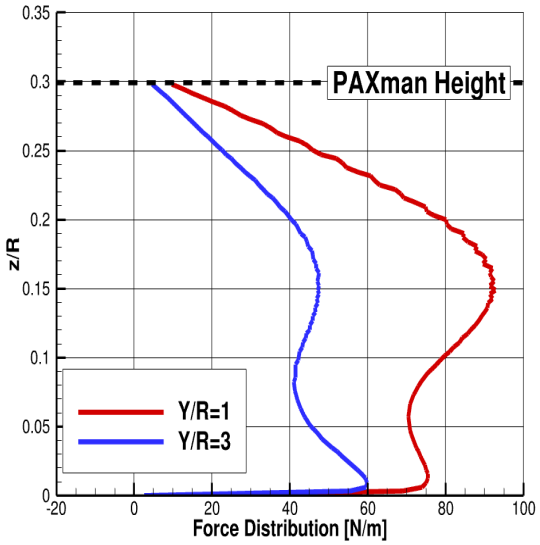


(b) PAXman force distribution

Figure 6.4: PAXman model forces calculated using the forward flight rotor. The rotor was operating at $V_\infty = 10kts$ (μ=0.023) and $C_T = 0.0125$



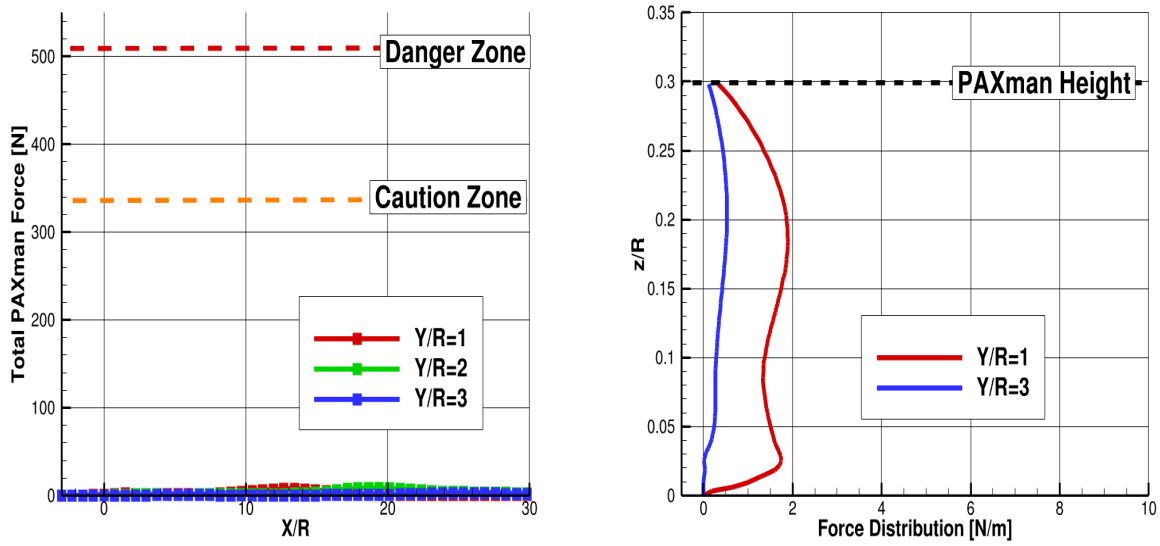
(a) Total PAXman forced flight



(b) PAXman force distribution

Figure 6.5: PAXman model forces calculated using the forward flight rotor. The rotor was operating at $V_\infty = 10kts$ (μ=0.023) and $C_T = 0.0175$

(V_{tip} and chord size) have been taken starting from the work of Curtiss et al. [32], which tested an isolated rotor in forward flight. PAXman results are compared in terms of total force and force



(a) Total PAXman forced flight

(b) PAXman force distribution

Figure 6.6: PAXman model forces calculated using the forward flight rotor. The rotor was operating at $V_\infty = 40kts$ (μ=0.0932) and $C_T = 0.0125$

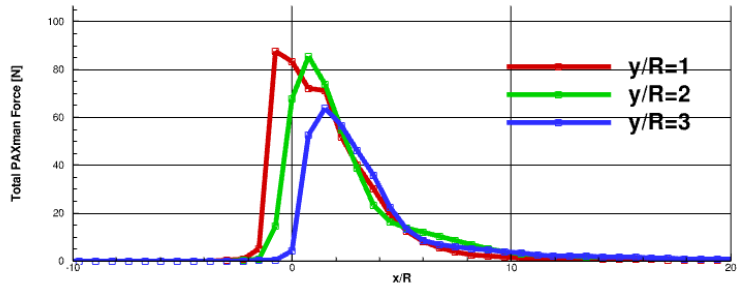
distribution.

	Model scale ms [26]	Full scale fs [159]
V_{tip} $[m/s]$	57	220
V_{infty} $[m/s]$	1.25	5.14
μ	0.022	0.022
Re	5308	618000
N blades	4	4
R [m]	0.8	5.5
c [m]	0.06	0.41
C_T	0.0125	0.0125

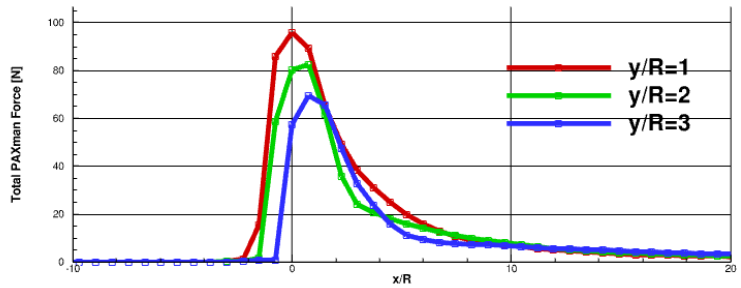
Table 6.4: Full and Model Scale rotors conditions.

Results in terms of total PAXman forces are shown in figure 6.7. Total PAXman forces are computed behind the rotor, at three distances 1R,2R and 3R. For both cases, the total force peaks are between 0 and 10 R. With peaks around 60 and 100 N. In general, the small scale rotor shows slightly lower forces, compared to the native full scale case. However, these differences are few, and for both cases the PAXman total forces are below the dangerous levels described in figure 6.4. In figure 6.8, the PAXman force distributions are shown for x/R=1 and x/R=2, where the PAXman total forces reach their peaks. Like in the previous case, the PAXman force distribution for native

full scale is slightly higher, however the distribution of the two methods is closer, suggesting again that the chest is the most influenced part of the human body. This result is a combination of the PAXman model (the chest area is wider, leading to a higher blocking area) and the outflow shape. Higher above the rotor, y/R=2.0 the results show a different behavior. Here, the most influenced part of the PAXman is the lower part, this is due to a different outflow shape, which has the peak at a very low altitude position. However, both models agree on the obtained results, showing a close PAXman force distribution over the PAXman body.



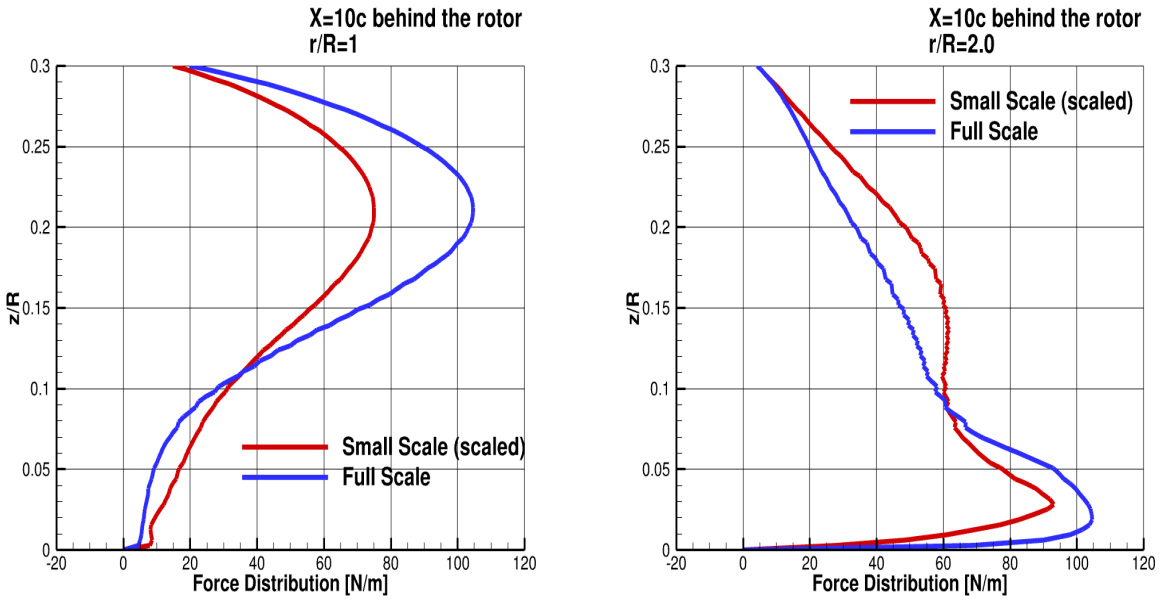
(a) Small Scale



(b) Full scale

Figure 6.7: Total PAXman model forces calculated for small to full scale, the rotor was operating in forward flight at μ=0.023, $C_T = 0.0125$.

To summarize, PAXman forces have been obtained by studying the outflow velocity of different helicopters operating in different scenarios. In general, rotors taken into account in this thesis do not seem to be a danger for ground personnel in terms of forces. In all cases considered, the obtained forces are under the caution limit defined by the PAXman model. However, a more dangerous zone can be defined between 1R and 3R. In this area, outflow forces reach their peaks before decreasing again. Hovering helicopters reach higher peaks for total forces on the human body with respect to taxiing rotors, especially when the disk loading is increasing. On the



(a) r/R=1.0 (b) r/R=2.0

Figure 6.8: PAXman model forces distribution calculated for small to full scale, the rotor was operating in forward flight at μ=0.023, $C_T = 0.0125$.

other hand, the forces generated by rotor in forward flight are weaker, and reduce with the forward speed. This is due to the shape of the wake in forward flight, that it is at a higher position from the ground compared to the PAXman height. In this way, the interaction between the PAXman and the taxing rotor wake is minimal. Considering the force distributions of forces over the PAXman human model, results suggest that the chest is the most stressed part. In this part of the human body, the blockage area is larger and in combination with the outflow shape, it creates larger forces than elsewhere on the PAXman. Furthermore, considering the obtained results, the 3 rotor diameter limit for wake encounters defined by FAA can be adopted also to define a safe area for ground personnel in terms of forces produced over the human body.

Chapter 7

Lagrangian particle tracking[1]

This chapter describes the particle tracking tool developed during the present work, and the results obtained to predict brownout clouds for hovering and forward flying rotors.

7.1 Algorithm description

The flow chart of the algorithm is presented in figure 7.1.

First, the method reads the CFD mesh and an input file, which contains information about the particle properties (density, radius, drag coefficient), starting positions, flowfield information and gravity. Then, it reads the CFD flowfield files and releases particles in the flowfield. When a new particle is released in the flowfield, it is assigned to a processor. Processors do not exchange information about their particles and, they track their particles from the first timestep to the end of the simulation. After generating particles, every processor searches and updates the position of every particle assigned to it. Finally, every processor writes an output file containing all particle positions at the timestep. For a new timestep, the cycle restarts reading a new flowfield file.

[1] Part of this chapter has been published in Rovere Federico, Barakos George, and Steijl Rene. "Safety analysis of rotors in ground effect." Aerospace Science and Technology (2022): 107655

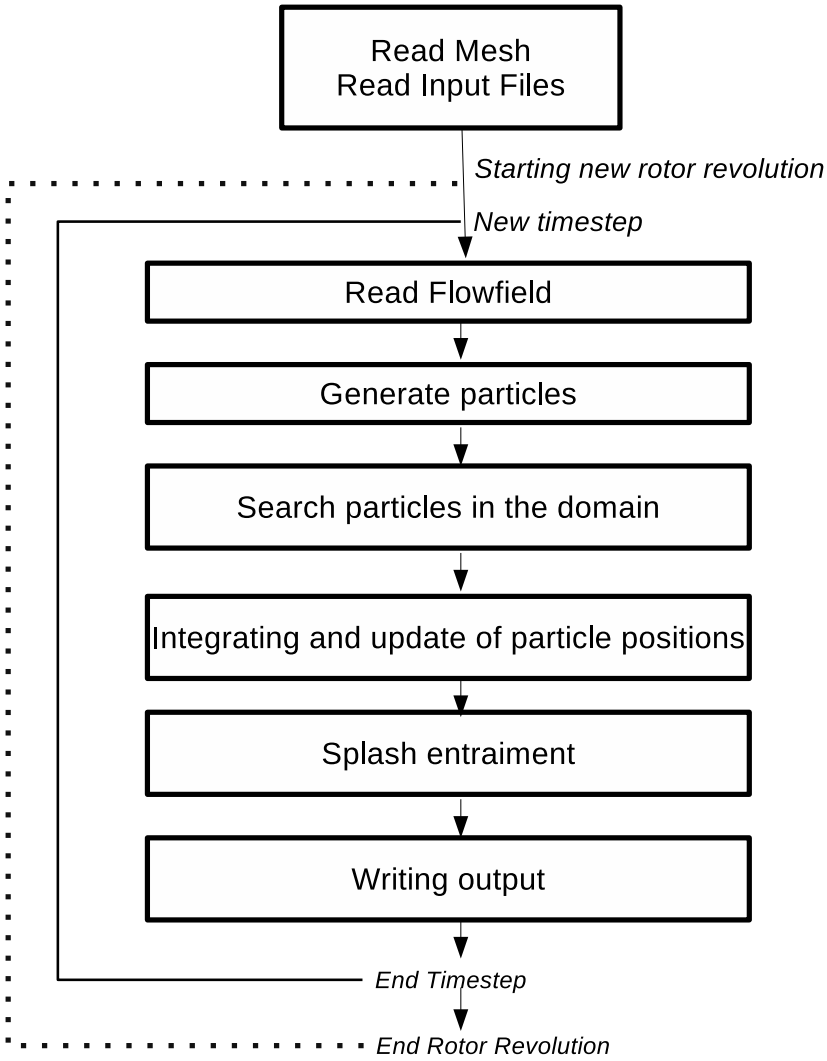


Figure 7.1: Particle tracking flow chart of the particle track tool.

7.2 Search algorithms

Once the position of every particle has been computed or initially, defined by the user, it is necessary to know the flowfield variables in the space occupied by the particle. To identify the position of the particle inside the domain three pieces of information are needed, the block number and the three indices i,j and k that define a single cell of the mesh. Two algorithms have been developed for this purpose. For both, the aim is to find the nearest cell centre to each particle. The first method, more computationally expensive, is used when there is no information about the previous position of the particles in the domain (considering the block number and the three indices i,j,k that define a single cell), in other words, when there is no initial information. The second, more efficient method, uses the previous particle position and searches in the neighboring cells only. Both methods are described in the following subsections.

7.2.1 Brute force search method

The brute force search method is expensive in terms of CPU time, and it is used only when strictly necessary. Using this method, a particle is searched over all the domain. The distance between the particle's position and cell centre is computed for every cell in the domain. Finally, the particle is assigned to the nearest cell. This method does not use an initial cue, it is based on the idea that there is no information about the position of the particles in the domain. This method is used at the beginning of the simulation, when the user defines the particle's position (x,y,z), but not its locations in the domain (number of block, and cell indices). This method is also used, when the particle changes grid hierarchy for chimera grids. It is also used when the distance between the particle position and the closest cell center is larger than the distance tolerance defined in the input file. In this case the particle is searched again using this method inside the original block, and its neighbors.

7.2.2 Fast search method

When the particle has already been localized once, the previous domain position (number of block, and cell indices that identify the nearest cell) is known, and this information can be used as an initial cue for updating the particle's position. To avoid the high computational cost, the search area is limited to the neighbor cells of the particle's position at time t_n. The search method is described in figure 7.2. The search area (the darkest area in the figure) is within by the maximum distance travelled by the particle during the Δt between time instances t_n and t_{n+1}. The area is delimited by the maximum range, r_{max}, computed as: $r_{max} = \mathbf{u}_p \Delta t$. A cell, i, is considered in the search area if $r_i < r_{max}$, where r_i is the distance between the cell center x_i, and the position of the particle at t_n, $r_i = ||\mathbf{x}_p^n - \mathbf{x}_i||$. The particle is then searched in the delimited area, comparing the distance between the cell center and the particle position at t_{n+1}. Finally, the particle is assigned to the nearest cell. Once a particle is found, interpolation is done to compute the flowfield properties at the locations of the particle. All cells around the particles are taken into account. The influence of these cells is weighted with respect to their distance. In all simulations, the grids were multi-block structured with hexahedal cells, so the cells surrounding particles plus the one that contains the particle, are 27 in total, except for cells near domain boundaries.

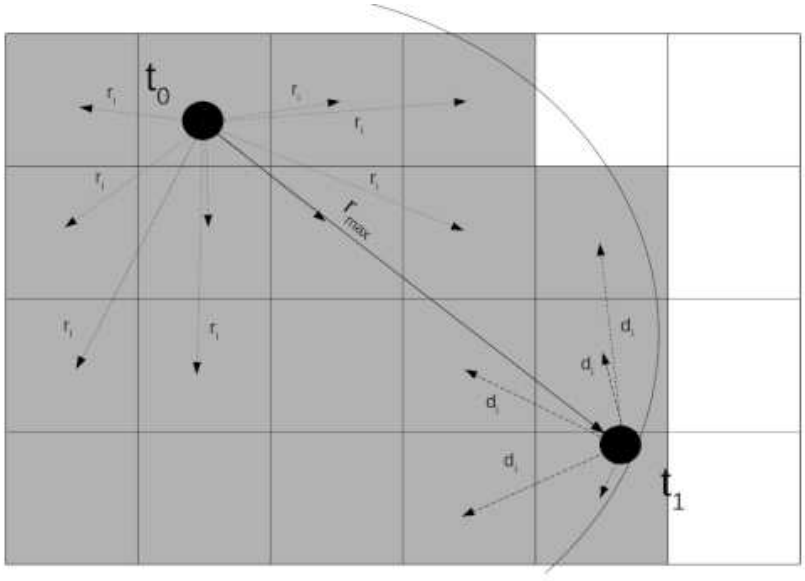


Figure 7.2: Search algorithm for particle tracking.

The employed equations are:

$$u_{flow}(x_p^{n+1}) = \sum_{i=0}^{27} u_{flow}(x_i) q_i, \tag{7.1a}$$

$$d_i = ||\mathbf{x}_i - \mathbf{x}_p^{n+1}||, \tag{7.1b}$$

$$q_i = \frac{\frac{1}{d_i}}{\sum_{i=0}^{27} \frac{1}{d_i}}. \tag{7.1c}$$

where $u_{flow}(x_p^{n+1})$ is the flowfield velocity as the position occupied by the particle and $u_{flow}(x_i)$ the flow velocity as the center of the i-th cell (this is done for all three velocity component, and flow density). $\mathbf{x}_i$ is the position of the i-cell center, while $\mathbf{x}_p^{n+1}$ is the particle position at timestep n+1. Where a particle is near the domain boundaries, the neighboring cells are fewer. In this case, the interpolation is using the "halo cells" of the mesh. Two layers of halo cells are added at the boundary of each block, and these contain the information of the first two layers of cells inside the domain of the neighbor block. This way, it is always possible to interpolate values from the 27 cells needed. When the code runs in parallel every particle is associated with a single processor, which searches only the particle assigned to it.

7.3 Validation test case - EH-60L Wong and Tanner experiments

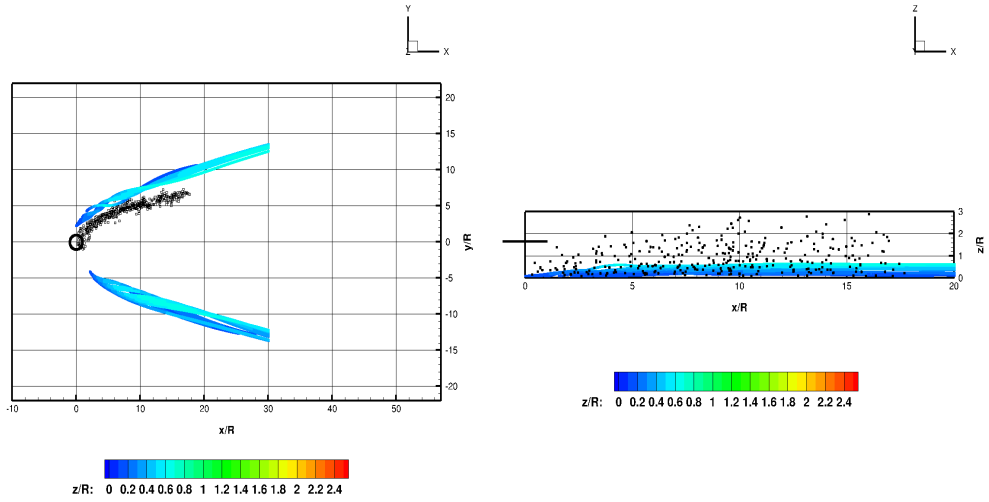
Lagrangian particle tracking has been validated using experimental results obtained by [12]. Experiments have been performed using photogrammetry to obtain qualitative results of the brownout cloud development of a EH-60L taxiing. The aircraft was operating at around 15m (around 1.8R) altitude and 12 m/s (approx 25 kts). A flowfield simulation has been performed for the same mesh, described previously, with a higher advance ratio ($\mu = 0.058$) and for a $C_T = 0.017$. The ground has been seeded considering a density of particles on the ground around 8 particles per m^2, with particles covering all areas where the friction threshold model is satisfied.

The experiment results show the development of the cloud at different time steps. Compared with the unsteady results with steady state simulations, experimental results have been collected in a single cloud shape. The aircraft position is moved to the origin of the domain, while the relative position between the aircraft position and the cloud points has been conserved. In these simulations, particles have mass properties, as listed in table 7.1.

$\rho_p\ (kg/m^3)$	$d_p\ (\mu m)$	C_D	B (kg/m^2)	$u_t^*\ (m/s)$
2650	9	1.048	0.03	0.58

Table 7.1: Properties of particles used in this work.

Results compared with experiments are presented in figure 7.3.



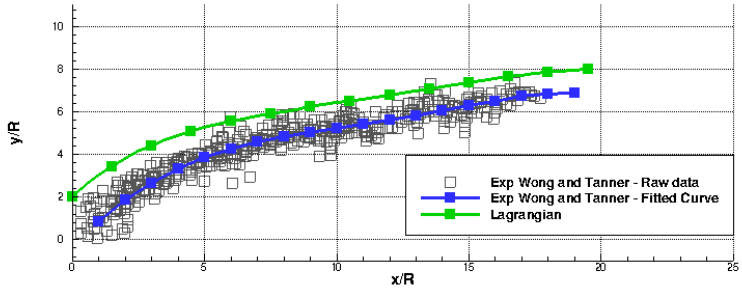
(a) Top View (b) Side View

Figure 7.3: Lagrangian particle tracking and Eulerian model validation. Experiments (black dots) [12] vs present study results (dots/isosurfaces with contours). The rotor was operating at $\mu = 0.058$, $C_T = 0.017$. The employed particles properties are listed in table 7.1.

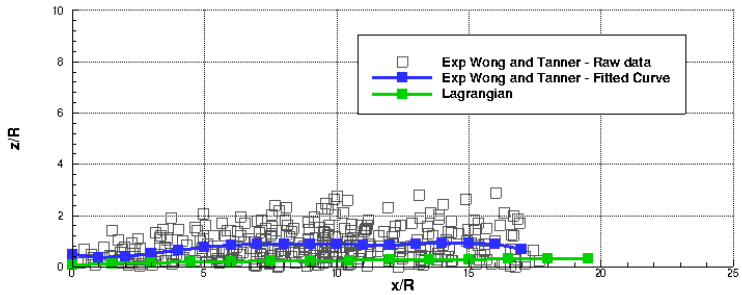
The results reveal how the Lagrangian particle tracking gives an approximation to the brownout cloud, see figure 7.3 (A). Computational and experimental results follow the wake shape, forming an almost parabolic front. A small offset can be noticed between experimental and computational results, however, this was due to the comparison between a steady state simulation work and unsteady experimental results, and the necessary transformations. The computational results,

underestimate the maximum height of the cloud, showing a maximum cloud altitude around 1R, while the maximum experimental cloud reaches almost 1.5R (figure 7.3 (B)). This could be due to the idealized numerical work vs the full scale, which measurements that also contains errors.

Figure 7.4, quantifies the differences between the experimental results, and the numerical simulations. Considering the difficulties on comparing different clouds in terms of shape and size, the clouds have been approximated with a polynomial curve obtained by the least squares method. As can be seen in figure 7.4 (A) the distance between the experiments and the numerical methods is about 1R. It is possible to notice, that this error is almost constant for the range of X/R values considered, and the shape of experimental curves is like to the numerical ones. From a side view (figure 7.4 (B)), the scatter in the experimental results is more evident, and the comparison between experimental and numerical method is more difficult. However, in both views, it is possible to notice that the differences between numerical model and experiments, are within the data scatter of experimental results. In the side view, the distance between the experiments is around 0.5R.



(a) Top View



(b) Side View

Figure 7.4: Lagrangian model validation. Experiments (gray dots) [12] vs present study results. The rotor was operating at $\mu = 0.058$, $C_T = 0.017$. Particle properties are listed in table 7.1.

7.4 Isolated hovering rotor IGE

Particle tracking has been used for different test cases, to understand how the brownout cloud develops and which factors are the more influencing. Initially, Lagrangian particle tracking has been used for the hovering IGE rotor described by Lee et al. [26], and described previously, in Chapter 4. As described for PAXman forces, this test case is not suitable for safety particle tracking, so the scaling factors have been adopted for the following results, using also the values already used for PAXman results and listed in the table 6.2. Furthermore, scaling factors allowed to study different aircraft categories, and define which may be more dangerous in terms of brownout clouds. Initially, an uplift study has been carried out, finding the positions where the particles are more liked to be uplifted, then particles have been released on the ground and tracked in the flowfield for 100 rotor revolutions.

7.4.1 Uplift criteria results

Figure 7.5 shows results for the threshold model. Using the computed shear stress on the ground of the domain, it was possible to calculate directly the friction velocity and using, the Bagnold threshold model, the particle uplift locations. In general, the results show that the peak of the uplift ratio u_t/u_t^* is in the proximity of 1 R distance from the rotor hub, and the area of uplift particles for the cases considered is at least between 1R and 2R. This is due to the presence of the starting vortex in the proximity of the ground shown in figure 5.13.

Higher DL helicopters produce stronger outflow, which may extend the uplift area up to 3R for h/R=0.5. The rotor distance from the ground has a key role in defining the area where particles may be uplifted. Results show that when the rotor operates at h/R=0.5, the particles can be uplifted up to a 3 rotor radius distance for high DL helicopter cases, while for higher rotor altitude the uplift area can be delimited by 2R. Furthermore, considering the low DL category (left column), in the case of h/R=0.5 and h/R=1, uplift ratio drops slowly, with values of uplift ratio around 0.5 after 3R, while for h/R=1.5 after the maximum the uplift drops quickly beyond 2R, reaching low uplift values at 3R. This leads to the result that the distance of the rotor from the ground has an

appreciable influence on the particle uplift.

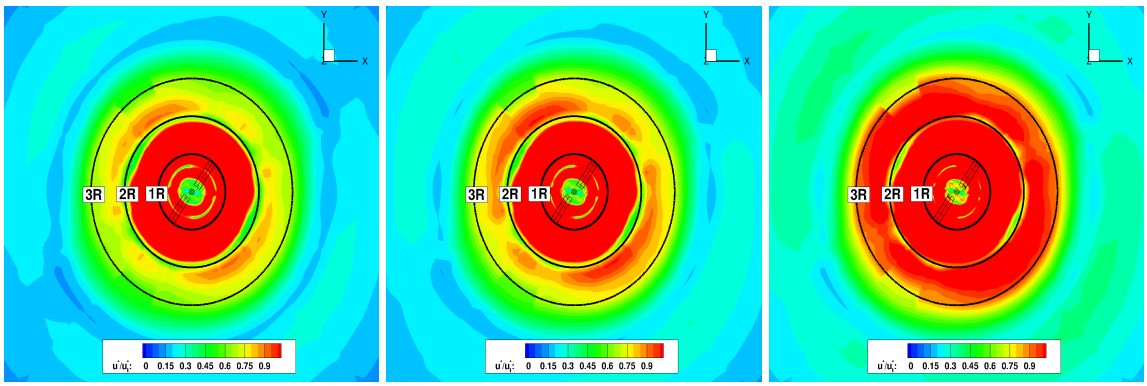
In the case of h/R=0.5, the uplift area for the lower DL helicopter is 2R, and grows to 3R for higher DL cases. On the other hand, for a rotor at h/R=1.5, the uplift area increment with higher DL scaling is not so effective. In general, the rotor height has a strong effect on particles uplift locations the uplift area is bigger with the rotor closer to the ground, and as for PAXman results, the higher DL helicopter produces the most dangerous scenarios, with bigger uplift areas.

Using this information it was possible to seed the ground with the particles that are more likely to be uplifted by the flowfield. Figure 7.6 shows the evolution of the particles with time for three helicopter categories taken into account, with the rotor at three different heights above the ground. Different seeding positions have been defined between 1R and 1.5R in proximity to the ground level. The bombardment effect has not been taken into account here. The simulations involve about 100 rotor revolutions, for a total hover time of 30 seconds. The flowfield generated by the model rotor was not able to lift particles.

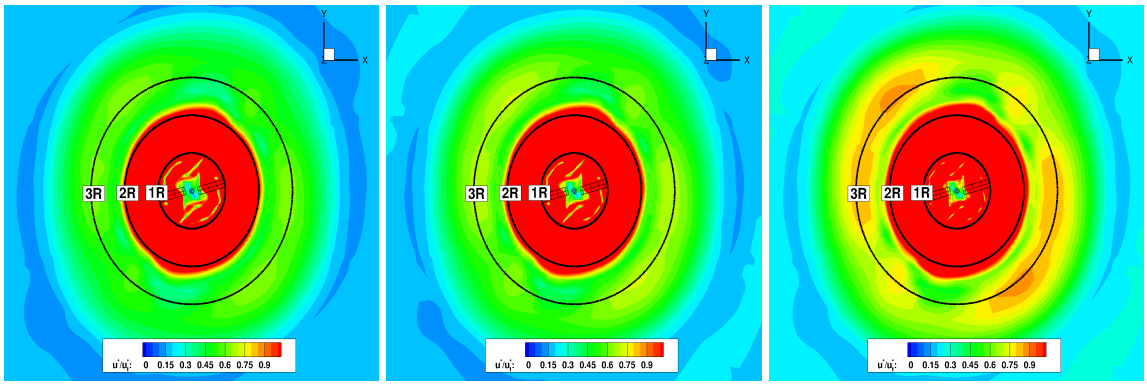
7.4.2 Brownout clouds

In figure 7.6, the tracking results are shown. Seeding points have been released in proximity to the ground, between 1 and 2R, as a result of the uplift model. A new particle is released in the flowfield from the ground, every 5deg of rotor azimuth. This is due to the 5deg timestep that separates every single flowfield file. The full simulation involves 100 revolutions (about 30 seconds for a full scale rotorcraft). Due to the high computational cost of the CFD simulation, the last revolution performed has been considered periodic, and has been repeated for the full duration of the particle tracking.

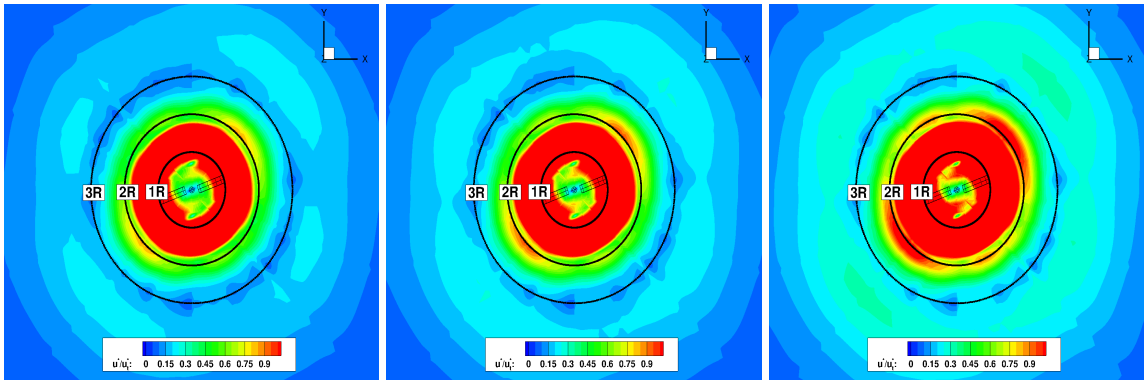
When the rotor is operating at h/R=1, the particles are uplifted by the flowfield, and move away from the rotor, in the radial direction. Particles are driven by the outflow that pushes them away from the rotor. However, depending on the strength of the outflow they reach different positions. Higher DL helicopters have a stronger outflow, and in that case, particles can reach



(a) h/R=0.5, Low DL helicopter (b) h/R=0.5, Medium DL helicopter (c) h/R=0.5, High DL helicopter



(d) h/R=1, Low DL helicopter (e) h/R=1, Medium DL helicopter (f) h/R=1, High DL helicopter



(g) h/R=1.5, Low DL helicopter (h) h/R=1.5, Medium DL helicopter (i) h/R=1.5, High DL helicopter

Figure 7.5: Uplift criteria results for rotors at different heights above the ground, and different scaling factors. The rotor was operating $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.08$.

a maximum radial distance of 8.5R, and a maximum height above the ground of 1R. Once the particles reach their maximum altitude, they fall again to the ground. Lower DL helicopter cases show lower values for maximum h/R and r/R. However, particles go further than the 3D separation

criteria used for wake encounters [161, 162]. In this case, the employed scaling has a minimum effect of particle paths increasing the maximum distance reached of about 1R between low and high DL helicopters, with a very limited influence on the maximum height reached by particles.

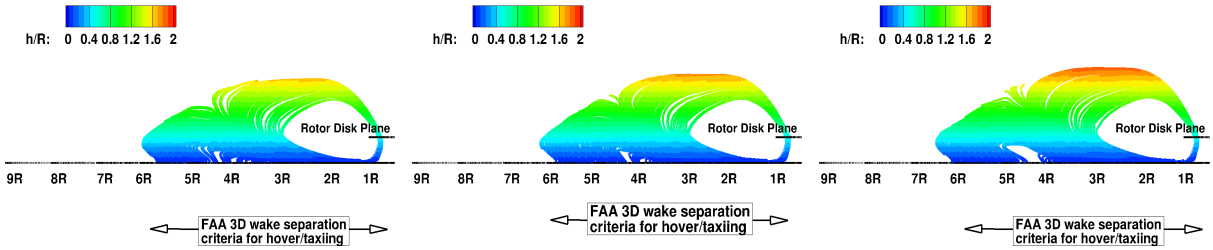
A similar path is followed in the case of h/R=1.5, where the particles reach a maximum height of h/R=1.5 for the high DL aircraft case. As for the previous case, the maximum height is reached at r/R=6. However, the flowfield seems weaker with respect to the previous case, and particles reach a maximum radial distance of 8R, in the most dangerous scenario.

On the other hand, when the helicopter is operating at h/R=0.5 the particles show different behavior. Initially, they are uplifted from the ground, and then two main branches appear. Some particles are reingested by the rotor, reaching the higher distance from the ground. These particles can be dangerous for the crew and the aircraft. The rest of the particles keep following the radial direction far from the rotor, and fall again on the ground at distance around 6.5R for the high DL scale helicopter.

