



University of
Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: Offshore Technology/ Risk Management	Spring semester, 2013 Open
Writer: Reyhaneh Ghahremani (Writer's signature)
Faculty supervisor: Bjørn H. Hjertager External supervisor: Knut Erik Giljarhus	
Title of thesis: CFD simulation of wind condition and thermal gradient around an offshore helideck	
Credits (ECTS): 30	
Key words: <ul style="list-style-type: none">• ABL• Hot plume• Offshore helideck• CFD• OpenFOAM®	Pages: 60 + enclosure: 40 + CD Stavanger, 12/06/2013

Abstract

With the development of offshore deep water fields, helicopter transportation and consequently helicopter deck becomes a vital part of these fields that located far from onshore. To ensure helicopter safe operations, aerodynamic hazards around an offshore platform and helideck need to be identified and risks of such disturbances should be minimized. The main sources of environmental hazards around an offshore platform are vertical air flow fluctuation (σ_z), and temperature increase above the helideck (Morrison, 2001). According to British CAP 437 standard, vertical air flow fluctuation should be less than 1.75 m/s above the helideck. Furthermore, temperature increase should be less than 2°C according to NORSOK C-004 and CAP 437 standards. In order to examine such aerodynamic disturbances, Atmospheric Boundary Layer (ABL) and hot air flow is simulated by OpenFOAM®2.1.1 as an open source Computational Fluid Dynamic (CFD) toolbox.

In this thesis, ABL simulation is first done for a simple geometry, and then it is developed for real offshore fixed platform geometry as wind environment disturbed in vicinity of offshore platform as a major obstacle, and turbulent eddies would be generated. Turbulence models applied for helideck simulations are k- ϵ and RSM turbulence model to assess which one going to present the real situation best. In addition, to examine thermal gradient, hot exhausted air flow is simulated for a simple geometry. To validate simulation results of this project as it is done by an open source code, the models are also simulated by KFX® CFD simulator in parallel by Dr. Giljarhus, and its results compared by simulation results of OpenFOAM.

In ABL simulations of simple geometry, results of horizontal velocity, turbulent kinetic energy, dissipation rate and turbulent eddy viscosity are plotted for different location at geometry length against height to assess how theses profile developed from inlet to outlet, and the OpenFOAM results compared with KFX results. In ABL simulation of real platform geometry, σ_z is calculated as it is one of the main missions of this report, and the results are plotted for coarse and fine mesh for k- ϵ and RSM turbulence models, and also turbulent kinetic energy pattern is shown and compared with KFX. For hot plume simulation, the location of 2°C isotherm is identified and the plume pattern and sizing is compared with KFX.

ABL simulation results for a simple geometry show a great homogeneity with KFX results especially for velocity profile. Regarding hot plume simulation, the plume pattern and its sizing validated as it is fairly similar compare to KFX result. And finally, In ABL helideck simulations, it is found that vertical airflow fluctuation is acceptable for the defined wind condition, as it is less than 1.75 m/s criterion. Furthermore, comparison between simulation results of k- ϵ and RSM turbulence models show that k- ϵ turbulence model generate more turbulent kinetic energy and consequently more vertical air flow fluctuation compare to RSM model.

Acknowledgements

CFD simulation of wind environment and hot plumes around an offshore helideck with OpenFOAM® toolbox was new for me, as it was not directly related to my background educations, and I did not have previous knowledge about OpenFOAM toolbox. In addition, a few articles were available related to my topic that increases the hardness of this master thesis project. Therefore, it was really challenging for me at the first glance, but my passion towards learning new things and importance of this thesis to increase safety of helicopter operations in offshore platforms lead me to apply for this topic which is defined by Scandpower AS.

To overcome these challenges, my thesis supervisor prof. Bjørn H. Hjertager and my co- supervisor, Dr. Knut Erik Giljarhus from Scandpower AS were always available to help me. Definitely, without their trust to me and guidance I can never start this challenging way.

Indeed, Prof. Hjertager held a crash course in UiS regarding to computational fluid dynamics and OpenFOAM toolbox which was crucial to me as it gave me valuable insights and background to start this thesis. I really want to appreciate him because of his assistance and sharing his valuable experience and knowledge of CFD simulations with me.

I also want to appreciate my co- supervisor Dr. Knut Erik Giljarhus. He always conducts me to the best possible way by his valuable knowledge and in-depth understanding of the concepts, OpenFOAM and KFX toolbox.

This thesis carried out in cooperation with Scandpower AS, so I also like to appreciate this company for giving me this opportunity to work on their defined topic, and I wish this thesis be fruitful for them. I also thank Scandpower AS personnel for their hospitality.

And finally, I appreciate my husband because of his love and support.

Table of Contents

Abstract.....	I
Acknowledgements.....	III
1 Introduction	1
1.1 Background.....	1
1.2 Problem definition	2
1.3 Goals and objective	4
1.4 Thesis layout.....	4
2 Theory	7
2.1 Offshore helideck and environmental hazards.....	7
2.1.1 Wind flow.....	8
2.1.2 Hot gas plume	9
2.1.3 Process gas release	10
2.1.4 Flared gas	11
2.1.5 Helideck management system.....	11
2.2 Turbulence	11
2.2.1 Turbulence at offshore helideck.....	12
2.2.2 Turbulence models	14
2.3 Computational fluid dynamics	18
2.3.1 Pre- processor	19
2.3.2 Solver	19
2.3.3 Post - processors.....	19
2.3.4 Governing equations of fluid dynamics.....	20
2.4 Atmospheric boundary layer (ABL)	21
2.5 OpenFOAM®	23
3 ABL Simulation of simple geometry.....	25
3.1 Pre- processing.....	25
3.1.1 Mesh generation.....	25
3.1.2 Initial fields.....	26
3.1.3 Fluid properties.....	27
3.1.4 Turbulence properties	27
3.2 Solving	27

3.2.1	Solution setups.....	27
3.2.2	Solving the case.....	29
3.3	Post - processing	29
4	Hot exhaust plume simulation.....	31
4.1	Pre – processing	31
4.1.1	Mesh generation.....	31
4.1.2	Initial fields.....	32
4.1.3	Fluid properties.....	33
4.1.4	Turbulence properties	33
4.2	Solving	33
4.2.1	Solution setups.....	34
4.2.2	Solving hot plume case	34
4.3	Post- processing	35
5	ABL simulation around an offshore platform and helideck.....	37
5.1	Pre- processing.....	37
5.1.1	Mesh generation.....	37
5.1.2	Initial fields.....	40
5.1.3	Fluid properties.....	40
5.1.4	Turbulence properties	40
5.2	Solving	41
5.2.1	Solution setups.....	41
5.2.2	Solving case 1 and 3, k- ϵ turbulence model	41
5.2.3	Solving case 2 and 4, LRR turbulence model	42
5.3	Post- processing	42
6	Results and discussion	43
6.1	Simulation results of atmospheric boundary layer.....	43
6.2	Simulation results of emission of hot plume	48
6.3	Simulation results of ABL simulation around the helideck.....	50
6.3.1	Comparison results of case 1 and 2 with coarse mesh to case 3 and 4 with finer mesh	50
6.3.2	Turbulent kinetic energy and vertical air flow fluctuation plots.....	52
7	Conclusion.....	57

7.1	Project conclusions.....	57
7.2	Future work.....	58
8	References	59
	Appendices.....	63
	Appendix A: ABL simulations of simple geometry.....	63
	Appendix A.1: blockMeshDict.....	63
	Appendix A.2: ABL conditions.....	64
	Appendix A.3: initial conditions.....	64
	Appendix A.4: side and top patches	64
	Appendix A.5: \mathbf{U} field.....	65
	Appendix A.6: k field.....	66
	Appendix A.7: ϵ field.....	67
	Appendix A.8: P field.....	68
	Appendix A.9: nut field.....	69
	Appendix A.10: controlDict.....	70
	Appendix A.11: fvSchemes	72
	Appendix A.12: fvSolution.....	73
	Appendix B: hot plume simulation	75
	Appendix B.1: blockMeshDict.....	75
	Appendix B.2: initial conditions	78
	Appendix B.3: side and top patches	78
	Appendix B.4: \mathbf{U} field.....	79
	Appendix B.5: k field.....	80
	Appendix B.6: ϵ field.....	81
	Appendix B.7: P field.....	82
	Appendix B.8: T field.....	83
	Appendix B.9: α field.....	84
	Appendix B.10: thermophysicalProperties.....	85
	Appendix B.11: fvSchemes.....	86
	Appendix B.12: fvSolution.....	87
	Appendix C: ABL simulations around helideck	89
	Appendix C.1: blockMeshDict.....	89

Appendix C.2: snappyHexMeshDict	90
Appendix C.3: ABL conditions	93
Appendix C.4: initial conditions	93
Appendix C.5: side and top patches	93
Appendix C.6: U field	94
Appendix C.7: k field	95
Appendix C.8: ε field.....	96
Appendix C.9: R field	97
Appendix C.10: fvSchemes	98
Appendix C.11: fvSolution.....	99
Appendix C.12:sampleDict	101
Appendix C.13:decomposeParDict	101
Appendix D: content of enclosed CD	102

1 Introduction

1.1 Background

Although there were no major fatalities resulting from aerodynamics hazards around an offshore helideck from 1976 to 1999, several incidents such as heavy landing of helicopter have been reported that turbulent wind environment and hot gases were the major reasons for the accident. The most serious report was the Claymore offshore platform accident, in 1995 that lead to very heavy landing of helicopter. This accident attracts the attentions toward an importance of aerodynamic disturbance as a risk factor for helicopter safe operation. However, this accident did not involve major fatalities. (Morrison, 2001)

In early 1960s, it has become apparent that it would be necessary to adapt some requirements for helicopter operations on offshore fixed and mobile installation. In 1964, the draft criteria for helicopter landing areas were published. This report was corrected in 1968 regarding to the proposed size of landing area (CAP 437, May 2012).

The first edition of CAP 437 was published in 1981 by the Civil Aviation Authority (CAA) of the UK. CAP 437 is a standard for offshore helicopter landing areas, and a principal instrument which explicitly imposes an obligation on the platform operator to minimize the occurrence of aerodynamic hazards. Up to now, this standard was revised and updated seven times. The seventh edition of CAP 437 incorporating amendment was published in February 2013.

This standard contains criteria for thermal gradient over the helideck, and limitation for vertical component of velocity fluctuation (Rowe et al.). The quantitative turbulence criterion for helideck design was first defined by CAP 437. In CAP 437 fifth edition that is published in August 2005, still as a general rule, the vertical mean speed above the helideck should not exceed ± 9 m/s for a wind speed up to 25 m/s (Safety Regulation Group, May 2009).

The research carried out by CAA paper 2008/02 to validate the ± 9 m/s wind speed criterion, concluded that this criteria cannot link the vertical airflow to any pilot workload or helicopter performance. Therefore, a new turbulence criterion should be defined (Safety Regulation Group, May 2009).

The above mentioned long lasting criterion was removed from CAP 437 material, and new turbulence criterion implemented in CAP 437- sixth edition published in December 2008. The new turbulence criterion is; as a general rule the standard deviation of the vertical airflow velocity should have values less than 1.75m/s (CAP 437, May 2012). In this standard, there is also a criterion for thermal gradient above the helideck which should have values less than 2°C.