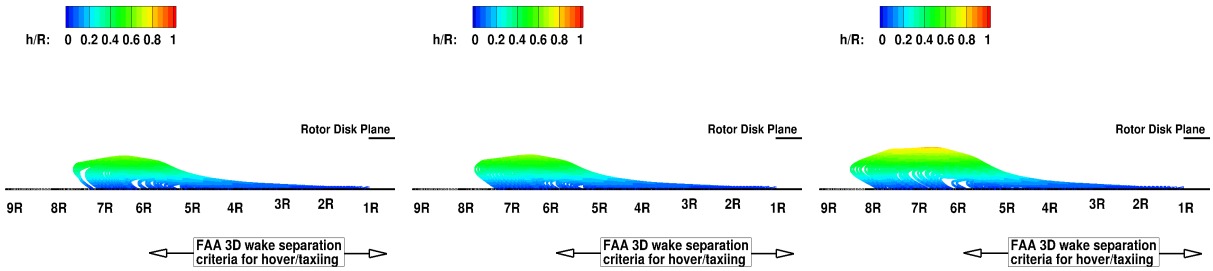
Like in the case of uplift results, the rotor position above the ground has a strong influence on particle path, defining the maximum height and distances reached by the particles, and whether particles are reingested or not. Scaling factors also have a role in defining the ”safe zone” for operations. Helicopters with higher disk loading push particles further, increasing the radial cloud dimension. In general, as expected, the higher DL helicopter shows the most dangerous scenario in all rotor configurations. When the helicopter is operating at h/R=1, the particles do not have a strong effect on pilot visibility. Up to a distance of 3-4 R, the particles are near the ground, and the pilot may see clearly the ground, avoiding ground obstacles and structures. On the other hand, particles may have an effect on the visibility of other helicopter operating nearby, entering the operational area. A similar scenario is presented at h/R=1.5, even if it is slightly worse than the previous. Here the particles stay close to the ground up to 2/3R, and reach higher positions. This way, the pilot visibility is slightly reduced with respect to the h/R=1 case. However, it may still be possible to see the ground in the rotor closer area.

Finally, the most dangerous scenario, is with the rotor was at h/R=0.5. In this case, the pilot’s

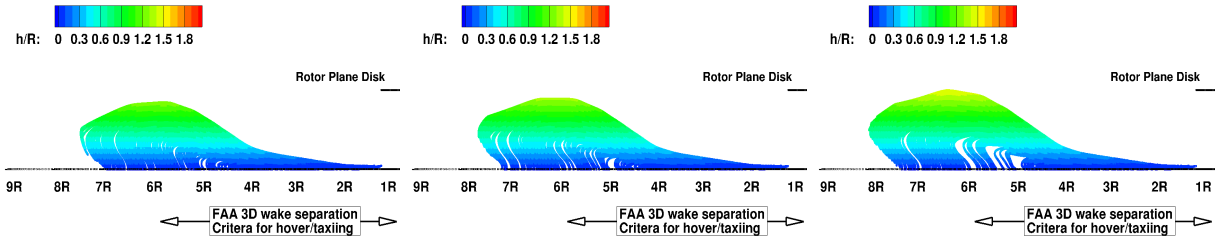
visibility is strongly reduced due to particle reingestion, generating the dangerous DVE condition described in the introduction. Furthermore, during the re-ingesting phase, particles can hit the blades and fuselage, damaging rotor structures or being ingested by the engines. Particles that move away from the rotor can be dangerous for ground personnel, equipment and other aircraft, while the recirculation of the particles creates risks for the helicopter and the crew itself due to the reingesting phase.



(a) h/R=0.5, Low DL helicopter (b) h/R=0.5, Medium DL helicopter (c) h/R=0.5, High DL helicopter



(d) h/R=1, Low DL helicopter (e) h/R=1, Medium DL helicopter (f) h/R=1, High DL helicopter



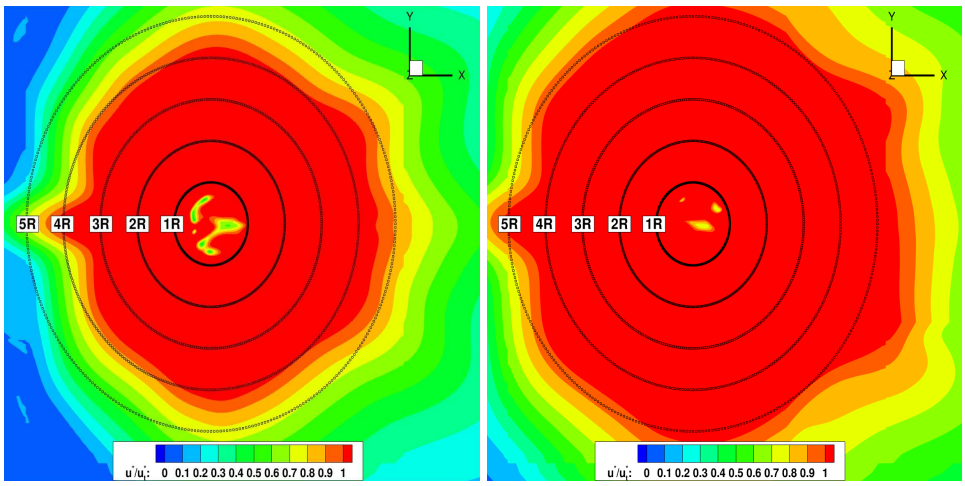
(g) h/R=1.5, Low DL helicopter (h) h/R=1.5, Medium DL helicopter (i) h/R=1.5, High DL helicopter

Figure 7.6: Particle paths for rotors at different heights above the ground and different scaling factors. The rotors were operating at $\theta_{75} = 12deg$, $Re_{tip} = 35000$ and $M_{tip} = 0.08$.

7.5 Forward flight rotor IGE

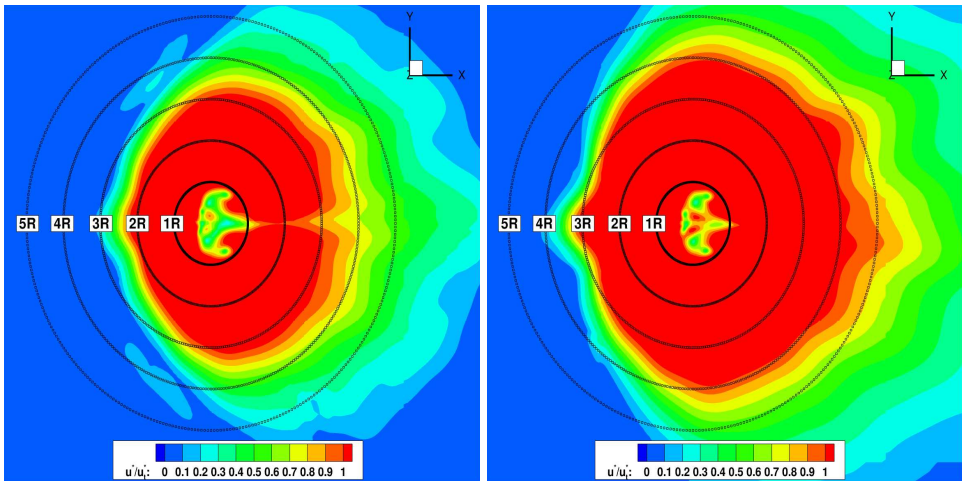
7.5.1 Uplift results

Like in the case of the isolated hovering rotor, results in terms of the uplift area are now presented. Figure 7.7, shows particle uplift results for two disk loading configurations and for all advance ratios simulated. In general, higher thrust coefficients lead to a wider uplift area. The uplift area increases by almost 1R when the C_T is higher, in all cases considered. For the hover condition, shown in figure 7.7 (A) and (B), the uplift area is 3420 m^2 (roughly $25A_{rotor}$) for $C_T = 0.0125$, while for $C_T = 0.0175$ it grows to 4928 m^2 (approx $36A_{rotor}$). For both forward flight cases, the uplift area decreases. For $V_\infty = 2.5m/s$ shown in figure 7.7 the uplift area is about 980 m^2 (approx $10A_{rotor}$) for the $C_T = 0.0125$ and 1536 m^2 (roughly 16 A_{rotor}) for $C_T = 0.0175$. The uplift area drops more by increasing the taxiing speed. In this case (figure 7.7 (E) and (F), $V_\infty = 5.14m/s$), the uplift area is 800 m^2 (roughly 8.4 A_{rotor}) for $C_T = 0.0125$, and 1240 m^2 (approx 13 A_{rotor}) for $C_T = 0.0175$. Furthermore, it is possible to notice, how the uplift area shape changes with increasing taxing speed. For the hover case, the uplift area is almost circular, however, it becomes stretched downstream, with increasing taxiing speed. For higher advance ratios, the uplift area extends mainly behind the rotor, while for the hover case it is almost equally distributed around the rotor forming a circle. This is due to the change of the rotor wake topology, that is influenced by the different advance ratios of the simulations. The wake topology changes due to the different taxiing speeds have been studied in previous works including [144], [163].



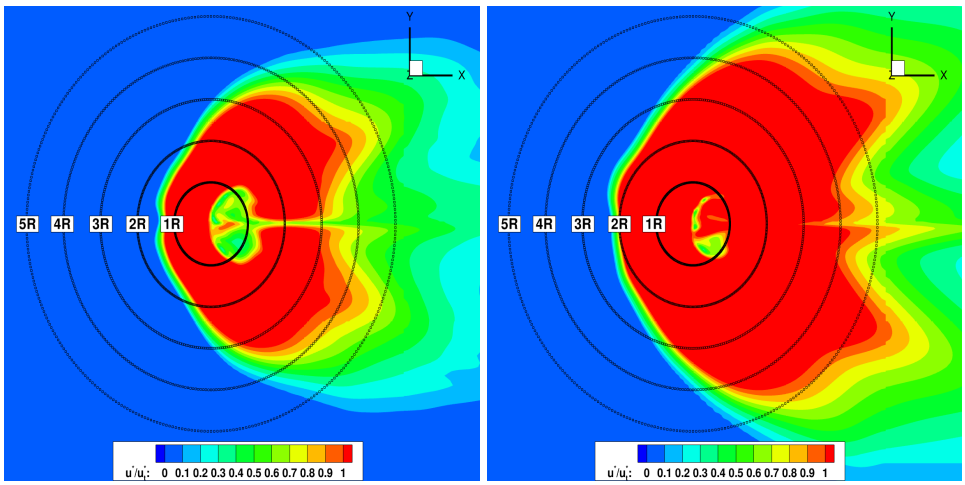
(a) Hover, $C_T = 0.125$

(b) Hover, $C_T = 0.175$



(c) $\mu = 0.011, C_T = 0.125$

(d) $\mu = 0.011, C_T = 0.175$



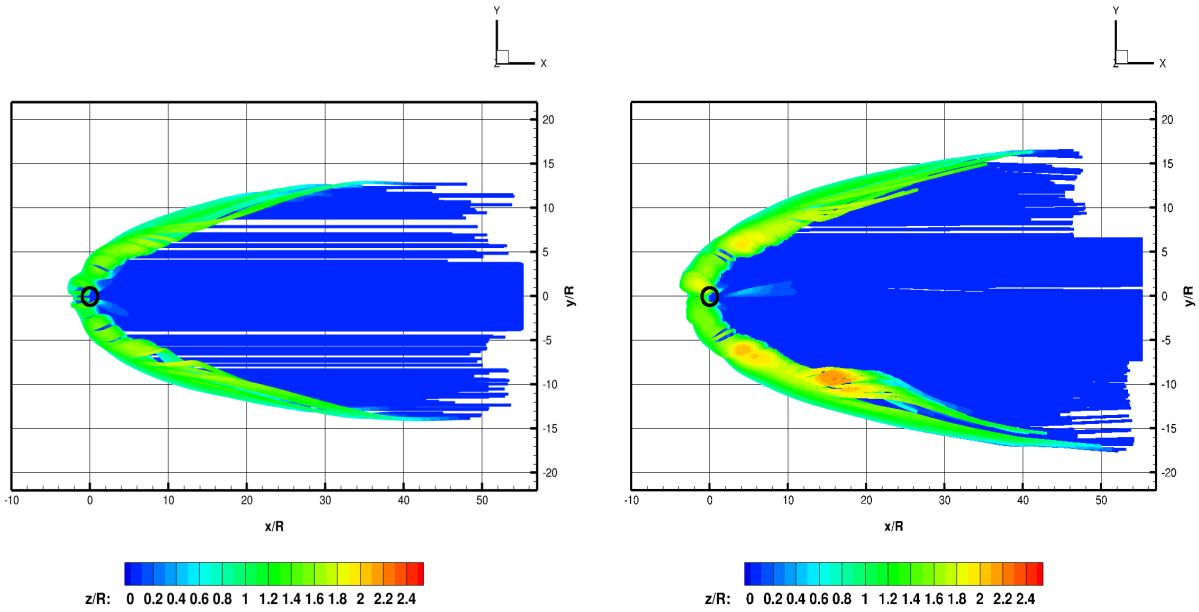
(e) $\mu = 0.022, C_T = 0.125$

(f) $\mu = 0.022, C_T = 0.175$

Figure 7.7: Particle uplift criteria results. The rotor was operating at h/R=1.66. The employes particle properties are listed in table 7.1.

7.5.2 Brownout clouds

In figures 7.8, the obtained brownout clouds are shown. Following the wake path, particles can reach long distances away from the aircraft. For all thrust coefficients studied, at a distance of 30R behind the rotor particles are still suspended in air, with an altitude of around 1.5R. At this height, the particles can affect the visual environment of other aircraft, and surround nearby structures or ground personnel. The velocity and mass of these particles were low, however, they may generate problems to unprotected eyes of ground personnel or decrease the life-time of structures due to dust erosion and impacts. The maximum lateral spread of the two main paths followed by the particles is around 20R in the case of $C_T = 0.0125$, while 30R for the higher disk loading configuration. In general, particles are uplifted in hover and low speed forward flight, and in all cases considered, a brownout cloud was generated. In the hover case, the cloud surrounding the rotor, reached an altitude of 1.6R and a maximum radial distance of 9R. In this scenario the hovering time maneuver had a direct impact on the visual environment of the pilot. For the forward flight cases, the cloud expands over a larger area behind the rotor. However, after a certain distance, particles fall again on the ground. In this case, the duration of the maneuver does not have a strong influence on the formed brownout cloud. When the rotor passes over the ground, it uplifts new particles, however, they do not accumulate around the rotor, but they are left behind it, following the rotor wake. In general, during hover, the DVE condition due to the brownout cloud, is more severe compare to the forward flight case. During forward flight, particles can reach larger distances, and affect structures and people far away from the rotor. Finally, comparing the two thrust coefficients for the forward flight case, the higher DL rotor generated the most dangerous scenario, pushing particles further and higher. Qualitatively speaking, the results just described, are close to what was presented by Tanner et al. [12] in their experimental work, and by Garrick et al. with an Eulerian model for brownout [14]. In these works, rotors IGE and forward flight have been considered. Even if the rotor configurations are dissimilar, the brownout clouds showed similar paths and sizes in terms of height and width with the present results.



(a) Forward flight, 10 kts, $C_T = 0.0125$

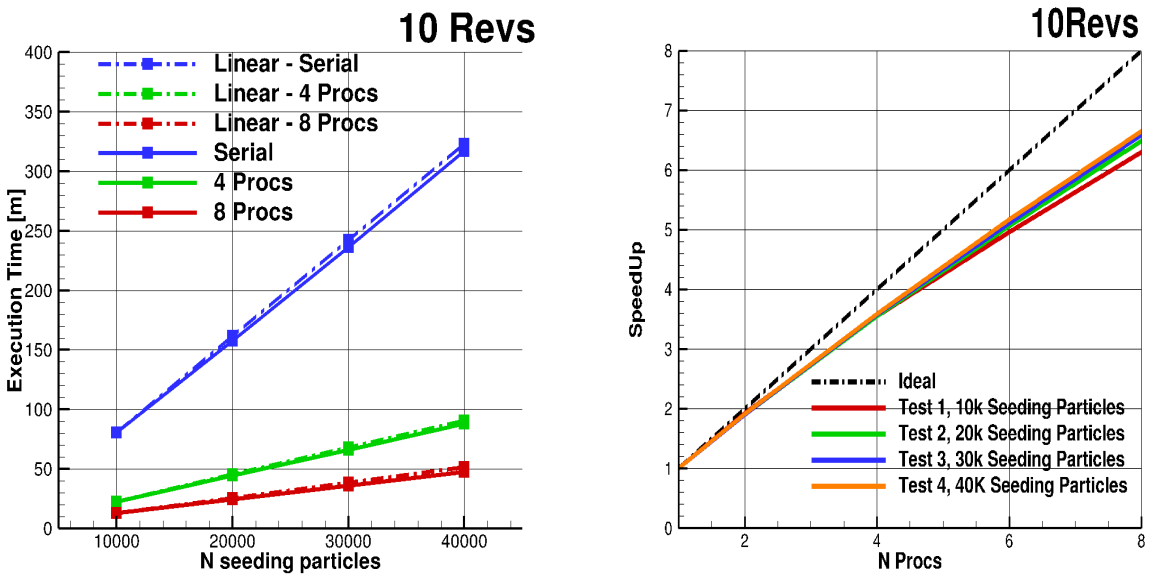
(b) Forward flight, 10 kts, $C_T = 0.0175$

Figure 7.8: Particle tracking results for forward flying rotors, 10 kts (μ=0.023), top view.

7.6 Timing and processors scaling

The total execution time is shown in figure 7.9 (a) for 10 rotor revolutions, using different numbers of seeding particles, and different numbers of processors. Results show a linear behavior. The execution time increases with the number of seeding particles tracked, however, the linear slope is lower for a bigger number of processors. All processors track the same amount of particles, improving the performance linearly with the number of processors used. For these tests, an unsteady hover rotor simulation has been used, with a coarse mesh (1.6 M). The particle tracking approach, as used, scales with the product of the number of particles used and the number of cell-centers checked. For the efficiency of the present method, the minimization of the number of cells in the CFD mesh involved in the search is a key factor. In figure 7.9 (b) the speedup of the total execution time is compared with the ideal linear behavior. Results show that the particle tracking timing is closer to the ideal case when a smaller amount of processors is used, this is in accordance with the Amdahl's law [164]. Figure 7.9 (b) compares the speedups of different tests, with different number of seeding particles. When the amount of particles is bigger, the obtained timing is slightly closer to the ideal case with respect to the case with a lower amount of seeding particles. The serial part of

the code does not change with the amount of particles, on the other hand the parallel part increases with it, resulting in better performance when more processors are used.



(a) Total Execution time for 10 Revs

(b) Code and Ideal Speedup

Figure 7.9: Total times and speedups for particle tracking.

To summarise, brownout clouds have been computed for different helicopters at different operational scenarios using a Lagrangian algorithm developed for this thesis work. The Lagrangian algorithm has been validated using experimental results. Considering the limits of steady simulations, computational results are within the range of the experimental scatter, with the mean distance between the CFD and experimental results around 1R for the top view, and 0.5R for a side view.

For hovering rotors, results suggest that the rotor distance from the ground has an important influence on cloud size and shape. When the rotor distance from the ground is reduced, particles are reingested in the rotor disk, defining a severe DVE condition. Furthermore, for all hovering rotors, particles are pushed further the 3D limit of FAA wake encounters, which cannot be adopted to define a safe area in terms of presence of particles.

On the other hand, particles uplifted by taxiing rotors generate a dense particle wall in front of the aircraft. This wall may spoil the pilot visibility. Furthermore, behind the rotor, particles follow the rotor wake and spread for huge distances behind the aircraft.

Finally, Lagrangian algorithm computational performances have been tested. Results show how the parallel strategy improves the overall computational performance of the algorithm, decreasing the final execution time depending by the number of processors used.

Chapter 8

Eulerian simulations[1]

As discussed in chapter 1, alongside of the Lagrangian approach, the Eulerian model can be used to compute brownout clouds. The Eulerian approach considers the dispersed phase as a continuum, usually described using a scalar transport equation, solved along with the flowfield model. This approach has also been used in this work. In general, this approach is more affordable in terms of computational cost, but may not be possible to track the particles with enough accuracy to evaluate their interactions with surrounding objects and the ground. However, it may still be possible to evaluate the pilot visual environment and the DVE severity due to the cloud. A full description of the Eulerian model has already been given in chapter 3. In this chapter, Eulerian results are presented; initially, they are compared with experimental results, and later with CFD results obtained using the Lagrangian method discussed previously. Furthermore, the two models are compared in terms of computational efficiency.

8.1 Algorithm

The HMB3 (Helicopter Multi-Block) [122, 165] is used for simulations using scalar transport equations. It solves the equations presented in chapter 2 in integral form with the ALE formulation

[1] Part of this chapter has been published in Rovere Federico, Barakos George, and Steijl Rene. "Rovere, F., Barakos, G. N. and Steijl, R. (2022) Eulerian and Lagrangian Brownout Simulations Methods, In: Aerospace Science and Technology.

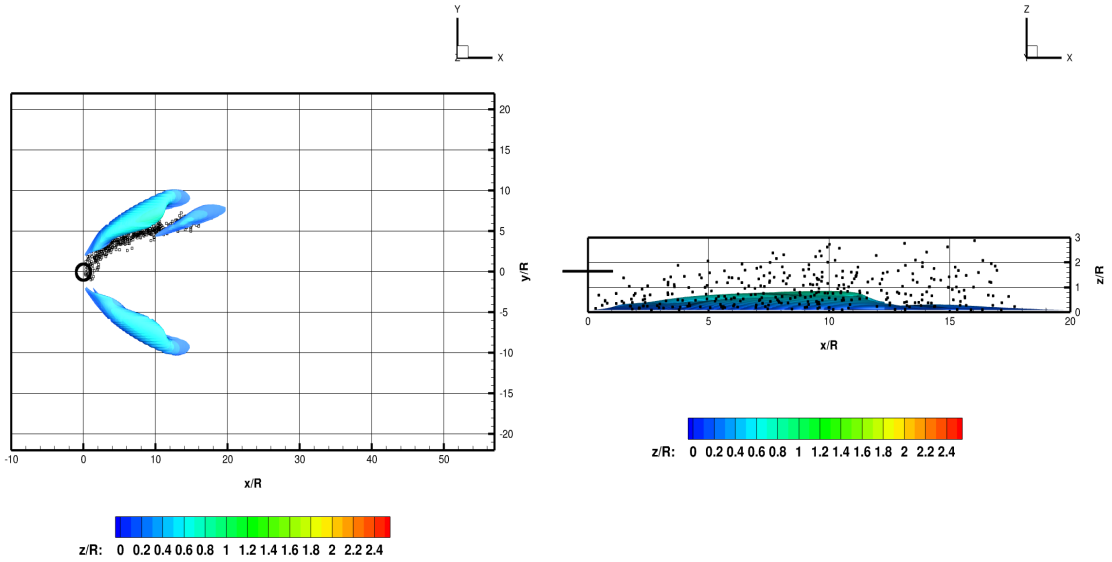
(Arbitrary Lagrangian Eulerian) for time-dependent domains (moving boundaries). Like for the URANS (Unsteady Reynolds Averaged Navier-Stokes) equations, the transport equations are discretised using a cell-centered finite volume approach on a multi block structured grid. HMB3 uses the Osher [130] and Roe [166] approximate Riemann solvers to evaluate the convective fluxes, and the viscous terms are discretised using second order central differencing. Third order accuracy in space is provided by the Monotone Upstream Centred Schemes for Conservation Laws (MUSCL) [167]. To avoid non-physical spurious oscillations, HMB3 uses the alternative form of the Van Albada limiter [131] where large gradients exist in the flowfield. An implicit dual time stepping method is employed to perform the temporal integration in time-accurate simulations.

8.2 Forward flying rotors IGE

In this section, Eulerian results will be discussed. Initially, validation has been performed with experiments by Wong and Tanner [12]. Later, test cases analyzed including aircraft in forward flight as already described in Chapter 3. Different rotor operational scenarios have been performed and results are compared.

8.2.1 Validation Test Case - EH-60L Wong and Tanner Experiments

The Eulerian model has been validated using experimental results obtained by Wong and Tanner [12]. Like in the Lagrangian validation proposed in Chapter 5, experiments have been performed using photogrammetry to obtain qualitative results of the brownout cloud development of a EH-60L taxiing. The aircraft was operating at around 15m (around 1.8R) of altitude and 12 m/s (approx 25 kts) of speed. A flowfield simulation has been performed for the same mesh, described in Chapter 5, with a higher advance ratio ($\mu = 0.058$) and for a $C_T = 0.017$. The ground has been seeded, imposing the value of $\phi = 1$ in the proximity of the ground. Computed results are compared to experimental data in figure 8.1.



(a) Top View

(b) Side View

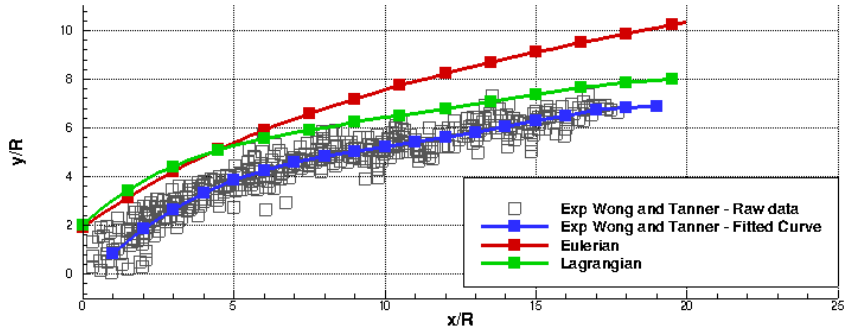
Figure 8.1: Eulerian model validation. Experiments (black dots) [12] vs present study results (iso-surfaces with contours). The rotor was operating at $\mu = 0.058, C_T = 0.017$. The employed particle properties are listed in table 4.1.

The experimental results show the development of the cloud at different time steps, compared with the unsteady results with steady state simulations. The experimental results have been collected in a single cloud shape. The aircraft position is moved to the origin of the domain, while the relative position between the aircraft position and the cloud points has been preserved. The results are presented as iso-surfaces of ϕ, number of particles density, as described in Chapter 3. The results reveal how the Eulerian iso-surface cloud gives an approximation to the brownout cloud in figure 8.1 (A). Computational and experimental results follow the wake shape, forming an almost parabolic dust front. A small offset can be noticed between experimental and computational results, however, this can be due to the comparison between a steady state simulation work and unsteady experimental results, and the necessary transformations. The computational results underestimate the maximum height of the cloud, showing a maximum cloud altitude around 1R, while the maximum experimental cloud reaches almost 1.5R (figure 8.1 (B)). These results are in agreement with the Lagrangian particle tracking validation.

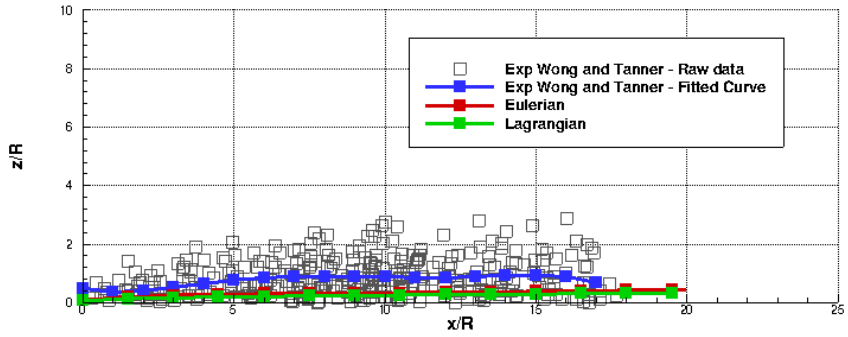
Figure 8.2, quantifies the distance between the experimental results and the numerical sim-

ulations. Considering the difficulties of comparing different clouds in terms of shape and size, the clouds have been approximated with a polynomial curve obtained by the least squares method.

Figure 8.2 (A) is a top view. As can be seen in 8.2 (A), closer to the rotor (for $X/R < 5$), the two numerical methods present similar results, with a distance of $0.3R$ at $X/R = 1.5$. While, the distance between the experiments and the numerical methods is about 1R for $X/R < 5$. Furthermore, when the distance from the rotor increases, the two numerical methods slightly disagree. For $X/R > 5$, the Lagrangian method is more accurate, preserving an error with the experimental curve of 1R. It is possible to notice, that this error is almost constant for all the range of X/R considered, and the shape of experimental results is close to the numerical. On the other hand, Eulerian results worsen further from the rotor, showing a distance from the experimental curve of 3R at around $X/R = 20$. Figure 8.2 (B) shows the side view. Here, the scatter in the experimental results is more evident, and the comparison between experimental and numerical method is more difficult. However, in both view, it is possible to notice that the distance between numerical models and experiments are within the data scatter of experimental results. Like in the previous case, the two numerical methods show closer results, with the distance between the two numerical methods around 0.1R, while the distance between the experiments is around 0.5R. From the top view, it is possible to say that the Lagrangian is more accurate than the Eulerian method for $X/R > 5$. However, this method is limited by the amount of particles it can handle.



(a) Top View



(b) Side View

Figure 8.2: Eulerian and Lagrangian models validation. Experiments (gray dots) [12] vs present study results. The rotor was operating at $\mu = 0.058$, $C_T = 0.017$. Particle properties are listed in table 4.1.

8.2.2 Lagrangian and Eulerian results comparisons

Several operational scenario have been tested. Different advance ratios and disk loading configurations have been investigated to define the impact of these aspects on brownout clouds. For these tests the following operational conditions have been used.

A direct comparison between Lagrangian particle tracking and Eulerian results is now presented. Eulerian results are plotted using iso-surfaces of volume fraction occupied by dust particles (iso-surfaces at $\phi = 10^{-3}$) and compared with massless Lagrangian particle tracking. For this comparison, the gravity has not been taken into account in both models. These simulations are performed as steady, imposing at first, $3cm$ sand thickness layer on the ground. The wake was

V_∞ $[kts]$	Mesh	C_T	Steady/Unsteady
0 (Hover)	low μ	0.0125	Steady
0 (Hover)	low μ	0.0175	Steady
5	low μ	0.0125	Steady
5	low μ	0.0175	Steady
10	low μ	0.0125	Steady
10	low μ	0.0175	Steady
10	low μ (without fuselage)	0.0125	Steady

Table 8.1: Simulations performed for Eulerian particle tracking.

steady. On the other hand, Lagrangian particle tracking has been set up using the information extracted from the uplift analysis. To avoid an exponential increase in execution time, the ground has been seeded with the particles only where they are more likely to be uplifted by the flowfield. The ground has been seeded considering the same density of particles on the ground of 8 particles per m^2, with particles covering all areas where the friction threshold model is satisfied. The number of particles per square unit has been chosen as a trade-off between the finite computing resources available and the need to have enough particles in the flowfield to represent the brownout cloud. Even if the number may appear small, the final cloud, counts millions of particles. Furthermore, different layers of particles have been placed at different heights on the ground, from zero up to $3cm$ thickness with a Gaussian distribution. As a final result, the low speed forward flight with $C_T = 0.0125$, has been seeded with 6110 particles, resulting in a final cloud of 91 million particles, while the case with $C_T = 0.0175$ the number of particles on the ground was 10,000, resulting in a final cloud of 150 million particles. For this comparison, the particles are considered massless, and gravity has not been taken into account.

Initially, a comparison in terms of effects is presented. Lagrangian and Eulerian results are compared from a side view, focusing on the cloud outline. The two methods have been tested considering different terms separately. Eulerian results have been compared with and without terminal velocity, while Lagrangian results are compared for particles with mass (and gravity) and massless particles (without gravity). Overall, the cloud outlines give a good comparison in terms of height reached by the particles. In general the gravity effect is considered in a similar way by the two methods. Results are presented in figure 8.3. For these simulations, the no-fuselage set-up has

been used. The rotor was operating at a forwards speed of 10kts ($\mu = 0.022$), and $C_T = 0.0125$. The outlines show similar behavior in proximity to the rotor, where the convection effects are stronger. Here, the gravity and mass effects are secondary, and the two methods agree well. This trend continues far from the rotor for the two Eulerian cases, the terminal velocity has a minimum effect, and the two outlines are close. However, when the mass of particles is taken into account, the Lagrangian approach shows a slightly stronger effect and the cloud drops when x/R is around 40. This may suggest that the gravity effect in the Eulerian model is somehow underestimated, with respect to the Lagrangian case, and careful matching is needed in the parameters of the two models to be fully comparable.

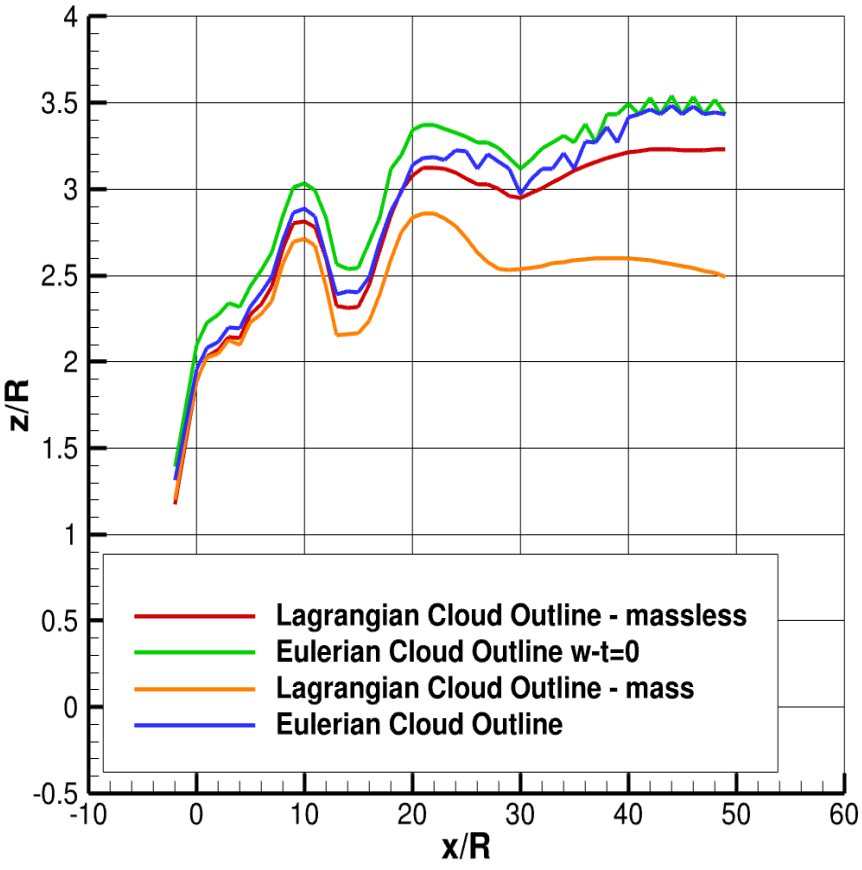


Figure 8.3: Lagrangian and Eulerian cloud outline, side view. The rotor operating in taxiing, $\mu = 0.022$, h/R=1.66, $C_T = 0.0125$. No fuselage setup.

Brownout clouds computed with Eulerian approach are now presented. These simulations are performed at different taxiing speeds, and they include the fuselage.

In figures 8.4 (A) and (B) the clouds generated for the lowest advance ratio condition are shown. In these results, the forward speed is very low, and the case can be considered as almost in hover. At this advance ratio, the freestream velocity effect is minimum, and the cloud surrounds the rotor in almost every direction, leaving a small clearance only behind it. For both thrust coefficients, the cloud reaches 20R distance in the lateral direction and around 20R behind the rotor. However, particles also spread ahead of the aircraft, generating a dense recirculation region that reaches around 15R. Furthermore, the particles can reach height of 6R above the ground. In this case, the cloud surrounds totally the aircraft, generating a severe DVE condition for all thrust coefficients tested here. The two thrust configurations show very similar results. However, when the thrust coefficient is higher, the cloud spreads more, reaching higher and further.