In Norway, there is also a standard for design, arrangement and engineering of helideck, called Norsok standard C-004 for helicopter deck on offshore installation. In this standard, there is also an implication of the importance of analyzing the wind environment and turbulent hot gases around an offshore helideck.

In Norsok C-004, the hot air flows are considered as a major risk factor for helicopter operation, and the position of 2°C isotherm shall be verified by CFD analysis (Norsok Rev. 2, Feb. 2013). This criterion is the same as one required by UK CAP 437.

In addition, according to Norsok C-004, helicopter operation is prohibited in strong turbulent wind conditions because it may affect safety margin and lift capacity of helicopter (Norsok Rev. 1, 2004). According to Norsok C-004 Rev 2, velocity field and velocity gradient together with turbulence plot shall be provided at different levels above the helideck, and limitations on helicopter operations may be considered for values of the vertical velocity fluctuation larger than 2.4 m/s, and restrictions for vertical velocity fluctuation exceeding 3.6 m/s (Norsok Rev. 2, Feb. 2013). Comparing Norsok criterion with CAP 437, UK standard seems to be more conservative as it defined values of vertical air flow fluctuation less than 1.75 m/s for helicopter safe operations.

To measure the wind environment around an offshore installation as well as thermal gradient above the helideck, CFD approach should be applied. The international standard ISO 15138 second edition called HVAC in oil and gas industry and offshore production installation published in 2007, established some guidance and requirement regarding the preparation for CFD analysis. In this standard, there is some requirement for construction of computation model and mesh generation as well as modeling of boundary layers (ISO 15138, 2007). There are also requirements for modeling of wind environment around the helideck and gas turbine exhaust plume dispersal (ISO 15138, 2007). According to this standard and also new version of Norsok C-004, simulation of wind environment around the helideck shall be done by using non-isotropic turbulence models such as Reynolds stress model (ISO 15138, 2007) and (Norsok Rev. 2, Feb. 2013).

1.2 Problem definition

With the development of offshore deep water fields, helicopter transportation becomes a vital part of these fields that located far from onshore. Consequently, offshore helicopter deck becomes an essential part of most offshore platforms. There is also an extensive use of helicopter to support offshore oil and gas fields in Norwegian continental shelf.

To ensure safe approach of helicopter, aerodynamic hazards around an offshore platform need to be identified and the probability of occurrence and consequences of such disturbance should be minimized. Turbulence associated with the wake region downwind of an offshore platform, recognized as the source of highest workload for the pilots according to the survey carried out by Civil Aviation Authority of the UK (Rowe et al.).

The main aerodynamic disturbance around an offshore platform, fall into two main categories. Wind flow around the platform and helideck, and hot gas turbulence. In other

words, the most important sources of environmental hazards are vertical turbulence fluctuation of wind and temperature increase above the helideck (Morrison, 2001).

Simulation of Atmospheric boundary layer (ABL) over a complex terrain is essential to examine the wind load on the offshore structure (Balogh et al., 2012). Wind environment disturbed in vicinity of offshore platform as a major obstacle. Therefore, the strength and direction of wind flow could be changed. In addition, turbulent eddies would be generated, especially at the sharp edges of the platform.

To ensure safe operation of helicopter, vertical turbulence fluctuation needs to be estimated. Turbulence criterion for helideck design is defined by CAP 437 standard. Therefore, care should be taken to ensure that standard deviation of vertical airflow velocity is less than 1.75 m/s above the helideck according to UK standard.

Another important risk factor around an offshore installation is the presence of hot turbulent gases. The major sources of exhaust gases in a platform are hot exhaust plume, release of process gases and flared gas. Hot plumes exhausted by gas turbines, are highly turbulent and buoyant with temperature around 400°C. These turbulent plumes mixed with air and travel by wind direction to helideck. Temperature increase should be less than 2°C averaged over a three seconds interval according to CAP 437 (CAP 437, May 2012). This criterion has a great importance for safe helicopter operation, due to effect of temperature on engine power of helicopter. Increasing temperature causes reduction of air density that lead to reduction of lift capacity of helicopter. Hot plume exhausted from flare tower also could cause the similar problem above the helideck the same as hot plume exhausted from gas turbine. However, they are more visible to pilots compared to gas turbine plume.

Release of gases from process area is another source of hazard for offshore helideck. These released gases can pose a hazard for helicopter operation because of risk of ingestion of hydrocarbon gas mixture into the helicopter engine. This could also be a probability of ignition due to the presence of helicopter as source of ignition.

Prediction of path of these exhaust plume is extremely difficult because it influenced by lots of factors such as wind speed and its direction, level of turbulence, geometry of the platform, velocity of plume and so on (Kulkarni et al., 2006).

1.3 Goals and objective

This master thesis has three main objectives:

- The first goal is simulation of atmospheric boundary layer (ABL) for a simple geometry with OpenFOAM® toolbox which is a computational fluid dynamic (CFD) code. In this phase of project, standard k- ϵ turbulence model is chosen to simulate ABL.

Since, OpenFOAM is an open source code there is no quality certification, and its results needs validation. Therefore, it is tried to validate the result of the modeling by comparing them with the results of simulation by Kameleon FireEx (KFX®) CFD code with the same initial conditions.

- The second goal of this project is simulation of hot exhaust plume with OpenFOAM, and comparison of the results with KFX.
- And finally the third goal is to generate mesh and simulate real platform geometry by `snappyHexMesh` utility. Then, vertical air flow fluctuation would be verified around the platform and helideck by standard k- ϵ turbulence model and Reynolds stress models for a coarse and fine mesh, so totally 4 cases would be simulated, and then their results would be compared with KFX results.

1.4 Thesis layout

Present report contains eight chapters:

The first chapter is introduction that mentioned above.

The 2nd chapter reviews theories relevant to the topic, Such as offshore helideck and different aerodynamics hazards around it, turbulence around offshore helideck and different kinds of turbulence models, theory about computational fluid dynamics (CFD) in conjunction with CFD governing equation, there is also some theory about atmospheric boundary layer, and last part of theory chapter is assigned to OpenFOAM.

In chapter 3, the ABL simulation set up of simple geometry is explained, divided in to three main sections, pre -processing, solving and post - processing.

Chapter 4 covers simulation set ups of hot exhaust plume simulation. The sub-sections of these chapters are the same as chapter 3.

Mesh generation and simulation setups of ABL simulation around a fixed offshore platform and helideck are explained in chapter 5.

In chapter 6, all simulation results are presented and discussed. First, OpenFOAM ABL simulation results of simple geometry are presented and then simulation results compared to KFX results. Then, hot plumes simulation results presented and compared with KFX

results, and finally results of ABL simulation around helideck is presented for all 4 cases and compared with KFX results.

The conclusion and future works are presented in chapter 7

In chapter 8, the materials and references used in this project is listed by means of endNote X6 citation software with Harvard citation style.

And finally, appendices cover the details of fields and setups of OpenFOAM for ABL simulation of simple geometry, hot exhaust plume simulation, ABL simulation around the helideck, and finally the content of attached CD.

2 Theory

2.1 Offshore helideck and environmental hazards

With the development of offshore deep water fields, helicopter transportation becomes a vital part of these fields that located far from onshore. Consequently, offshore helicopter deck is an essential part of most of offshore platforms.

In Norwegian Continental Shelf (NCS), there is also an extensive use of helicopters to support offshore oil and gas fields. There are 74 offshore installations in the Norwegian sector classified as Permanent Rigs with helidecks approved for passenger transport operations according to Jeppesen¹. Many of these permanent rigs were dimensioned for the Sikorsky S-61N helicopter type and have been approved accordingly with those D-value and weight limitations. There are also 38 mobile rigs with offshore helicopter deck. Totally, there are 166 vessels with offshore helideck in Norway² (MAP, 2009). Regarding the above mentioned statistics, helideck is an integral part of offshore Norway sector as well. Therefore, the safety of helicopter transportation has a great importance.



Figure 1: an offshore helideck (Helideckinspect, 2013)

The design of helideck is an integral part of the platform overall design, and placement of offshore facilities such as gas turbine, processing, flare tower, cranes and so on could affect the flows over helideck a great deal.

¹ Source of aeronautical navigation data

² Statistics is for 2009

An accident occurred in 1995 in the Claymore accommodation platform draw an attentions toward environmental hazards to helicopter operation. The accident involved an uncontrollable decent which resulted in very heavy helicopter landing. Fortunately, the accident did not lead to major injuries, but caused very serious damage to helicopter. It is assumed that the helicopter was trapped in the plume exhausted from the gas turbine (Rowe et al.).

Helideck operation is considered as one of the risky part of flight in pilot point of view due to turbulence within the wake region downwind of an offshore platform (Rowe et al.). Therefore, identifying the aerodynamic hazards around an offshore platform has a great importance and would help to define characteristics and magnitude of the turbulence around the helideck. The following hazards show in figure 2, are known as possible aerodynamic disturbance for an offshore helideck.

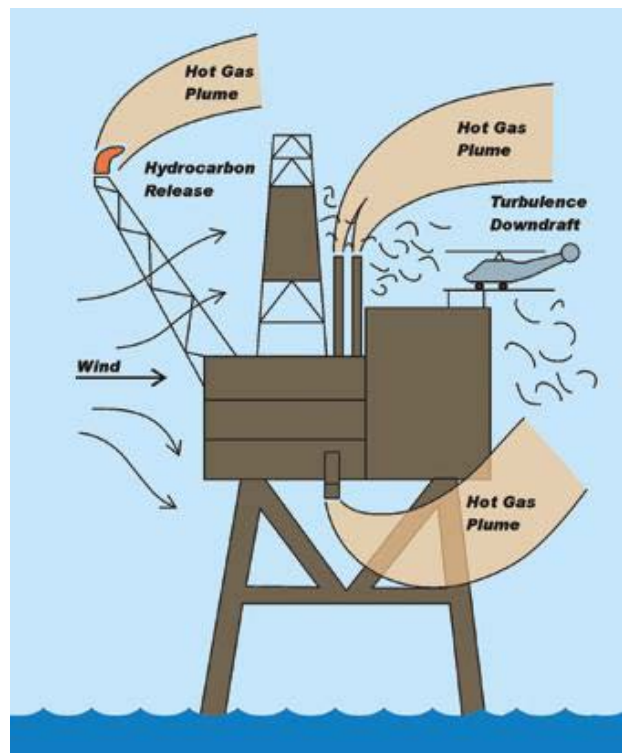


Figure 2: the main aerodynamic hazards around an offshore platform helideck (Rowe et al., 2002)

2.1.1 Wind flow

Wind velocity and wind direction is an important factor for helicopter safe operation and its impacts should be carefully evaluated. Wind flow is affected by structures and obstacles in a platform and its magnitude and direction might be disturbed by them and consequently regions of turbulence would be created, especially near the major edges and corner of platform. Flows are also generally separated at the front edges of the helideck.