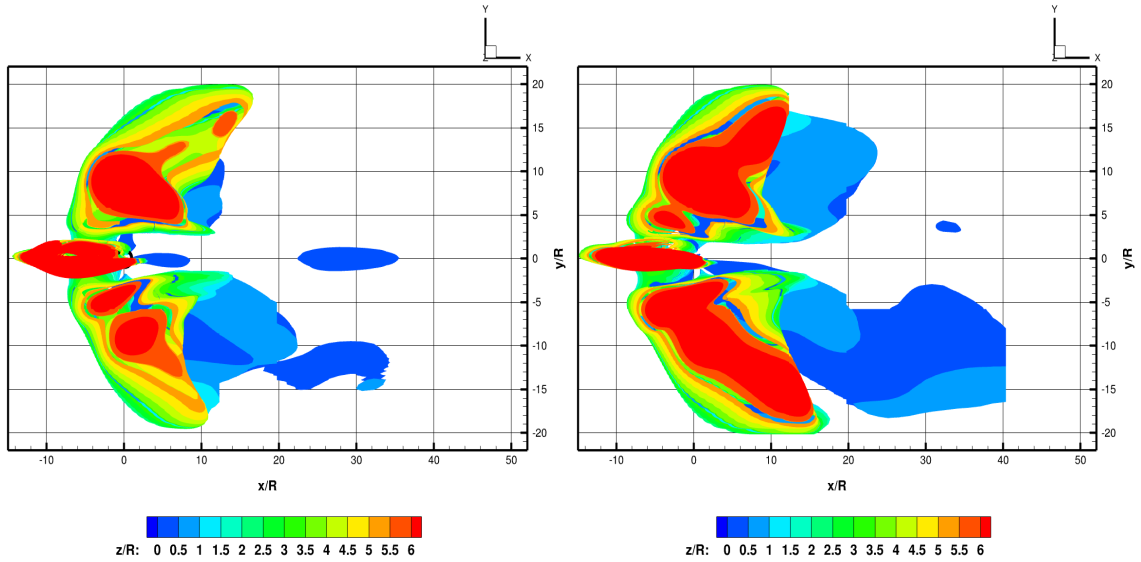
In general, the Lagrangian particle tracking agrees with the Eulerian results and, the main cloud features can be seen also in figure 8.4 (C) and (D). However, some differences can still be noticed. The Lagrangian results show more the asymmetry of the flowfield. In the Eulerian case, the two sides of the cloud look similar in terms of dimensions, while for the Lagrangian results, the cloud side at $Y/R < 0$ expands more in the X/R direction with respect to the other side ($Y/R > 0$). This may be due to the differences on ground seeding between the two models, and the limited number of particles in the Lagrangian case. Some asymmetry was expected in the flowfield due to the presence of the fuselage and the usage of a non-uniform actuator disk in these simulations.

A comparison between the two models was also performed for higher advance ratio cases, in figures 8.5 (C) and (D). At this taxiing speed, the cloud spreads head of the rotor, following the rotor wake. Behind the rotor, the cloud spreads in two branches, that can reach a distance of up to 20R in the lateral direction, and 2.5R in height. This part of the brownout cloud may affect other aircraft operating nearby, spoiling their visuals, or the driven dust may be ingested by their engines. Furthermore, particles may damage structures and give problems to ground personnel. As seen in previous works, higher disk loading configurations are able to generate bigger clouds [168, 57]. For the high advance ratio case the two models agree better than for the low advance

ratios. The branches of the cloud are quite symmetric for all cases studied, and the cloud shape and dimensions agree between the two computational models.

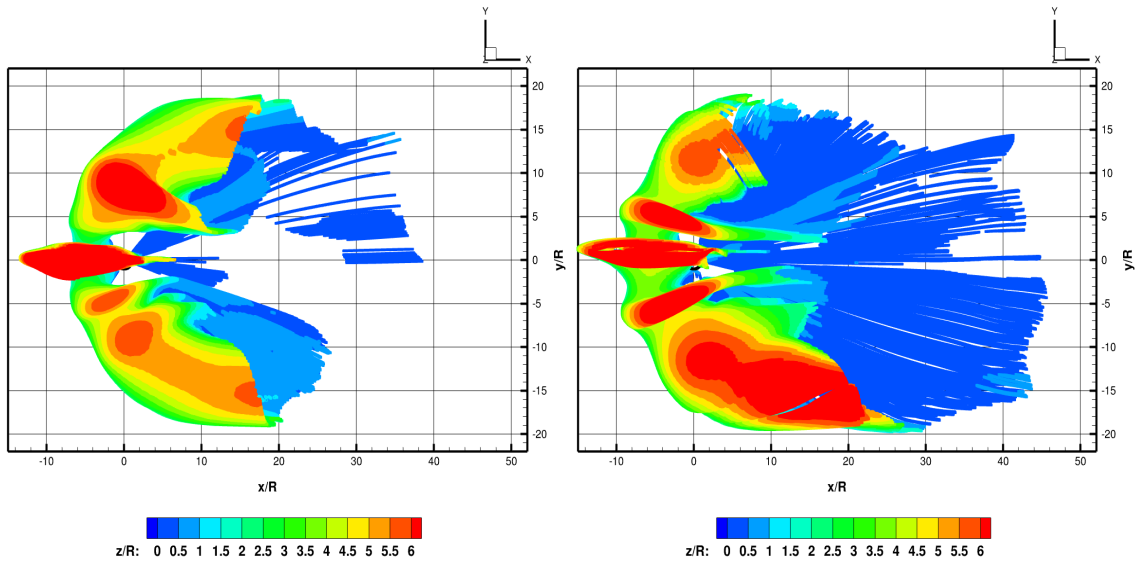
When the taxiing speed is further increased, the cloud is strongly influenced by the wake topology, like in the previous cases. At $\mu = 0.022$ (figures 8.6 (A) and (B)) the cloud is formed by a dense region in front of the rotor and by two branches behind the aircraft. However, some differences can still be noticed. First of all, the region in front of the rotor is smaller for the higher advance ratio, decreasing the severity of the DVE condition, with respect to the lower advance ratio case. In this case, the recirculation region stops at 4R in front of the rotor. As before, a higher thrust coefficient leads to larger and higher brownout clouds. Figures 8.6 (A) and (B) can be compared with figures 8.6 (C) and (D) for a direct contrast of the results obtained with the two different approaches. Results are generally comparable throughout the domain. Particles can reach similar results for the cloud shape, maximum heights and distance reached. The final cloud is 15R wide and with a maximum heights of around 3R. Furthermore, both approaches show larger clouds in the case of higher rotor thrust coefficient configurations.

Comparing figures 8.6 (B) and 8.6 (C), it is possible to notice some differences in proximity to the rotor. In the Lagrangian case, some particles reach high positions during recirculation. This number of particles is low and they are not visible in the Eulerian results which show an iso-surfaces of $\phi = 10^{-3}$. Figure 8.7 compares the Eulerian and Lagrangian results for the no fuselage configuration. In both cases, it is possible to notice that asymmetric of the previous clouds was due to the fuselage. The clouds shown in figure 8.7 (a) and (b) look very similar. Both clouds follow the wake topology, with a limited recirculation region in front of the rotor. In this part of the cloud, the particles are not reingested by the rotor itself, however, their height is enough to spoil the visual of the pilot on the ground.



(a) Eulerian, $C_T = 0.125$

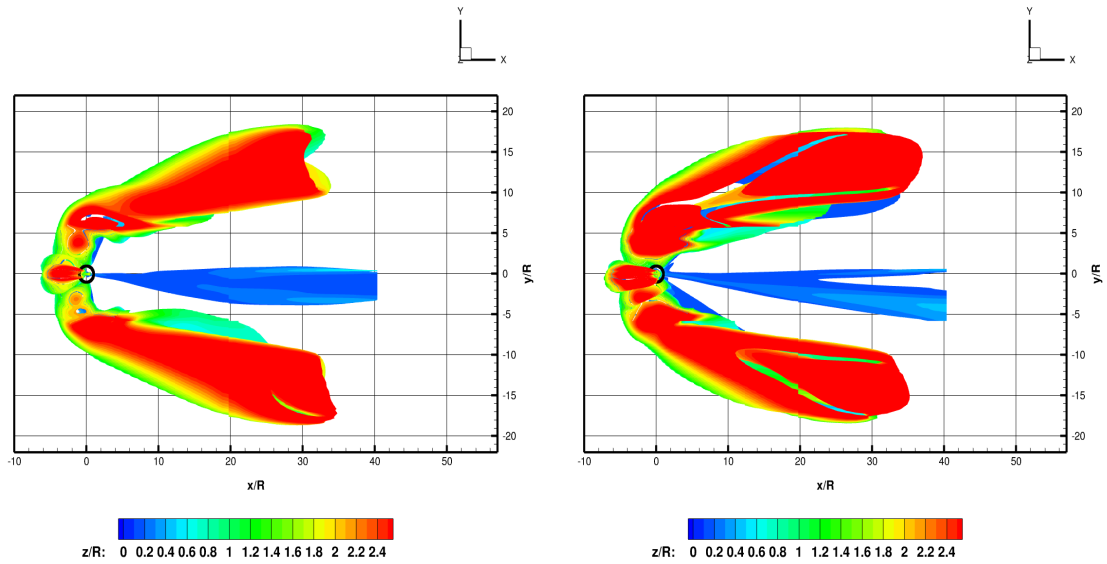
(b) Eulerian, $C_T = 0.175$



(c) Lagrangian, $C_T = 0.125$

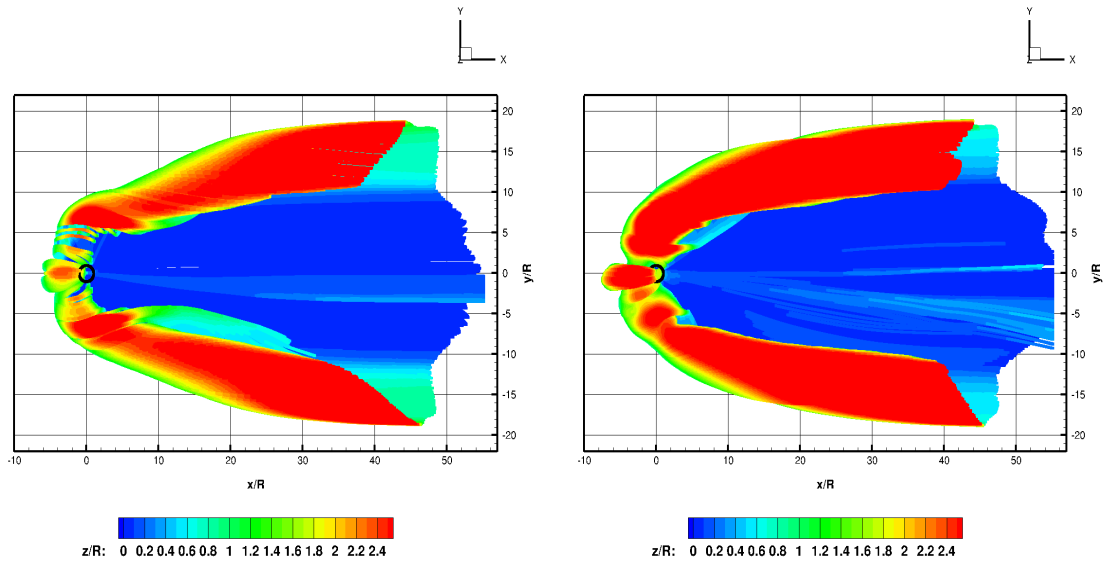
(d) Lagrangian, $C_T = 0.175$

Figure 8.4: Lagrangian particle tracking and cloud iso-surfaces, with the rotor operating in hover, h/R=1.66. Massless particles.



(a) Eulerian, $C_T = 0.125$

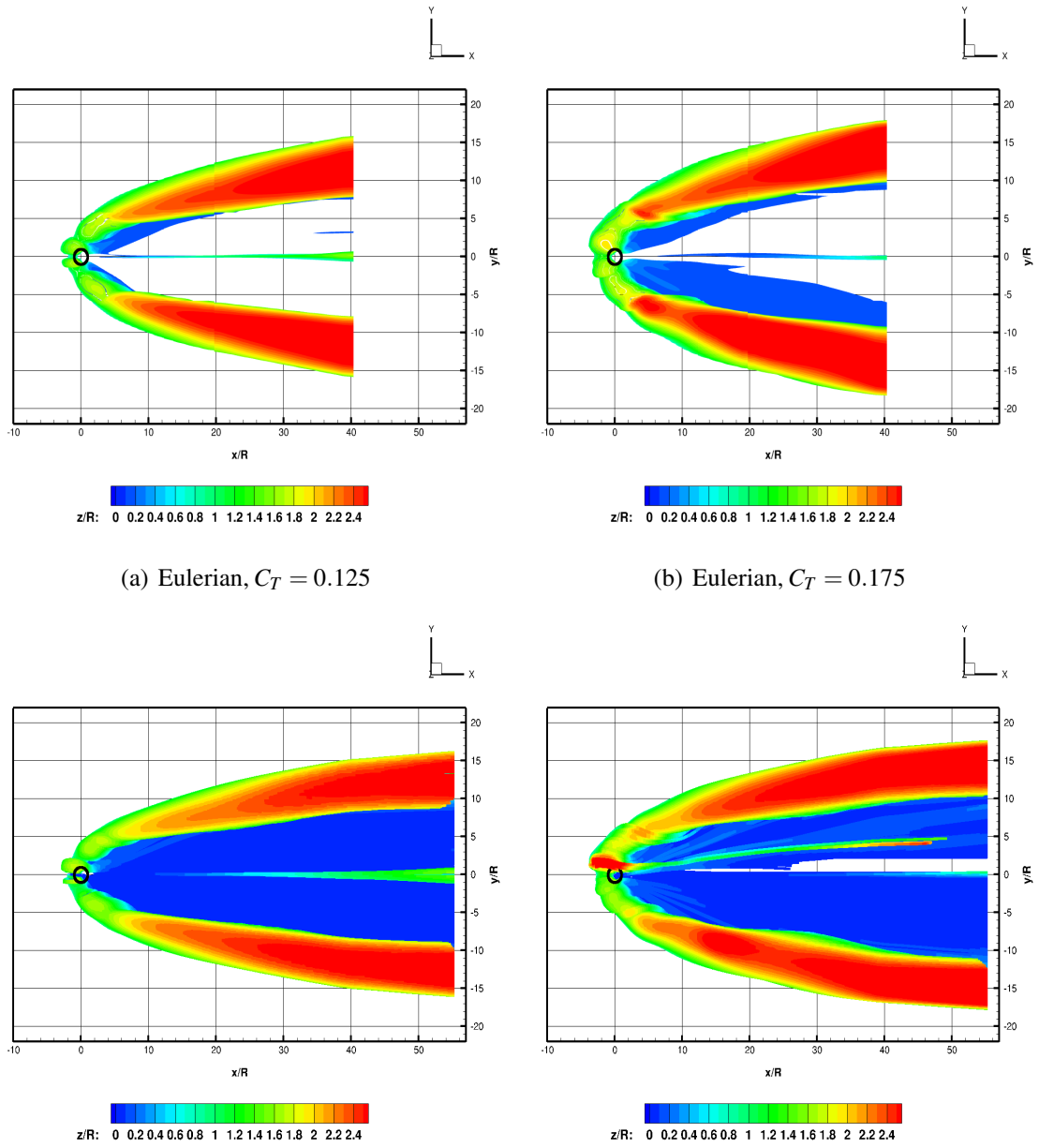
(b) Eulerian, $C_T = 0.175$



(c) Lagrangian, $C_T = 0.125$

(d) Lagrangian, $C_T = 0.175$

Figure 8.5: Lagrangian particle tracking and cloud iso-surfaces, with the rotor operating in forward flight $\mu = 0.011$, h/R=1.66. Massless particles.



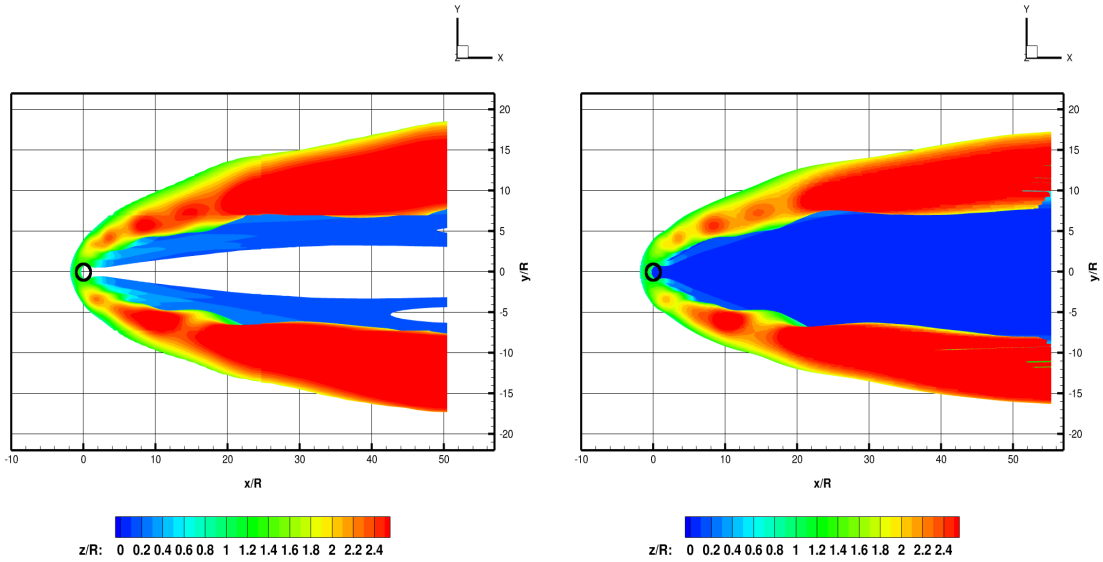
(a) Eulerian, $C_T = 0.125$

(b) Eulerian, $C_T = 0.175$

(c) Lagrangian, $C_T = 0.125$

(d) Lagrangian, $C_T = 0.175$

Figure 8.6: Lagrangian particle tracking and cloud iso-surfaces, with the rotor operating in forward flight $\mu = 0.022$, h/R=1.66. Massless particles.



(a) Eulerian Method

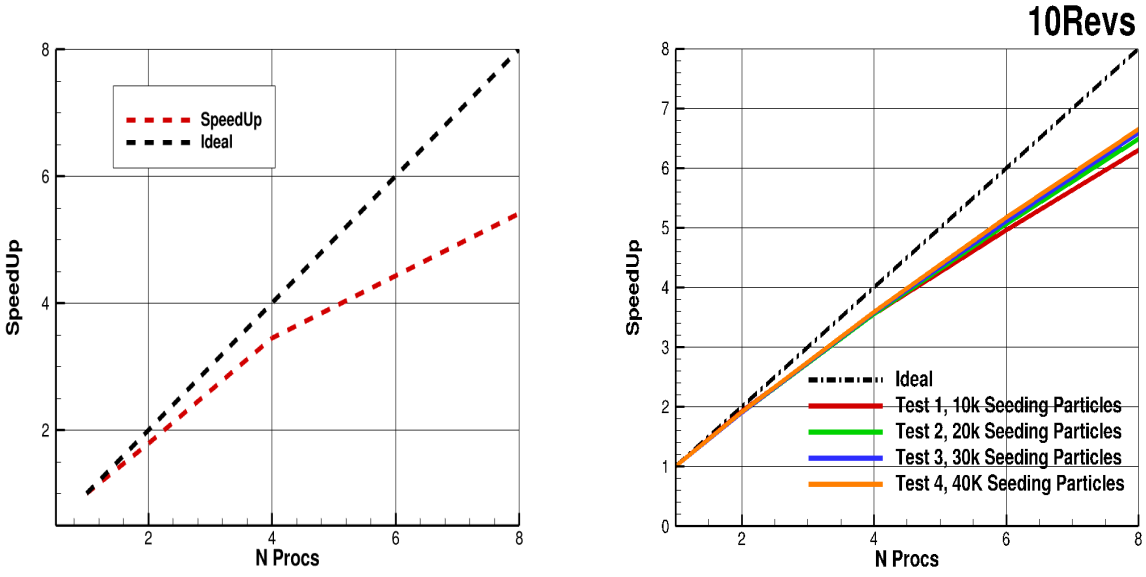
(b) Lagrangian Method

Figure 8.7: Lagrangian and Eulerian results, without fuselage. The rotor was operating in forward flight $\mu = 0.022$, h/R=1.66. Massless particles.

8.3 Timing and performance comparison of the methods

In this section, a detailed performance analysis is shown, comparing Eulerian and Lagrangian execution times and parallel performances. Both codes can run on serial and parallel computers. As previously described, in the Lagrangian particle tracking, the total amount of particles is shared among processors. In general, the size of the mesh has an influence on the total execution time of Lagrangian code. Searching particles in finer meshes is computationally more expensive. However, the mesh size does not have a strong influence on parallel scaling, all processors have to read the input flowfield files. On the other hand, in the Eulerian method, the code takes advantage of the multi-block approach used in HMB3 and processors solve the transport equation for the assigned number of blocks. Blocks are shared among the processors to obtain an equal amount of work. Due to the nature of the Eulerian approach, the total amount of particles is not known a priori. A single diffusion equation is computed, independently from the number of particles that a specific test case may take into account. In figure 8.8 it is possible to see how performances improve with increasing number of processors. The speed up is close to the ideal-linear behavior for a very lim-

ited number of processors in both cases. However, the Lagrangian case is closer to the ideal cases with a higher number of processors. When 8 processors are used, the speed-up of the Lagrangian code is around 6.3 and 6.6 (depending on the number of particles used), while Eulerian speed up stops at 5.5. This is a function of the employed number of blocks. Dividing further the domain, and using more CPUs can lead to better performances. It is important to say that in the Eulerian case, the amount of memory used for the simulation is limited, and shared among the processors, while in the case of the Lagrangian code, when mode processors are used, more memory is needed to store the flowfield. In the Lagrangian code, to limit the number of information exchanges between processors, the flowfield is read by every processor, increasing the total amount of memory needed for the simulation, with increasing number of processors.



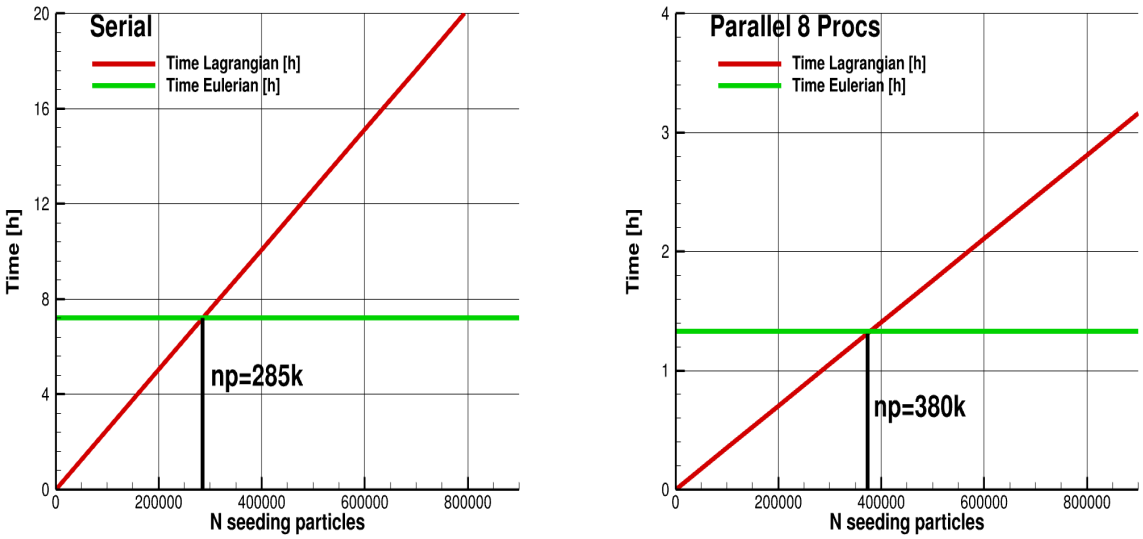
(a) Eulerian

(b) Lagrangian

Figure 8.8: Speedups Lagrangian vs Eulerian methods.

As mentioned previously, in the Eulerian approach the number of particles has no influence on timing performance, in general, when the amount of particles is limited the Lagrangian approach is better than the Eulerian, it allows a faster and more accurate calculation, however as this amount grows, the total execution time may become prohibitive. In figure 8.9 the total execution time for the Lagrangian and Eulerian code are compared. The total execution time for Lagrangian code can

be considered to scale linearly with the total number of particles in the flowfield, and this execution time is computed starting from the linear behavior computed for a smaller amount of particles [108]. In contrast, the Eulerian execution time is constant with the number of particles. Every sub-plot of figure 8.9, emphasizes a number of particles uplifted. When the total amount of particles in the simulation is greater than the threshold, the Lagrangian execution time overcomes the Eulerian, and it became less efficient. It is also possible to notice that when the number of processors increases, the threshold grows. This is due to the fact that the Lagrangian method speed-up is faster with respect to the Eulerian, as shown in figure 8.8. The linear slope of the Lagrangian curve decreases faster than the Eulerian fixed value, as show by figure 8.9. This is due to the larger amount of memory needed to run the Lagrangian test case. Furthermore, the Lagrangian output files may contain a very large number of points, and may become difficult to plot.



(a) Serial

(b) Parallel 8 Procs

Figure 8.9: Timing Lagrangian vs Eulerian for different numbers of processors.

To summarize, brownout clouds obtained using the Eulerian model show small difference with experimental and Lagrangian results closer to the rotor. Under 5R of distance from the rotor, the cloud width error is around 1R, while the height error is about 0.1R. However, when the distance from the rotor increases, the width error grows up to 3R, while the height error increase

to 0.5R. Considering the final size of the clouds and the values of the available experimental data, these errors are acceptable. Comparing different rotor operational scenarios, the clouds generated by the two different numerical models are similar in terms of height and side. Furthermore, a computational analysis compared the efficiency of the two models. The Eulerian model was more convenient in terms of computational costs when the amount of particles to be tracked was high.

Chapter 9

Conclusions and future work

This thesis investigated the safety of aircraft operating IGE, starting from the analysis of the outflow generated by the rotor and operating IGE at different conditions. This was followed by the prediction and model brownout cloud that such outflow may generate when a rotorcraft is operating over loose ground sediments. To obtain the brownout cloud a Lagrangian particle tracking tool and an Eulerian transport model were developed. The two models were compared in terms of results and timing performance. Initially, the CFD code was validated against experimental results in terms of rotor performance and flowfield, later the work focused on the safety analysis in terms of PAXman forces acting on ground personnel, and brownout clouds generated by the interaction between the wake and a loose sediment bed. Investigation on the rotor flight parameters was carried out, comparing results for the rotor hovering at different heights above the ground. Several disk loading configurations, and different rotor advance ratios were used. Results obtained have been compared with other safety distance criteria, defining a possible ”safe area for ground personnel operation”.

9.1 Conclusions

In general, CFD results obtained in this investigation agrees with experimental data in terms of rotor performance and flowfield for OGE and IGE. For OGE simulations, CFD results for thrust and

torque slightly underestimate the experimental results at the collective angle and Reynolds number used. For the IGE configurations, HMB3 can predict rotor performance in terms of C_T^{IGE}/C_T^{OGE} and C_Q^{IGE}/C_Q^{OGE} with errors that do not exceed 3%. However, some differences in the outflow predictions for radial stations far away from the rotor can be noticed. The differences can be important if used to evaluate safety regions near the helicopter. However, CFD slightly overestimates the outflow, leading to a more conservative approach.

Eulerian and Lagrangian approaches have been used in the past for brownout simulations. For the first time, in this work these two approaches have been compared for the same test cases, in accuracy and efficiency. In general, brownout clouds follow the wake topology in both approaches. For the taxiing speeds taken into account, a dust wall is formed in front of the rotor, following the recirculation region ahead of the fuselage. However, the cloud also spreads behind the rotor, forming two branches that cover a wide area around the flight path. Both methods agree on the cloud size, defining different clouds for different configurations of taxiing speed and disk loading. Furthermore, both numerical methods have been compared with experimental results. The differences in terms of size have found to be 0.1R closer to the rotor, and up to 2R far from the rotor in terms of width and up to 0.5R in terms of height. A direct comparison between the final clouds computed by the two computational methods shows minimum differences in cloud shape and size. Convection of particles is the main effect with gravity and diffusion having a secondary role for the tested conditions. However, the computational cost to simulate brownout clouds with Lagrangian particle tracking can be prohibitive. The execution time needed to simulate a brownout cloud is influenced by the total amount of particles in the flowfield and it may become very high. For these reasons the Eulerian method may be considered a more efficient method to predict brownout even with a cost in terms of accuracy and level of physical detail.

In terms of safety, aerodynamic forces exerted by the outflow on ground personnel have been studied with the PAXman model. Results show that a safety distance criterion based on the PAXman forces, can be deducted by the outflow analysis. For the first branch of results, a hovering small-scale rotor has been used. Due to the small scale of the rotor, scaling factors have been applied to the flowfield, to take into account the stronger outflow generated by a full

scale rotor. For all aircraft configurations, taken into account in this work, there is no risk due to the outflow forces for ground personnel. Several configurations were considered in this work, the most dangerous scenario in terms of total forces acting on human body is generated by an hovering rotor at h/R=1.5, which reach 280N between 1 and 2R. This maximum value, however, is under the caution limit of PAXman model. In general, results show that the PAXman total forces are higher in an area between 1R and 2R away from the rotor, but after 3R it is safer. It is clear that the FAA wake encounter criterion of 3D can be adopted for ground operation if the presence of particles on the ground can be excluded. The force distributions suggest that the chest is the part of the human body that is most influenced by the force due to the outflow and this result can be part of the evaluation on definite safety regulations for people acting in the proximity of the rotor. Disk loading has a strong influence on the PAXman forces for the hovering and forward flying rotors. Higher DL helicopters can generate stronger outflow, and create more dangerous scenarios for ground personnel, nearby equipment and structures. On the other hand, the height above the ground, for hovering rotor, has a limited influence on PAXman forces. Furthermore, PAXman forces are strongly influenced by advance ratio. Hovering rotors generate higher forces. However, when the rotor is in forward flight, forces drop significantly, and when the taxiing speed increases they drop even more, resulting in a safe situation. This can be due to the rotor wake topology at different forward speeds, which is closer to the ground (and ground personnel) for low forward speed.

Particle tracking results show that particles can be uplifted and reach large distances away from the rotor, exceeding the limit of 3D. In general, it appears that the FAA limit for wake encounters cannot be used as safety guarantee in presence of particles on the ground. It is also clear that to define a particle free zone, it is necessary to take into account the rotor operating conditions due to the strong influence of the disk loading and in general of the size of the aircraft on the particle paths. Particle paths are also strongly influenced by the position of the rotor with respect to the ground. Results show that the uplift of particles is influenced by the taxiing speed, and higher advance ratios lead to a smaller uplift area on the ground. Furthermore, the advance ratio also has an influence on the uplift area shape. The almost circular uplift area shown in the hover case

changes with increasing taxiing speed. Furthermore, uplift and brownout simulations indicate that the disk loading configurations increase the uplift area and also increase the cloud size, potentially leading to more dangerous scenarios for the crew and the ground personnel.

9.2 Future work

To guide future research work on safety for rotor operating IGE and brownout simulations, the following could be explored:

- Brownout simulations using Eulerian models could be explored more. Results so far are limited to one-way coupling and transport models;
- The possibility to use two-way coupling should be investigated. Furthermore, different models of gravity should be considered. This of course is useful for cases with significant uplift;
- Evaluate other rotors geometric and operational features that may influence brownout/white-out, such as the number of blades, presence of tail rotor and different manoeuvres that may affect particles uplift and cloud;
- Investigate the impact of brownout on the aircraft structure, evaluating how the life-time of rotor blades and structures change in case of extensive and frequent brownout operations;
- Further research on the effect of particles in the visuals of the pilot, evaluating the severity of DVE generated by the concentration of particles, defining different severity levels based on specific particle concentration levels;
- Particle tracking models described in this work can be used for other applications. Flow involved with spray or modelling the experiments like PIV in wind tunnel, where seeding of the flow is required.

Bibliography

[1] "Helicopter Assault: Putting SOF on Target," https://fightersweep.com/1635/helicopter-assault-putting-sof-on-target/, Accessed: 18-10-2017.

[2] Collins, P. and Moore, C., "Solutions to helicopter blade erosion, improving aircraft availability and reducing costs," *40th European Rotorcraft Forum 2014*, 2014, pp. 109–118.

[3] Feuerstein, A. and Kleyman, A., "Ti–N multilayer systems for compressor airfoil sand erosion protection," *Surface and Coatings Technology*, Vol. 204, No. 6, 2009, pp. 1092–1096.

[4] Trexler, M. and Kopchik, J., "Development of a novel erosion resistant coating system for use on rotorcraft blades," *Annual Forum Proceedings - AHS International*, Vol. 3, 2012, pp. 1889–1897.

[5] Weigel, W. D., "Advanced Rotor Blade Erosion Protection System." Tech. rep., KAMAN AEROSPACE CORP BLOOMFIELD CT, 1996.

[6] Nissen, J., Holemans, P., and Venezia, J., "Rotor shield advanced rotor blade erosion protection, application to V-22," *Annual Forum Proceedings - AHS International*, Vol. 4, 2015, pp. 2478–2489.

[7] Calvert, M. and Wong, T.-C., "Aerodynamic impacts of helicopter blade erosion coatings," *30th AIAA Applied Aerodynamics Conference*, 2012, p. 2914.

[8] Smialek, J. L., "The chemistry of Saudi Arabian sand: A deposition problem on helicopter turbine airfoils," 01 1991, Gordon Conference on Corrosion; 14 Jul. 1991; New London, NH; United States.

[9] Mapes, P., Kent, R., and Wood, R., "DoD helicopter mishaps FY85-05: findings and recommendations," *US Air Force*, Vol. 13, 2008.

[10] "INTRO NH-90 HELICOPTER NAVY," http://www.dutchaviationsupport.eu/00-HTMS/a&darcticzone-2-pv.htm, Accessed: 18-10-2017.

[11] Rodgers, S. J., "Evaluation of the Dust Cloud Generated by Helicopter Rotor Downwash," Tech. rep., USAAVLABS Technical Report 67-81, March 1968, 1968.

[12] Wong, O. D. and Tanner, P. E., "Photogrammetric Measurements of an EH-60L Brownout Cloud," *Journal of the American Helicopter Society*, Vol. 61, No. 1, 2016, pp. 1–10.

[13] Nathan, N. and Green, R., "Flow visualisation of the helicopter brown-out phenomenon," *The Aeronautical Journal*, Vol. 113, No. 1145, 2009, pp. 467–478.

[14] Garrick, D. P., Rajagopalan, R. G., and Guntupalli, K., "Simulation of landing maneuvers of rotorcraft in brownout conditions," *International Powered Lift Conference, Los Angeles, CA*, 2013-4266.

[15] Ghosh, S., Lohry, M. W., and Rajagopalan, R. G., "Rotor configurational effect on rotorcraft brownout," *28th AIAA Applied Aerodynamics Conference, Chicago, Illinois, USA*, 2010 - 4238.

[16] Phillips, C. and Brown, R. E., "Eulerian simulation of the fluid dynamics of helicopter brownout," *Journal of Aircraft*, Vol. 46, No. 4, 2009, pp. 1416–1429.

[17] Alfred, J., Celi, R., and Leishman, J. G., "Flight Path Optimization for Brownout Mitigation Using a High-Fidelity Simulation Model," *Journal of the American Helicopter Society*, Vol. 62, No. 3, 2017, pp. 1–15.

[18] Preston, J. R., Troutman, S., Keen, E., Silva, M., Whitman, N., Calvert, M., Cardamone, M., Moulton, M., and Ferguson, S. W., "Rotorwash Operational Footprint Modeling," Tech. rep., Missile Research Development and Engineering Center Redstone Arsenal AL Missile Guidance directorate, 2014.

[19] Silva, M. and Riser, R., "CH-47D tandem rotor outwash survey," *67th Annual Forum of American Helicopter Society , Virginia Beach, Virginia, USA*, 2011.

[20] Garcia-Dorado, I., Aliaga, D. G., Bhalachandran, S., Schmid, P., and Niyogi, D., "Fast Weather Simulation for Inverse Procedural Design of 3D Urban Models," *ACM Transactions on Graphics*, Vol. 36, No. 2, 2017, pp. 21.

[21] Wadcock, A. J., Ewing, L. A., Solis, E., Potsdam, M., and Rajagopalan, G., "Rotorcraft Downwash Flow Field Study to Understand the Aerodynamics of Helicopter Brownout," Tech. rep., 10 2008.

[22] Sugiura, M., Tanabe, Y., Sugawara, H., Matayoshi, N., and Ishii, H., "Numerical Simulations and Measurements of the Helicopter Wake in Ground Effect," *Journal of Aircraft*, 2016.

[23] Cowherd Jr, C., "Sandblaster 2 Support of See-Through Technologies for Particulate Brownout," Tech. rep., MIDWEST RESEARCH INST KANSAS CITY MO, 2007.

[24] Zehnder, J. A., Zhang, L., Hansford, D., Radzan, A., Selover, N., and Brown, C. M., "Using digital cloud photogrammetry to characterize the onset and transition from shallow to deep convection over orography," *Monthly weather review*, Vol. 134, No. 9, 2006, pp. 2527–2546.

[25] Ramasamy, M. and Yamauchi, G. K., "Using Model-Scale Tandem-Rotor Measurements in Ground Effect to Understand Full-Scale CH-47D Outwash," *Journal of the American Helicopter Society*, Vol. 62, No. 1, 2017, pp. 1–14.

[26] 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, 2010, pp. 22005.

[27] Milluzzo, J., Sydney, A., Rauleder, J., and Leishman, J., "In-ground-effect aerodynamics of rotors with different blade tips," *66th Annual Forum Proceedings of the American Helicopter Society, Phoenix, AZ*, 2010.

[28] Milluzzo III, J. I. and Leishman, J. G., "Vortical sheet behavior in the wake of a rotor in ground effect," *AIAA Journal*, Vol. 55, No. 1, 2016, pp. 24.

[29] alakrishnan Ganesh, B. and Komerath, N., "Unsteady aerodynamics of rotorcraft in ground effect," *Journal of Aircraft*, 2005.

[30] Saijo, T., Ganesh, B., Huang, A., and Komerath, N., "Development of unsteadiness in a rotor wake in ground effect," *21st AIAA Applied Aerodynamics Conference*, 2003, p. 3519.

[31] Bourne, K., Reddy, K., Kumar, C., and Ooi, A., "An investigation of fundamental flow structures in ground effect with application to the development of brownout conditions in hover," 2014.

[32] Curtiss, H., Sun, M., Putman, W., and Hanker, E., "Rotor aerodynamics in ground effect at low advance ratios," *Journal of the American Helicopter Society*, Vol. 29, No. 1, 1984, pp. 48–55.

[33] Tanner, P. E., Overmeyer, A. D., Jenkins, L. N., Yao, C.-S., and Bartram, S. M., "Experimental Investigation of Rotorcraft Outwash in Ground Effect," 2015, AHS International Annual Forum & amp; Technology Display; 71st 2015; Virginia Beach, Virginia, USA.

[34] Rauleder, J. and Leishman, J. G., "Flow Environment and Organized Turbulence Structures Near a Plane Below a Rotor," *AIAA journal*, Vol. 52, No. 1, 2014, pp. 146–161.

[35] Sydney, A., Baharani, A., and Leishman, J., "Understanding brownout using near-wall dual-phase flow measurements," *67th Annual Forum Proceedings of the American Helicopter Society, Virginia Beach, VA*, 2011.

[36] Ramasamy, M., Johnson, B., and Leishman, J. G., "Understanding the Aerodynamic Efficiency of a Hovering Micro-Rotor," *Journal of the American Helicopter Society*, Vol. 53, No. 4, 2008, pp. 412–428.

[37] Johnson, B., Leishman, J. G., and Sydney, A., "Investigation of Sediment Entrainment Using Dual-Phase, High-Speed Particle Image Velocimetry," *Journal of the American Helicopter Society*, Vol. 55, No. 4, 2010, pp. 42003–42003.

[38] Glucksman-Glaser, M. S., *The Effects of Model Scaling on Sediment Transport in Brownout*, Ph.D. thesis, 2013.

[39] Rauleder, J. and Leishman, J. G., "Particle-fluid interactions in rotor-generated vortex flows," *Experiments in Fluids*, Vol. 55, No. 3, Mar 2014, pp. 1689.

[40] Rauleder, J. and Leishman, J. G., "Interactions of a Vortical Flow with Dispersed Particles Below a Rotor," *AIAA Journal*, 2016.

[41] Whitehouse, G. R., Wachspress, D. A., and Quackenbush, T., "Aerodynamic design of helicopter rotors for reduced brownout," *International Powered Lift Conference Proceedings, Philadelphia, PA*, 2010.

[42] Baharani, A. K., *Investigation into the Effects of Aeolian Scaling Parameters on Sediment Mobilization Below a Hovering Rotor*, Ph.D. thesis, 2011.

[43] Glucksman-Glaser, M. S., *The Effects of Model Scaling on Sediment Transport in Brownout*, Ph.D. thesis, University of Maryland, 2013.

[44] Perrotta, G., Glucksman-Glaser, M., and Jones, A. R., "Similarity Parameters for the Characterization of Sediment Mobilization by Unsteady Rotor Wakes," *Journal of Aircraft*, 2015.

[45] Leishman, G. J., *Principles of helicopter aerodynamics with CD extra*, Cambridge University Press, 2006.

[46] Fradenburgh, E. A., "Aerodynamic factors influencing overall hover performance," *Aerodynamics of Rotary Wings*, Vol. 7, No. 1, 1972.

[47] Sydney, A. and Leishman, J. G., "Time-Resolved Measurements of Rotor-Induced Particle Flows Produced by a Hovering Rotor," *Journal of the American Helicopter Society*, Vol. 59, No. 2, 2014, pp. 1–16.

[48] Sydney, A., Leishmat, J., and Martin, G., "Measurements of the plume-like three-dimensionality of rotor-induced dust fields," *Annual Forum Proceedings - AHS International*, Vol. 2, 2014, pp. 873–892.

[49] Sydney, A. and Leishman, J., "Measurements of Rotor/Airframe Interactions in Ground Effect Over a Sediment Bed," *69th Annual Forum of American Helicopter Society, Phoenix, Arizona, USA*, 2013.

[50] Wachspress, D., Whitehouse, G., Keller, J., Yu, K., Gilmore, P., Dorsett, M., and McClure, K., "A high fidelity brownout model for real-time flight simulations and trainers," *in: 65th Annual Forum of American Helicopter Society, Grapevine, Texas, USA, 2009*, pp. 1281–1304.

[51] Gerlach, T., "Visualisation of the brownout phenomenon, integration and test on a helicopter flight simulator," *The Aeronautical Journal*, Vol. 115, No. 1163, 2011, pp. 57–63.

[52] Harris, M. J. and Lastra, A., "Real-time cloud rendering for games," *Proceedings of Game Developers Conference*, 2002, pp. 21–29.

[53] Harris, M. J. and Lastra, A., "Real-Time Cloud Rendering," *Computer Graphics Forum*, Vol. 20, Wiley Online Library, 2001, pp. 76–85.

[54] Yusov, E., "High-performance rendering of realistic cumulus clouds using pre-computed lighting," *Proceedings of High Performance Graphics*, Eurographics Association, 2014, pp. 127–136.

[55] Kallweit, S., Müller, T., McWilliams, B., Gross, M., and Novák, J., "Deep Scattering: Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks," *arXiv preprint arXiv:1709.05418*, 2017.

[56] Tritschler, J. K., Celi, R., and Leishman, J. G., "Methodology for Rotorcraft Brownout Mitigation Through Flight Path Optimization," *Journal of Guidance, Control, and Dynamics*, 2014.

[57] Milluzzo, J. and Leishman, J. G., "Assessment of rotorcraft brownout severity in terms of rotor design parameters," *Journal of the American Helicopter Society*, Vol. 55, No. 3, 2010, pp. 32009.

[58] Keller, J. D., Whitehouse, G. R., Wachspress, D. A., Teske, M. E., and Quackenbush, T. R., "A physics-based model of rotorcraft brownout for flight simulation applications," *62th Annual Forum Proceedings of the American Helicopter Society, Phoenix, AZ*, 2006.

[59] Teske, M., Thistle, H., and Ice, G., "Technical advances in modeling aerially applied sprays," *Transactions of the ASAE*, Vol. 46, No. 4, 2003, pp. 985.

[60] Ferguson, S. W., "Rotorwash Analysis Handbook Volume I - Development and Analysis," Tech. rep., FAA, DOT/FAA/RD-93/21, 1994.

[61] Wenren, Y., Walter, J., Fan, M., and Steinhoff, J., "Vorticity Confinement and Advanced Rendering to Compute and Visualize Complex Flows," *44th AIAA Aerospace Sciences Meeting and Exhibit Reno*, 2006–0945.

[62] Smith, M. J., Hoang, R. V., Sgambati, M. R., Dascalu, S., and Harris Jr, F. C., "A Dynamic Multi-contextual GPU-based Particle System using Vector Fields for Particle Propagation." *CAINE*, 2008, pp. 203–208.

[63] Ebert, D. S., *Texturing & modeling: a procedural approach*, Morgan Kaufmann, 2003.

[64] de Reus, A., Bakker, R., and Nijland, T., "A simulation environment for helicopter flight in degraded visual environments," 2010.

[65] Syal, M., Govindarajan, B., and Leishman, J., "Mesoscale sediment tracking methodology to analyze brownout cloud developments," *66th Annual Forum of the American Helicopter Society, Phoenix, Arizona, USA, 2010*, pp. 1644–1673.

[66] Syal, M. and Leishman, J. G., "Predictions of Brownout Dust Clouds Compared to Photogrammetry Measurements," *Journal of the American Helicopter Society*, Vol. 58, No. 2, 2013, pp. 1–18.

[67] D'Andrea, A., "Development and application of a physics-based computational tool to simulate helicopter brownout," 2011.

[68] Bilanin, A., Teske, M., and Hirsh, J., "Neutral atmospheric effects on the dissipation of aircraft vortex wakes," *AIAA Journal*, Vol. 16, No. 9, 1978, pp. 956–961.

[69] Govindarajan, B. and Leishman, J., "Predictions of Rotor and Rotor/Airframe Configurational Effects on Brownout Dust Clouds," *Journal of Aircraft*, 2015.

[70] Pulla, D. P. and Conlisk, A., *The long time structure of the rotor wake in ground effect*, Ph.D. thesis, Ohio State University, 2005.

[71] Xin, J., Chen, R., and Li, P., "Time-stepping free-wake methodology for rotor flow field simulation in ground effect," *Aircraft Engineering and Aerospace Technology: An International Journal*, Vol. 87, No. 5, 2015, pp. 418–426.

[72] Zhao, J. and He, C., "Physics-Based Modeling of Viscous Ground Effect for Rotorcraft Applications," *Journal of the American Helicopter Society*, Vol. 60, No. 3, 2015, pp. 1–13.

[73] 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, 2002, pp. 759–775.

[74] Griffiths, D. A., Ananthan, S., and Leishman, J. G., "Predictions of Rotor Performance in Ground Effect Using a Free-Vortex Wake Model," *Journal of the American Helicopter Society*, Vol. 50, No. 4, 2005, pp. 302–314.

[75] He, C. and Zhao, J., "Modeling rotor wake dynamics with viscous vortex particle method," *AIAA journal*, Vol. 47, No. 4, 2009, pp. 902.

[76] Haehnel, R., Moulton, M., Wenren, Y., and Steinhoff, J., "A Model to Simulate Rotorcraft-Induced Brownout," *67th Annual Forum Proceedings of the American Helicopter Society, Montreal, Canada*, 2008, pp. 589–601.

[77] Phillips, C., Kim, H. W., and Brown, R. E., "The flow physics of helicopter brownout," *66th American Helicopter Society Forum: , Phoenix, AZ, USA, 2010*, pp. 1273–1291.

[78] Rajagopalan, R. G. and Fanucci, J. B., "Finite difference model for vertical axis wind turbines," *Journal of Propulsion and Power*, Vol. 1, No. 6, 1985, pp. 432–436.

[79] McAlpine, J. D., Koracin, D. R., Boyle, D. P., Gillies, J. A., and McDonald, E. V., "Development of a rotorcraft dust-emission parameterization using a CFD model," *Environmental fluid mechanics*, Vol. 10, No. 6, 2010, pp. 691–710.

[80] Morales, F. and Squires, K. D., "Simulation of Rotor Vortex Interactions with a Particle-Laden Turbulent Boundary Layer," *29th AIAA Applied Aerodynamics Conference, Honolulu, HI*, 2011.

[81] 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*, 2011.

[82] Jasion, G. and Shrimpton, J., "Prediction of brownout inception beneath a full-scale helicopter downwash," *Journal of the American Helicopter Society*, Vol. 57, No. 4, 2012, pp. 1–13.

[83] Lakshminarayan, V. K., Kalra, T. S., and Baeder, J. D., "Detailed Computational Investigation of a Hovering Microscale Rotor in Ground Effect," *AIAA journal*, 2013.

[84] Friedmann, L., Ohmer, P., and Hajek, M., "Real-Time Simulation of Rotorcraft Downwash in Proximity of Complex Obstacles using Grid-Based Approaches," *70th Annual Forum of the American Helicopter Society*, 2014.

[85] Kutz, B., Günther, T., Rumpf, A., and Kuhn, A., "Numerical examination of a model rotor in brownout conditions," *in: 70th Annual Forum of American Helicopter Society, Montréal, Canada, 2014*, pp. 2462–2484.

[86] Oruc, I., Horn, J. F., and Shipman, J., "Coupled Flight Dynamics and Computational Fluid Dynamics Simulations of Rotorcraft/Terrain Interactions," *Journal of Aircraft*, 2017, pp. 1–15.

[87] Kalra, T. S. and Baeder, J. D., "Modeling Subscale Rotor Wake in Ground Effect with Accurate Turbulent Length Scales," *AIAA Journal*, 2017, pp. 1–10.

[88] Kalra, T. S., Lakshminarayan, V. K., and Baeder, J. D., "CFD validation of micro hovering rotor in ground effect," *66th Annual Forum American Helicopter Society 66th, Phoenix, Arizona, USA*, 2010.

[89] Thomas, S., Kalra, T. S., and Baeder, J. D., "A hybrid CFD methodology to model the two phase flowfield beneath a hovering laboratory scale rotor," *Proceedings of the 42nd AIAA Fluid Dynamics Converence (June 2012)*, Vol. 3, 2012.

[90] Milluzzo, J., *Effects of Blade Tip Shape on Rotor In-ground-effect Aerodynamics*, Ph.D. thesis, 2011.

[91] Dacles-Mariani, J., Zilliac, G. G., Chow, J. S., and Bradshaw, P., "Numerical/experimental study of a wingtip vortex in the near field," *AIAA journal*, Vol. 33, No. 9, 1995, pp. 1561–1568.

[92] Di Giacinto, M., Sabetta, F., and Piva, R., "Two-way coupling effects in dilute gas-particle flows," *Journal of Fluids Engineering*, Vol. 104, No. 3, 1982, pp. 304–311.

[93] Norouzi, H. R., Zarghami, R., Sotudeh-Gharebagh, R., and Mostoufi, N., *Coupled CFD-DEM modeling: formulation, implementation and application to multiphase flows*, John Wiley & Sons, 2016.

[94] Crowe, C. T., Schwarzkopf, J. D., Sommerfeld, M., and Tsuji, Y., *Multiphase flows with droplets and particles*, CRC press, 2011.

[95] Phillips, C., Kim, H., and Brown, R., "Helicopter brownout - Can it be modelled?" *Aeronautical Journal*, Vol. 115, No. 1164, 2011, pp. 123–133, cited By 11.

[96] Ryerson, C. C., Haehnel, R. B., Koenig, G. G., and Moulton, M. A., "Visibility enhancement in rotorwash clouds," *43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, 2005, pp. 10–13.

[97] Ahn, G., Jung, K., Myong, R., Shin, H., and Habashi, W., "Numerical and experimental investigation of ice accretion on rotorcraft engine air intake," *Journal of Aircraft*, 2015.

[98] Son, C. and Yee, K., "Procedure for Determining Operation Limits of High-Altitude Long-Endurance Aircraft Under Icing Conditions," *Journal of Aircraft*, 2017, pp. 1–16.

[99] Kim, J. W., Dennis, P. G., Sankar, L. N., and Kreeger, R. E., "Ice accretion modeling using an Eulerian approach for droplet impingement," *AIAA Paper 2013*, Vol. 246, 2013.

[100] Govindarajan, B., Leishman, J. G., and Gumerov, N. A., "Particle-Clustering Algorithms for the Prediction of Brownout Dust Clouds," *AIAA journal*, 2013.

[101] Hu, J., Xu, G., Shi, Y., and Wu, L., "A numerical simulation investigation of the influence of rotor wake on sediment particles by computational fluid dynamics coupling discrete element method," *Aerospace Science and Technology*, Vol. 105, 2020, pp. 106046.

[102] Hu, J., Xu, G., Shi, Y., and Huang, S., "The influence of the blade tip shape on brownout by an approach based on computational fluid dynamics," *Engineering Applications of Computational Fluid Mechanics*, Vol. 15, No. 1, 2021, pp. 692–711.

[103] Tan, J. F., Er Gao, J., Barakos, G. N., Lin, C. L., Zhang, W. G., and Huang, M. Q., "Novel approach to helicopter brownout based on vortex and discrete element methods," *Aerospace Science and Technology*, Vol. 116, 2021, pp. 106839.

[104] D'Andrea, A. and Scorcelletti, F., "Enhanced numerical simulations of helicopter landing maneuvers in brownout conditions," *66th Annual Forum of American Helicopter Society, Phoenix, Arizona, USA*, 11-13 May 2010, pp. 1084–1101.

[105] Syal, M., Rauleder, J., Tritschler, J., and Leishman, J. G., "On the possibilities of brownout mitigation using a slotted-tip rotor blade," *29th AIAA applied aerodynamics conference*, 2011, p. 3183.

[106] Bagnold, R., "The physics of blown sand and desert dunes. London: Methuen." 1941.

[107] Shao, Y. and Lu, H., "A simple expression for wind erosion threshold friction velocity," *Journal of Geophysical Research: Atmospheres*, Vol. 105, No. D17, 2000, pp. 22437–22443.

[108] Rovere, F., Barakos, G., and Steijl, R., "Safety analysis of rotors in ground effect," *Aerospace Science and Technology*, Vol. 129, 2022, pp. 107655.

[109] Polzin, W., Guntupalli, K., and Rajagopalan, R. G., "Discrete Blade Model for Rotorcraft Brownout," *29th AIAA Applied Aerodynamics Conference*, 2011-3182.

[110] Rabbott Jr, J. P., "Static-thrust measurements of the aerodynamic loading on a helicopter rotor blade," Tech. rep., NASA, 1956.

[111] Phillips, C., Kim, H. W., and Brown, R. E., "The effect of rotor design on the fluid dynamics of helicopter brownout," *35th European Rotorcraft Forum, Hamburg, Germany, 2009*, 2009.

[112] Calvert, M. E. and Wenren, Y., "Rotorcraft Outwash and Human Stability," *AIAA Scitech 2019 Forum*, 2019, p. 1095.

[113] Papadakis, M., Hung, K. E., Vu, G. T., Yeong, H. W., Bidwell, C. S., Breer, M. D., and Bencic, T. J., "Experimental investigation of water droplet impingement on airfoils, finite wings, and an S-duct engine inlet," 2002.

[114] Aksamit, N. O. and Pomeroy, J. W., "Near-surface snow particle dynamics from particle tracking velocimetry and turbulence measurements during alpine blowing snow storms," *The Cryosphere*, Vol. 10, No. 6, 2016, pp. 3043–3062.

[115] Clifton, A., Rüedi, J.-D., and Lehning, M., "Snow saltation threshold measurements in a drifting-snow wind tunnel," *Journal of Glaciology*, Vol. 52, No. 179, 2006, pp. 585–596.

[116] Sato, T., Kosugi, K., and Sato, A., "Saltation-layer structure of drifting snow observed in wind tunnel," *Annals of Glaciology*, Vol. 32, No. 1, 2001, pp. 203–208.

[117] Sugiura, K. and Maeno, N., "Wind-tunnel measurements of restitution coefficients and ejection number of snow particles in drifting snow: determination of splash functions," *Boundary-Layer Meteorology*, Vol. 95, No. 1, 2000, pp. 123–143.

[118] Guala, M., Manes, C., Clifton, A., and Lehning, M., "On the saltation of fresh snow in a wind tunnel: profile characterization and single particle statistics," *Journal of Geophysical Research: Earth Surface*, Vol. 113, No. F3, 2008.

[119] Tominaga, Y., "Computational fluid dynamics simulation of snowdrift around buildings: Past achievements and future perspectives," *Cold Regions Science and Technology*, 2017.

[120] Chirico, G., Szubert, D., Vigevano, L., and Barakos, G. N., "Numerical modelling of the aerodynamic interference between helicopter and ground obstacles," *CEAS Aeronautical Journal*, Vol. 8, No. 4, 2017, pp. 589–611.

[121] Boutanios, Z. and Jasak, H., "Two-way coupled Eulerian-Eulerian simulations of drifting snow with viscous treatment of the snow phase," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 169, 2017, pp. 67–76.

[122] Steijl, R., Barakos, G., and Badcock, K., "A framework for CFD analysis of helicopter rotors in hover and forward flight," *International Journal for Numerical Methods in Fluids*, Vol. 51, No. 8, 2006, pp. 819–847.

[123] Barakos, G., Steijl, R., Badcock, K., and Brocklehurst, A., "Development of CFD capability for full helicopter engineering analysis," 2005.

[124] Garcia, A. J. and Barakos, G. N., "Numerical simulations on the ERICA tiltrotor," *Aerospace Science and Technology*, Vol. 64, 2017, pp. 171–191.

[125] Leble, V. and Barakos, G., "A coupled floating offshore wind turbine analysis with high-fidelity methods," *Energy Procedia*, Vol. 94, 2016, pp. 523–530.

[126] Higgins, R. J., Barakos, G. N., Shahpar, S., and Tristanto, I., "A computational fluid dynamic acoustic investigation of a tiltwing eVTOL concept aircraft," *Aerospace Science and Technology*, Vol. 111, 2021, pp. 106571.

[127] Fitzgibbon, T. A., *Advanced rotor blade design based on high-fidelity computational fluid dynamics*, Ph.D. thesis, University of Glasgow, 2021.

[128] Versteeg, H. K. and Malalasekera, W., *An introduction to computational fluid dynamics: the finite volume method*, Pearson education, 2007.

[129] White, F. M. and Majdalani, J., *Viscous fluid flow*, Vol. 3, McGraw-Hill New York, 2006.

[130] Osher, S. and Chakravarthy, S., "Upwind schemes and boundary conditions with applications to Euler equations in general geometries," *Journal of Computational Physics*, Vol. 50, No. 3, 1983, pp. 447–481.

[131] Van Albada, G., Van Leer, B., and Roberts, W., "A comparative study of computational methods in cosmic gas dynamics," *Upwind and High-Resolution Schemes*, Springer, 1997, pp. 95–103.

[132] Axelsson, O., *Iterative solution methods*, Cambridge university press, 1996.

[133] Cantariti, F., Dubuc, L., Gribben, B., Woodgate, M., Badcock, K., and Richards, B., "Approximate Jacobians for the solution of the Euler and Navier-Stokes equations," *Aerospace Engineering Report*, Vol. 5, 1997.

[134] Jameson, A., "Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings," *10th Computational fluid dynamics conference*, 1991, p. 1596.

[135] Reynolds, O., "On the dynamical theory of incompressible viscous fluids and the determination of the criterion," *Philosophical Transactions of the Royal Society of London. A*, Vol. 186, 1895, pp. 123–164.

[136] Menter, F. R., "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA journal*, Vol. 32, No. 8, 1994, pp. 1598–1605.

[137] Wilcox, D. C., "Reassessment of the scale-determining equation for advanced turbulence models," *AIAA journal*, Vol. 26, No. 11, 1988, pp. 1299–1310.

[138] Greeley, R. and Iversen, J. D., *Wind as a geological process: on Earth, Mars, Venus and Titan*, Vol. 4, CUP Archive, 1987.

[139] Tanner, P. E., "Photogrammetric characterization of a brownout cloud," *67th Annual Forum of the American Helicopter Society, Virginia Beach, Virginia, USA*, 2011, pp. 3039–3055.

[140] Crowe, C. T., Schwarzkopf, J. D., Sommerfeld, M., and Tsuji, Y., *Multiphase flows with droplets and particles*, CRC press, 2011.

[141] Syal, M. and Leishman, J. G., "Modeling of bombardment ejections in the rotorcraft brownout problem," *AIAA journal*, Vol. 51, No. 4, 2013, pp. 849–866.

[142] Lu, H. and Shao, Y., "A new model for dust emission by saltation bombardment," *Journal of Geophysical Research: Atmospheres*, Vol. 104, No. D14, 1999, pp. 16827–16842.

[143] Marticorena, B. and Bergametti, G., "Modeling the atmospheric dust cycle: 1. Design of a soil-derived dust emission scheme," *Journal of Geophysical Research*, Vol. 100, No. D8, 1995, pp. 16,415–16,430.

[144] Sugiura, M., Tanabe, Y., Sugawara, H., Barakos, G. N., Matayoshi, N., and Ishii, H., "Validation of CFD Codes for the Helicopter Wake in Ground Effect," *in: 43th European Rotorcraft Forum, Milan, Italy, 2017*, 2017.

[145] Zhang, T. and Barakos, G. N., "High-fidelity CFD validation and assessment of ducted propellers for aircraft propulsion," *Journal of the American Helicopter Society*, Vol. 66, No. 1, 2021, pp. 1–28.

[146] Zhang, T., *Numerical investigation of ducted propellers for novel rotorcraft configurations*, Ph.D. thesis, University of Glasgow, 2022.

[147] ICEM CFD, "ver. 17.2," *ANSYS Inc., Southpointe*, 2016.

[148] Jimenez-Garcia, A., *Development of predictive methods for tiltrotor flows*, Ph.D. thesis, University of Glasgow, 2018.

[149] Jarkowski, M., Woodgate, M., Barakos, G., and Rokicki, J., "Towards consistent hybrid overset mesh methods for rotorcraft CFD," *International Journal for Numerical Methods in Fluids*, Vol. 74, No. 8, 2014, pp. 543–576.

[150] Roache, P. J., "Perspective: a method for uniform reporting of grid refinement studies," 1994.

[151] Roache, P. J., "Verification of codes and calculations," *AIAA journal*, Vol. 36, No. 5, 1998, pp. 696–702.

[152] Bhagwat, M. J. and Leishman, J. G., "Measurements of bound and wake circulation on a helicopter rotor," *Journal of AIRCRAFT*, Vol. 37, No. 2, 2000, pp. 227–234.

[153] Hunt, J. C., Wray, A. A., and Moin, P., "Eddies, streams, and convergence zones in turbulent flows," 1988.

[154] Shaidakov, V., "Helicopter design," *Proceedings of Moscow Aviation Institute*, Vol. 381, 1978, pp. 36–68.

[155] Crozon, C., Steijl, R., and Barakos, G. N., "Numerical study of helicopter rotors in a ship airwake," *Journal of Aircraft*, Vol. 51, No. 6, 2014, pp. 1813–1832.

[156] Kusyumov, A., Mikhailov, S., Romanova, E., Garipov, A., Nikolaev, E., and Barakos, G., "Simulation of flow around isolated helicopter fuselage," *EPJ Web of Conferences*, Vol. 45, EDP Sciences, 2013, p. 01103.

[157] Cavallo, S., Ducci, G., Steil, R., and Barakos, G. N., "CFD Analysis of Helicopter Wakes in Ground Effect," *44th European Rotorcraft Forum, Delft, The Netherlands*, 2018.

[158] Jimenez-Garcia, A., Barakos, G., Treve, V., Rooseleer, F., Cappellazzo, V., and Graham, R., "Helicopter Wake Encounters in the Context of RECAT-EU," *43th European Rotorcraft Forum, Milan*, No. 512, CFAS, 2017.

[159] "Vertipedia," https://vertipedia.vtol.org/, Accessed: 23-09-2019.

[160] Cheeseman, I. and Bennett, W., "The effect of the ground on a helicopter rotor in forward flight," 1955.

[161] CAA, "CAP 490: Manual of Air Traffic Services Part 1," 2015.

[162] Jimenez Garcia, A. and Barakos, G., "Accurate Predictions of Rotor Hover Performance at Low and High Disc Loadings," *Journal of Aircraft*, 2017, pp. 1–22.

[163] Köpp, F., "Wake-vortex characteristics of military-type aircraft measured at Airport Oberpfaffenhofen using the DLR Laser Doppler Anemometer," *Aerospace Science and Technology*, Vol. 3, No. 4, 1999, pp. 191.

[164] Amdahl, G. M., "Validity of the single processor approach to achieving large scale computing capabilities," *Spring Joint Computer Conference*, 1967.

[165] Lawson, S., Woodgate, M., Steijl, R., and Barakos, G., "High performance computing for challenging problems in computational fluid dynamics," *Progress in Aerospace Sciences*, Vol. 52, 2012, pp. 19–29.

[166] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[167] Van Leer, B., "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method," *Journal of Computational Physics*, Vol. 32, No. 1, 1979, pp. 101–136.

[168] Rovere, F., Steijl, R., and Barakos, G. N., "CFD Analysis of a micro rotor in ground effect," *AIAA Scitech Forum, Orlando, Florida, USA*, 2020-1793.

Appendix A

Particle Tracking in Tecplot

A.1 Test case description

In this appendix, Tecplot is used to calculate particle paths using flowfield file(s) obtained using HMB3. The test case used has been described by Lee et al. [26], it consists in a hovering rotor IGE. The particle tracking will be performed over 10 revolutions using solution files that describe half revolution. In this test case a full revolution is completed in 1440 time steps, 4 time steps per 1 degree of azimuth.

A.2 Particle positions

First of all, the flowfield files need to be uploaded in Tecplot. If the flowfield is periodic, it is necessary to upload the minimum number of files for a portion of the full revolution with the desired frequency of files. Figure A.1 shows the option window for uploading the files. In this case are uploaded 9 files, which correspond to half revolution taken during the second revolution performed by the simulation, with a frequency of a new file every 20 degrees.

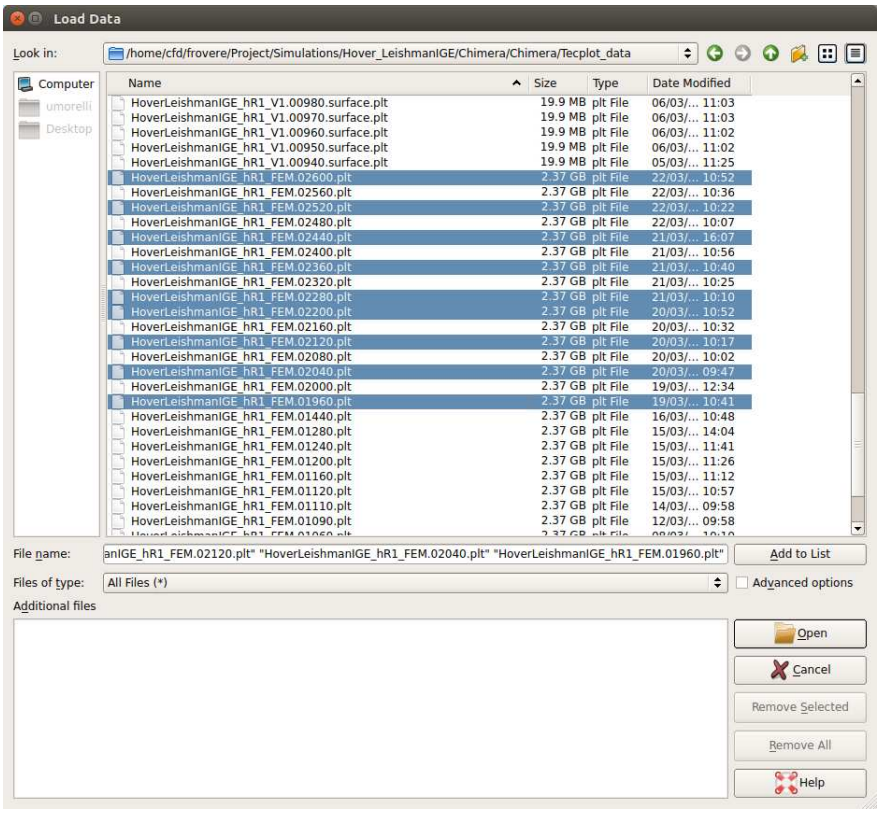


Figure A.1: Uploaded unsteady flowfield data

Once the files are uploaded, it is necessary to give tecplot further instructions. Following the path: *Analyze - Field Variable*. It is necessary to specify the velocity components, and the pressure and density of the flowfield as shown in figure A.2.

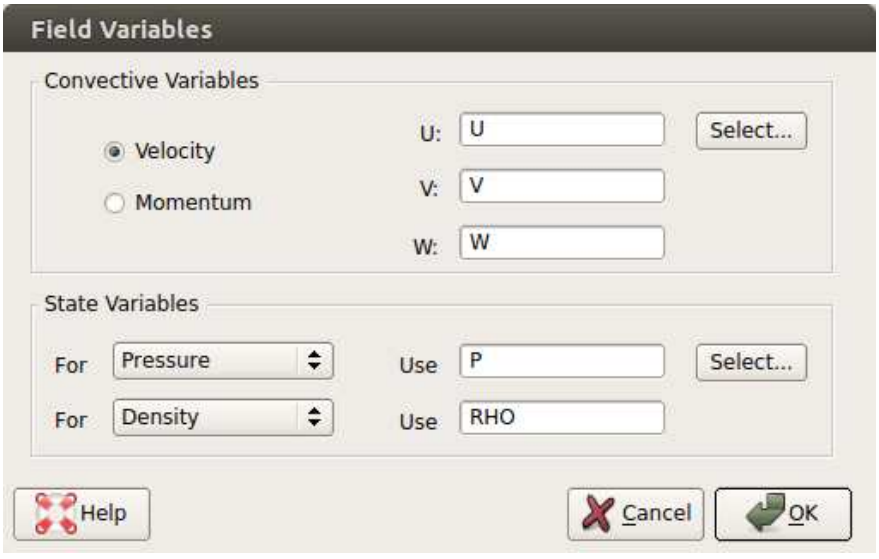


Figure A.2: Setting Field Variables in Tecplot.

The uploaded files consist of different solutions at different time steps, but for Tecplot, they are just zones appended one after the other. It is necessary to arrange the zones in separate time

steps, allowing Tecplot to use the zones the correct time steps when calculating the particle positions. To achieve this, the simulation time and the number of zones for each time step are needed. If the zones are named properly is it possible to see where ”the time step changes” opening the Zone style window, as shown in figure: A.3. When the counter of zones re-starts the solution rappresents a new time step, as highligted in figure A.3.

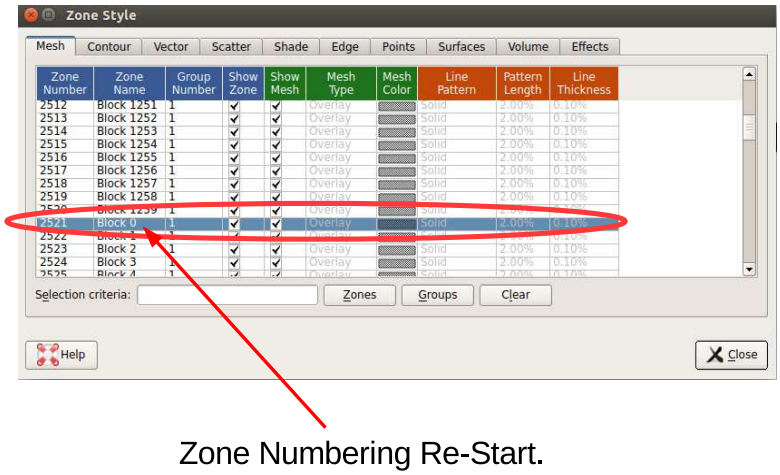


Figure A.3: Numbering of zones per timestep

If the number of zones does not change with the time step the total number of zones uploaded will be:

Total number of zones = (number of zones for time step) (Number of time steps)

When this is done, it is necessary impose the correspondence between time step and zones. If the solution is periodic, flowfield solution files can be re-used to obtain a simulation, that is longer than one revolution. Following the path *Analyze - UnsteadyFlow Options*, it is necessary to remove the steady flow option to access the unsteady flow options. As shown in figure: A.4.