Wind velocity generally increased with the height over the sea and contains unsteady turbulent component. In general, a wind flow for a given point in time can be defined as

mean velocity (U) plus the fluctuating part $u'(t)$. This equation (eq 1) is also known as Reynolds decomposition.

$$u(t) = U + u'(t) \quad \text{eq (1)}$$

In above equation, the turbulent part $u'(t)$ typically defined by statistical quantities such as root mean square is mainly used. Velocity fluctuations $u'(t)$ are typically quite large in a region between the upstream edges of helideck and center part of it. Furthermore, the mean velocity (U) sometimes has a turn point in center of helideck that might cause instability (Chen et al., 1995).

In CAP 437, 6th edition a new turbulence criterion is defined for helicopter safe operations. As a general rule, standard deviation of the vertical airflow velocity should not exceed 1.75 m/s. Therefore, helicopter operator should be informed about the wind condition to make sure that this criterion is met before any helicopter operations (CAP 437, May 2012).

In Norway, there is also a standard for design, arrangement and engineering of helideck, called NORSOK standard C-004 for helicopter deck on offshore installation. In this standard, there is also an implication of the importance of analyzing the wind environment. According to this standard, helicopter operation is prohibited in strong turbulent wind conditions because it may affect safety margin and lift capacity of helicopter (NORSOK Rev. 1, 2004). According to NORSOK C-004 Rev 2, velocity field and velocity gradient together with turbulence plot shall be provided at different levels above the helideck, and limitations on helicopter operations may be considered for values of the vertical velocity fluctuation larger than 2.4 m/s, and restrictions for vertical velocity fluctuation exceeding 3.6 m/s (NORSOK Rev. 2, Feb. 2013). Comparing NORSOK criterion with CAP 437, UK standard seems to be more conservative as it defined values of vertical air flow fluctuation less than 1.75 m/s for helicopter safe operations.

However, some questions were raised regarding the above mentioned criterion defined by CAP 437 by the research recently done by a Brazilian group (Silva et al., 2010). First, according the wind tunnel turbulence measurement they have done on the side of ship model in a non-disturbed flow condition, standard deviation of vertical air flow found 1.45m/s, so how it is possible to have standard deviation of 1.75 m/s in a real condition with turbulent flow? Second, how is it possible to assess only vertical turbulence component since frontal velocity gradient has significant effect on helicopter performance? (Silva et al., 2010). Therefore, additional research is required to answer these questions.

2.1.2 Hot gas plume

Gas turbines are responsible for power generation on offshore platforms. They produce hot exhaust streams that reach a temperature of 400°C. They are buoyant, unsteady and less predictable because of plume temperature fluctuations. This turbulent hot air travels mostly upwards and dilutes with air, loose its momentum and gradually travels by local wind direction (Rowe et al.).

In fixed platform hot exhaust plume reach over the helideck by wind direction; however in floating production unit, helideck usually located downwind and steps should be taken to ensure lateral separation between hot plume and helideck. The height of the plume that reached over the helideck depends on height of exhaust, strength of wind, exhaust temperature and its momentum (Rowe et al.).

According to CAP 437, standard for offshore helicopter landing areas, in excess of 2°C temperature average over 3 seconds' time interval above the ambient temperature, care should be account for helicopter operation. Typically, it took about 2 to 5 seconds for a helicopter to respond to the changes, which imply that 3 seconds disturbance should be considered. According to CAP 437, 2°C averages over 3 seconds correspond to hover thrust/margin of approximately 0.4% (Rowe et al.). This criterion is also in accordance with NORSOK C-004 rev 2 requirements, as the position of 2°C isotherm in relation to helideck shall be verified by CFD analysis because presence of hot air flows are considered as a major risk factor for helicopter operation according to this standard (NORSOK Rev. 2, Feb. 2013).

Due to fact that helicopter draws the air downward through its rotor, hot gas plume combination with sudden change in air temperature lead to significant reduction of helicopter lift capacity due to reduction of air density as a result of increasing air temperature. It might also lead to possible momentary stalling of helicopter engines (NORSOK Rev. 1, 2004).

According to CAA paper 2007/02; installing gas turbine exhaust plume visualization would be fruitful by highlighting the hazards for the pilot, especially for the platforms that produce significant amount of gas exhaust. Furthermore, a monitoring program is recommended to monitor the temperature around the platform continuously (CAP 437, May 2012).

2.1.3 Process gas release

Sometimes process gases vented to air for some reasons and sometimes it might release accidentally. The effects of these released gases depend on their momentum, density, temperature and location on platform. In excess of process gases, there is a risk of ingestion of hydrocarbon gas mixture in to the helicopter engine which result in loss of engine control or even flameout. Analysis of engine performance showed that mixture of hydrocarbon- air with concentration of 10% lower flammability limit (LFL) is safe condition in engine control point of view (Rowe et al.). Concentration above 10% LFL might lead helicopter's engine surge or flameout with risk to the helicopter and its passengers (CAP 437, May 2012).

Furthermore, the possibility of ignition should be considered as a result of helicopter which can be a potential ignition source (Rowe et al.).

In addition, operation of emergency blow down system may result in excessive gas concentration that could lead to engine surge and flameout, so routine cold flaring also should consider as a risk factor (CAP 437, May 2012).

Gas release problem is considered as a safety case for a platform and it need some sort of emergency preparedness such as immediate alert to helicopter to stay clear, rather than establishing criterion (Rowe et al.).

2.1.4 Flared gas

Flare towers are normally long structures that design to vent excess and unburned gases of the platform as far as possible. The hot gas plume of flare tower need the same consideration as mentioned for hot gas plume exhausted from the gas turbine, but they have the advantage that they are more visible for pilots (Rowe et al.).

2.1.5 Helideck management system

To reduce the probability of occurrence of above mentioned aerodynamic hazards and reduce the consequence of such disturbance, CAP 99004 proposed the following issues for a helideck:

- Essence of an unobstructed airflow around the helideck
- Location of gas turbine and diesel units as exhausted plume might reach to helideck
- Flared gas and venting of flammable gas
- Location, operation and maintenance of wind recording equipment
- Combined operations in offshore installation with the potential to disturb the airflow or to emit hot exhaust into the helideck path (Morrison, 2001).

Minimizing above mentioned issues by a helideck management system could increase the safety of helicopter operations in offshore installations. Therefore, an audit and control system which monitors compliance with a set of established operational requirements could be designed to minimize aerodynamics hazards (Morrison, 2001).

2.2 Turbulence

Turbulent flow is defined as a three - dimensional highly unsteady flow that contains random change in its quantities such as velocity, pressure and density with significant amount of vorticity. Due to unpredictable nature of turbulence, modeling of a turbulence flow is still one of an unresolved problem that needs further improvement despite significant progress about it until today (Zienkiewicz et al., 2005).

In figure 3, a turbulent flow with random variation in its velocity $u'(t)$ is shown. Where is $u(t) = U + u'(t)$.

eq(1)

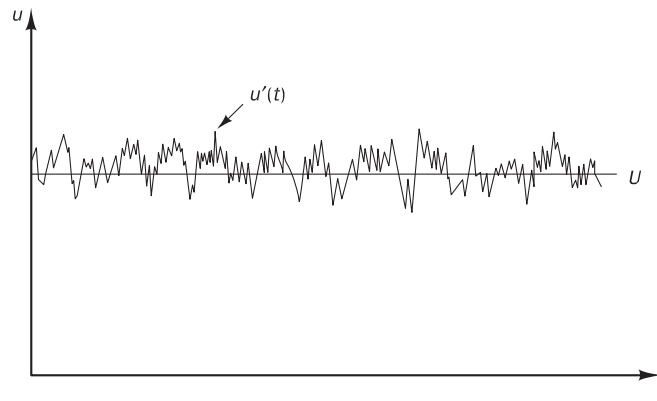


Figure3: typical point velocity measurement in turbulent flow (Hjertager, January 2002)

Dimensionless Reynolds number is widely applied to define the characteristic of three dimensional flows in transition from laminar or turbulent flow. Above a certain amount of Reynolds number a flow become turbulent. In this high Reynolds numbers, the flow governs mostly by inertial force rather than viscous forces. On the other hand, laminar flows occur at low Reynolds numbers where the viscos force dominates mostly. The equation of Reynolds number is:

$$Re = UL / \nu \quad \text{eq (2)}$$

Where U and L are characteristic velocity and length scale respectively and ν is kinematic viscosity.

Generally, turbulence leads to generation of eddies at very different scales, extending from large eddies down to very small ones. Large eddies, are dominated by inertia effects while viscous effects are negligible. These large turbulent eddies are feed by mean flow and consequently they are highly anisotropic. These large scale eddies are more energetic than small ones and transport their properties more effectively.

Kinetic energy transfer from large eddies to smaller ones which is called energy cascade. Smaller turbulent eddies (Kolmogorov scale) are dominated by viscosity, they rotate faster and their energy dissipated and transfer to thermal energy due to work done against viscous stress. Small eddies are universally irrelevant to flows (Kano, 1999) and they are non-directional or isotropic.

2.2.1 Turbulence at offshore helideck

Generally, at a given point wind speed varies randomly with time due to turbulence that leads to atmospheric mixing process which is also affected by the friction of the surface wind blow on it. Therefore, natural winds contains eddies that lead to variation of wind speed both in space and time. The frequency of generated eddies depended on wind speed and size of eddies (Rowe et al.).

In an offshore platform, the number of eddies significantly increase due to separation generated because of offshore platform as an obstacle, especially on sharp edges (Rowe et al.).

To categorize the response of helicopter to generated turbulence, turbulence frequency is used as an index. High frequency turbulence has lower length scale compared to dimension of helicopter. They are also smoother and more integrated with small amplitude, so helicopter responds to them easily. Frequency content above 10 rad/s could cause vibration and more uncomfortable ride to helicopter, and make it difficult for the pilot to read instrument. However, this disturbance is manageable and does not affect the flight path of helicopter (Rowe et al.).

In the turbulence frequency range between 1 to 10 rad/s, which means longer turbulence length scale, the helicopter should start to respond to the turbulence. In this range, the generated disturbance can rotate and push the helicopter around that pilot should normally adapt a control strategy that minimize the disturbance effect and keep the flight on path or position (Rowe et al.).

In low frequency turbulence and larger amplitude, below about 1 rad/s, flight path of helicopter is extremely affected by generated disturbance unless the pilot adopts some corrective measures to keep the flight on track (Rowe et al.).

Obviously, the greater the severity of turbulence happens in final stage of landing, the higher work load the pilot would have to keep the helicopter stabilized and guide to safe landing (Rowe et al.).

Pilot work load could be considered as an inverse measure of the safety. Owing to fact that the higher work load the pilot has, he could have less attention towards other important tasks. To quantify the pilot work load, the Cooper-Harper (C-H) rating method could be applied which show success in practice see figure 4 (Rowe et al.).

- Level 1: corresponds to low work load and good flight quantities that enable pilot to achieve desired performance.
- Level 2: corresponds to tolerable deficiencies that need pilot attention to achieve an adequate performance standard.
- Level 3: in this level there are major deficiencies that need pilot's attention and ability to handle the situation and keep the flight standard performance (Rowe et al.).

Task performance	Pilot rating	Workload	
desired	1	not a factor to minimal	LEVEL 1
	2		
	3		
adequate	4	moderate to extensive	LEVEL 2
	5		
	6		
unacceptable	7	extensive to intense	LEVEL 3
	8		
	9		
untenable	10	uncontrollable	

Figure 4: Cooper-Harper rating scale (Rowe et al.)

Prediction of the helicopter response to the turbulent flow and pilot reaction to this situation is a very complicated process that needs a fully dynamic model of the helicopter aerodynamic behavior as well as pilot behavior (Rowe et al.).

2.2.2 Turbulence models

It is impossible to compute all turbulent flow at one time even with supercomputers. Therefore, turbulence models should be applied (Kano, 1999) somehow that introduce the minimum amount of complexity while representing the essence of the relevant physics (Wilcox, 2006). There is no single turbulence model that could be applied for all kind of problems due to lack of computer resources and storage capacity for highly fluctuating turbulence model. Therefore, different kind of techniques and models are developed by the researchers to describe turbulence as real as possible and defining unknown terms of turbulence equations. These kinds of turbulence models contain some differential equation and algebraic constants which in conjunction with Navier- Stokes equation can simulate the real behavior of turbulent flow (Unhale, 2004). Turbulence models covered in this report are as follow:

2.2.2.1 RANS models (Two- equation models)

The whole range of two- equation models and stress transport models are considered as Reynolds Average Navier- Stokes (RANS) models. In RANS based models, mean quantities are calculated and the fluctuating part calculated from additional modeled variables, consequently these models work with coarser grid that need less computational effort, which means that these models are more economical (Unhale, 2004).

Time – averaging of Navier- Stokes equation generate 6 unknown variables called Reynolds stress. The Reynolds stress tensor should be modeled by relating the Reynolds stresses to the time averaged velocity. According to Boussinesq suggestion, for 3D flows the equation for Reynolds stress is as follow:

$$-\overline{\rho u'_i u'_j} = \mu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \rho k \delta_{ij}, \quad \text{eq(3)}$$

In above formula δ_{ij} is Kroenecker delta, and μ_t is turbulent viscosity.

According CAP 437, special care should be account for vertical air flow velocity fluctuation component (σ_z), so we are interested in formulas for calculation of vertical Reynolds stress component:

$$-\overline{\rho u_z'^2} = 2\mu_t \frac{\partial \bar{u}_z}{\partial z} - \frac{2}{3} \rho k \quad \text{eq (4)}$$

$$-\overline{\rho u_z'^2} = 2C_\mu \rho \frac{k^2}{\varepsilon} \frac{\partial \bar{u}_z}{\partial z} - \frac{2}{3} \rho k \quad \text{eq (5)}$$

$$\overline{u_z'^2} = \frac{2}{3} k - 2C_\mu \frac{k^2}{\varepsilon} \frac{\partial \bar{u}_z}{\partial z} \quad \text{eq (6)}$$

Equation 6 represent vertical Reynolds stress component. From above formulas, expression for standard deviation of vertical velocity component (σ_z) derived as follow:

$$\sigma_z = \sqrt{\frac{2}{3} k - 2C_\mu \frac{k^2}{\varepsilon} \frac{\partial \bar{u}_z}{\partial z}}. \quad \text{eq (7)}$$

The above formula has two main problems. First, σ_z value is dependent on gradient of mean vertical flow velocity. It means that the results would be different for positive and negative value of this gradient. Second, the value under square root expression might become negative leading to imaginary number for standard deviation (Giljarhus, 2013).

In addition to above mentioned formula, turbulent kinetic energy defined as follow:

$$k = \frac{1}{2} (\overline{u_x'^2} + \overline{u_y'^2} + \overline{u_z'^2}) \quad \text{eq (8)}$$

By defining eq 3 and eq 8 the 6 unknown Reynolds stress tensor is decrease to two new unknowns called turbulent viscosity (μ_t) and turbulent kinetic energy (k).

2.2.2.1.1 Standard k- ε turbulence model

Standard k- ε model is the simplest and most validated and robust model of turbulence modeling. k- ε models give large reduction in convergence times, so they make parametric study possible (Murakami, 1993). Furthermore, k- ε turbulence model shows an excellent performance for many industrials flow, such as prediction of mean wind speed (Murakami, 1993). However, this model overestimates in prediction of turbulent kinetic energy around sharp edges (Murakami, 1993). In addition, k- ε model assumes isotropic eddy viscosity and

has poor performance on rotating flows, some unconfined flows and curved boundary layers (Hjertager, January 2002).

Both of turbulent kinetic energy and dissipation rate differential equation can derived from Navier- Stokes equation. Transport equations for turbulent kinetic energy and dissipation are as follow:

For turbulent kinetic energy k

$$\frac{\partial \rho k}{\partial t} + \frac{\partial}{\partial x_i} (\rho U_i k) = \frac{\partial}{\partial x_i} \left[\frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_i} \right] - \rho \overline{u_i u_j} \frac{\partial U_i}{\partial x_j} - \rho \cdot \varepsilon \quad \text{eq (9)}$$

I II III IV V

In above formula:

- I: transient term
- II: convective term
- III: diffusive transport
- IV: production of turbulent kinetic energy
- V: rate of dissipation of turbulent kinetic energy
- σ_k is a constant turbulent Schmidt number for k

For dissipation rate ε

$$\frac{\partial \rho \varepsilon}{\partial t} + \frac{\partial}{\partial x_i} (\rho U_i \varepsilon) = \frac{\partial}{\partial x_i} \left[\frac{\mu_t}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_i} \right] + C_1 \frac{\varepsilon}{k} \left[-\rho \overline{u_i' u_j'} \frac{\partial U_i}{\partial x_j} \right] - C_2 \rho \frac{\varepsilon}{k} \varepsilon \quad \text{eq (10)}$$

I II III IV V

Where:

- I: transient term
- II: convective term
- III: diffusive transport
- IV: production of dissipation of rate of turbulent kinetic energy
- V: rate of dissipation of turbulent kinetic energy dissipation
- σ_ε , C_1 and C_2 are constant

In k- ε turbulence models, turbulence is assumed fully developed and isotropic. It means that velocity fluctuation components in x, y and z directions, can be assumed equal (Silva et al., 2010). This means that the second term ($\frac{\partial \overline{u_z}}{\partial z}$) in equation 7, could be eliminated as it has small values. Therefore for a steady state regime, standard deviation of vertical velocity component (σ_z) is defined as follow:

$$\sigma_z = \sqrt{\frac{2}{3} k} \quad \text{eq(11)}$$

As mentioned before, according to CAP 437 vertical velocity fluctuation (σ_z) should have values less than 1.75 m/s (CAP 437, May 2012).

2.2.2.2 Reynolds Stress Model (RSM)

In Reynolds Stress Model (RSM), instead of eddy viscosity approach, Reynolds stress is calculated directly. RSM is the most general model of all classical turbulence models that has high performance for many complex flows including non-circular ducts and curved flows. However, it is computationally expensive due to addition of new unknowns' variable that leads to more partial differential equations. Furthermore, this model is not widely validated and has some similar problem like as K- ϵ model due to problem with epsilon (Hjertager, January 2002).

According to ISO 15138 standards and NORSOK C-004 Rev 2, CFD simulation of wind environment around the helideck shall be done by applying non-isotropic model, such as RSM model. The non-isotropic models enable the turbulence to be resolved into its vertical and along-wind components which is necessary for the analysis of helicopter safe operations (ISO 15138, 2007).

As RSM models are used to simulate non- isotropic situation, the velocity fluctuation components are not equal in turbulent kinetic energy equation (eq 8) as mentioned for k- ϵ models, so this means:

$$\tau'_{ij} = -\rho \overline{u'_i u'_j} \neq \mu_t \left[\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right] - \frac{2}{3} \delta_{ij} \cdot \rho \cdot k \quad \text{eq(12)}$$

In these situations with non- isotropic transport coefficients, Reynolds stress model defines the unknown correlations of above equation as follow:

$$\frac{\partial}{\partial t} (\rho \cdot \overline{u'_i u'_j}) + \frac{\partial}{\partial x_i} (\rho \cdot U_i \cdot \overline{u'_j u'_j}) = D_{ij} - \rho \cdot \left[\overline{u'_i u'_k} \cdot \frac{\partial U_j}{\partial x_k} + \overline{u'_j u'_k} \cdot \frac{\partial U_i}{\partial x_k} \right] + \epsilon_{ij} + R_{ij} \quad \text{eq(13)}$$

In above equation, the terms on the left hand side are transient and convection of the correlations, and the terms on the right hand side are diffusion, production, dissipation and redistribution of correlations. This equation introduces seven more unknowns, which mean that seven more partial differential equation should be solved and this is computationally expensive.

2.2.2.3 Large Eddy Simulation (LES)

In Large Eddy Simulation (LES), large eddies are computed and smallest eddies known as sub grid scale (SGS) are modeled. Considering the fact that larger eddies are directly influenced by boundary conditions and they transfer most of Reynolds stress, so they should be computed. On the other hand, smaller scale eddies are weaker and their contribution to Reynolds stresses are less than large one, so they have less influence (Wilcox, 2006).

Large eddy simulations are three dimensional and time dependent simulation. Generally, LES simulation is the preferred method because of its accuracy, but due to its expensive

cost, it is not always feasible. However, computational costs of LES are less than Direct Numerical Simulation (DNS) method (Hjertager, January 2002).

In the flow with too high Reynolds number or in complex geometries that application of DNS is difficult, so LES is a preferred method (Hjertager, January 2002).

A major difficulty with LES simulation happens near a solid surface where all small eddies should be simulated. To solve this problem, the grid spacing and timestep required by LES should be decreased which is falls toward using (DNS) as the surface approached. This problem happens because of serious limitation of Reynolds number for LES (Wilcox, 2006).

In LES simulation, it is essential to define which quantities should be calculated precisely, to achieve this goal filtering is used. Filtering helps to have a velocity field that contains only large scale component of total field (Hjertager, January 2002). The filtered velocity is defined as follow:

$$\bar{u}_i(x) = \int G(x, x') u_i(x') dx' \quad \text{eq (14)}$$

The model momentum equation for resolved scale is as follow:

$$\frac{\partial(\rho \bar{u}_i)}{\partial t} + \frac{\partial(\rho \bar{u}_j \bar{u}_i)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \tau_{ij}^S}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right] \quad \text{eq (15)}$$

This equation is derived from Navier-Stokes equation, when it is filtered for constant density which is means that it is for incompressible fluid.

In this equation, τ_{ij}^S is a new unknown called Sub-Grid Scale (SGS) Reynolds stress, which the large scale momentum flux is caused by the action of the small or unresolved scales (Hjertager, January 2002). The models applied to estimate SGS Reynolds stress are called sub-grid scale model.

2.3 Computational fluid dynamics

When a fluid flow needed in depth analysis and study, normally there is three common way towards this problem. First, it can be done by means of experimental fluid dynamics (EFD) which applies experimental methodology and procedures for solving fluid such as wind tunnel experiment. Second approach is theoretical (analytical) fluid dynamics by use of mathematics for physical problem which is only capable to find exact solutions for simple geometries, and the last way is numerical approach or computational fluid dynamics (CFD). CFD simulation could be well applied in physical phenomena when experimental measurements are not possible such as large scales cases, hazardous situation such as explosion simulation as well as physical modeling such as Atmospheric Boundary Layer (ABL) simulation (Stern et al., 2005).