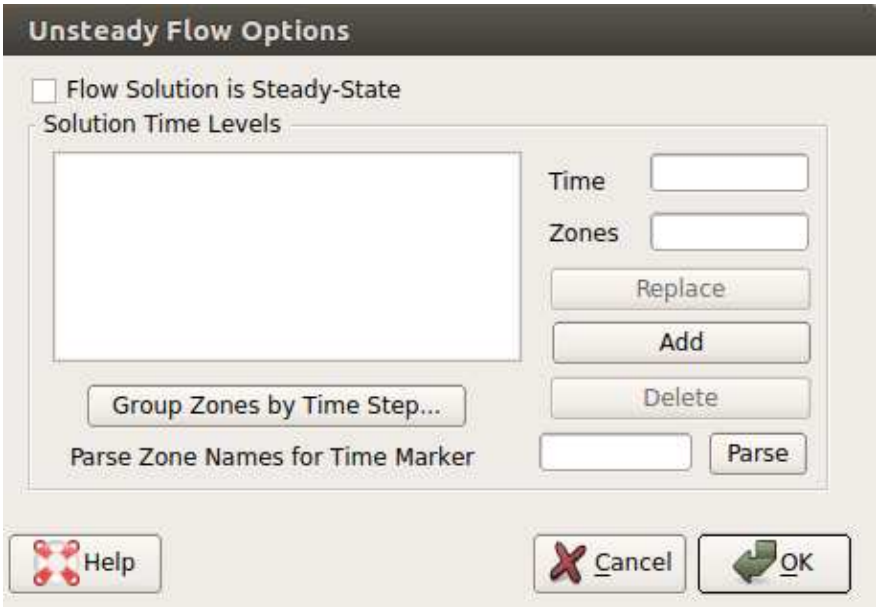


Figure A.4: Unsteady time steps

Using the the buttom *Group zones per time step* the user can generate the timesteps necessary for the unsteady solution and associate them at the proper group of zones, as shown in figure A.5 The *time step* is the timestep between two different flowfield files, (the same in un file for HMB3).

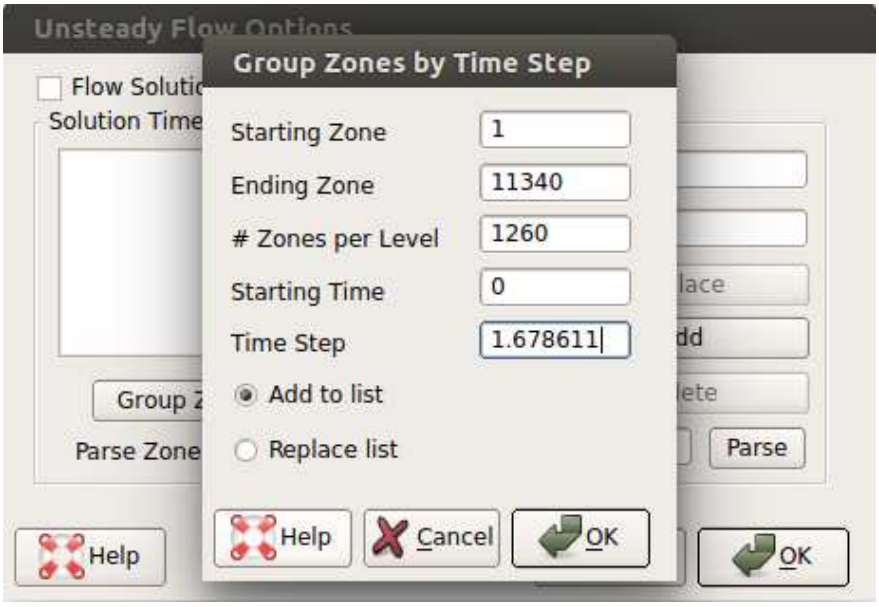


Figure A.5: Generate time steps for half revolution

The first and the second fields are the starting and the ending zones of the unsteady simulation. The latter have to be the last zone of the periodic period considered. In this example, the last zone belongs to the time step of the first half of a revolution. The third field is the number of zones per time step. Finally, Tecplot requires the first starting time and delta time step. Pressing ”ok”, Tecplot creates a number of time step that uses for the uploaded zones, and the period of time for the portion of revolution chosen for the case. (in this example half revolution). To generate a full

revolution or more, it is necessary to repeat this process over the whole desired period updating the starting time. See figure A.6. Every time that this routine is run, time steps for the portion the revolution chosen, are generated and associated with the zones.

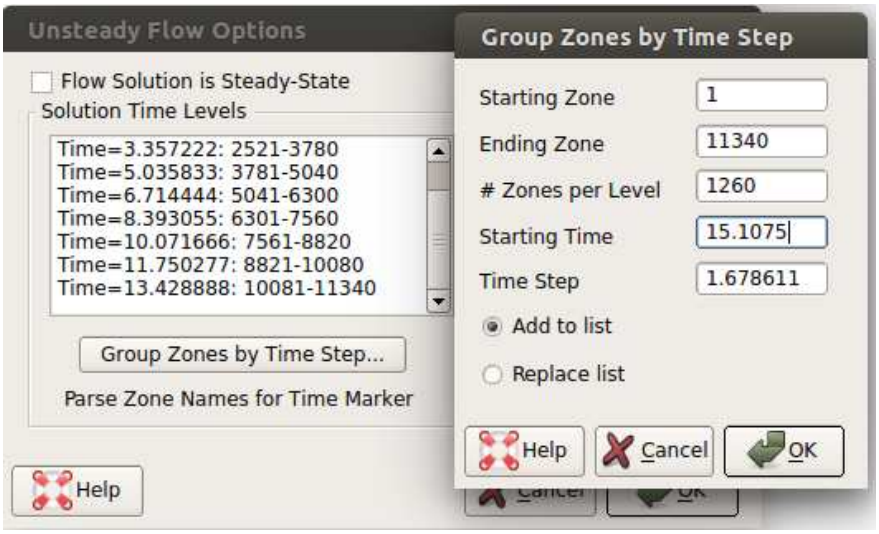


Figure A.6: Generate time step for the full solution

The next step consist generating the starting points of the particles. Selecting *streamtracer* a window appears as shown in figure A.7.

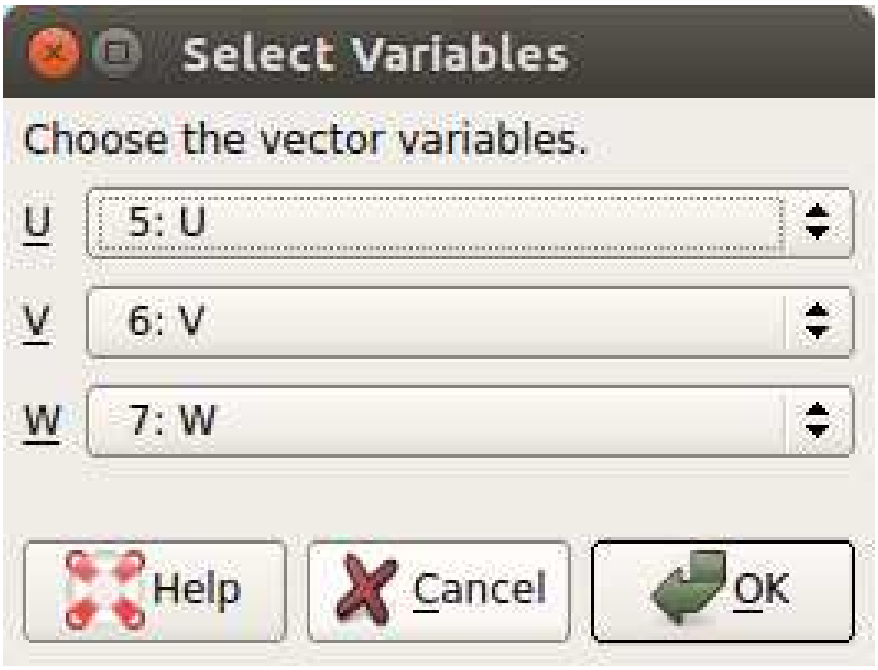


Figure A.7: Velocity components

Using the *By entering X Y Z* option, which also includes the possibility to generate a rake, it is possible to add particles in the domain. In the tab *Timing* it is possible to select *Show Makers* (figure A.9), to visualize the starting points of the particles.

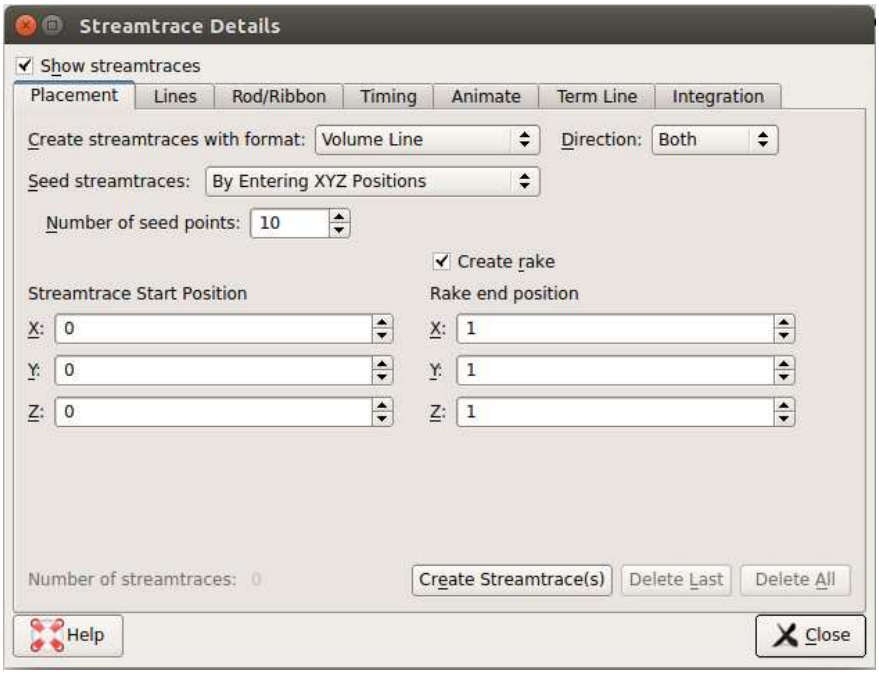


Figure A.8: Streamtracer option window.

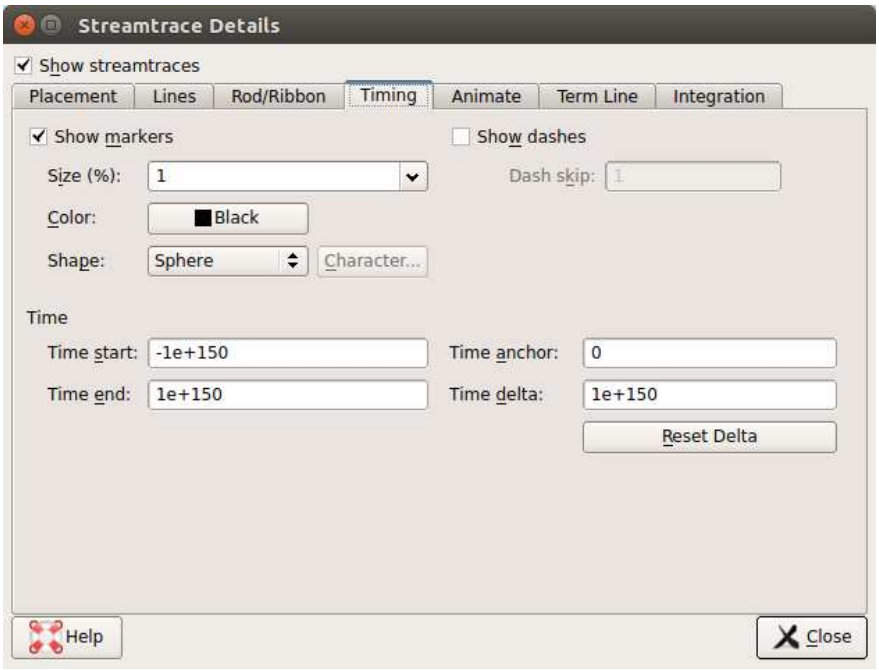


Figure A.9: Markers for streamtracer.

Then, following the path: *Analyze - Calculate particle tracking and Streak lines*, the particle tracking option is found, as shown in figure A.10.

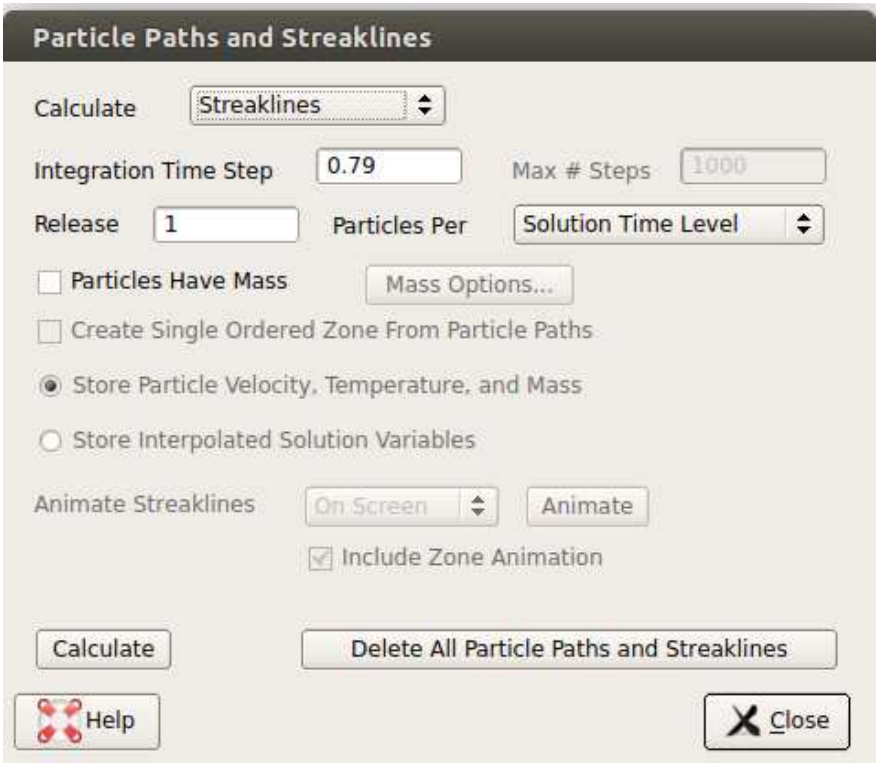


Figure A.10: Streamtracer option window.

Tecplot suggests an integration time step that can be left as is. It is possible to define particles with or without mass, and then the release time. With this option, it is possible set how many particles from every starting point (defined in stream trackers) are released for every solution time level or every solution unit time. Once the particles paths are computed, it is necessary to select the option *scatter* in the plotting option menu. A good way to visualize the particles is coloring them with a specific coordinate (in this case Z) or a velocity field components (figure: A.12). Tecplot generates the particles as zones as showed in figure A.11.

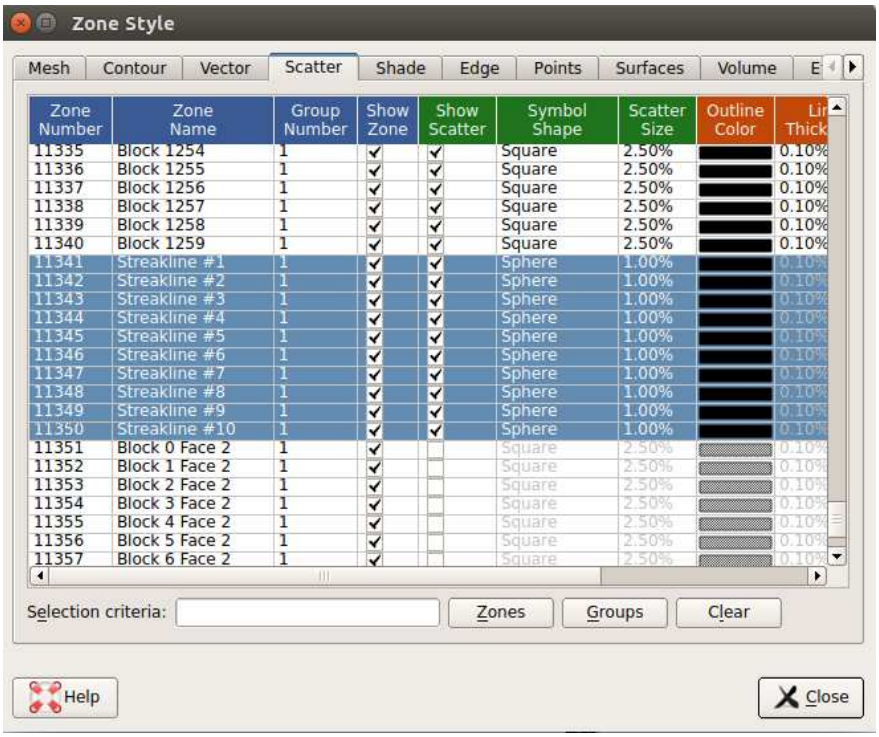


Figure A.11: Particles zone in the zones window.

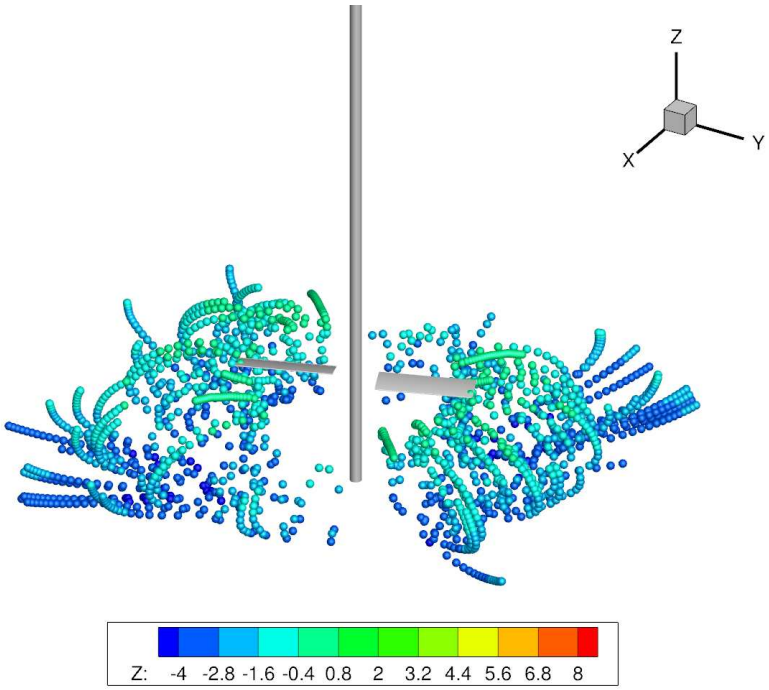


Figure A.12: Visualization.

After this, it is possible animate the particles on the screen or generate a video file using the option *animate*. It is possible to combine the particles animation with an animation done using the

flowfield. The user should pay attention on this possibility, because the two different animations have to start and end at the same time step. Moreover the option that defines the time steps for the animations follow different paths in Tecplot, and for them it is not possible to assign several timesteps at the same solution zones. So it is not possible to produce a periodic animation without duplicating the solution zones.

Appendix B

Particle tracking tool for HMB3

This technical note details the Lagrangian particle tracking tool in HMB3. The Lagrangian particle tracking is a postprocessing tool of CFD solutions and delivers particle paths in steady or unsteady flowfields. The current model includes the effects of particle mass and ballistics. On the other hand, user define scalars are used within HMB3 in steady or unsteady modes.

B.1 Lagrangian numerical model

B.1.1 Massless particles

The particle tracking tool computes the paths of particles with or without mass. If the user imposes $\rho_p = 0$ in the input file for a specific kind of particle, the tool calculates the path for massless particles (and the other particle proprieties are ignored). In the case of massless particles, there is no slip between a particle and the flowfield. In other words, particle and flowfield velocities are equal ($\mathbf{u} = \mathbf{u_p}$), where $\boldsymbol{u}_p$ is particle velocity and $\boldsymbol{u}$ is the velocity of the flowfield at each position of the particle. Particles have first to be localized in the computational domain to obtain the velocity of the flowfield at their positions. Then, to update the particle position it is necessary to integrate the particle motion equation. The integration is performed by a forth order Runge-Kutta method. The position of a particle x_p is computed by:

$$\frac{\partial \mathbf{x}_p}{\partial t} = \mathbf{u}(t_n, \mathbf{x}_p^n), \tag{B.1}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \frac{\Delta t}{6}(K_1^p + 2K_2^p + 2K_3^p + K_4^p), \tag{B.2}$$

where th K_i is the i-th intermediate step. Namely:

$$K_1^p = \mathbf{u}(t_n, \mathbf{x}_p^n), \tag{B.3a}$$

$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2} K_1^p, \tag{B.3b}$$

$$K_2^p = \mathbf{u}(t_n + \frac{\Delta t}{2}, \mathbf{x}_p^{n+\frac{1}{2}}), \tag{B.3c}$$

$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2} K_2^p, \tag{B.3d}$$

$$K_3^p = \mathbf{u}(t_n + \frac{\Delta t}{2}, \mathbf{x}_p^{n+\frac{1}{2}}), \tag{B.3e}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t K_3^p, \tag{B.3f}$$

$$K_4^p = \mathbf{u}(t_n + \Delta t, \mathbf{x}_p^{n+1}), \tag{B.3g}$$

B.1.2 Particles with mass

In the case of particles with mass, the user has to define the relevant proprieties in the input file *st.expert.particles*. The first model equation has been proposed by Kutz [85], and we refer to this with *formulation 0*:

$$\mathbf{a}_p = 0.5\rho_{air}\frac{(\mathbf{u} - \mathbf{u}_p)}{\tau_p} - \mathbf{g}, \tag{B.4}$$

Here, $\boldsymbol{a}_p$ is the acceleration of the particle, $\boldsymbol{u}_p$ is its velocity, $\boldsymbol{u}$ is the velocity of the flowfield in the position of the particle, and $\tau_p = \dfrac{\rho_p d_p}{18\mu_\infty}$ is the relaxation time. This is also solved with the fourth order Runge-Kutta. The user may also opt for *formulation 1* in *st.expert.particles*. The particles are then driven by the flowfield velocities and their positions in time are obtained by integrating their equations of motion like in the previous case. Once more, the integration method used is the fourth order Runge-Kutta, and the equation for particle tracking acceleration is:

$$\mathbf{a}_p = \frac{0.5\rho_{air}(\mathbf{u} - \mathbf{u}_p)||(\mathbf{u} - \mathbf{u}_p)||}{B} - \mathbf{g} \tag{B.5}$$

Here $\boldsymbol{a}_p$ is the acceleration of the particle, $\boldsymbol{u}_p$ is its velocity, $\boldsymbol{u}$ is the velocity of the flowfield in the position of the particle, and B the particle ballistic coefficient, $B = \dfrac{m_p}{S_p C_D}$. Here, m_p is the particle mass, $S_p = \pi d_p^2/4$ is the particle frontal area (particles are assumed spherical), C_D is the particle drag coefficient, and $\mathbf{g}$ is the acceleration of gravity. To integrate the particle acceleration, the fourth order Runge-Kutta method has been used. To obtain the velocity and position of particles at any timestep, it is necessary to integrate two vector equations, one to obtain the velocities and one for the positions:

$$\begin{aligned} \frac{\partial \mathbf{u}_p}{\partial t} &= \mathbf{a}_p(t_n, \mathbf{u}_p^n), \\ \frac{\partial \mathbf{x}_p}{\partial t} &= \mathbf{u}_p(t_n, \mathbf{x}_p^n). \end{aligned} \tag{B.6}$$

Following the Runge-Kutta, method, the solution can be written as:

$$\begin{aligned} \frac{\partial \mathbf{u}_p}{\partial t} &= \mathbf{a}_p(t_n, \mathbf{u}_p^n), \\ \mathbf{u}_p^{n+1} &= \mathbf{u}_p^n + \frac{\Delta t}{6}(K_1^v + 2K_2^v + 2K_3^v + K_4^v), \end{aligned} \tag{B.7}$$

$$\begin{aligned} \frac{\partial \mathbf{x}_p}{\partial t} &= \mathbf{u}_p(t_n, \mathbf{x}_p^n), \\ \mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \Delta t(K_1^p + 2K_2^p + 2K_3^p + K_4^p), \end{aligned} \tag{B.8a}$$

where th K_i is the i-th intermediate step. Namely:

$$K_1^p = \mathbf{u}_p(t_n, \mathbf{x}_p^n), \quad K_1^v = \mathbf{a}_p(t_n, \mathbf{u}_p^n), \tag{B.9a}$$

$$\mathbf{u}_p^{n+\frac{1}{2}} = \mathbf{u}_p^n + \frac{\Delta t}{2} K_1^v, \quad \mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2} K_1^p, \tag{B.9b}$$

$$K_2^p = \mathbf{u}_p(t_n + \frac{\Delta t}{2}, \mathbf{x}_p^{n+\frac{1}{2}}), \quad K_2^v = \mathbf{a}_p(t_n + \frac{\Delta t}{2}, \mathbf{u}_p^{n+\frac{1}{2}}), \tag{B.9c}$$

$$\mathbf{u}_p^{n+\frac{1}{2}} = \mathbf{u}_p^n + \frac{\Delta t}{2} K_2^v, \quad \mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2} K_2^p, \tag{B.9d}$$

$$K_3^p = \mathbf{u}_p(t_n + \frac{\Delta t}{2}, \mathbf{x}_p^{n+\frac{1}{2}}), \quad K_3^v = \mathbf{a}_p(t_n + \frac{\Delta t}{2}, \mathbf{u}_p^{n+\frac{1}{2}}), \tag{B.9e}$$

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^n + \Delta t K_3^v, \quad \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t K_3^p, \tag{B.9f}$$

$$K_4^p = \mathbf{u}_p(t_n + \Delta t, \mathbf{x}_p^{n+1}), \quad K_4^v = \mathbf{a}_p(t_n + \Delta t, \mathbf{u}_p^{n+1}). \tag{B.9g}$$

Once the intermediate timesteps are known, it is necessary to locate the particles in the flowfield, and update the fluid velocities and density values in equation B.5. The search algorithms used for this purpose are described later in this Appendix.

B.2 FlowChart of the tracking methods

The flow chart of the algorithm is presented in figure B.1.

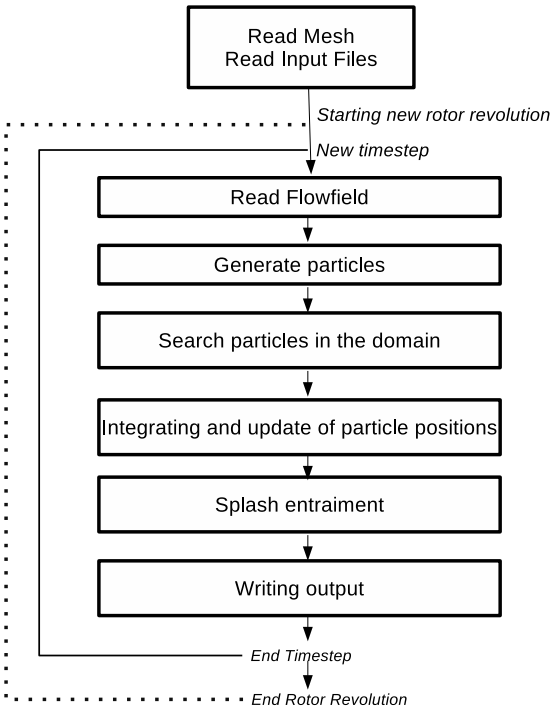


Figure B.1: Particle tracking tool flowchard

First, the method reads the mesh and the input file, which contains information about the particle proprieties (density, radius, drag coefficient), seeding start positions, flowfield information and gravity. Then, it reads the flowfield files, and releases particles in the flowfield. When a new particle is released in the flowfield, it is assigned to a processor. Processors do not exchange information about their particles and, they track their particles from the first timestep to the end of the simulation. After generating particles, every processor searches and updates the position of every particle assigned to it. Finally, every processor writes an output file containing all particle positions at the timestep.

B.3 Splash entraiment

To model bombardment (the process by which particles hit the ground and uplift more particles) a specific methods is used. In the *st.expert.particles* the user may activate the *bomb* flag to 1 in the first line, as listed in code B.1. This allows to generate particles also from the splash entrainment, and not only from the seeding entities. If the bomb flag is set 0, no new particles are emitted. However, the particles still interact with the ground, rebounding. When an entrained particle hits the ground, it may have sufficiently high energy to launch more particles. The kinetic energy gained by the hit particles may overcome the cohesive forces and lead to uplift. Once the particle hits the ground, it rebounds and enters again the flowfield. This phenomenon is called splash entrainment. It has been modeled using a probabilistic approach, proposed in [65] and [85]. Before describing how the splash entrainment works, it is necessary to model the particle wall interaction

using the hard sphere model, which is described in [140]. It is based on the integrated form of the equations of motion, and instantaneous deformations of the particles do not appear in the formulation explicitly. Considering the difference of the momentum between two generic time-steps (a and b). The momentum difference is equal to the impulsive force acting on the particle during that time period.

$$\mathbf{J}^{(b)} = m(\mathbf{u}_p^{(b)} - \mathbf{u}_p^{(a)}) \tag{B.10}$$

Computing the impulsive force acting before and after the collision, it is possible to estimate the post-collisional translation velocities, solving the momentum equations. The hard sphere model introduces the following assumptions: 1) Particle deformations are neglected. During the collision process, the particle radii are constant. 2) Coulomb's friction law applies to particles that are sliding along a wall. Considering three different timesteps: t=0 pre-collision time, t=1 collision time, and t=2 post collision time, as described in figure B.2. It is possible to define two different impulses acting on the particles during impact:

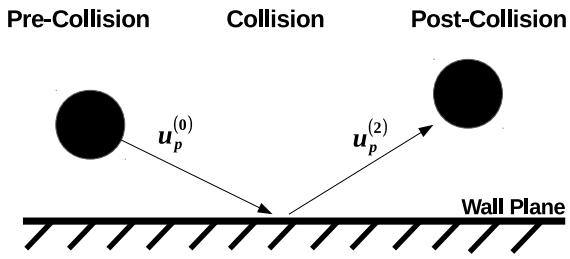


Figure B.2: Schematic representation of particle-wall interaction

$$\mathbf{J}^{(1)} = m(\mathbf{u}_p^{(1)} - \mathbf{u}_p^{(0)}), \tag{B.11a}$$

$$\mathbf{J}^{(2)} = m(\mathbf{u}_p^{(2)} - \mathbf{u}_p^{(1)}). \tag{B.11b}$$

Considering a 3D case, with mass-point particles, and the wall perpendicular to the Z axis. It is possible to define a boundary condition at the wall as:

$$u_{pZ}^{(1)} = 0. \tag{B.12}$$

In [140] the definition of the coefficient of *restitution* e is given by:

$$J_Z^{(2)} = eJ_Z^{(1)}. \tag{B.13}$$

A typical value used is $e = 0.6$. This coefficient represents the loss of momentum due to inelastic effects. From Coulomb's friction law, we have:

$$J_X^{(1)}\mathbf{i} + J_Y^{(1)}\mathbf{j} = -\varepsilon_X f J_Z^{(1)}\mathbf{i} - \varepsilon_Y f J_Z^{(1)}\mathbf{j}, \tag{B.14a}$$

$$J_X^{(2)}\mathbf{i} + J_Y^{(2)}\mathbf{j} = -\varepsilon_X f J_Z^{(2)}\mathbf{i} - \varepsilon_Y f J_Z^{(2)}\mathbf{j}. \tag{B.14b}$$

Here, f is the *coefficient of kinetic friction*, ε_X and ε_Y are factors indicating the proportion of the velocity in each wall plane component direction, defined as

$$\varepsilon_X = \frac{u_{PX}}{\sqrt{u_{pX}^2 + u_{pY}^2}}, \varepsilon_Y = \frac{u_{pY}}{\sqrt{u_{pX}^2 + u_{pY}^2}}. \tag{B.15}$$

From their definition, it is possible to see that $\varepsilon_X^2 + \varepsilon_Y^2 = 1$. In this work, these values are constant during the collision process. In other words: $\varepsilon_X^{(0)} = \varepsilon_X^{(1)} = \varepsilon_X^{(2)}$, and the same for ε_Y. Considering the velocity of an impacting particle, and the friction and restitution coefficients it is possible to solve the system for the 12 unknowns (three components of velocity and three components of the impulse force acting on a particle at times (1) and (2)). The rebound velocity components of the particles are:

$$u_{Rx} = u_{px}^{(0)} + \varepsilon_X f(1+e)u_{pz}^{(0)}, \tag{B.16a}$$

$$u_{Ry} = u_{py}^{(0)} + \varepsilon_Y f(1+e)u_{pz}^{(0)}, \tag{B.16b}$$

$$u_{Rz} = -e u_{pz}^{(0)}. \tag{B.16c}$$

In the case of no emitted particles during the impact ($N_s = 0$), the rebound velocity is the velocity of the rebounding particles, in other words, $\mathbf{u}_R(N_s = 0) = \mathbf{u}_p$. However, if other particles are emitted, the kinetic energy, due to the rebound velocity, is shared among all emitted particles. After computing N_s, using the momentum and energy conservation laws, it is possible to compute the average and standard deviation of particles' velocity. Hu and Shao [107] described the model to compute the number of particles emitted by splash entrainment. The main assumption of their model is the volume of the void-shaped crater created on the ground by an impacting particle. Following their approach, the volume of the crater is equal to the volume of the sand emitted from the bed. The crater volume, V_c, excavated by an impacting particle is a function of the impacting particle mass, diameter, velocity and soil proprieties. V_c can be approximated as:

$$V_c = \frac{\pi \rho_p d_p^3 |\mathbf{u}_p^2|}{12 p_s} \left(sin(2\alpha_0) - 4sin^2(\alpha_0) + \frac{7.5\pi |\mathbf{u}_p| sin^3(\alpha_0)}{\beta_v d_p} \right), \tag{B.17}$$

where α_0 is the angle of the impacting particle velocity and the ground plane, $\beta_v = \sqrt{\frac{2p_s d_p}{m_p}}$, and p_s is the plastic pressure of the soil exerted on the particle general, it depends on the mechanical properties of the sediment bed. In this work, $P_s = 10^3 Nm^{-2}$, following the approach in [141] and [142] and so,

$$N_s = \frac{V_c}{\pi d_p^3/6}. \tag{B.18}$$

The number of particles ejected in this way, may still change considerably with the velocity and angle of the impacting particle. In other works, different types of particles have been taken into account, and results show that N_s may reach high values ($10^5 - 10^6$), when particles of larger diameter (around $d_p = 10\mu m$), eject small diameter particles ($d_p = 1\mu m$) [141]. Using the rebound

velocity defined previously, it is possible to compute the velocity of the emitted particles following a probabilistic approach. The splash-entrained particles are released with a velocity that follows a normal distribution. To compute the average and variance velocities of emitted particles, it is necessary to impose momentum and energy conservation among the emitted particles. The starting velocity of the emitted particle is defined with $\mathbf{u}_S$, and $\mathbf{u}_{Si}$ is the velocity vector of the i-th emitted particle. The relationship between $\mathbf{u}_S$ and $\mathbf{u}_R$ can be obtained using the momentum conservation law,

$$m_R \mathbf{u}_R = \sum_{i=1}^{N_{s+1}} m_i \mathbf{u}_{Si}. \tag{B.19}$$

The multivariate normal distribution for the emitted particle velocity is $\Phi(\mu_B, \Sigma_B)$:

$$\Phi = \frac{1}{\pi^{3/2} det(\Sigma_B)^{3/2}} exp^{-\frac{1}{2}(\mathbf{u}_s - \mu_B)^T \Sigma_B^{-1} (\mathbf{u}_s - \mu_B)}. \tag{B.20}$$

The mean and the covariance matrices of the normal distribution are:

$$\mu_B = \frac{1}{N_s + 1} \sum_{i=1}^{N_s+1} \mathbf{u}_{Si}, \tag{B.21a}$$

$$\Sigma_B = \begin{bmatrix} \sigma_{xx}^2 & 0 & 0 \\ 0 & \sigma_{yy}^2 & 0 \\ 0 & 0 & \sigma_{zz}^2 \end{bmatrix}, \tag{B.21b}$$

where the variance is:

$$\sigma_{ij}^2 = \frac{1}{N_s + 1} \sum_{i=1}^{N_s+1} (\mathbf{u}_{Si} - \mu_B)(\mathbf{u}_{Si} - \mu_B)^T. \tag{B.21c}$$

The mean variance can be estimated using the momentum and energy conservation laws. In general, impacting particles may eject different types of particles with different density and diameter with respect to the impacting particle. However, in this work the particles are assumed to be equal in mass: $m_R = m_i$. Following this assumption, it is possible to obtain:

$$\mu_B = \frac{1}{N_s + 1} \sum_{i=1}^{N_s+1} \mathbf{u}_{Si} = \frac{\mathbf{u}_R}{N_s + 1}. \tag{B.22}$$

Furthermore, it is to assumed that the total kinetic energy of the rebounding particles is conserved, with no extra losses, and it is computed as:

$$E_R = \frac{1}{2} m_R |\mathbf{u}_R|^2 = \frac{1}{2} \sum_{i=1}^{N_s+1} m_i |\mathbf{u}_{Si}|^2. \tag{B.23}$$

Assuming a diagonal covariance matrix, with only σ_{xx}^2,σ_{yy}^2 and σ_{zz}^2 non-zero, and using the equation B.21c, it is possible to obtain the value of the variance from the kinetic energy and the rebound velocity as:

$$\sigma_{xx}^2 = \frac{E_R}{m_R(N_s+1)} \frac{u_{Rx}^2}{(u_{Rx}^2+u_{Ry}^2+u_{Rz}^2)} - \frac{u_{Rx}^2}{(N_s+1)^2},$$
$$\sigma_{yy}^2 = \frac{E_R}{m_R(N_s+1)} \frac{u_{Ry}^2}{(u_{Rx}^2+u_{Ry}^2+u_{Rz}^2)} - \frac{u_{Ry}^2}{(N_s+1)^2}, \tag{B.24}$$
$$\sigma_{zz}^2 = \frac{E_R}{m_R(N_s+1)} \frac{u_{Rz}^2}{(u_{Rx}^2+u_{Ry}^2+u_{Rz}^2)} - \frac{u_{Rz}^2}{(N_s+1)^2}.$$

The splash entrainment model is now closed, using the impacting particle velocity and mass. It is possible therefore to define the number of particles emitted and their initial velocity. This will allow to properly simulate the interaction between the ground and the impacting particle.

B.4 Search methods

During the particle tracking process it is necessary to know the velocity and density values of the flowfield in the space occupied by the particle. In other words, the particles have to be localized inside the computational domain. To do this, search methods have been developed. A "brute force" search method, that is expensive in terms of computational cost, is used only during the first time step to obtain the initial starting particle positions. A new search method, used for faster localization, using the previous positions.

B.4.1 Brute force method

Initially, the method needs to localize the particles inside the computational domain. At the beginning of the simulation, the code finds the initial position of the particles, relating the particle coordinate to the block number , i, j and k indices of the nearest cell centers. The code computes the distance between the particle position and all cell centers of the mesh. This requires a substantial effort if the number of particles is large, or if the mesh is fine. Due to its computational cost this method is used only at the beginning of the simulation, where there are no initial information about the position of particles in the computational domain.

B.4.2 New search method

To avoid the high computational cost, the search area is limited to the neighbor cells of each particle position at time t_n. The search method is described in Figure B.3. The search area (the darkest area in the Figure) is within by the maximum distance travelled by the particle during the Δt between time instances t_n and t_{n+1}. The area is delimited by the maximum range, r_{max}, computed as: $r_{max} = \mathbf{u}_p \Delta t$. A cell, i, is considered in the search area if $r_i < r_{max}$, where r_i is the distance between the cell center $\mathbf{x_i}$, and the position of the particle at t_n, $r_i = ||\mathbf{x}_p^n - \mathbf{x}_i||$. The particle is then searched in the delimited area, comparing the distance between the cell center and the particle position at t_{n+1}. Finally, the particle is assigned to the nearest cell. Once a particle is found, linear interpolation is used to compute the flowfield proprieties at locations of the particle. All cells around the particles are taken into account. The influences of all cells are weighted with respect to their distance. In all simulations, the grids were multi-block structured with hexahedal

cells, so the cells around the particles plus the one that contains the particle, are 27 in total, except for cells near domain boundaries.

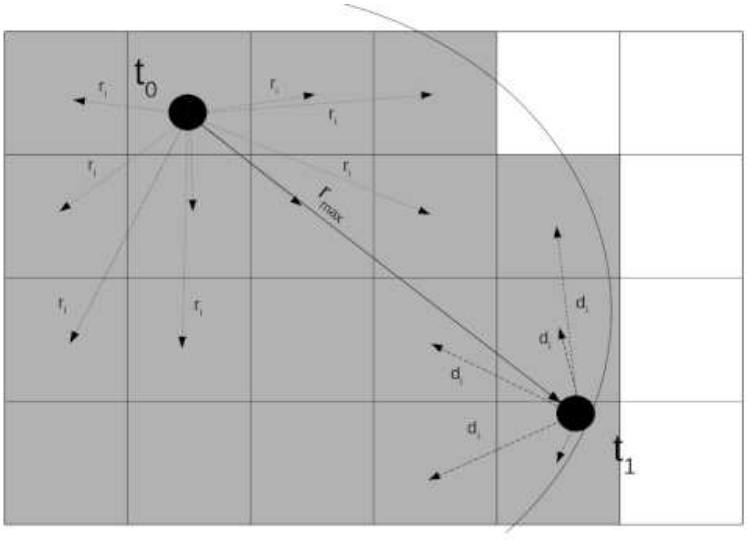


Figure B.3: Search algorithm for particle tracking

The employed equations are:

$$u_{flow}(x_p^{n+1}) = \sum_{i=0}^{27} u_{flow}(x_i) q_i, \tag{B.25a}$$

$$d_i = ||\mathbf{x}_i - \mathbf{x}_p^{n+1}||, \tag{B.25b}$$

$$q_i = \frac{\frac{1}{d_i}}{\sum_{i=0}^{27} \frac{1}{d_i}}, \tag{B.25c}$$

where $u_{flow}(x_p^{n+1})$ is the flowfield velocity at the position occupied by the particle and $u_{flow}(x_i)$ the flow velocity at the center of the i-th cell. $\mathbf{x}_i$ is the position of the i-cell center, while $\mathbf{x}_p^{n+1}$ is the particle position at timestep n+1. Where a particle is near the domain boundaries, the neighbour cells are fewer. In this case, the interpolation is using the "halo cells" of the mesh. Two layers of halo cells are added at the index-boundary of the domain, and these contain the information of the first two layers of cells inside the domain of the neighbor block. This way, it is always possible to interpolate values from 27 cells. When the code runs in parallel every particle is associated with a single processor, which searches only the particle, assigned to it.

B.4.3 Chimera search

The flowfield files contain the iblank information for the overset grids. When the particle is in a portion of a background mesh shared with a foreground one, the iblank value is 0. In this case, the particle is searched in the higher level chimera grids with the brute force method. If the particle is in a portion of the background mesh not shared with the foreground or directly in the foreground

mesh, the iblank value is non 0, and the particles are usually searched with the new method. When the particle is a higher level mesh, and it crosses the chimera boundary (passing a face with the boundary flag 12001), it is searched again with the brute force method in the background mesh. This may slow down a lot the code if particles keep going in and out of the chimera levels.

B.4.4 Check search

In the *st.expert.particles* the user can define a distance tolerance. This tolerance defines the maximum distance allowed between a particle and the nearest cell center. When the distance between the particle and the cell center is greater of this tolerance, the code uses the brute force method to find the particles in the 27 blocks near the original block. This check allows the code to correct the indices of the particles, in case the fast search method did not work properly. It is important to use a distance tolerance which is comparable to the cell size, in the employed grid.

B.5 Input files

B.5.1 st.expert.particles

A sample input file *"st.expert.particles"* is presented here.

Listing B.1: *st.expert.particles* file example

```
n grids | read all (y=1,n=0) | Ncycle | bomb option (y=1,n=0)| dist_tol |
2 0 1 0 3.0
| Grid-path | Re | Mach | model turbo |
/home/jupiter/frovere/EC145_Steady/low_mu/GRID/low_mu_v2.grd 144510.0 0.015 3000
Seeding types (N bc (flags) | N planes | N rakes | N points | N seeding prop)
0 0 390 0 1
Info seeding (see tn)
-17.50 7.0 -22.186679 -17.5 -7.0 -22.186679 3 1 1
particle prop TypeID | density | radius | cl | cd | u_t (all dimensionless)
1 2163.26 0.000011 0.0 1.046 0.0
gravity (x y z component) (dimensionless minus sign already in the code)
0.0 0.0 0.152239
Velocity scaling
1.0 1.0 1.0
Equation
1
Space limits
10000.0 10000.0 -22.30
Chimera levels option (if Ngrid>1 else put 0 or none)
0 1
Chimera levels option (if Ngrid>1 else put 0 or none)
192 764
skipping blocks (N tranches of blocks to skip + tranches to skip)
0
stop seeding revolution
1
ref values length | vel | density
0.410 5.14 1.225
```

The *st.expert.particles* file is used by the particles tracking tool. First of all, the code needs information about the number of grids that the grid file contains. In other words, if the grid file is a chimera case. *Read all* option allows the code to read all the flowfield files at once. Reading all flowfield files in the unsteady case may require a lot of memory, especially for rotor simulations. In the case, of steady simulation, there is only one flowfield file, and the default option (False) is used.

Ncycle allows the code to repeat the same set of flowfield files during simulation. In an unsteady, but the periodic case, it is possible to use the same repetition of flowfield for a longer time than a single period. If the simulation is steady, set the default option (1). The bombardment option is following, which allows creating more particles when the ground is hit by particles. Without this option, particles rebound when hitting the ground, without emitting new particles. This option is available in steady and unsteady modes. The last option, on the first line, is the *disttol*, which indicates the max distance tolerance allowed by the fast searching method. When a particles is found, but the distance between it and the cell center is larger than this threshold, the particle is searched again in all blocks near the present block with the brute force method. Specifically, the particle distance from every cell contained in the 27 blocks is computed. This may slow down the code, and becomes very inefficient if the cells are big. The second line of *st.expert.particles* reads information already contained in the st file. The third line contains the numbers of seeding entities and the number of particles types. First, the seeding surfaces, then in order: number of seeding planes, the number of seeding rakes and the number of seeding points. The last is the number of the types of particles. Starting from the second row, the lines are grouped by the type of seeding entities. The first lines are dedicated to surfaces of the mesh, then after the mesh-surfaces are specified, the user can define the information about the seeding planes, seeding rakes and seeding points. The mesh-surfaces lines are formed by columns, each contains information about the seeding surface. The first column is the flag assigned to the surface, the next three indicate the distributions in the three directions, in other words, the ratio between the particles and the number of nodes of the surface. The fifth column is the frequency for releasing new particles, in other words, it specifies after how many time steps a new particle is released from the same position. The last column gives information about the type of particles, and which proprieties are provided, after all seeding entities are specified. This number is the type ID of particle type. Everywhere, where it is necessary to specify one or more points, the coordinates are put in brackets. The lines dedicated to seeding planes contain the four points which can define a finite plane, and then the distributions of the four rakes that define the plane, the time frequency and the type of particles. The lines dedicated to seeding rakes contain just two points, followed by the distribution of particles over the rake. Again the frequency in time and the ID of particles are the last two columns. The lines dedicated to seeding points contain the coordinates of the point, the frequency and finally, the ID of the type of particle. A scheme of how the seeding entities need to be listed is presented:

Surface (boundary flag)	9999 Np1 Np2 Np3 fr Type
Plane	x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4 Np1 Np2 Np3 Np4 fr Type
Rake	x1 y1 z1 x2 y2 z2 Np fr Type
Point	x y z Np fr Type

Table B.1: Seeding entities input format

The last part of the file *st.expert.particles* contains information about each particles type.

- ρ_s, density of the particle material
- radius, particle radius
- C_l, particle lift coefficient
- C_d, particle drag coefficient

- u_τ. is a parameter for the uplift model

So far, C_l and u_τ are not used. Each line is dedicated to a specific kind of particle and it contains the particle ID number (as specified in the seeding entities), then mass, radius, C_l, C_d and u_τ. Gravity is set in the next line, considering three components. The minus sign is already considered in the code and, the gravity value need to be dimensionaless, $g = \frac{9.81 l_{ref}}{V_{ref}^2}$. It is also possible to define scaling factors for velocity. The parameter multiplies the three velocities component. An extra line is defined for a different equation, the default is 1. Space limits are the limit of the ground boundary. When exceeded, the particles are no more tracked, nor they interact with the ground (if it is modelled as a solid boundary in the grid). Other chimera information needed, includes the chimera levels and the number of blocks that every grid contains. The flowfield file must contain the iblank value, which is used to search for the particles between different chimera levels. The next line allows the user to skip tranches of blocks. The number of trances and the number of blocks are to be given. If the solution is periodic, the user can stop the seeding after the first revolution. Finally, there are the reference values for the flowfield length, velocity and density. They are needed for the bombardment model.

B.5.2 Flowfield files

Reading the flowfield can be the longest part of the particle tracking execution time. A specific version of flowfield files must be generated before running the Lagrangian particle tracking tool. Initially, a Tecplot macro is needed. *"extract_dat.mcr"*, which extracts the .dat version of the HMB Binary file. In this file (.dat), are only the three coordinates, the three velocity components, the flow density and the iblank variable are stored.

Listing B.2: Sample input "extract_dat.mcr" file.

```
$!VarSet |MFBD| = './'
$!VarSet |start| = 145000
$!VarSet |delta| = 10
$!VarSet |nb| = 1
$!PAGECONTROL CREATE
$!LOOP |nb|
$!VarSet |step| = (|start|+((|loop|-1)*|delta|))
$!READDATASET '"./Flowfield_Binary.|step\%5.5d|.plt" '
  READDATAOPTION = NEW
  RESETSTYLE = YES
  VARLOADMODE = BYNAME
  ASSIGNSTRANDIDS = YES
  VARNAMELIST = '"X" "Y" "Z" "RHO" "U" "V" "W" "P" "K" "O" "Ret" "iBlank"'
$!AlterData
  ValueLocation = Nodal
  Equation = '{iblank_v}={iblank}'
$!WriteDataSet " ./ FlowfieldBinary .| step\%5.5d|.dat"
  IncludeText = No
  IncludeGeom = No
  IncludeCustomLabels = No
  VarList = [1-7,13]
  Binary = No
  UsePointFormat = Yes
  Precision = 9
  TecplotVersionToWrite = TecplotCurrent
$!ENDLOOP
$!RemoveVar |MFBD|
```

Once the dat file is obtained, a c script needs to be run to obtain the flowfield binary file. The new file contains an integer which is the total number of points in the flowfield, then the flowfield variables x,y,z,u,v,w,ρ and iblank. In the case of non chimera mesh, iblank must to be set to 1.

Listing B.3: Sample input "create_ff" file.

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <math.h>
#include <assert.h>
#include <time.h>
struct block
{
        int number,nx,ny,nz;
};
void skip_this_line (FILE *fp);
int main(int arg_count, char *argv[]) {
  if (arg_count !=6) {
        printf (" Incorrect number of parameters: \\n <file root> <first timestep> <last timestep> <delta> <topo file>\\n");
        exit (0);
  }
  FILE *output,* input ,* topFile ;
  int nblocks,bs,ib,t,t_max, start_file , end_file , delta_file , new_file ,nx,ny,nz,nx1,ny1,nz1,i,j,k,index,nmax,index0;
  char filename [300], filename_o [300],append[7];
  double r1,r2,r3,i1,i2,i3,i4,i5,*data;
  int * data_int ,nmax_2,iskip;
  size_t result ;
  struct block *blocks;
  bs=7+1; /* number of equations */
  topFile =fopen(argv [5],"r");
  nmax=0;
  nmax_2=0;
  data_int =calloc (1, sizeof (int));
  result = fscanf ( topFile ,"\%d\\n",&nblocks);
  if ( result != 1) { fputs ("Reading error : number of blocks\\n",stderr); return (-1);}
  blocks=calloc (nblocks, sizeof ( struct block));
  for (ib=0;ib<nblocks;ib++){
     result = fscanf ( topFile ,"\%d \%d \%d\\n",&nx,&ny,&nz);
     blocks[ib].nx=nx;
     blocks[ib].ny=ny;
     blocks[ib].nz=nz;
     nmax+=(nx+2)*(ny+2)*(nz+2)*bs;
     nmax_2+=(nx)*(ny)*(nz)*bs;
          }
     index0=0;
      start_file =atoi (argv [2]);
     end_file =atoi (argv [3]);
      delta_file =atoi (argv [4]);
     data_int [0]=nmax;
     t_max=( end_file - start_file )/ delta_file ;

     for (t=0;t<t_max+1;t++){
              strcpy ( filename , argv [1]);
              strcpy ( filename_o , argv [1]);
              new_file = start_file +( delta_file *(t));
              sprintf (append,"\%05d",new_file);
              strcat ( filename , ".");
              strcat ( filename , append);
              strcat ( filename , ".dat");
              strcat ( filename_o , ".");
              strcat ( filename_o , append);
              strcat ( filename_o , ".plt");
              printf ("t:\%d reading this file :\%s\\n",t,filename);
              input=fopen(filename,"r");
              if (! input ) { printf ("Cannot find \%s\\n",filename); exit (0);}
              output=fopen(filename_o,"w");
         for( iskip=0;iskip<9;iskip++) skip_this_line ( input );
              data=calloc (nmax,sizeof(double));
              fwrite ( data_int , sizeof ( int ) ,1, output );
              index0=0;
              for(ib=0;ib<nblocks;ib++){
                      for( iskip==0;iskip<5;iskip++) skip_this_line ( input );
                      nx=blocks[ib].nx;
                      ny=blocks[ib].ny;
                      nz=blocks[ib].nz;
                      nx1=nx+2;
                      ny1=ny+2;
                      nz1=nz+2;
                              for(k=0;k<nz1;k++){
                              for(j=0;j<ny1;j++){
                              for(i=0;i<nx1;i++){
                                      /* read in points */
                                      if ( fscanf ( input ,"\%lf \%lf \%lf \%lf \%lf \%lf \%lf \%lf\\n",&r1,&r2,&r3,&i1,&i2,&i3,&i4,&i5) != 8) {
                                              fprintf ( stdout ,"Error Reading the flowfield , ibl:\%d i:\%d,j:\%d,k:\%d\\n",ib,i,j,k);
                                              exit (-123);
                                      }
                                      index=(ny1*(i+nx1*k)+j)*bs;
                                      data[index0+index]=r1;
                                      data[index0+index+1]=r2;
                                      data[index0+index+2]=r3;
                                      data[index0+index+3]=i2;
                                      data[index0+index+4]=i3;
                                      data[index0+index+5]=i4;
                                      data[index0+index+6]=i1;
                                      data[index0+index+7]=i5;

                                      if (data[index0+index+6]==0.0){ printf ("rho zero!\\n"); exit (0);}
                              }
                              }
```

```
                                }
                        index0+=(nx+2)*(ny+2)*(nz+2)*bs;
                    }
            fwrite(data,sizeof(double),nmax,output);
            printf("End writing\\n");
            printf("Closed input\\n");
            fclose(output);
            printf("Closed output\\n");
            free(data);
            printf("ended timestep:\%d\\n",t);
            }
free(blocks);
}
void skip_this_line (FILE *fp)
{
        char dummy;
        while( fscanf(fp,"\%c",&dummy) && dummy!='\\n'){}
}
```

B.6 Usage of the particle tracking tool

The particle tracking code can be used in steady and unsteady mode. To run the **steady** mode the command is:

```
particle_tracking <flowfield root> <Flowfield timestep> <N iterations> <Steady delta t> <none>
```

The flowfield file used for the steady simulation has to be: <flowfield root> . <Flowfield timestep>.plt, generated as described in the flowfield section. <N iterations> is the number of steady iterations that the code must perform, and <Steady delta t> is the steady timestep. It is important to emphasize that steady particle tracking, means that the flowfield is fixed and does not change, however the particle tracking simulation is unsteady. The total dimensionless time of the simulation is : <N iterations> $\times$ <Steady delta t>. The particle tracking results are the final position traveled by the seeding particles during the total amount of time due to the steady flowfield. For the **unsteady** cases the command line is:

```
particle_tracking <flowfield root> <first timestep> <last timestep> <delta time step> <un>
```

In this case, the flowfield changes with the time step. As for the steady case, the flowfield files root need to be provided, and the first and the last timestep of the simulation. <delta time step> is the step between flowfield files. The last input file is the *un* file used for the HMB simulation. In the case the user added the $Ncyles > 1$ in the input file *st.expert.particles*, the code will loop the flowfield files listed in that line.

B.6.1 Output of the Particle tracking tool

When the code finds the particles for the first time, every processor prints a file containing the particles initial position. The output files are called particles_position.seedings.0000000.dat.XXXXX, where XXXXX is the number of the processor. Every processors saves only the positions of the particles assigned to it, so to obtain the full seeding file it is necessary to concatenate the files using:

```
cat particles_position.seedings.0000000.dat.0* > particles_position.seedings.0000000.dat
```

The head of the obtained output file is:

```
VARIABLES = "timestep" "x" "y" "z" "u" "v" "w" "u_flow" "v_flow" "w_flow" "dist" "Particle_ID"
     "Seeding_ID" "Seeding_ID_type" "i" "j" "k" "ibl" "bomb" "proc"
```

The first variable stored in the output file is the timestep, followed by particle coordinates and velocities components. Later the flowfield velocities of the cells where the particle is assigned and stored. Last float number is the distance of the particles position and the cell center. Particle ID, the number of particle. Then, the seeding ID and the types are given. Finally, the positions of the particles in the computational domain are registered, i,j,k are the indices of the cell contained in the block ibl. If the particles has been generated by the bombardment "bomb" value is 1, and also the processor number is registered. The output variables are listed in table B.2. After this, it is possible

timestep	output timestep
x,y,z	particle position coordinates
u,v,w	particles velocities components
u_flow,v_flow,w_flow	flow velocities components at the particle position
dist	distance between the particle position and nearest cell center
Particle_ID	Particle Number
Seeding_ID	Seeding entities Number that generated the particle
Seeding_ID_type	Seeding entities type that generated the particle (surface, plane etc.)
ibl,i,j,k	Block number and cell indices of the particle's nearest cell center
bomb	if bomb=1 particle created by bombardment, otherwise 0.
proc	processor that tracked the particle

Table B.2: Variables contained in the particle tracking tool output files

to preplot them, using the executable in the hmb/bin folder. In steady mode the code prints the output file at the end, with the full path of every particles. *particles_position.YYYYYYY.dat.XXXXX*, where YYYYYYY is the final timestep. Like before it is necessary to concatenate the files together, and preplot them, before opening with Tecplot. In the case of unsteady flow, the code prints an output at every time step. Output file contains the timestep, x,y,z and u,v,w of the particle. Then are stored the three velocity components of the flow at the position of the particle. The last float number is the distance of the particle position, with the cell center assigned to it. Finally, extra information regarding the particles and the computational domain is stored. First are the particle ID and Seeding ID. The univocal number assigned to the particle, and the single seeding entity (surface, plane, rake or point). The Seeding_ID_type is also defined (0 for surfaces, 1 for planes, 2 for rakes and 3 for points). After this, the indices and the block number of the closest cell to the particle position follow. Finally, the "bomb" variable defines the particle has been created by bombardment effect or not. (bomb=0 no, bomb=1 yes). The last integer is the processor number that computed the particle path.

B.6.2 Tacplot and Lagrangian tool comparison

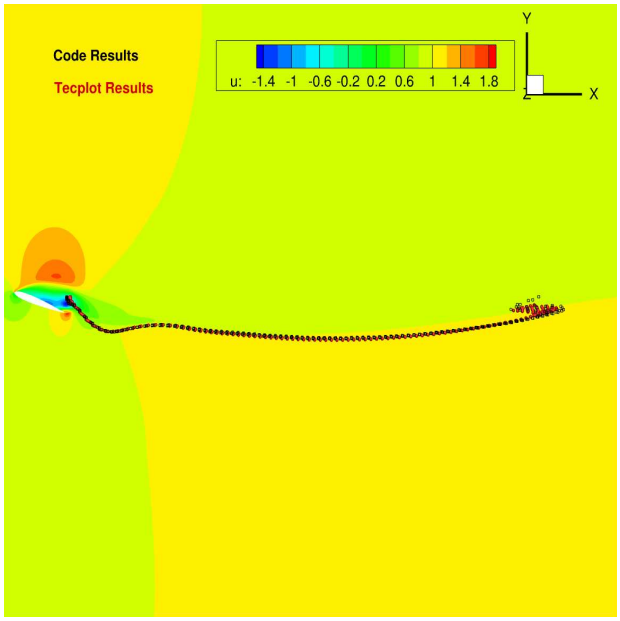
Tecplot and Lagrangian tool comparison

A direct comparison between Lagrangian particle tracking and Tecplot particle tracking is performed. Tests are performed for mass particles, using a pitching NACA0012 airfoil. Geometrical and operational conditions of the airfoil are presented in tab: B.3. Particles are released in proximity of the trailing edge. Results shown in figure B.4 suggest that particles follow the wake of the airfoil. It is possible to notice an concentration of particles, far from the airfoil. New particles are

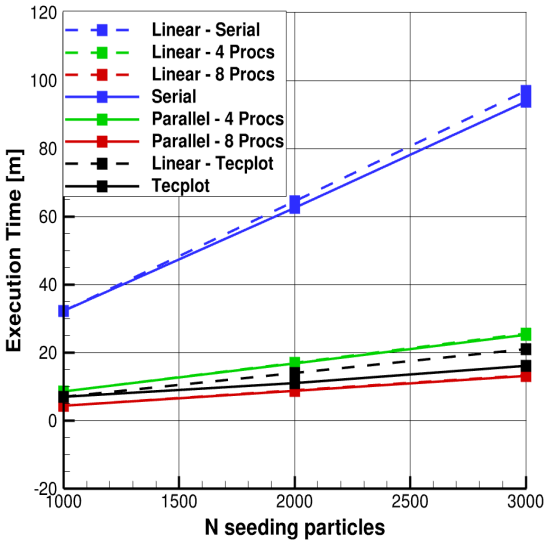
released every timestep, when particles are released in the flowfield during the stall, their starting X velocity is negative, influenced by the recirculation on the upper part of the airfoil. In these few timesteps, particles recirculate near the airfoil, and agglomerate, however when the stall condition stops, all the particles are convected away. Tecplot and particle tracking agree in particle paths result, showing a very close behavior. In figure B.4 (B), the total execution time between the two methods is given. Tecplot shows good timing performance, with a small increase in the execution time with the increase of particles inside the flowfield. The particle method shows better timing performance only for a high number of processors used.

Freestream Mach number	0.38
Reynolds number	3.8e6

Table B.3: Airfoil condition.



(a) Comparison results Naca



(b) Total Execution time Naca

Figure B.4: Total time for particle tracking

B.6.3 Particle tracking examples

Steady flow - Rotor in taxiing

A test case for rotor in taxiing is now presented to give an example to how to run the Lagrangian particle tracking in steady mode. In the next section, the Lagrangian particle tracking is demonstrated for an unsteady case. The test case consists in a rotor in forward flight, the main geometrical and operational conditions are listed in table B.4. For this test case, the rotor is approximated as an actuator disk. The mesh does not contain a fuselage.

V_∞ $[kts]$	10
V_∞ $[m/s]$	5.14
M_∞	0.0151
Re	144'500
μ	0.0233
N blades	4
R [m]	5.5
c [m]	0.41
DL $[kg/m^2]$	370
C_T	0.0125

Table B.4: Forward flight conditions and rotor geometric data.

As described previously, the Lagrangian particle tracking is a post-processing tool. For this reason, initially, it is necessary to run HMB3, to obtain a suitable flowfield solution. For the single phase simulation, the necessary files are listed below.

- st
- st.expert.actuator_disk
- st.expert.turbulence_model
- st.expert.output
- st.expert.freestream
- solids

Listing B.4: *st* file for rotor in steady forward flight.

```
File root       : rotor_ff
Mesh root       : Grid/bg.grd

Incidence       : 0.0
Sideslip        : 0.0
Mach number     : 0.0151
Reynolds number : 144510

Explicit steps  : 10
Explicit cfl    : 0.4
Implicit steps  : 100000
Implicit cfl    : 2.0
Convergence     : 1.0e-8

Model           : 3000
```

Listing B.5: *st* file for rotor in steady forward flight.

```
x-velocity                  : 1.0
y-velocity                  : 0.0
z-velocity                  : 0.0
```

Listing B.6: *st.expert.actuator_disk* file for rotor in steady forward flight.

```
Steady/Unsteady          : 0
Number of Actuator Disks : 1

Disk Model               : 1
Number of blades         : 4
Ct                       : 0.00625
Radius                   : 13.41
Chord                    : 1.0
Cutout                   : 3.53
Mu                       : 0.0233
Omega                    : 0.79688
Centre of Disk           : 0.0 0.0 0.0
Disk tilt roll and yaw   : -3.0 0.0 0.0
Tolerance                : 0.80
Solidity                 : 0.095

Print Gaussian (0/1)     : 0
Chimera level            : 0
Boundary Condition       : -1
```

Listing B.7: *st.expert.output* file for rotor in steady forward flight.

```
grid reference length                    : 1.0
reference area                           : 1.0
reference length for moment about x-axis : 1.0
reference length for moment about y-axis : 1.0
reference length for moment about z-axis : 1.0
location on grid of moment centre (x)    : 0.0
location on grid of moment centre (y)    : 0.0
location on grid of moment centre (z)    : 0.0
Output convergence of steady loads/output: 1
Output frequency for steady solutions    : 5000
Output frequency for steady loads        : 500
Output maximum cell residuals (0/1)      : 0
Output cell centre residuals (0/1)       : 0
Output cell centre solution  (0/1)       : 0
Output cell vertex solution  (0/1)       : 0
Output cell vertex FEM solution (0/1)    : 0
Output cell centre FEM solution (0/1)    : 0
Output cell centre gradients (0/1)       : 0
Output cell centre surface (0/1)         : 0
Output cell centre FEM surface (0/1)     : 0
Output cell centre viscous surface (0/1) : 1
Output cell centre SAS solution (0/1)    : 0
Output cell centre DES solution (0/1)    : 0
Output cell centre SA  solution (0/1)    : 0
Binary Output (0/1) CC Chimera Halo      : 1 0 0 1
Output surface only with RHO and P (0/1) : 0
Blocks to write starting from 1          : 0
```

Listing B.8: *st.expert.turbulence_model* file for rotor in steady forward flight.

```
--------------------
Two-equation models:
--------------------
Mach number correction : 0
Vortex limiter         : 0
Sustain ambient values : 0
Transition modelling   : 0
Roughness              : 0
```

```
SST blending           : 1
SST clip               : 1
Non-Linear terms       : 0
SAS clip               : 0
limit SAS mu_t         : 0
Quadratic stress       : 0
Cubic stress           : 0
DES/LNS clip           : 0
Cubic root volume DES  : 0
DES model              : 0
DDES model             : 0
IDDES model            : 0
R/C correction         : 0
R/C scale              : 1.0
Apply Pk Clip          : 0

New 2eq Model          : 1

--------------------
One-equation models:
--------------------
DES model   : 0
DDES model  : 0
IDDES model : 0
SALSA model : 0
EDW   model : 0
DES-CDES    : 0.65
Secundov correction    : 0
R/C correction         : 0

-----------------
Algebraic models:
-----------------
Smagorinsky constant : 0.18
```

Before run HMB3, few preparation steps are needed. The first is loadbalance. HMB3 is a multi-block fluid solver. Every block of the grid is assigned to a single processor, which revolve the fluid equations. *loadbalance* assign every block to a single processors. To run *loadbalance* use the following command:

```
loadbalance st X
```

Where X, is the number of processors that it will be used for the simulation. The second routine that it is necessary to launch HMB3, is pre-grid. This routine split the grid file, in single-block files. To use pregrid, use the following command:

```
pre_grid_parallel_GCC_4.5.1_OpenMPI_1.3.3_19-10-18 Grid/bg.grd Grid/solids none
```

solids file, contains the boundary flag used in the grid to identify the solid walls. For this test case, the *solids* file is listed below.

Listing B.9: *solids* file for rotor in steady forward flight.

```
1
110000
```

The test case used does not contain a fuselage, and the only solid wall is the ground. As said in the beginning, the rotor, approximated as actuator disk, is operating IGE. The boundary flag *110000* indicates a moving solid wall, and its velocity is the same defined in *st.expert.freestream*. Once that these preliminary routines have been used, it is possible to run HMB3. The command to use is:

```
mpiun -np X hmb_parallel_method3_3.0.0_COUPLED_GCC_... st none none none
```

Once the single-phase simulation is done, output files need to be concatenated. HMB3 writes sigle-block files. To use the particle tracking tool, the Binary output is necessary. The command to use is:

```
multi_Binary rotor_ff 100000 100000 1 X 1
```

Where X, is again the number of processors used in the simulation. The simulation is steady, and the particle tracking simulation will use a single flowfield file. Now, it is necessary to re-write the flowfield files in the format suitable for the particle tracking tool. In the previous section the Tecplot macro and the c code necessary for this were described. To run the macro, it may be necessary to change few information contained in the head of the file, like file root and iteration, while for the c code this is not necessary. To obtain the flowfield file two commands are necessary.

```
tec360 -b extract_dat.mcr
```

```
./create_ff rotor_ff_Binary 100000 100000 1 Grid/bg.grd.top
```

These two commands write two new files, a .dat and a .plt file. Only the .plt is used by the tool, the .dat file can be deleted (it is an ASCII version of the .plt file, than is note useful anymore). To properly run the particle tracking tool, the *st.expert.particles* file is necessary. For this test case, this is listed below.

Listing B.10: *st.expert.particles* file for rotor in steady forward flight

```
n grids | read all (y=1,n=0) | Ncycle | bomb option (y=1,n=0)| dist_tol |
1 0 1 0 10.0
| Grid-path | Re | Mach | model turbo |
background.grd 144510.0 0.015 3000
Seeding types (N bc (flags) | N planes | N rakes | N points | N seeding prop)
0 0 17 0 1
Info seeding (see tn)
-17.50 7.000 -22.18 -17.50 -7.000 -22.18 3 1 1
-15.00 10.72 -22.18 -15.00 -10.72 -22.18 5 1 1
-12.60 14.45 -22.18 -12.60 -14.45 -22.18 7 1 1
-10.25 18.18 -22.18 -10.25 -18.18 -22.18 9 1 1
-7.833 21.91 -22.18 -7.833 -21.91 -22.18 10 1 1
-5.416 25.64 -22.18 -5.416 -25.64 -22.18 12 1 1
-3.000 29.37 -22.18 -3.000 -29.37 -22.18 14 1 1
-0.583 33.10 -22.18 -0.583 -33.10 -22.18 16 1 1
 1.833 34.73 -22.18  1.833 -34.73 -22.18 17 1 1
 4.250 35.70 -22.18  4.250 -35.70 -22.18 17 1 1
 6.666 36.66 -22.18  6.666 -36.66 -22.18 18 1 1
 9.083 37.63 -22.18  9.083 -37.63 -22.18 18 1 1
 11.50 38.60 -22.18  11.50 -38.60 -22.18 19 1 1
 13.91 39.56 -22.18  13.91 -39.56 -22.18 19 1 1
 16.33 40.53 -22.18  16.33 -40.53 -22.18 20 1 1
 18.75 41.50 -22.18  18.75 -41.50 -22.18 20 1 1
 21.16 41.60 -22.18  21.16 -41.60 -22.18 20 1 1
particle prop TypeID | density | radius | cl | cd | u_t (all dimensionless)
1 2163.2 0.000011 0.0 1.046 0.0
gravity (x y z component) (dimensionless minus sign already in the code)
0.0 0.0 0.1522
Velocity scaling
1.0 1.00 1.00
Equation
1
Space limits
10000.0 10000.0 -22.30
```

```
Chimera levels option (if Ngrid>1 else put 0 or none)
none
Chimera levels option (if Ngrid>1 else put 0 or none)
none
skipping blocks (N tranches of blocks to skip + tranches to skip)
0
stop seeding revolution
1
ref values length | vel | density
0.41 5.14 1.22500
```

For this test case, the flowfield is seeded with 17 rakes in proximity of the ground. In this case, the particles have mass (otherwise the density had to be set to 0.). Particle soil material density can be $\rho_p^* = 2650[kg/m^3]$ and radius can be settled to $r_p^* = 4.5 \cdot 10^{-6}$ [33]. These values and gravity were adimenzionalized using reference values defined in table B.4 as following:

$$\rho_p = \frac{\rho_p^*}{\rho_{ref}} = \frac{2650}{1.255} = 2160, \quad r_p = \frac{r_p^*}{L_{ref}} = 0.000011, \quad g = \frac{9.81 L_{ref}}{V_{ref}^2} = 0.1522 \tag{B.26}$$

In the mesh, the ground is at Z=-22.3, which define the Z_{min}, and the limit of the domain. The chimera options are not used for this test case, and neither the *stop seeding revolution*. Full information on specific entries in the file, are described in the previous chapter.

To run the particle tracking code, it is necessary to use the following command:

```
mpirun -np X ./particle_tracking_parallel rotor_ff_Binary 100000 15000 0.05 none
```

In this case, it is supposed that the single phase solution performed 100000 iterations, but this may be different for other cases. The code performs 15000 steady iterations, with a $\Delta t = 0.05$. For a steady simulation, the *un* file it is not necessary. When the code finds the particles for the first time, every processor writes a file contains the particles initial position. The output files are named *particles_position.seedings.0000000.dat.XXXXX*, where XXXXX is the number of the processor. Every processor saves only the particles position assigned to it. To obtain the full seeding file, it is necessary to concatenate the files, using:

```
cat particles_position.seedings.0000000.dat.0* > particles_position.seedings.0000000.dat
```

For a steady simulation, the code only writes the last iteration. Like in the previous case, the output files are *particles_position.0015000.dat.XXXXX*. To obtain the final output file, these files need to be concatenated, and using preplot_hmb, they can be converted into binary files.