CFD is a very powerful technique that helps to solve wide ranges of physical problems such as fluid flow, heat transfer, and chemical reaction and so on by means of computational methods including mathematical and physical modeling numerical methods such as grid generations, solvers and etc.

Although, helideck assessments are usually done by wind tunnel experiment rather than CFD, due to complexity of platform geometry and difficulties of turbulence modeling with CFD simulation. Therefore, wind tunnel tests are considered more reliable and economic for this case (Rowe et al.). However, it does not mean that CFD simulations are not applicable in assessment of aerodynamic hazards around the helideck.

Every CFD codes contain three main elements:

2.3.1 Pre-processor

In this step required inputs such as geometry definition and computational domain specification, grid generation, initial boundary condition, appropriate physical model, and fluid properties for a given flow problem is defined. These setting would be later used by solver.

CFD solution accuracy is highly depended on number of cells, the larger the number of cells, the higher accuracy. However, it should be considered that with increasing the number of cells, the computation cost increased as more powerful computer is needed. Therefore, the optimal approach is to use non- uniform mesh. It means that use finer mesh where there is large variation from point to point and coarse mesh in regions with little change (Versteeg and Malalasekera, 2007).

2.3.2 Solver

The numerical solution techniques used by CFD codes, is finite volume method. This method has three main steps.

- First, integration of governing equations of fluid flows over all finite control volume of domain.
- Second, conversion of resulting integral equation into algebraic equations, called discretization.
- Finally, solution of the algebraic equations by an iterative method. Normally, physical phenomena chosen to solve a complex and non- linear case need number of iterations to get converged (Versteeg and Malalasekera, 2007).

2.3.3 Post - processors

In this phase, there are some visualization tool that display grid and geometry, vector and contour, and animations.

2.3.4 Governing equations of fluid dynamics

In fluid dynamic there are eight unknown dependent variables. They are pressure (p), three velocity components (u, v, w), temperature (T), viscosity (μ), heat conductivity (k) and density (ρ). To find these eight unknowns, eight equations are needed. These equations are as follow:

2.3.4.1 Continuity equation

This is an equation for conservation of mass.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0. \quad \text{eq (16)}$$

For incompressible fluid, density is constant. Therefore, the second term also would be zero ($\nabla \cdot \vec{u} = 0$). This means that incompressible fluids are divergence free.

2.3.4.2 Momentum balance

The Navier-Stokes equation derived from the Newton's second law which implies that the sum of the forces act on a body is equal to time change of the momentum of the body.

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{f}. \quad \text{eq (17)}$$

In above equation \mathbf{v} is the flow velocity, ρ is the fluid density, p is the pressure, \mathbf{T} is the total stress tensor, and \mathbf{f} represents body forces (per unit volume) acting on the fluid. Above equation is in a vector form.

2.3.4.3 Energy balance

Energy equation is governed by first law of thermodynamic, which implies that the change in the internal energy of a closed system is equal to the amount of heat supplied to the system, minus the amount of work done by the system on its surroundings

$$\frac{\partial(\rho h)}{\partial t} + \frac{\partial(\rho u_j h)}{\partial x_j} = -\frac{\partial q_j}{\partial x_j} + \frac{Dp}{Dt} + \tau_{ij} \frac{\partial u_i}{\partial x_j} + \dot{S} \quad \text{eq (18)}$$

In above equation, S is a source term and h is enthalpy of system.

2.3.4.4 Equation of state

For an ideal gas the equation of state is as follow

$$pV = nRT. \quad \text{eq (19)}$$

In above equation V is volume which is equal to mass divided by density.

2.3.4.5 Empirical relation for μ and k

Division of fluids in Newtonian and non-Newtonian fluids is done on the basis of relation between stress and strain. In Newtonian fluids the relation between stress and corresponding rate of strain is direct and linear. The Newton's law of viscosity is as follow:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \cdot \frac{\partial u_i}{\partial x_i} \cdot \delta_{ij} \quad \text{eq (20)}$$

The last equation is for finding the last unknown variable called heat conductivity (k). Second law of thermodynamics or Fourier's law relates the heat flux (q_i) to the local temperature gradient.

$$q_i = -k \frac{\partial T}{\partial x_i} \quad \text{eq (21)}$$

2.4 Atmospheric boundary layer (ABL)

Simulation of Atmospheric Boundary Layer (ABL) flows over a complex terrain is essential to estimate the wind load on an offshore platform (Balogh et al., 2012).

Computational Fluid Dynamics (CFD) codes are widely applied in simulation of ABL. In simulation of ABL by CFD codes, RANS turbulence modeling and standard K- ϵ model are mostly used. Although, LES models can provide more accurate simulation for the turbulent flow compare to RANS model due to better characterization of turbulent flow, but because of their expensive cost, RANS model is still the preferred model. RANS model in conjunction with two equation models provide fast answers. To achieve reliable solution with this model, turbulence model should sufficiently be defined for the problem in a question (Balogh et al., 2012).

However, application of two- equation RANS model without modification of wall roughness often result in unwanted decay in the mean wind speed profile and especially turbulent kinetic energy at the inlet of computational domain, due to an inconsistency between fully developed inflow condition for ABL, wall function formulation and turbulence model (Balogh et al., 2012).

To remove above mentioned inconsistency, "a reformulation of the wall function base on the aerodynamic roughness and on the derivation of the kinetic energy inlet profile from the solution of turbulent kinetic energy transport equation" is proposed by Parente et al. recent work on 2011 (Balogh et al., 2012).

In a CFD simulation of ABL, the mean wind speed profile and turbulence quantities that define for inlet of the computational domain are generally fully developed, equilibrium profiles. These profiles should represent the roughness of that part of upstream terrain that is not included in the computational domain, such as terrain upstream of the inlet plane. Therefore, this is expressed by means of appropriate aerodynamic roughness length (z_0) (Blocken et al., 2007). Almost in all CFD codes, accurate description of the flow near the ground surface is essential part of simulation of lower part of ABL (Blocken et al., 2007). Theoretically, z_0 is a height at which the wind speed becomes near zero, when there is a logarithmic wind profile. Typical aerodynamic roughness lengths for different terrain types are present in table 1.

Table1: surface roughness lengths for different type of terrain (Ramli et al., 2009)

Terrain description	z_0 (m)	Surface
Open sea, fetch at least 5 km	0.0002	Sea
Mud flats, snow; no vegetation, no obstacles	0.005	Smooth
Open flat terrain; grass, few isolated obstacles	0.03	Open
Low crops; occasional large obstacles	0.10	Roughly Open
High crops; scattered obstacles	0.25	Rough
parkland, bushes; numerous obstacles	0.5	Very Rough
Regular large obstacle coverage (suburb, forest)	1.0	Closed
City center with high- and low-rise buildings	≥ 2	Chaotic

The wind profile for different aerodynamic roughness is shown in figure 5 which is plotted according to below formula. From this formula, wind profile would have a logarithmic shape as it is typical for atmospheric boundary layers.

$$\bar{U}_w(z) = \frac{u_\tau}{k} \ln\left(\frac{z}{z_0}\right) \quad \text{eq (22)}$$

Here, κ is Von Karman constant equal to 0.41, and u_τ is friction velocity or u star calculated at reference wind velocity of 3m/s for 5 meter height above ground. As illustrated in figure 5, the wind profile is more vertical for small value of roughness compare to higher values of the roughness length. (Osenbroch, July 2006)

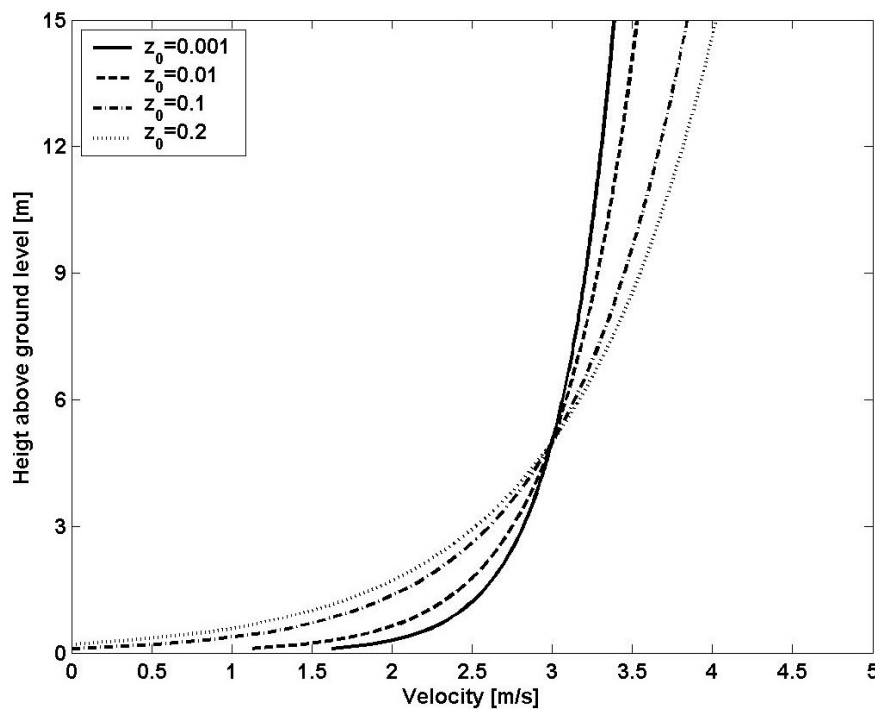


Figure 5: the relation between roughness length and shape of velocity profile applied at boundaries (Osenbroch, July 2006)

Generally due to importance of the surface roughness and high Reynolds number associated with ABL flow which is because of low kinematic viscosity of air flow, the use of wall function is required for near wall modeling (Blocken et al., 2007). In figure 6 the universal law of wall for a smooth surface is shown. In this figure the dash line represent smooth surface.

In below figure k_s^+ is dimensionless equivalent sand-grain roughness height.

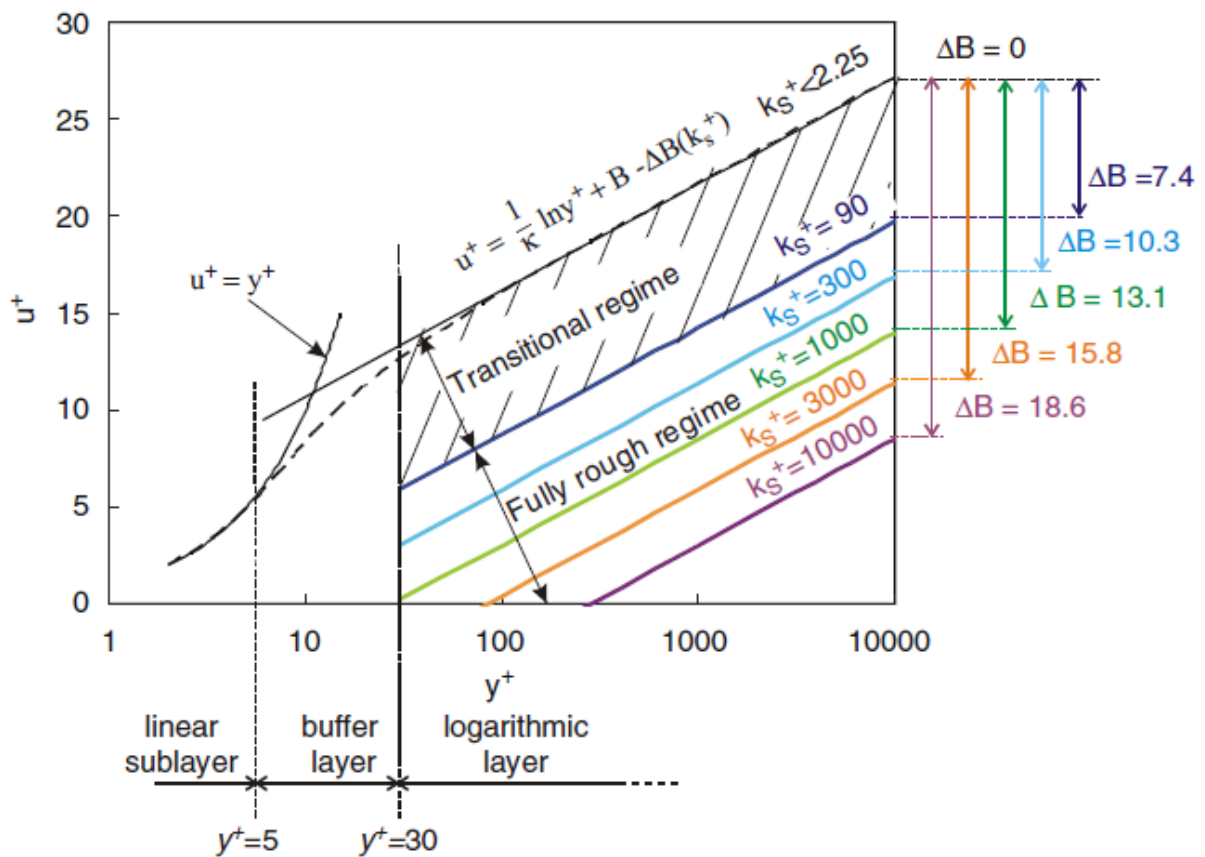


Figure 6: law of wall for smooth and sand-grain roughness surface with k_s^+ as a parameter (Blocken et al., 2007)

In above figure, the laminar law ($u^+ = y^+$) is valid for values less than $y^+ = 5$, and logarithmic law ($u^+ = 1/\kappa \ln(y^+) + B$) is applied for fully turbulent regime for y^+ values about $y^+ = 30$ up to $y^+ = 500-1000$. In this formula B is the integration constant around 5- 5.4 (Blocken et al., 2007).

2.5 OpenFOAM®

OpenFOAM® is an acronym of Open source Field Operation and Manipulation. OpenFOAM toolbox was developed at Imperial College of London during 1990-1999. First, it was issued as a commercial code between 1999 to 2004, but in 2005 it released as an open source code under GPL license until now (Hjertager, December 2009a).

OpenFOAM library is written in C++ and design to solve complex physics problem using finite volume method and unstructured discretization grid effectively. Variety range of fluid dynamics problem can be solved by this software. However, it is not a CFD solver, but a finite volume toolbox. Although OpenFOAM is a tool box, it contains lots of solver for compressible and incompressible fluid in steady state, transient and turbulent flows as well as laminar. It has also the ability for parallel calculation which is an advantage for large and complex cases (Churchfield et al., 2010).

However, regarding to fact that OpenFOAM is an open source program, there is lack of high quality documentation and updated guides which lead to absence of quality certification (Lysenko et al., 2012). Therefore, the simulation results need validation. In this thesis, it is tried to validate the result of the modeling by comparing them with the results of modeling by other CFD code, so the models are also simulated with Kameleon FireEx KFX^{®3} in parallel. The KFX simulations are done by Dr. Giljarhus.

OpenFOAM is executed on Linux machine, so to be able to run it on windows computer it needs another platform that in my case it is virtual box. In this project, OpenFOAM version 2.1.1 is used due to its new boundary condition type for ABL simulation of nut field.

Every OpenFOAM case contains three main folders. 0 folder, contain initial field information. Constant folder, contain poly mesh folder, turbulence property and material property, and finally system folder that contains solution setting, time step and solver setting for each field. The details of each of above mentioned folders would be explained for my cases in the next chapters.

³ Kameleon FireEx KFX[®] is a 3D transient finite volume CFD simulator, dedicated to gas dispersion, flare and fire simulation. It is also used for other industrial analysis for more than 20 years (ComputIT,2013).

3 ABL Simulation of simple geometry

To find scientific literature relevant to simulation ABL and thermal gradient above an offshore helideck, different websites such as scienceDirect, Scopus, and EBSCO were searched, but a few relevant topics for verification of aerodynamic hazards around an offshore helideck founded. The most relevant article was “CFD simulation and wind tunnel investigation of a FPSO offshore helideck turbulent flow” which simulate wind flow around helideck by means of ANSYS CFX software (Silva et al., 2010). However, no relevant article was founded regarding to simulation of thermal gradient or even heat transfer.

Simulation with OpenFOAM needs lots of patience, due to fact that sometimes trial and error is necessary to find optimum values for relaxation factors or even selection of boundary condition type and initial values for each fields. In addition, finding the source of error happens during the simulation need lots of concentration and patience as well.

As mentioned before, to estimate wind load on a structure simulation of ABL is necessary. Turbulence model chosen to simulate ABL in this phase of project is standard k- ϵ turbulence model, as this model is one of the most verified turbulence models and shows reasonable prediction for many flows.

To simulate ABL with OpenFOAM as a CFD code, three main steps are needed. Pre - processing, Solving and finally post- processing of the results, the work done in each step would be explained in following.

3.1 Pre- processing

For this phase of master thesis project, TurbineSiting tutorial used as starting point, and some change applied to it to adapt it for defined purposed.

3.1.1 Mesh generation

The geometry used for this phase of project is a simple 2D box with dimension of 600m length and 100m heights with 300 cells in x direction and 75 cells in z direction. As in this case a simple geometry is chosen, hexahedron meshes is applied which is provide high quality solutions with fewer cells. The accuracy of CFD simulation is highly influenced by number of cells, which means that the higher the number of cells, the better accurate result. However, by increasing number of cells, the time and costs of simulation increased as more powerful computer is needed (Versteeg and Malalasekera, 2007).

In this project, simulations were carried out on a 32 bit operating system laptop computer with limited processing speed, so it is tried to choose number of cells somehow that have reasonable computational time as well as accurate results. In this case, the total number of cells is 22500 which are generated by `blockMesh` utility that generate structured multi-blocks meshes. The information for mesh setup and number of point coordinates and lists of points defining boundaries in OpenFOAM, are defined on `blockMeshDict` which is

located under `polymesh` on constant folder. The detail of `blockMeshDict` definition is on appendix A.1 .

To ensure about mesh quality, `checkMesh` command executed. Some of the most important mesh quality parameters are non- orthogonality and skewness (Hjertager, December 2009a). In this case, the mesh was definitely OK with maximum non-orthogonality of zero and maximum skewness of 2.27e-13.

The mesh geometry and its domain are shown in figure 7. As seen from mesh figure, to increase accuracy of results, finer meshes is applied near ground.

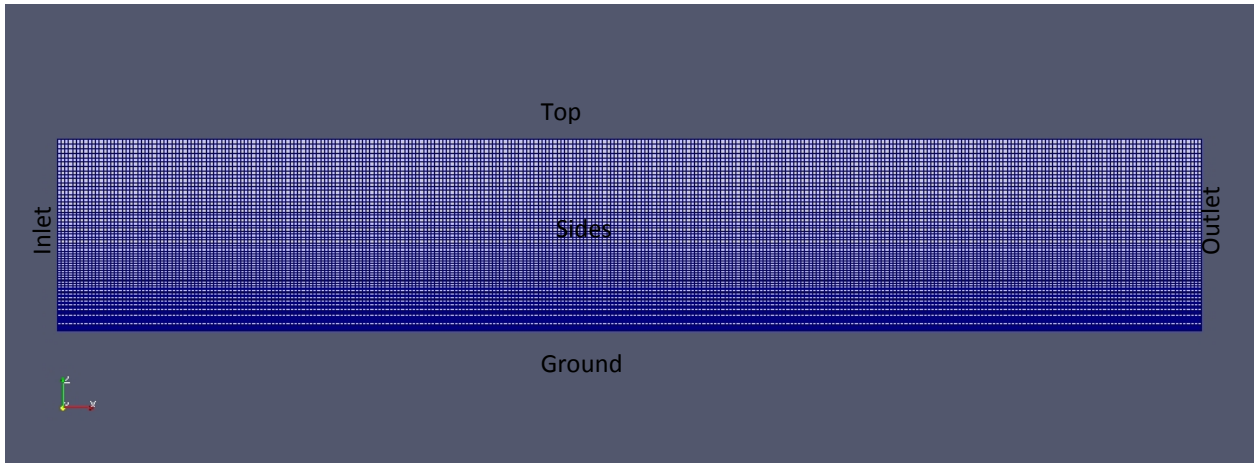


Figure 7: mesh and computational domains

3.1.2 Initial fields

After mesh generation, the next step is to define initial fields and boundary condition types for each of fields.

As in this phase the aim is simulation of ABL, the initial values needed to be calculated at inlet are: friction velocity (u_τ), turbulent kinetic energy (k) and epsilon (ϵ). The inlet boundary condition formulas for ABL are proposed by Richards and Hoxey and they are widely applied in simulation of neutral ABL as follow (Richards and Hoxey, 1993):

$$u = \frac{u_\tau}{\kappa} \ln\left(\frac{z+z_0}{z_0}\right) \quad \text{eq(23)}$$

In this report the horizontal velocity (u) is 8 m/s, reference height (z) is 10m, roughness height (z_0) is assumed 0.0002 as in our case we have calm sea, and κ is Von Karman constant equal to 0.41. Therefore, the friction velocity (u_τ) or u star calculated from above formula is 0.303.

After calculation of u_τ , the epsilon value is calculated by equation 24.

$$\epsilon = \frac{u_\tau^3}{\kappa(z+z_0)} \quad \text{eq(24)}$$

Turbulent kinetic energy value would be calculated by below formula. In this project the constant value of C_μ is considered 0.033, as this value also used in KFX simulation.

$$k = \frac{u_\tau^2}{C_\mu^{0.5}} \quad \text{eq(25)}$$

The inlet boundary type for velocity and epsilon are `atmBoundaryLayerInletVelocity` and `atmBoundaryLayerInletEpsilon` respectively and for k it is `fixedValue`. In outlet, the boundary type is `zeroGradient` for U, k and ϵ . The detailed definitions of each of these fields are available in appendices.

In this report, OpenFOAM version 2.1.1 is used for simulation. This version has a new boundary type for nut for modeling of ABL compare to its old version (OpenFOAM, 2011-2013). Hence, `nutkAtmRoughWallFunction` boundary type is chosen for ground in nut fields. See appendix A.9.

3.1.3 Fluid properties

Physical properties of the fluid define in `transportProperties` file which is located under constant folder in OpenFOAM. In this case our fluid is air with kinematic viscosity of around $1.5e-05 \text{ m}^2/\text{s}$ at 300K temperature (engineeringtoolbox, 2013). As air is a Newtonian fluid with constant value of viscosity, transport model in `transportProperties` file is selected as Newtonian.