```
cat particles_position.0015000.dat.0* > particles_position.0015000.dat

hmb/bin/preplot_hmb particles_position.0015000.dat
```

Now they can be plotted, using Tecplot and visualizated. Considering R/c=13.41, it is possible to obtain a top view, plotting the cloud on x/R,y/R and z/R. These values can be computed, starting from the x,y,z variables contained in *particles_position.YYYYYYY.plt* as shown in figures B.5 (go to $Data -> Alter$) and then imposed ad plotting coordinates in $Plot -> Axis\ XYZ$ B.6.

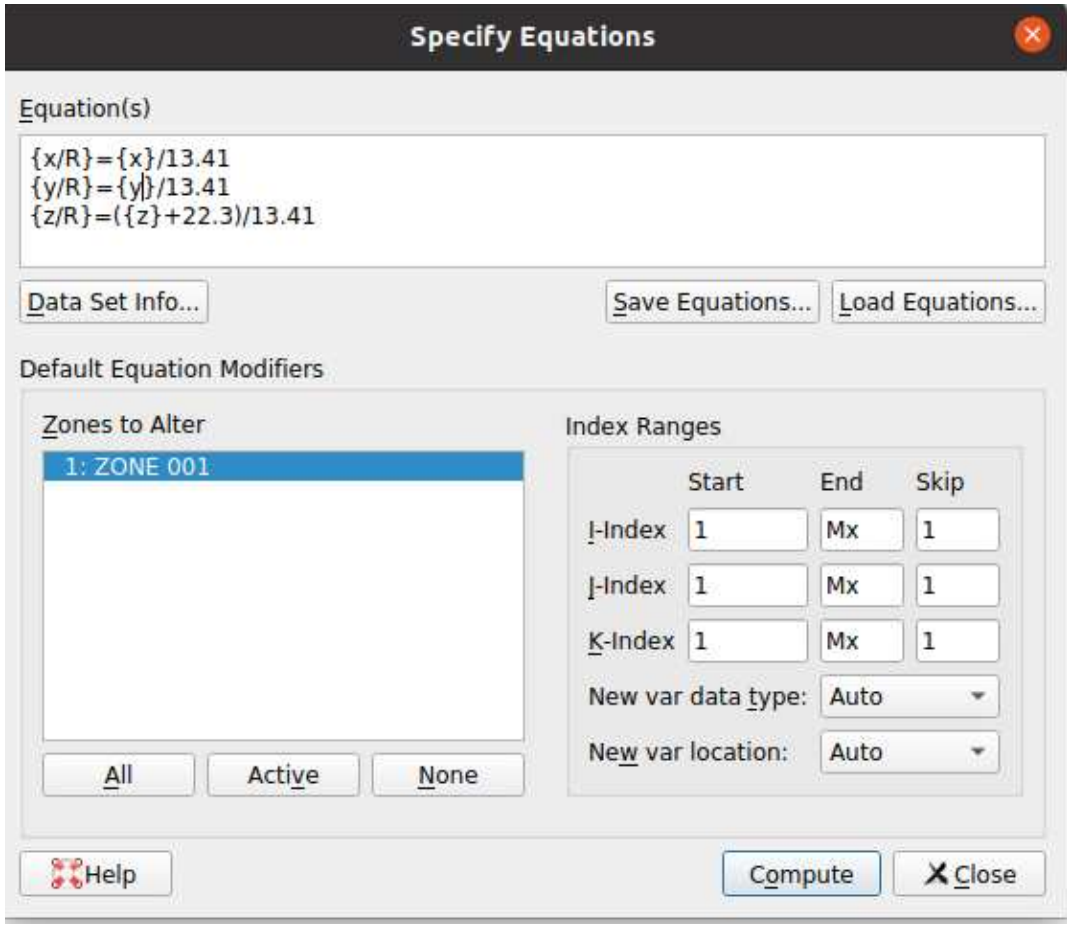


Figure B.5: Computing normalized coordinates.

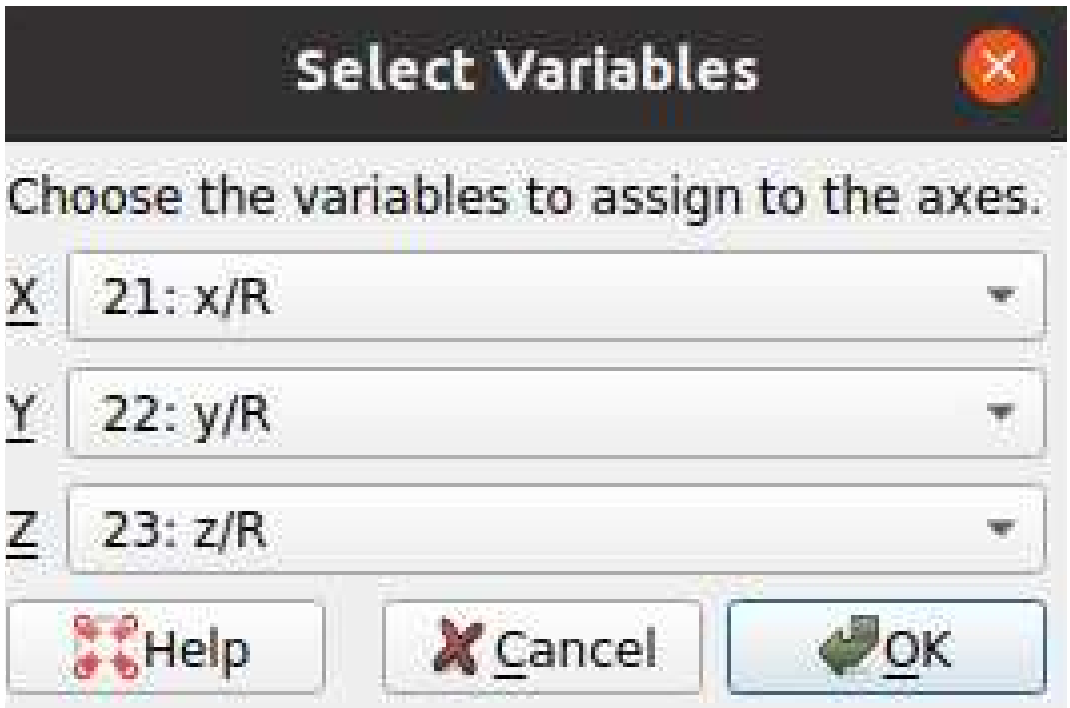


Figure B.6: Imposing normalized coordinates for plots

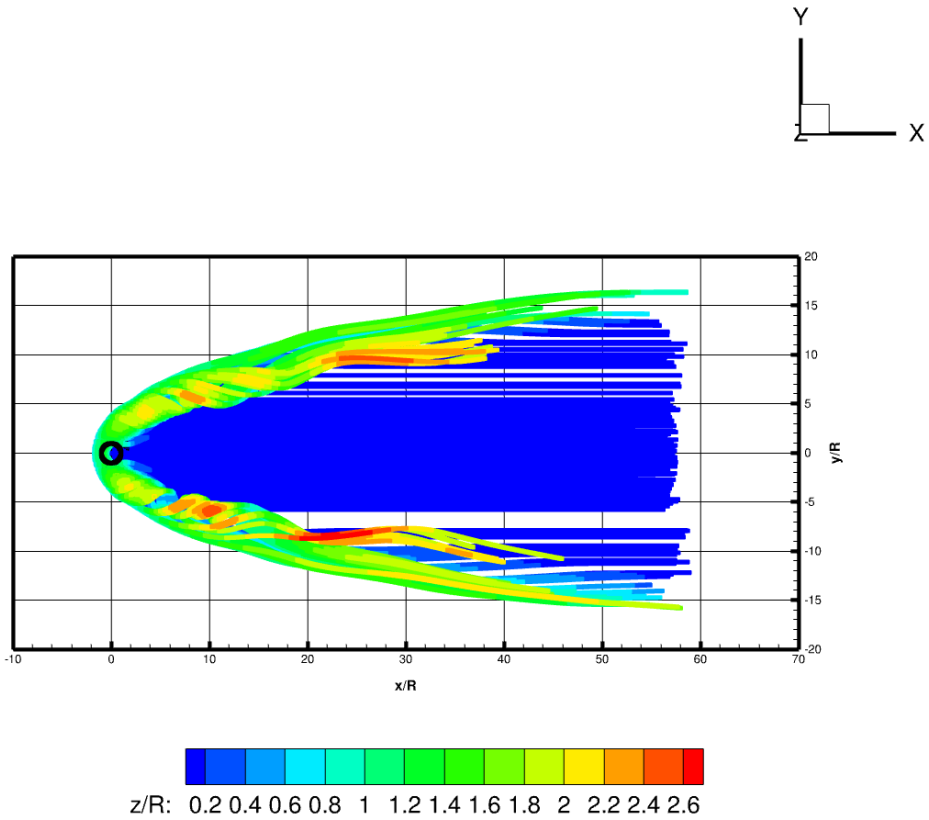


Figure B.7: Lagrangian particle tracking results for steady forward flying rotor. The rotor is operating at 10 kts (μ =0.023), $C_T = 0.0125$, top view

Unsteady flow - Pitching NACA0012 Airfoil

For the unsteady mode, the pitching NACA0012 test case is presented. As in the previous case, it is necessary to run the single-phase simulation. For this, the needed files are:

- st
- st.expert.chimera
- st.expert.turbulence_model
- st.expert.output
- st.expert.motions
- un
- un.Component (one for each overset grid component)

This test case is unsteady and use a chimera grid composed by background and foreground mesh. To generate this mesh, an example is given in TN16-005-QuickGuideTestCases. The physical conditions of the aerofoil are listed in table B.5.

Re_∞	$3.8 \cdot 10^6$
M_∞	0.38
α_{mean}	0.0 deg
α_{max}	20.0 deg
α_{min}	-20.0 deg

Table B.5: Aerofoil conditions

The necessary files are listed below:

Listing B.11: *st* file for pitching NACA0012 aerofoil.

```
File root       : naca0012_chim
Mesh root       : ./GRID/NACA0012_chim.grd

Incidence       : 0.0
Sideslip        : 0.0
Mach number     : 0.38
Reynolds number : 3.8e6

Explicit steps  : 200
Explicit cfl    : 0.3
Implicit steps  : 500
Implicit cfl    : 40.0
Convergence     : 1.0e-8

Model           : 3000
```

Listing B.12: *st.expert.chimera* file for pitching NACA0012 aerofoil.

```
Number of grids  : 2
Blocks pre Grid
8 8
```

```
Level of Grid
0 1

Solids Files                  : ./GRID/solids
Write bin flags (0/1)         : 0
Read bin flags (0/1/2)        : 0
Recalculate the weights (0/1) : 1
Recalculate distances (0/1)   : 0
Test Chimera weights (0/1)    : 0
Spool up for forward flight (0/steps) : 0
Point in tetra tolerance      : 1.0e-12


Interpolation method          : 10
Additional solid treatment    : 0
Use dot product criterion     : 0
Internal walls                : 0
Overlap extent            : 0
Very thin elements (0/1)      : 0
Points out of domain (0/1)    : 0
MVBB cleanup (0/1)            : 0
MVBB cleanup tolerance        : 1.0e-18
RT direction (vector)         : 0 1 0
Extend solid region (0/1)     : 0
MVBB & boundary inspect (0/1) : 0
Patch projection (0/1)        : 1
Geometry flag to project on   : -1
Projection direction (vector) : 0 0 1
White levels                  : 0
List of white levels

Boundary layer correction     : 0
Remove vel. from bndr. layer  : 0
Embedded solids (0/1)         : 0

Boundary layer thickness per solid flag:
```

Listing B.13: *st.expert.motions* file for pitching NACA0012 aerofoil.

```
Number of components: 2

Name    : Background
Embedded: 0
Origin  : 0.0 0.0 0.0
Axis 1  : 1.0 0.0 0.0
Axis 2  : 0.0 1.0 0.0
Axis 3  : 0.0 0.0 1.0

Name    : Airfoil
Embedded: 1
Origin  : 0.25 0.0 0.0
Axis 1  : 1.0 0.0 0.0
Axis 2  : 0.0 1.0 0.0
Axis 3  : 0.0 0.0 1.0
```

Listing B.14: *st.expert.output* file for pitching NACA0012 aerofoil.

```
grid reference length                   : 1.0
reference area                          : 1.0
reference length for moment about x-axis : 1.0
reference length for moment about y-axis : 1.0
reference length for moment about z-axis : 1.0
location on grid of moment centre (x)    : 0.0
location on grid of moment centre (y)    : 0.0
location on grid of moment centre (z)    : 0.0
Output convergence of steady loads/output: 1
Output frequency for steady solutions    : 5000
Output frequency for steady loads        : 100
```

```
Output maximum cell residuals (0/1)      : 0
Output cell centre residuals (0/1)       : 0
Output cell centre solution  (0/1)       : 0
Output cell vertex solution  (0/1)       : 0
Output cell vertex FEM solution (0/1)    : 0
Output cell centre FEM solution (0/1)    : 0
Output cell centre gradients (0/1)       : 0
Output cell centre surface (0/1)         : 0
Output cell centre FEM surface (0/1)     : 0
Output cell centre viscous surface (0/1) : 0
Output cell centre SAS solution (0/1)    : 0
Output cell centre DES solution (0/1)    : 0
Output cell centre SA  solution (0/1)    : 0
Binary Output (0/1) CC Chimera Halo      : 1 0 1 1
Output surface only with RHO and P (0/1) : 0
Blocks to write starting from 1          : 0
```

To generate the *un*, *un.Background un.Airfoil* files, it is necessary to use the following command:

```
./set_rigid_motions un
```

where `un` is the name of the main unsteady input file. For *un.Airfoil* file, the parameters are then entered as follows:

Listing B.15: input for un files

```
Enter name:
Airfoil
Are the rigid-body motions periodic? (1=yes, 0=no)
1
Periodic motions
Enter reduced frequency of 1st harmonic :
0.075
Enter number of cycles :
10
Enter number steps per cycles :
400
Enter maximum number of pseudo-time steps/time step
200
Enter tolerance of unsteady residual
0.001
Every how many steps is 3D output file required :
10
Every how many steps is surface output file required :
10
Every how many steps is checkpoint output file required - without solution :
10
Rotation about local x-axis ? (1=yes, 0=no)
0
Translation in local x-direction ? (1=yes, 0=no)
0
Rotation about local y-axis ? (1=yes, 0=no)
0
Translation in local y-direction ? (1=yes, 0=no)
0
Rotation about local z-axis ? (1=yes, 0=no)
1
Is rotation oscillatory ? (1=yes, 0=no)
1
Enter number of harmonics :
1
Enter mean angle (degs):
0.0
Enter sine and cosine harmonics (degs):
-20 0.0
0.3490655 0.000000
```

```
Translation in local z-direction ? (1=yes, 0=no)
0
```

While for *un.Background*:

Listing B.16: input for un files

```
Enter name:
Background
Are the rigid-body motions periodic? (1=yes, 0=no)
1
Periodic motions
Enter reduced frequency of 1st harmonic :
0.075
Enter number of cycles :
10
Enter number steps per cycles :
400
Enter maximum number of pseudo-time steps/time step
200
Enter tolerance of unsteady residual
0.001
Every how many steps is 3D output file required :
10
Every how many steps is surface output file required :
10
Every how many steps is checkpoint output file required - without solution :
10
Rotation about local x-axis ? (1=yes, 0=no)
0
Translation in local x-direction ? (1=yes, 0=no)
0
Rotation about local y-axis ? (1=yes, 0=no)
0
Translation in local y-direction ? (1=yes, 0=no)
0
Rotation about local z-axis ? (1=yes, 0=no)
0
Translation in local z-direction ? (1=yes, 0=no)
0
```

Before run HMB3, few preparation steps are needed. The first is loadbalance. HMB3 is a multi-block fluid solver. Every block of the grid is assigned to a single processor, which revolve the fluid equations. *loadbalance* assign every block to a single processors. To run *loadbalance* use the following command:

```
loadbalance st X
```

Where X, is the number of processors that it will be used for the simulation. The second routine that it is necessary to launch HMB3, is pre-grid. This routine split the grid file, in single-block files. To use pregrid, use the following command:

```
pre_grid_parallel_GCC_4.5.1_OpenMPI_1.3.3_19-10-18 GRID/naca0012_chim.grd GRID/solids none
```

solids file, contains the boundary flag used in the grid to identify the solid walls. For this test case, the *solids* file is listed below.

Listing B.17: *solids* file for the NACA0012

```
1
7000
```

Once that these preliminary routines have been used, it is possible to run HMB3. The command to use is:

```
mpiun -np X hmb_parallel_method3_3.0.0_COUPLED_GCC_... st un none none
```

Once the single-phase simulation is done, output files need to be concatenated. HMB3 writes sigle-block files. To use the particle tracking tool, the Binary output is necessary. The command to use is:

```
multi_Binary naca_chimera0012 0 4000 10 X 1
```

Where X, is again the number of processors used in the simulation. As described previously, the particle tracking tool is a post processing step. When the single-phase simulation is done, it is necessary to transform the flowfield file in the format suitable for the particle tracking tool. In the previous section the Tecplot macro and the c code necessary for this were described. so they will not repeated. To run the macro it is necessary to change few information contained in the head of the file, namely file root and iterations, while for the c code this is not necessary. To obtain the flowfield file two commands are necessary.

```
tec360 -b extract_dat.mcr
```

```
./create_ff naca0012_chim_Binary 0 4000 10 GRID/NACA0012_chim.grd.top
```

These two commands write two new files, a .dat and a .plt file for each time-step. However, only the .plt is used by the tool, the dat file can be deleted (it is necessary to generate the plt file, than is not more useful). Once the flowfield files are ready, the *st.expert.particles* is needed:

Listing B.18: *st.expert.particles* file for pitching NACA0012 aerofoil

```
n grids | read all (y=1,n=0) | Ncycle | bomb option (y=1,n=0)| dist_tol |
2 0 1 0 0.50
| Grid-path | Re | Mach | model turbo |
NACA0012_chim.grd 3.8e6 0.38 3000
Seeding types (N bc (flags) | N planes | N rakes | N points | N seeding prop)
0 0 1 0 1
Info seeding (see tn)
1.01 0.01 1.0 1.01 -0.01 1.0 5 1 1
particle prop TypeID | density | radius | cl | cd | u_t (all dimensionless)
1 2163.2 0.00005 0.0 1.046 0.0
gravity (x y z component) (dimensionless minus sign already in the code)
0.0 0.0 0.0
Velocity scaling
1.0 1.0 1.0
Equation
1
Space limits
10000.0 10000.0 10000.0
Chimera levels option (if Ngrid>1 else put 0 or none)
0 1
Chimera levels option (if Ngrid>1 else put 0 or none)
8 8
skipping blocks (N tranches of blocks to skip + tranches to skip)
0
stop seeding revolution
1
ref values length | vel | density
1.0 1.0 1.0
```

In this case, it is necessary to specify the number to the grids in the first row of the *st.expert.particles*, and later how blocks are shared among grids. For this test case, the chimera grid is composed by 2 grids, each with 8 blocks and the first is the background. For this example, only a single rake of particles with mass is introduced in the flowfield, in proximity of the trailing edge. Space limits are larger than the grid size, so no ground effect has been taken into account. Without bombardment, the reference values are not used. Gravity has not been taken into account for this test case, only particle with mass. Particle soil material density is $\rho_p^* = 2650[kg/m^3]$ and radius can be settled to $r_p^* = 4.5 \cdot 10^{-6}[m]$ [33]. These values were adimenzionalized using air density $\rho_{ref} = 1.225[kg/m^3]$ and $L_{ref} = 0.5[m]$.

Finally, particle tracking tool can run. As previously stated, the particles are shared among processors, so the number of processor should be at lease smaller than the amount of particles. The command to run the unsteady particle tracking tool is:

```
mpirun -np X ./particle_tracking_parallel naca0012_chim_Binary 0 4000 10 un
```

Here *naca0012_chim_Binary*, is the file root of the flowfield files, while the *un* is the same file used for the HMB3 unsteady simulation. At each unsteady timestep, the tool generate an output. The output files are particles_position.00YYYYY.dat.XXXXX. Where XXXXX is the processor number, while YYYYY is the time-step. To obtain the final output files, these files need to be concatenated, and then using preplot_hmb, they can be converted into binary files.

```
cat particles_position.YYYYYYY.dat.0* > particles_position.YYYYYYY.dat
```

```
hmb/bin/preplot_hmb particles_position.YYYYYYY.dat
```

Considering each time-step a separate particle tracking file, it is possible to use the *edit time strands* option, to animate the particle tracking. To obtain this, flowfield and particle tracking file for each time step need to be uploaded in tecplot one after the other, as shown in figure B.8. Then, using *edit time strands* option,as described in figure B.9, it is possible to define a certain amount of zones as a single timestep. Finally, using the scatter option, it is possible to obtain the animation like in figure B.10.

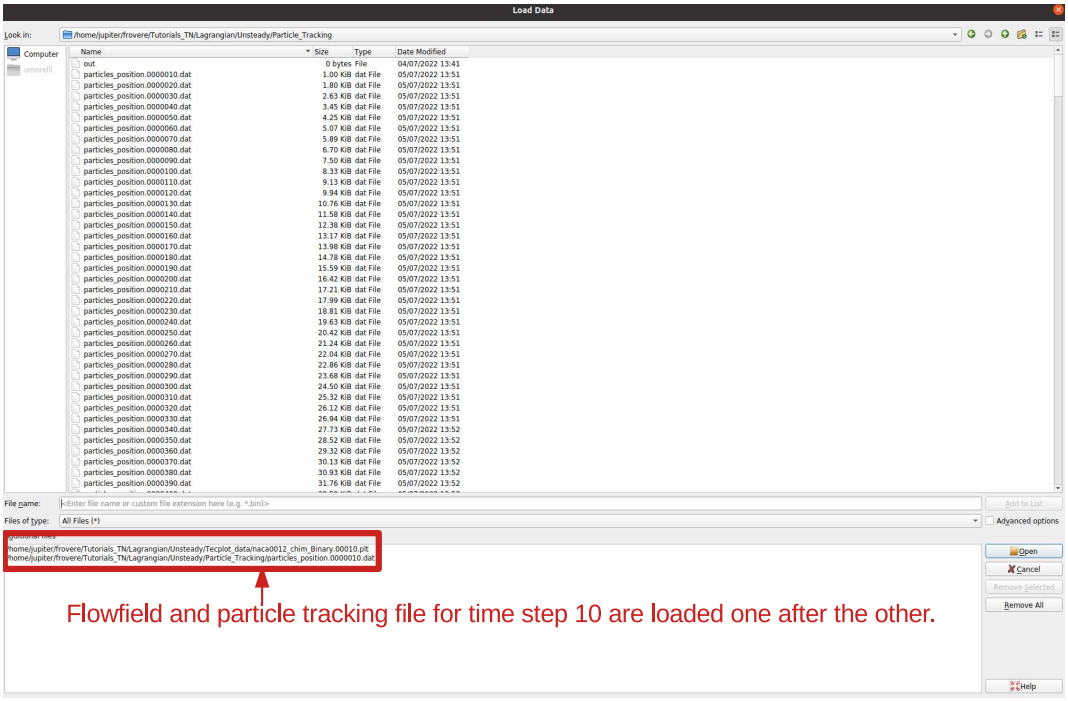


Figure B.8: To animate flowfield and particle tracking it is necessary to list properly the files.

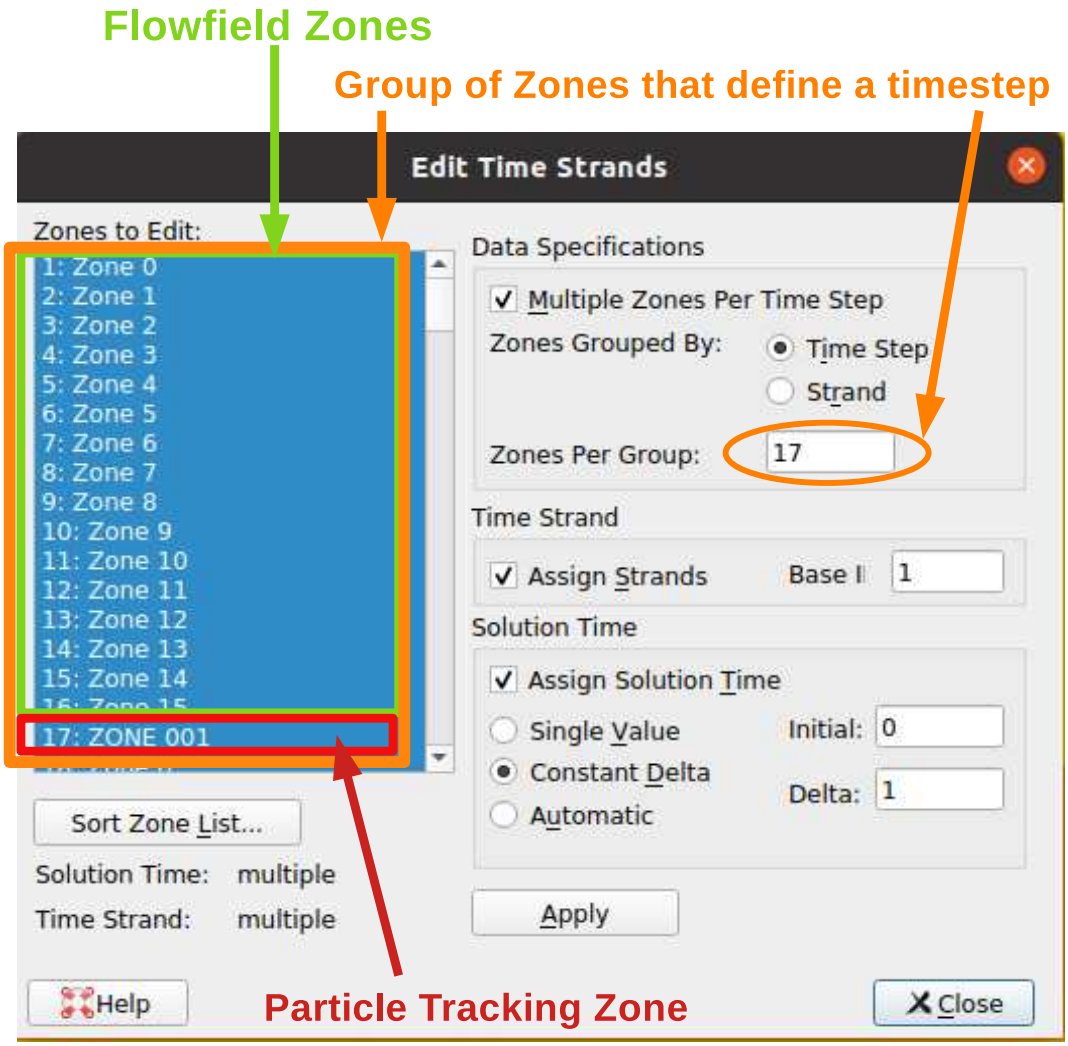


Figure B.9: Tecplot edit time strands window.

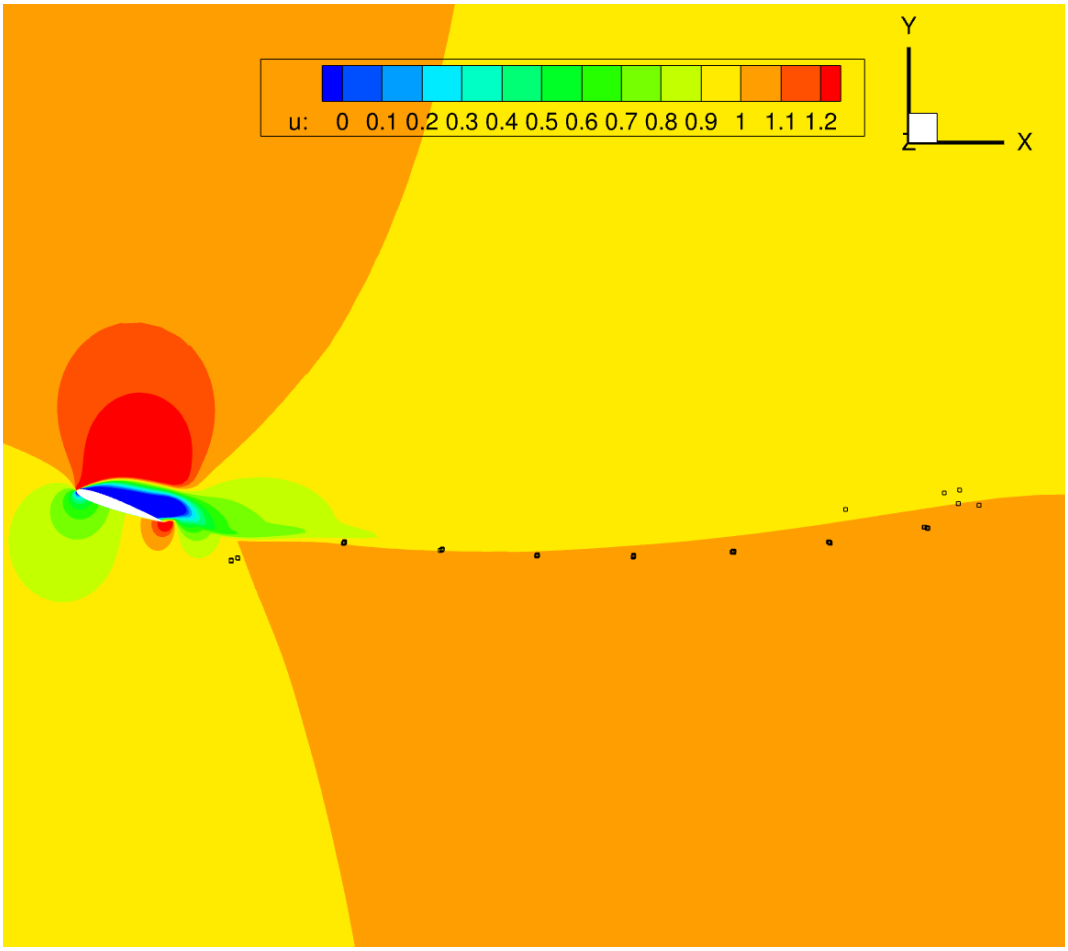


Figure B.10: Lagrangian particle tracking results for naca0012 unsteady.

Appendix C

Eulerian model of HMB3

C.1 Eulerian numerical model

The particle dust transport model used here is:

$$\int_V \frac{\partial \rho_d}{\partial t} dV + \int_V (u + w_t) \cdot \nabla \rho_d dV = \int_S \vec{S_d} \cdot \vec{n} dS + \int_V \nu_p \nabla^2 \rho_d dV \tag{C.1}$$

It is possible to start using a number density: $\rho_d = \frac{N_p}{V}$ $[N_p/m^3]$ where w_t is the fall out velocity, which take into account the effect of gravity, ν_p is the particle diffusion coefficient, while S_d is the source term, computed starting from the uplift Bagnold method. The fall out velocity is a vector, considering the gravity direction as z and written as $w_t = [0\ 0\ \hat{w}_t]^T$ Starting from the 2nd Newton law, for a falling particle where the gravitational force and the aerodynamic drag are equal:

$$(\rho_p - \rho) g \frac{\pi d_p^3}{6} = C_D \frac{\pi d_p^2}{4} \frac{\rho w_t^2}{2} \tag{C.2}$$

For $Re_p < 1$, $C_D = 24/Re_p$. Considering $Re_p = \frac{d_p w_t}{\nu_{ref}}$ it is possible to obtain:

$$w_t = \frac{g d_p^2}{18 \nu_{ref}} \frac{(\rho_p - \rho)}{\rho}. \tag{C.3}$$

In this situation, we assume that $Re_p < 1$ everywhere, which may not be strictly true, however, for the porpuse of this work it is acceptable. Furthermore, we consider that particles reach their terminal velocity instantaneously. The terminal velocity is given $[m/s]$, and it is necessary to be nondimenzionalized with U_{ref}. It is necessary to define $St = \frac{\tau_p}{\tau_f}$, where τ_p is the particle response time, defined as $\tau_p = \frac{\rho_p d_p^2}{18 \mu_{ref}}$, while $\tau_f = L_{ref}/U_{ref}$ is the fluid response time. Finally, we define the Froude number. $Fr^2 = \frac{U_{ref}^2}{g L_{ref}}$. With some algebra it is possible to reach:

$$\frac{w_t}{U_{ref}} = \frac{g d_p^2}{18 U_{ref} \nu_{ref}} \frac{\rho_p}{\rho} \left(1 - \frac{\rho}{\rho_p}\right). \tag{C.4}$$

Introducing the particle response time:

$$\hat{w_t} = \frac{g\tau_p}{U_{ref}}\left(1-\frac{\rho}{\rho_p}\right), \tag{C.5}$$

and introducing the Stokes number,

$$\hat{w_t} = \frac{StL_{ref}g}{U_{ref}^2}\left(1-\frac{\rho}{\rho_p}\right), \tag{C.6}$$

and the the Froude number, we can rewrite:

$$\hat{w_t} = \frac{St}{Fr^2}\left(1-\frac{\rho}{\rho_p}\right). \tag{C.7}$$

Once that the terminal velocity has been modelled, it is necessary to define the source terms. The source term can be modelled using the mass fluxes defined by Marticorena [143], as horizontal (Q) and vertical (F). In this case, the x and y axis define the ground plane, while z is normal to it. We define $\vec{F_m} = [Q\ Q\ F]^T$. It is necessary start with the horizontal flux, computed using the threshold friction velocity.

$$Q = Ecu_*^3\frac{\rho}{g}\left(1-\frac{u_{*t}}{u_*}\right)\left(1+\frac{u_{*t}^2}{u_*^2}\right) \tag{C.8}$$

and u_{*t} is the threshold friction velocity (Bagnold model [106]), while $u_{*t} = \sqrt{\frac{\tau_w}{\rho}}$ is the friction velocity, with $c = 0.261$ a constant of the model. ρ is the flowfield density, and E is the ratio of erodible to total surface area, taken as 1 for simplicity. From the horizontal flux, it is possible to compute the vertical flux [143] as:

$$F = Qe^{13.4f-6.0} \tag{C.9}$$

In this work f=0.1 [143]. The vertical flux F is defined in [143] as the mass of the particles passing through a horizontal unit area per unit time. The equation derived by Marticorena [143], is defined as a mass flux, with dimensions of $[\frac{kg}{ms}]$. To use it for the source term it is necessary to convert to number of particles per unit volume. In this work, the mass flux has been modelled as $\vec{S_d} = [Q\ Q\ F]^T\frac{d_p}{m_pA_p}$. Here m_p is the particle mass, d_p is the particle diameter, and A_p the particle frontal area, which can be considered the smallest area through a particle can enter the flowfield from the ground.The equation reads:

$$Q\frac{d_p}{m_pA_{ref}} = \frac{Ecd_p}{m_pA_{ref}}u_*^3\frac{\rho}{g}\left(1-\frac{u_{*t}}{u_*}\right)\left(1+\frac{u_{*t}^2}{u_*^2}\right) = \frac{pu_*^3}{m_pA_{ref}}\frac{\rho d_p}{g}, \tag{C.10}$$

where $p = Ec\left(1-\frac{u_{*t}}{u_*}\right)\left(1+\frac{u_{*t}^2}{u_*^2}\right)$, and contains only dimensionless values. It is important to notice that if $u_{*t} > u_*$, the threshold model is not satisfied, in other words, particles cannot be uplifted, and this may lead to a zero flux. However, this is only a numerical problem, leading to $S_d = 0$, in case of $u_{*t} > u_*$. We now define ϕ, the volume fraction occupied by the particles as: $\phi = \rho_d V_p$. To obtain this, we can multiply both sides of equation C.10 with V_p, the volume of the

particle. It also necessary to apply the divergence theorem to the source term in the equation C.1,to obtain:

$$\int_V \frac{\partial \phi}{\partial t} dV + \int_V (u + w_t) \cdot \nabla \phi dV = V_p \int_V \nabla \cdot \vec{S}_d dV + \int_V \nu_p \nabla^2 \phi dV \tag{C.11}$$

Considering an arbitrary volume, we can obtain:

$$\frac{\partial \phi}{\partial t} + (u + w_t) \cdot \nabla \phi = V_p \nabla \cdot \vec{S}_d + \nu_p \nabla^2 \phi \tag{C.12}$$

It is now necessary to non-dimenzionalize equation C.12, using.

$$\begin{aligned} \widetilde{t} &= \frac{t}{l_{ref}/V_{ref}}, \\ \widetilde{x} &= \frac{x}{l_{ref}} \end{aligned} \tag{C.13}$$

It is also necessary to define $\sigma = \dfrac{\rho}{\rho_p}$, which is the ratio between the density of the fluid and the density of particles material soil ($\rho_p = 2650 kg/m^3$). We also define the ratio between the particle diameter and the reference length: $\hat{d_p} = \dfrac{d_p}{l_{ref}}$.

Rewriting the expression C.12, with the dimensionless variables, we obtain:

$$\frac{\partial \phi}{\partial \widetilde{t}} \frac{V_{ref}}{l_{ref}} + (\widetilde{u} + \widetilde{w_t}) \cdot \nabla \phi \frac{V_{ref}}{l_{ref}} = \frac{V_p}{l_{ref}} \nabla \cdot \vec{S}_d + \nabla^2 \phi \frac{1}{l_{ref}^2} \frac{\nu}{Sc} \tag{C.14}$$

and with algebra, it is possible to read:

$$\frac{\partial \phi}{\partial \widetilde{t}} + (\widetilde{u} + \widetilde{w_t}) \cdot \nabla \phi = \frac{V_p}{V_{ref}} \nabla \cdot \vec{S}_d + \nabla^2 \phi \frac{1}{Sc} \frac{1}{Re}. \tag{C.15}$$

$\vec{n}$ can be defined as $\vec{n} = [0\ 0\ 1]^T$, in this way

$$\nabla \cdot \vec{S}_d = \left(\frac{\partial}{\partial x} \left(p \widetilde{u_*}^3 \right) + \frac{\partial}{\partial y} \left(p \widetilde{u_*}^3 \right) + \frac{\partial}{\partial z} \left(p \widetilde{u_*}^3 \right) e^{13.5f - 6.0} \right) \frac{V_{ref}^3 \rho V_p d_p}{g m_p A_p},$$

$$\frac{\partial \phi}{\partial \widetilde{t}} + (\widetilde{u} + \widetilde{w_t}) \cdot \nabla \phi = \left(\frac{\partial p \widetilde{u_*}^3}{\partial x} + \frac{\partial p \widetilde{u_*}^3}{\partial y} + \frac{\partial p \widetilde{u_*}^3}{\partial z} e^{13.4 - 6.0} \right) V_{ref}^3 \frac{V_p \rho}{g} \frac{1}{\rho_p V_p \dfrac{\pi \hat{d}_p l_{ref}}{4}} \frac{1}{V_{ref}} + \nabla^2 \phi \frac{1}{ScRe} \tag{C.16}$$

Defining $S = \left(\dfrac{\partial p \widetilde{u_*}^3}{\partial x} + \dfrac{\partial p \widetilde{u_*}^3}{\partial y} + \dfrac{\partial p \widetilde{u_*}^3}{\partial z} e^{13.4 - 6.0} \right)$, for simplicity, and using the first term on the right of equation we obtain:

$$S V_{ref}^3 \frac{\rho}{g} \frac{V_p}{\rho_s V_p \dfrac{\pi \hat{d}_p^2 l_{ref}^2}{4}} \frac{l_{ref}}{V_{ref}} = S V_{ref}^2 \frac{4 \sigma V_p}{g \pi \hat{d}_p}. \tag{C.17}$$

Introducing $\sigma = \dfrac{\rho}{\rho_s}$ and the Froude number as $Fr^2 = \dfrac{V_{ref}^2}{g l_{ref}}$, we have:

$$\frac{\partial \phi}{\partial \widetilde{t}} + (\widetilde{u} + \widetilde{w_t}) \cdot \nabla \phi = S \frac{4Fr^2 \sigma}{\pi \hat{d}_p} + \nabla^2 \phi \frac{1}{ScRe} \tag{C.18}$$

The *Sc* can be modelled using the Einstein-Stokes equation for Brownian motion [140].

$$\nu_p = \frac{kT}{3\pi\mu d_p}, \tag{C.19}$$

where k is the Boltzmann constant, T is the temperature, μ is the fluid viscosity, and as usual d_p particle diameter.

C.2 Input files

C.2.1 st.expert.scalars

Listing C.1: *st.expert.scalar* example