3.1.4 Turbulence properties

Turbulence property is defined in `RASProperties` file under constant directory. Turbulence model selected for this phase of simulation is k- ϵ turbulence model.

3.2 Solving

In this phase as we have incompressible, isothermal and steady flow, `simpleFoam` solver is chosen.

3.2.1 Solution setups

Solution setups are defined under system folder within three main files called `controlDict`, `fvSolution` and `fvSchemes`.

In `controlDict` file, number of iteration is set to 1000. As the case is steady state the time step setting does not make any changes, so `deltaT` is set to 1, and writing interval for results is set to 50.

Functions are defined in `controlDict` to write down information from simulation. In ABL simulation, to be able to plot U, k, ϵ and nut values against height, five sets are defined in `controlDict` under functions to store results of simulation at different x locations. In this case 0, 150, 300, 450 and 600 points in the length selected, to monitor how U, k, ϵ and nut values are developed along x axis and 5 diagrams plotted for each of mentioned variables against height. Detail of `controlDict` file is available on appendix A.10.

Solvers and various control parameters for every unknown variable are defined under `fvsolution` file under system directory (Hjertager, December 2009a). For this case, unknown variables are p , U , k and ϵ .

In OpenFOAM 3 types of solvers implemented: preconditioned conjugate or bi-conjugate gradient called `PCG/PBiCG`, algebraic multi-grid solver called `CAMG`, and finally solver with smoothing called `smoothSolver` (Hjertager, December 2009a).

For this case, `CAMG` and `smoothSolver` are selected for unknown variables mentioned above, as they are recommended solvers due to their speed (Hjertager, December 2009a). In general, `CAMG` solver is used for pressure and `smoothSolver` are applied for other variables. These `smoothSolver` needs a smoothing scheme and the recommended one is `GaussSeidel` (Hjertager, December 2009a) which is used in this project.

Furthermore, solver tolerance and relative tolerance need to be defined for each of mentioned unknown variables. Solver tolerance defines how accurately the equation is solved and if the initial residual of an equation is lower than this value, the equation is not solved anymore (Hjertager, December 2009a). Relative tolerance defines how accurately an equation is solved inside each iteration loop. In the case with convergence problem this value can set to zero to force the solver to converge (Hjertager, December 2009a). However, in ABL simulation of this case there is no convergence problem, so `relTol` for U , p , k and ϵ is set to 0.1.

Pressure- velocity coupling method applied in steady state cases is `SIMPLE`. `SIMPLE` stands for Semi Implicit Method for Pressure Linked Equation. `SIMPLE` algorithm is applied to make sure there is a correct linkage between pressure and velocity (Versteeg and Malalasekera, 2007).

In steady state cases the solutions need to be under-relaxed. Under relaxation factor are well-known tuning tools that force the steady state cases converge (Hjertager, December 2009a). In this project, typical relaxation factor are used, 0.3 for pressure and 0.7 for other variables. The detail of `fvsolution` file is on appendix A.12.

Conversion of resulting integral equation in to system of algebraic equation is called discretization (Versteeg and Malalasekera, 2007) which is necessary for all equation terms needed to be solved. In OpenFOAM discretization is controlled in a dictionary called `fvschemes` which is located under system directory (Hjertager, December 2009a).

Time derivative scheme should be indicated in `fvschemes`, as this case is steady state `ddtSchemes` is chosen. Then, `divSchemes` should be defined. `divSchemes` represent the convection scheme and it is the most important discretization scheme in CFD (Hjertager, December 2009a). In this project, Gaussian integration used for both gradient and divergence term with different interpolation schemes. For this case, a 2nd order upwind

interpolation scheme, `linearUpwindV`, is used for velocity vector to improve handling of it.

Before start to solve the case, the source code of below file is changed according to Dr. Giljarhus idea: `atmBoundaryLayerInletVelocityFvPatchVectorField.C`. The change is made on decreasing of the code default roughness length value of 0.001 as it was bigger than Z_0 value of 0.0002 used in this project, and consequently it did not take into account Z_0 value of 0.0002 assumed in this project. This change also made on `atmBoundaryLayerInletEpsilonFvPatchVectorField.C`, and Z_0 valued decreased with the same reason. In addition, U profile default definition in `atmBoundaryLayerInletVelocityFvPatchVectorField.C` file is corrected due to fact that for the values bigger than H_{ref} , it plots a straight line for velocity rather than a curve profile. After these changes are made on above mentioned files, the utility is compiled with `./Allwmake` command.

3.2.2 Solving the case

After above mentioned settings, the case is solved by executing below command:

```
simpleFoam | tee out
```

In this case, the simple solution converged after 594 iterations. In steady state cases, only result of last iteration is correct and acceptable and time steps solutions are only intermediate un-converged results (Hjertager, December 2009a). The above command writes the outputs on a text file named "out".

3.3 Post - processing

After solving the case, the results should be checked for convergence. To achieve this goal, individual residuals can be extracted from output text file with the following command:

```
foamLog out
```

The above command creates a directory `logs` containing several files. In this file, the most important values are initial residuals of p , k , U_x , ϵ , and local continuity error (Hjertager, December 2009b) which indicate mass balance error in the solution (Hjertager, December 2009a). Initial residuals should approach zero when a solution is converged, and local continuity error should have very small value, otherwise it indicates problems in model (Hjertager, December 2009a). To check these values, they are plotted with `xmgrace` in the following diagram.

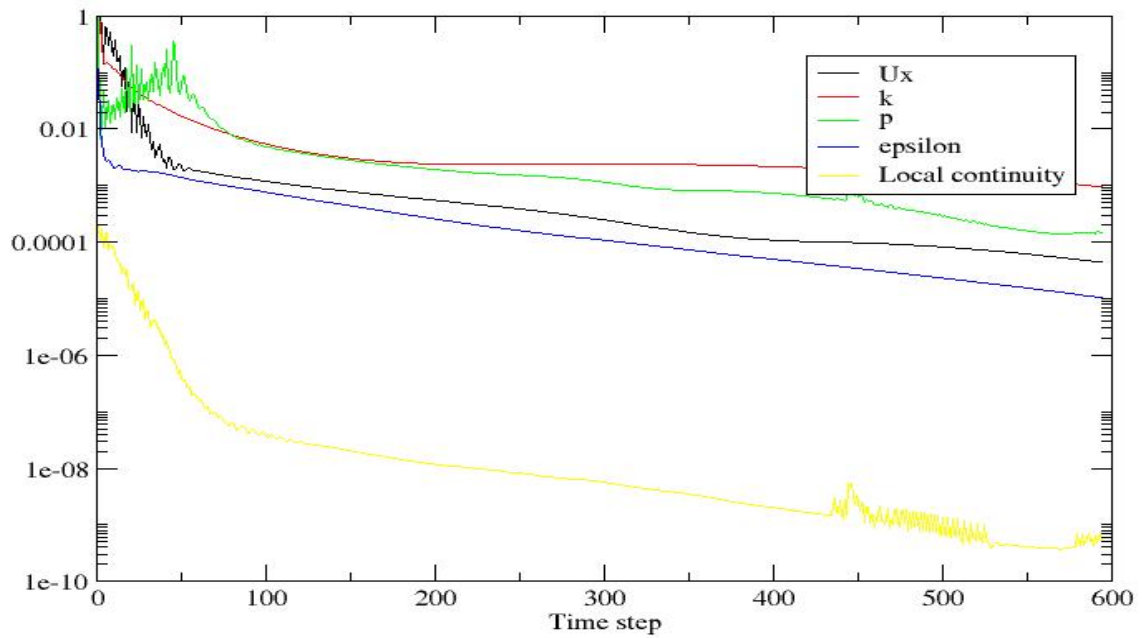


Figure 8: initial residuals and local continuity error

As seen from above figure, initial residuals and local continuity error have small acceptable values for this case.

4 Hot exhaust plume simulation

Prediction of hot smoke flow exhausted from an offshore installation and its path is a challenging task as it affects by different factors such as smoke flow rate, smoke temperature, wind velocity and its direction, level of turbulence and geometry of the platform. Therefore, different initial conditions such as different smoke flow velocity and smoke temperature can lead to different smoke flow or plume patterns (Gong and Li, 2010).

The use of CFD modeling in case of smoke flow movement is increasingly popular (Deckers et al., 2012). By computational fluid dynamic method, hot plume could be modeled numerically. According to NORSOK standard CFD analysis should be applied to verify the correct location and sizing of 2°C isotherm above the helideck (NORSOK Rev. 2, Feb. 2013).

Smoke flow in an offshore installation could be wind - dominated, buoyancy - dominated caused by the reduced density of the smoke, or both of them. Smoke flow dominated by wind if the exhausted hot gas source has relatively small power. In sufficiently powerful plume source, the flow would be dominated by buoyancy (Gong and Li, 2010).

In this phase of master thesis project, a three-dimensional, steady state, turbulent, buoyancy-driven, hot compressible air flow would be simulated by means of OpenFOAM toolbox. In this case, an ABL flow with reference velocity of 8m/s and temperature of 283°K blow from left side of the computational domain or inlet, and air flow rise up with velocity of 100m/s and temperature of 483°K from a secondary inlet located at the bottom of the geometry.

No single tutorial available in OpenFOAM which can use as starting point for our defined purpose, so finding appropriate initial boundary type for each field was quite difficult. In addition, there was no article or relevant scientific literature for hot smoke simulation with OpenFOAM, so there was low inspiration at beginning of this phase. Therefore, some trial and error is used to find suitable boundary condition type and initial values for each field. Simulations set ups of this phase are as follow:

4.1 Pre - processing

4.1.1 Mesh generation

The geometry used for this phase of project is a 3D box with a secondary inlet on its bottom with 50m distance from zero point of box in both x and y directions. The dimension of computational domain is 350m x100m x 100m with 100 cells at length, 60 cells at width and 70 cells at height. Dimension of secondary inlet is 1m in length and 1m in width, with 2x2 numbers of cells.

To find appropriate length of computational domain mentioned above, some trial and errors were needed as with 250 meter length which was defined at first run it seems that the plume path is tend to be longer than the computational domain, so longer computational

domain was needed. Finally, 350 meter length seems to be enough to show the entire plume length.

The above mentioned numbers of cells is divided in to nine hex blocks. The detail definition of vertices, blocks and boundary are available in `blockMeshDict` in appendix B.1. Total numbers of cells in this phase are 442680 that are generated by `blockMesh`.

Simple grading is also applied to have finer cells near the secondary inlet and near wall (near wall treatment), to have higher accuracy in the simulation and its results.

The boundary types for sides of computational domain are defined as `symmetryPlane`, and for ground it is defined as `wall`, and the others are `patch`.

The geometry and computational domain is shown in figure 9. In this computational domain the air flow enters form the left side of the geometry or inlet, it mixed with the exhausted plume comes from the secondary inlet at the bottom and goes out from right side of geometry or outlet.