```
Number of User Defined Scalars      : 1
User Defined Scalars Scheme order   : 3
Explict (0) or Implict (1) scheme   : 1
Restart File                        : 1
No source term (1)                  : 1
Unsteady time step                  : 0.1
Dual-Time inner iteration           : 40
Dual-Time Residual (Res/Update)     : 0
Dual-Time Tolerance                 : 1.0e-4
CFL number                          : 10.0
Number of iterations                : 7000
Fixed flow variable solve           : 0
Local time stepping                 : 1
User Defined Scalars output         : 100
Schmidt number                      : 1e6
Froude number (squared)             : 6.568
Particle diameter (adimensionalized): 21.9e-6
fluid and soil density ratio r_f/r_s: 0.00046
friction threshold (adimensionalized):0.1128
Terminal velocity (1=yes 0=no)      : 1.0
Lref                                : 0.41
Vref                                :5.14
Stokes number               : 0.01
```

The first line defines the number of user defined scalars. This should be 5 or less as currently the Jacobian reuses the Navier-Stokes Jacobian. The scalar equations will normally be independent of each other but it is possible to set up systems or equations with the correct changes. Apart from the functions in the library *libHMBuds.so* extra space will be needed for the left and right Jacobains since currently it is assumed they are indepe indent, and use twice the number of user defined scalar variables, (twice since both the left and right Jacobians are stored in the same vector). The second line is the order of the scheme. This can be either first-order or third-order with MUSCL extrapolation. The following line sets up either an explicit or implicit solve. A time-accurate explicit solve is not recommended since the stable time-step can be of the order of 10^{-8} resulting in millions of iterations required for each scalar to be advected one non-dimensional unit. The implicit solve obtained using dual time stepping in exactly the same way the unsteady Navier-Stokes equations are solved. The 4th line states where a restart file is used while the 5th is just a performance enhancement for when there is no source term. The next four lines relative to the setup of the parameters in the dual time stepping method for the user defined scalars. If the unsteady time step is set to zero then the user defined scalars equations are solved to steady state. For any other positive value the non-dimensional time step is used in the dual time-stepping motion. Note that if the flow variables are changing, i.e. the unsteady Navier-Stokes equations are being solved, this time step should be the same as the one used for the solution of the unsteady Navier-Stokes equations. Otherwise, if the flow variables are fixed then a value can be selected based on the time accuracy requirement. The next input is the maximum number of inner iterations. This is normally smaller than the number required for the Navier-Stokes equations since the source term is not normally as stiff. Then follows if the update norm should be based on the unsteady residual (0) or the change in solution from one inner iteration to the next (1). The second method is one used for the unsteady Navier-Stokes equations. The final number is the tolerance used in the dual time

stepping method. In this case 0.0001 means the update norm must drop 4 orders of magnitude from the initial one. The CFL is either the CFL number of the explicit scheme (maximum of 0.5) or the implicit scheme. The maximum CFL for the implicit scheme greatly depends on the underlying user defined scalars equations but 10.0 should be fine for nearly all of them. However, if the scalars equations have no source term and are hence of convection-diffusion type, a CFL in the low hundreds are possible. The number of iterations are to be used when there is a fixed flow solution for either the explicit or implicit scheme. The Fixed flow variable solve describes the behavior of the velocity field for the Navier-Stokes equations. If it is fixed, then a steady state Navier-Stokes solve is used, else it is varying and hence an unsteady solve. Local time stepping means the time step is NOT set to the smallest time-step for all cells. This means that the solution is no time accurate for the explicit solve but will converge in many fewer time-steps to the steady state solution. If the implicit dual time-stepping method is being used, the time step is always local. The last time defines how often the solution is outputted. Then the user must define variables for the Eulerian model: Schmidt number and Froude number as defined in the previous section, particle diameter and density ratio between fluid and soil, the friction velocity threshold and the flag for the terminal velocity. Finally the reference values of length and velocity in $[m]$ and $[m/s]$.

C.2.2 Checkpoint files

HMB3 uses checkpoint files also for UDS. In the same folder where the checkpoint files for the flowfield are saved, the checkpoint_UDS files are stored too. They are stored as: $< File_root >$_checkpoint_UD When HMB3 is restarted, if $< File_root >$_UDS is inside the folder, the checkpoint files are read, and used a starting value for UDS. For this reason, to introduce an initial value of UDS in the domain, it is also possible creating a checkpoint file ($< File_root >$_checkpoint_UDS). Due to variety of configurations that meshes and test cases may have, a generic code that imposes a value of UDS in portions of the domain cannot be provided.

C.3 Eulerian modelling examples

C.3.1 Steady flow - NACA0012 aerofoil

For this steady test case the NACA0012 aerofoil is used. The aerofoil is operating at Re=$3.8 \cdot 10^6$, $M_\infty = 0.038$ and angle of attack $\alpha = 10deg$. To add the initial scalar quantity to the flowfield, at the inflow, flowfield and USD values are imposed, using the boundary condition 10007 (profile boundary values). The farfield boundary conditions has been imposed to the top, right and button of the domain, ad described in figure C.1.

For this specific case, this boundary condition has been imposed on blocks 1 and 6, on faces 3 and 1, respectively. The input files used for this simulations are:

- st
- st.expert.profile
- st.expert.turbulence_model
- st.expert.output

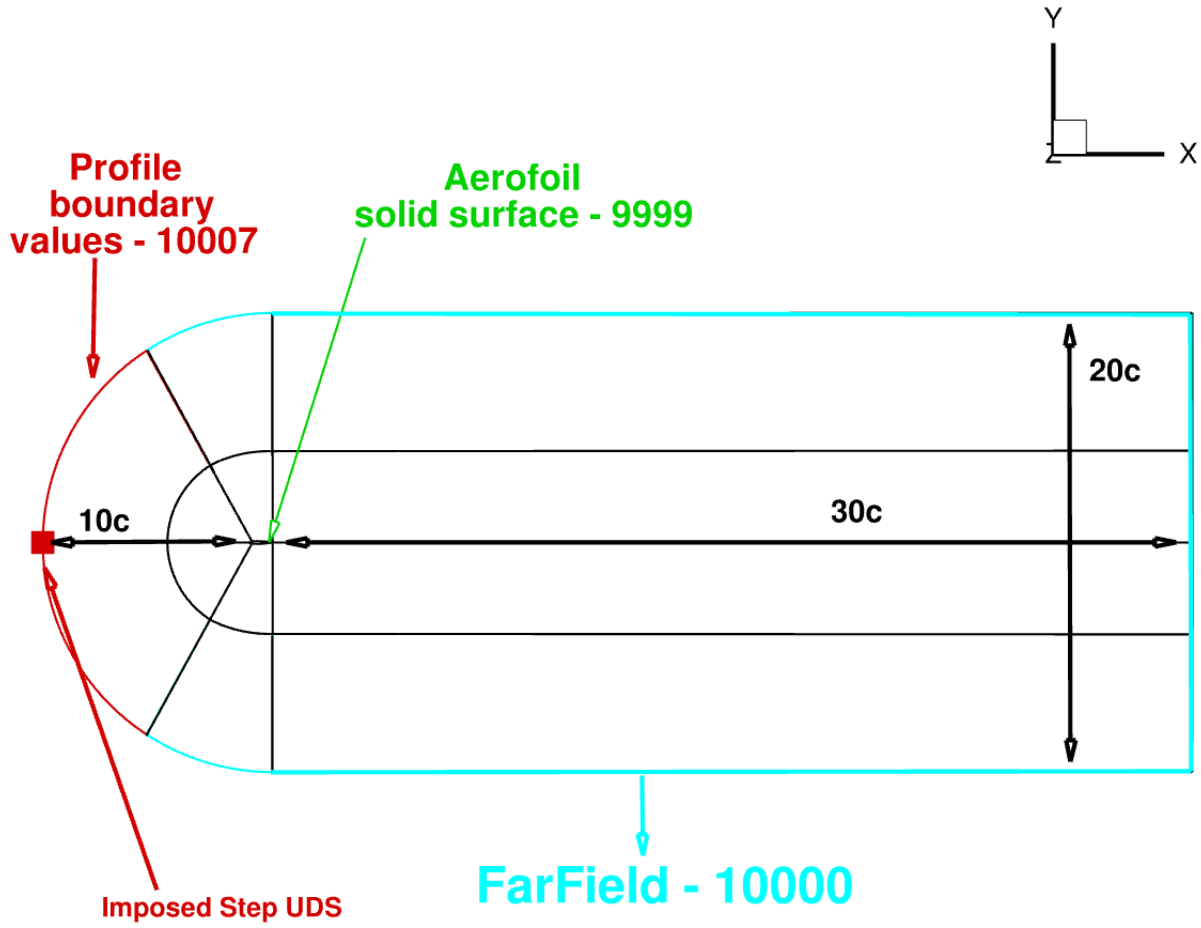


Figure C.1: Grid size and boundary conditions for steady NACA0012 aerofoil.

- st.expert.scalars
- profile_inflow
- profile_UDS

Listing C.2: *st* file for the steady NACA0012 aerofoil.

```
File root       : naca0012
Mesh root       : ./GRID/naca0012.grd

Incidence       : 10.0
Sideslip        : 0.0
Mach number     : 0.38
Reynolds number : 3.8e6

Explicit steps  : 500
Explicit cfl    : 0.3
Implicit steps  : 5000
```

```
Implicit cfl    : 40.0
Convergence     : 1.0e-8

Model           : 3000
```

Listing C.3: *st.expert.turbulence_model* file for the steady NACA0012 aerofoil.

```
--------------------
Two-equation models:
--------------------
Mach number correction : 0
Vortex limiter         : 0
Sustain ambient values : 1
Transition modelling   : 0
Roughness              : 0
SST blending           : 1
SST clip               : 1
Non-Linear terms       : 0
SAS clip               : 0
limit SAS mu_t         : 0
Quadratic stress       : 0
Cubic stress           : 0
DES/LNS clip           : 0
Cubic root volume DES  : 0
DES model              : 0
DDES model             : 0
IDDES model            : 0
R/C correction         : 0
R/C scale              : 1.0
Apply Pk Clip          : 0

New 2eq Model          : 1

--------------------
One-equation models:
--------------------
DES model   : 0
DDES model  : 0
IDDES model : 0
SALSA model : 0
EDW   model : 0
DES-CDES    : 0.65
Secundov correction    : 0
R/C correction         : 0

-----------------
Algebraic models:
-----------------
Smagorinsky constant : 0.18
```

Listing C.4: *st.expert.output* file for the steady NACA0012 aerofoil.

```
grid reference length                   : 1.0
reference area                          : 1.0
reference length for moment about x-axis : 1.0
reference length for moment about y-axis : 1.0
reference length for moment about z-axis : 1.0
location on grid of moment centre (x)    : 0.0
location on grid of moment centre (y)    : 0.0
location on grid of moment centre (z)    : 0.0
Output convergence of steady loads/output: 1
Output frequency for steady solutions    : 5000
Output frequency for steady loads        : 100
Output maximum cell residuals (0/1)      : 0
Output cell centre residuals (0/1)       : 0
Output cell centre solution  (0/1)       : 0
Output cell vertex solution  (0/1)       : 0
Output cell vertex FEM solution (0/1)    : 0
```

```
Output cell centre FEM solution (0/1)    : 0
Output cell centre gradients (0/1)       : 0
Output cell centre surface (0/1)         : 0
Output cell centre FEM surface (0/1)     : 0
Output cell centre viscous surface (0/1) : 0
Output cell centre SAS solution (0/1)    : 0
Output cell centre DES solution (0/1)    : 0
Output cell centre SA  solution (0/1)    : 0
Binary Output (0/1) CC Chimera Halo      : 0 0 1 1
Output surface only with RHO and P (0/1) : 0
Blocks to write starting from 1          : 0
```

Listing C.5: *st.expert.scalars* file for the steady NACA0012 aerofoil.

```
Number of User Defined Scalars      : 1
User Defined Scalars Scheme order   : 3
Explict (0) or Implict (1) scheme   : 1
Restart File                        : 0
No source term (1)                  : 0
Unsteady time step                  : 0.10472
Dual-Time inner iteration           : 20
Dual-Time Residual (Res/Update)     : 0
Dual-Time Tolerance                 : 1.0e-4
CFL number                          : 50.0
Number of iterations                : 20
Fixed flow variable solve           : 0
Local time stepping                 : 1
User Defined Scalars output         : 10
Schmidt number                      : 1e6
Froude number (squared)             : 330
Particle diameter (adimensionalized): 9e-6
fluid and soil density ratio r_f/r_s: 0.00046
friction threshold (adimensionalized):0.11
Terminal velocity (1=yes 0=no)      : 0.0
Lref                                : 1.0
Vref                                : 57.0
Stokes number                       : 0.045
```

Listing C.6: *st.expert.profile* file for steady NACA0012 aerofoil.

```
Given profile   : profile_inflow profile_UDS

EFP (1/0)       : 0
EFP pressure    : 1.0
Signature (1/0) : 0
Signature file  : 0
Start sig       : 0
Finish sig      : 0
```

The above file will use the content of file profile_inflow for boundary conditions as described below for the flow solution variables. An additional file is needed: profile_UDS for the boundary conditions for the user defined scalars but the formatting is the same as the flow solution variables file. A typical inlet profile file is shown below. The first line of the file is the number of blocks where the profile is applied to. This is followed by a set of lines which the block number and, the face, where the profile will be applied to, and the dimensions of the face. The lines after this block of data contain the 5 non dimensional flow variables ρ, u, v, w, p followed by the turbulent variables. In this case the values of k and ω. The values are read in the standard HMB3 jik ordering for the given face. Caution is need when compiling these files when non constant profiles are being used as the ordering will be depended on the orientation of the blocks.

Listing C.7: *profile_inflow* file for steady NACA0012 aerofoil.

```
2
```

```
1 3 64 1
6 1 64 1
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
...
```

The profile_UDS is similar to the previous, after the three first lines, the value of UDS is imposed for every node. In this case, three points were seeded with UDS=1, in the middle of the domain.

Listing C.8: *profile_UDS* file for steady NACA0012 aerofoil.

```
2
1 3 64 1
6 1 64 1
1.0
1.0
1.0
0.0
0.0
0.0
..
```

Before run HMB3, few preparation steps are needed. The first is loadbalance. HMB3 is a multi-block fluid solver. Every block of the grid is assigned to a single processor, which revolve the fluid equations. *loadbalance* assign every block to a single processors. To run *loadbalance* use the following command:

```
loadbalance st X
```

Where X, is the number of processors that it will be used for the simulation. The second routine that it is necessary to launch HMB3, is pre-grid. This routine split the grid file, in single-block files. To use pregrid, use the following command:

```
pre_grid_parallel_GCC_4.5.1_OpenMPI_1.3.3_19-10-18 GRID/naca0012_chim.grd GRID/solids none
```

solids file, contains the boundary flag used in the grid to identify the solid walls. For this test case, the *solids* file is listed below.

Listing C.9: *solids* file for the NACA0012

```
1
9999
```

Once that these preliminary routines have been used, it is possible to run HMB3. The command to use is:

```
mpiun -np X hmb_parallel_method3_3.0.0_COUPLED_GCC_... st none none none
```

HMB3 writes sigle-block files. To visualize use the User Define Scalar the output files have to be concatenated. The command to use is:

```
multi_new naca_chimera0012_CCFEM_UDS 300 300 1
```

Results can be plotted using Tecplot and contours option, UDS is a standard variable in Tecplot workframe.

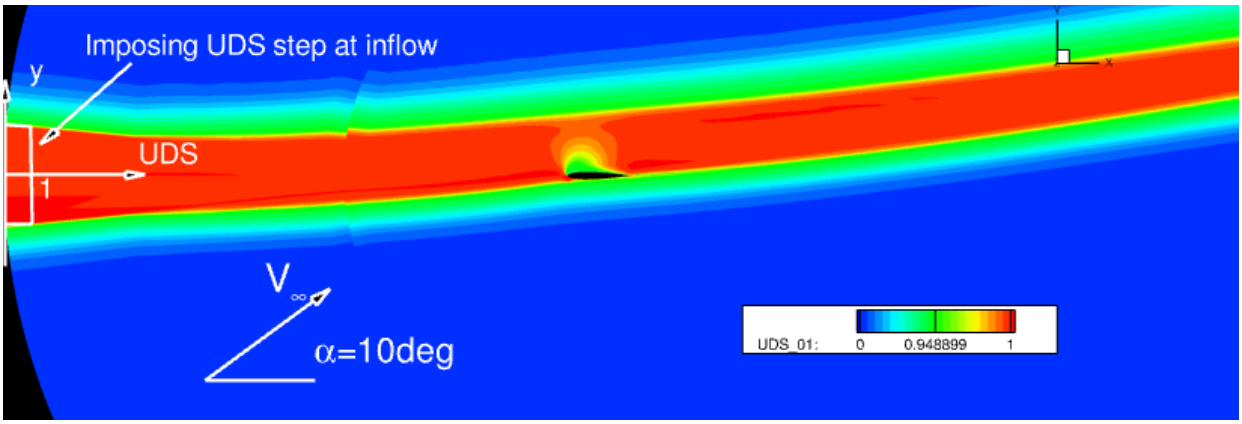


Figure C.2: Eulerian results for steady aerofoil. The aerofoil was operating at $Re = 3.8\dot{1}0^6$, $M_\infty = 0.38$, $\alpha = 10\,\text{deg}$

Unsteady flow - Pitching NACA0012 aerofoil

For the unsteady mode, the pitching NACA0012 test case presented in TN16-005 Quick guide test cases has been used. The physical conditions are listed in table C.1.

Re_∞	$3.8 \cdot 10^6$
M_∞	0.38
α_{mean}	10.3deg
α_{max}	18.4deg
α_{min}	2.2deg

Table C.1: Aerofoil conditions

For this unsteady simulation, the needed files are:

- st
- st.expert.chimera
- st.expert.turbulence_model
- st.expert.profile
- st.expert.output
- st.expert.motions
- st.expert.scalars
- profile_inflow
- profile_UDS
- un
- un.Component (one for each overset grid component)

The input files are listed below:

Listing C.10: *st* file for pitching NACA0012 aerofoil.

```
File root       : naca0012
Mesh root       : ./GRID/naca0012.grd

Incidence       : 0.0
Sideslip        : 0.0
Mach number     : 0.38
Reynolds number : 3.8e6

Explicit steps  : 500
Explicit cfl    : 0.3
Implicit steps  : 5000
Implicit cfl    : 40.0
Convergence     : 1.0e-8

Model           : 3000
```

Listing C.11: *st.expert.turbulence_model* file for pitching NACA0012 aerofoil.

```
--------------------
Two-equation models:
--------------------
Mach number correction : 0
Vortex limiter         : 0
Sustain ambient values : 1
Transition modelling   : 0
Roughness              : 0
SST blending           : 1
SST clip               : 1
Non-Linear terms       : 0
SAS clip               : 0
limit SAS mu_t         : 0
Quadratic stress       : 0
Cubic stress           : 0
DES/LNS clip           : 0
Cubic root volume DES  : 0
DES model              : 0
DDES model             : 0
IDDES model            : 0
R/C correction         : 0
R/C scale              : 1.0
Apply Pk Clip          : 0

New 2eq Model          : 1

--------------------
One-equation models:
--------------------
DES model   : 0
DDES model  : 0
IDDES model : 0
SALSA model : 0
EDW   model : 0
DES-CDES    : 0.65
Secundov correction    : 0
R/C correction         : 0

-----------------
Algebraic models:
-----------------
Smagorinsky constant : 0.18
```

Listing C.12: *st.expert.output* file for pitching NACA0012 aerofoil.

```
grid reference length                   : 1.0
reference area                          : 1.0
reference length for moment about x-axis : 1.0
reference length for moment about y-axis : 1.0
reference length for moment about z-axis : 1.0
location on grid of moment centre (x)    : 0.0
location on grid of moment centre (y)    : 0.0
location on grid of moment centre (z)    : 0.0
Output convergence of steady loads/output: 1
Output frequency for steady solutions    : 5000
Output frequency for steady loads        : 100
Output maximum cell residuals (0/1)      : 0
Output cell centre residuals (0/1)       : 0
Output cell centre solution  (0/1)       : 0
Output cell vertex solution  (0/1)       : 0
Output cell vertex FEM solution (0/1)    : 0
Output cell centre FEM solution (0/1)    : 0
Output cell centre gradients (0/1)       : 0
Output cell centre surface (0/1)         : 0
Output cell centre FEM surface (0/1)     : 0
Output cell centre viscous surface (0/1) : 0
Output cell centre SAS solution (0/1)    : 0
Output cell centre DES solution (0/1)    : 0
```

```
Output cell centre SA  solution (0/1)    : 0
Binary Output (0/1) CC Chimera Halo      : 0 0 1 1
Output surface only with RHO and P (0/1) : 0
Blocks to write starting from 1          : 0
```

Listing C.13: *st.expert.scalars* file for pitching NACA0012 aerofoil.

```
Number of User Defined Scalars      : 1
User Defined Scalars Scheme order   : 3
Explict (0) or Implict (1) scheme   : 1
Restart File                        : 0
No source term (1)                  : 0
Unsteady time step                  : 0.10472
Dual-Time inner iteration           : 20
Dual-Time Residual (Res/Update)     : 0
Dual-Time Tolerance                 : 1.0e-4
CFL number                          : 50.0
Number of iterations                : 20
Fixed flow variable solve           : 0
Local time stepping                 : 1
User Defined Scalars output         : 10
Schmidt number                      : 1e6
Froude number (squared)             : 330
Particle diameter (adimensionalized): 9e-6
fluid and soil density ratio r_f/r_s: 0.00046
friction threshold (adimensionalized):0.11
Terminal velocity (1=yes 0=no)      : 0.0
Lref                                : 1.0
Vref                                : 57.0
Stokes number                       : 0.045
```

Listing C.14: *st.expert.profile* file for pitching NACA0012 aerofoil.

```
Given profile   : profile_inflow profile_UDS

EFP (1/0)       : 0
EFP pressure    : 1.0
Signature (1/0) : 0
Signature file  : 0
Start sig       : 0
Finish sig      : 0
```

The above file will use the content of file profile_inflow for boundary conditions as described below for the flow solution variables. An additional file is needed profile_UDS for the boundary conditions for the user defined scalars but the formatting is the same as the flow solution variables file. A typical inlet profile file is shown below. The first line of the file is the number of blocks where the profile is applied to. This is followed by a set of lines which the block number the face that the profile will be applied to, and the dimensions of the face. The lines after this block of data contain the 5 non dimensional flow variables ρ, u, v, w, p followed by the turbulent variables. In this case the values of k and ω. Caution is need when compiling these files when non constant profiles are being used as the ordering will be depended on the orientation of the blocks.

Listing C.15: *profile_inflow* file for pitching NACA0012 aerofoil

```
2
1 4 44 1
2 4 44 1
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
1.000000 1.00000 0.000000 0.000000 7.9365079 0.001 1.0
..
```

The profile_UDS is similar to the previous, after the three first lines, the value of UDS is imposed for every node. In this case, four nodes were seeded with UDS=1, in the middle of the domain.

Listing C.16: *profile_UDS* file for pitching NACA0012 aerofoil.

```
2
1 4 44 1
2 4 44 1
0.0
0.0
0.0
0.0
0.0
0.0
..
1.0
1.0
1.0
1.0
```

To generate the *un*, *un.Background un.Aerofoil* files, it is necessary to use the following command:

```
./set_rigid_motions un
```

where `un` is the name of the main unsteady input file. For *un.Aerofoil* file, the parameters are then entered as follows:

Listing C.17: input for un files

```
Enter name:
Aerofoil
Are the rigid-body motions periodic? (1=yes, 0=no)
1
Periodic motions
Enter reduced frequency of 1st harmonic :
0.075
Enter number of cycles :
10
Enter number steps per cycles :
400
Enter maximum number of pseudo-time steps/time step
200
Enter tolerance of unsteady residual
0.001
Every how many steps is 3D output file required :
10
Every how many steps is surface output file required :
10
Every how many steps is checkpoint output file required - without solution :
10
Rotation about local x-axis ? (1=yes, 0=no)
0
Translation in local x-direction ? (1=yes, 0=no)
0
Rotation about local y-axis ? (1=yes, 0=no)
0
Translation in local y-direction ? (1=yes, 0=no)
0
Rotation about local z-axis ? (1=yes, 0=no)
1
Is rotation oscillatory ? (1=yes, 0=no)
1
Enter number of harmonics :
1
```

```
Enter mean angle (degs):
-10.3
Enter sine and cosine harmonics (degs):
-8.1 0.0
0.141372 0.000000
Translation in local z-direction ? (1=yes, 0=no)
0
```

While fo *un.Background*:

Listing C.18: input for un files

```
Enter name:
Background
Are the rigid-body motions periodic? (1=yes, 0=no)
1
Periodic motions
Enter reduced frequency of 1st harmonic :
0.075
Enter number of cycles :
10
Enter number steps per cycles :
400
Enter maximum number of pseudo-time steps/time step
200
Enter tolerance of unsteady residual
0.001
Every how many steps is 3D output file required :
10
Every how many steps is surface output file required :
10
Every how many steps is checkpoint output file required - without solution :
10
Rotation about local x-axis ? (1=yes, 0=no)
0
Translation in local x-direction ? (1=yes, 0=no)
0
Rotation about local y-axis ? (1=yes, 0=no)
0
Translation in local y-direction ? (1=yes, 0=no)
0
Rotation about local z-axis ? (1=yes, 0=no)
0
Translation in local z-direction ? (1=yes, 0=no)
0
```

When the input files are ready, it is possible to run HMB for the unsteady case:

```
hmb st un none none
```

HMB3 writes sigle-block files. To visualize use the User Define Scalar the output files have to be concatenated. The command to use is:

```
multi_new naca_chimera0012_CCFEM_UDS 0 4000 10
```

Now they can be plotted using Tecplot as shown in figure C.3. Considering each time-step a separate file, it is possible to use the *edit time strands* option, to animate the solutions.

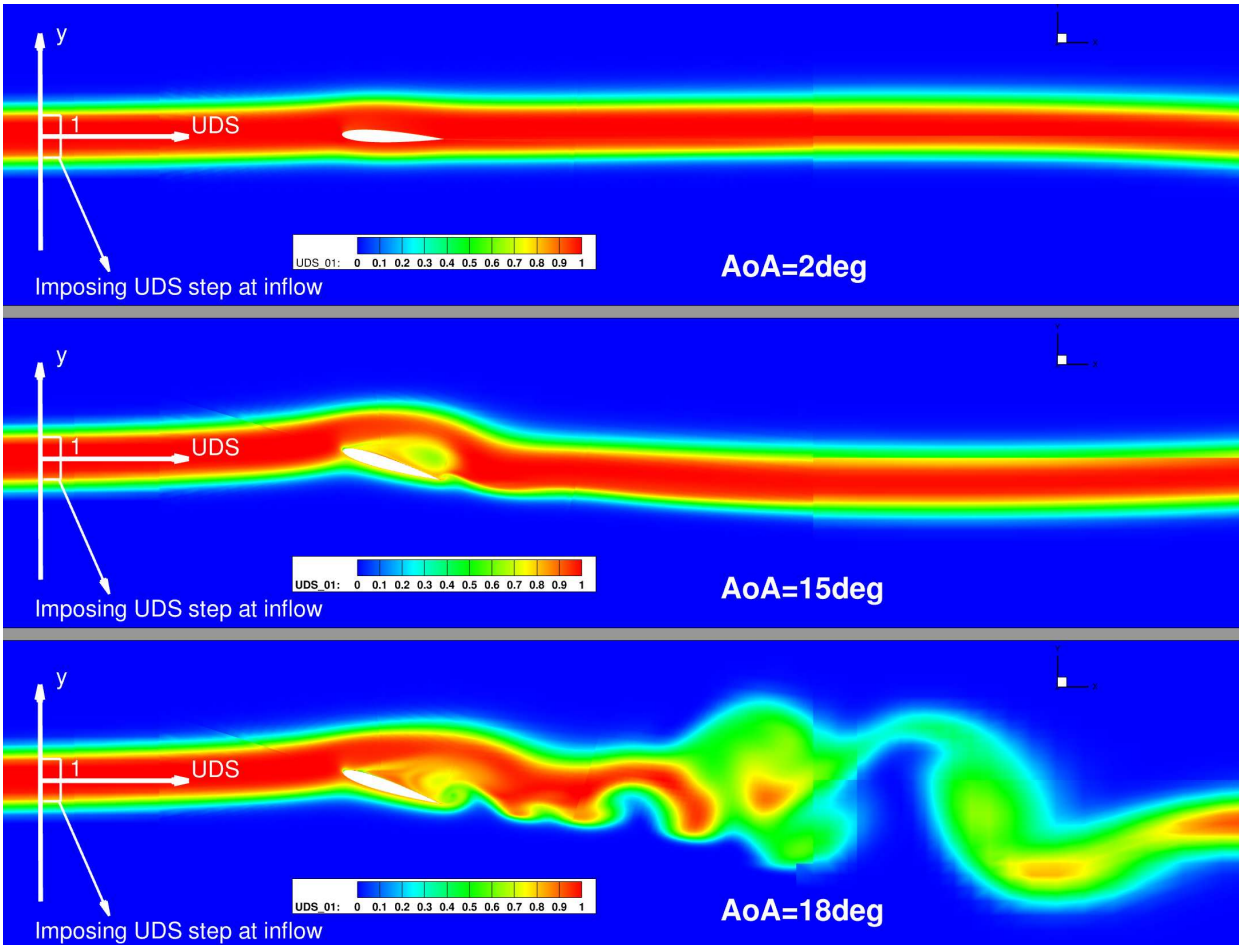


Figure C.3: Eulerian results for unsteady aerofoil. The aerofoil was operating at $Re = 3.8 \cdot 10^6$ and M_∞=0.38.