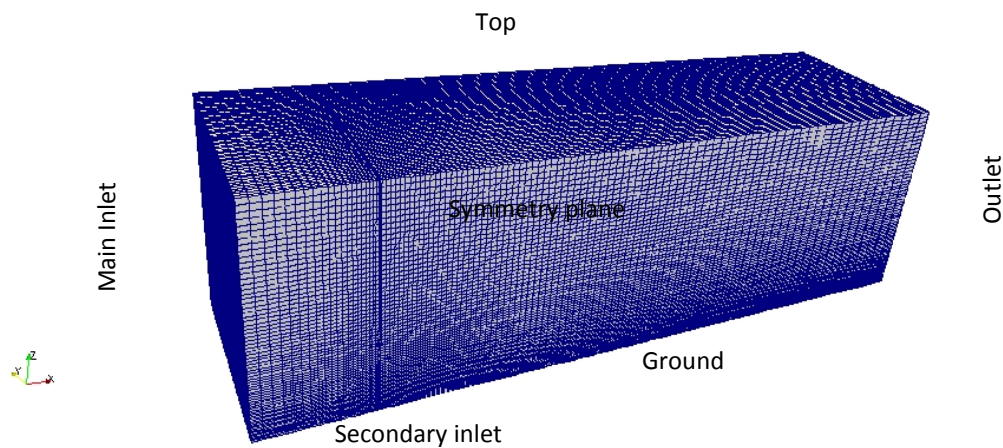


Figure 9: mesh and computational domains

4.1.2 Initial fields

This phase of simulations showed some sort of sensitivity regarding to initial values of turbulence kinetic energy and dissipation rate; somehow that for some initial values of k and ε the simulation diverges after few iterations. Therefore, appropriate initial values for k and ε is provided by Dr.Giljarhus from KFX simulation result.

This phase of simulation has new initial fields. They are temperature (T) with dimension of K, α and μ because energy equation needs to be solved, as compressible fluids are non-isothermals and always there is heat transfer.

To be able to have ABL velocity profile instead of uniform fixed velocity at the main inlet, `groovyBC` type is applied at main inlet of U field. Definition of `valueExpression` is

according to Richards and Hoxey proposed equation for ABL velocity at inlet (eq 23). Velocity at secondary inlet has fixed value of 100 m/s.

As `groovyBC` boundary condition applied at main inlet for velocity, it is also should be applied for turbulent dissipation rate. However, with this approach the simulation diverged after some iterations. Therefore, the model first solved only with `groovyBC` boundary condition type for velocity, and after it got converged, the epsilon main inlet boundary type changed to `groovyBC` type as well. With this method the solution get converged after few more iterations.

In T field, `fixedValue` boundary condition type is used for the wall, as ground is not considered as an adiabatic wall without any heat transfer. Therefore, `zeroGradient` boundary condition is not used for the ground.

In all of fields top of geometry is defined as `slip` and sides have `symmetryPlane` boundary type. The details of initial field definition are available in appendices B.2 to B.9.

4.1.3 Fluid properties

As in this phase of project there is a heat transfer, thermal material properties for the fluid is needed to define instead of transport properties under constant directory. The models used to calculate thermal properties are defined after `thermoType` in `thermophysicalProperties` file (Hjertager, December 2009a). In this case, hot plumes properties are assumed to be constant, and ideal gas equation of state would be solved to calculate the density variation of hot air rise up from secondary inlet.

Numerical value of fluid properties which is air in this model is defined after `mixture`. Detail of `thermophysicalProperties` file is available on appendix B.10.

In heat transfer models, the other necessary input is gravitational acceleration defined in `g` file which is located under constant directory.

4.1.4 Turbulence properties

As exhaust plumes are usually turbulent, so models are required to examine the effects of unresolved small scales movements. The most common turbulence model applied for simulation of smoke flow is $k-\epsilon$ model. However, this model has a disadvantage of low temporal resolution as it time- averaged eddies (Hu and Fukuchi, 2003). In this phase, standard $k-\epsilon$ turbulence model is employed in `RASProperties`.

4.2 Solving

Usually, in heat transfer models there is a compressible fluid. In our case, the hot air rise upward with velocity 100m/s. Therefore, Mach numbers have values around 0.34 which implies subsonic regime. In this case also there is buoyancy, as with increasing the temperature, density would be decreased and the hot smoke flow would be rise up.

To sum up, as this case is assumed steady state, and we have subsonic regime with buoyancy, the most appropriate solver for this case is `buoyantSimpleFoam`.

4.2.1 Solution setups

Number of iteration for this case set to 1000 with writing interval of 50 in `controlDict` file. In this phase as `groovyBC` type is applied at velocity and epsilon inlet field, following definition is added in `controlDict` file before the application line:

```
libs (
    "libOpenFOAM.so"
    "libsimpleSwakFunctionObjects.so"
    "libswakFunctionObjects.so"
    "libgroovyBC.so"
);
```

The solvers used for this phase are preconditioned conjugate (PCG) for pressure and `p_rgh`, and bi-conjugate gradient (PBiCG) for the others variables which is defined under `fvSolution` file, as in cases with convergence problem these solvers are the preferred ones, however the execution time would be longer.

The SIMPLE algorithm is used for Pressure- velocity coupling as our case is steady state, and Relaxation factors are defined to force the solution gets converged. Heat transfer simulation usually converges slowly, so it is recommended to use under-relaxation factors 0.9 or above for enthalpy (Bakker, 2002-2006b). However, the optimal combination of relaxation factors for this case found by trial and error approach to achieve a balance between execution time and results accuracy. The detail of `fvSolution` file is on appendix B.12.

Time derivative scheme; `ddtSchemes`; is set to steady state under `fvSchemes`, and second-order upwind scheme was employed for velocity in discretizing of the convection terms. Detail of `fvSchemes` setting is on appendix B.11.

4.2.2 Solving hot plume case

After above mentioned solution setups, the case is solved by executing the following command:

```
buoyantSimpleFoam | tee out
```

In this case the solution with `groovyBC` for velocity converged after 830 iterations and execution time was around 3.5 hours. Then, the entire folder copied and the epsilon boundary condition type changed to `groovyBC`, and SIMPLE solution converged after 9 more iterations.

4.3 Post-processing

To verify the simulation results graphically, `paraFoam` post-processing utility is used to visualize the result. To observe smoke flow pattern, a contour is applied for T at isotherm of 285°K . As shown in figure 10, the smoke flow first rise up due to momentum buoyancy, and then spreads longitudinally to the right side of computational domain due to wind flow.

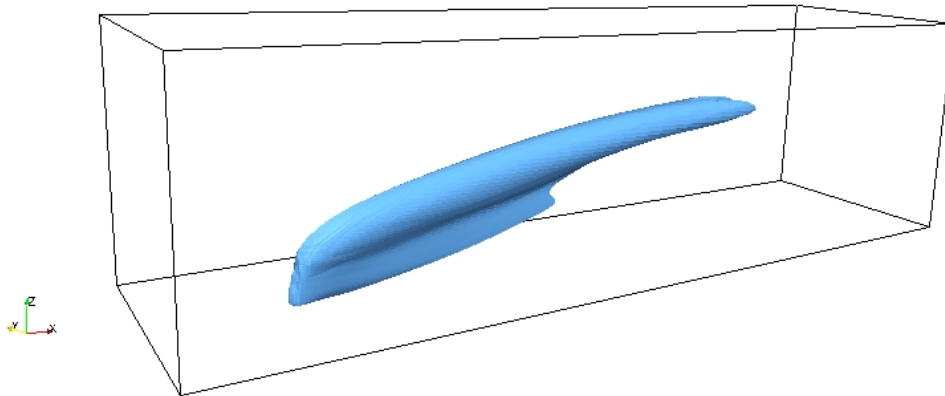


Figure 10: exhausted plume pattern

To check the residuals of simulation results, `FoamLog` command executed to extract the residuals from the out text file as follow:

```
foamLog out
```

Initial residuals of U_x , ϵ , k , p_rgh , h , and local continuity error are plotted in the below figure.

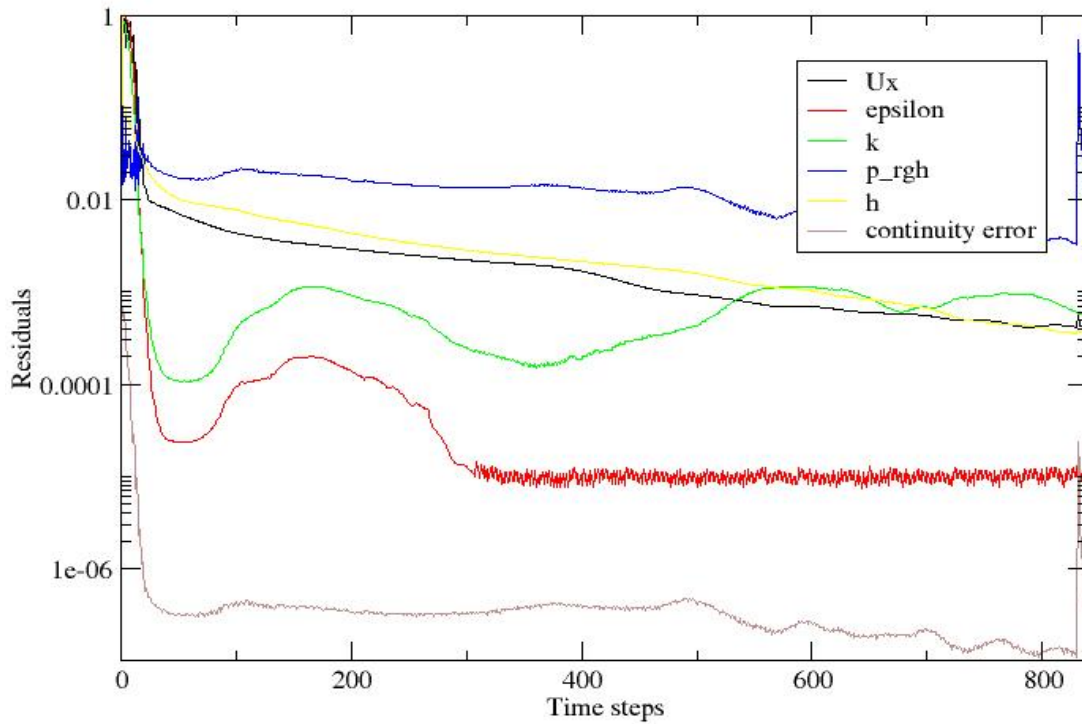


Figure 11: initial residuals and local continuity error

In above figure, all initial residuals and local continuity error have small values which confirm that the results are converged. The peak seen in above figure for p-rgh and local continuity error at time step 830 is due to fact that the boundary condition type is changed to `groovyBC` for epsilon after the case got converged in 830 time step. However, these initial residuals drop again after few more iterations.

5 ABL simulation around an offshore platform and helideck

The main objective of this thesis is to ensure helicopter safe approach on an offshore installation. To achieve this goal, wind environment around an offshore platform should be verified by CFD simulation of atmospheric boundary layer.

Offshore platforms have very complex geometries with lots of utilities and equipment on it. Structures located on platform disturb both wind flow and hot gases coming out from the platform, and therefore turbulence and eddies would be generated especially at sharp edges. Consequently, to verify the vertical velocity fluctuation above the helideck, in this phase of project real fixed platform geometry is used to access the wind environment around the platform and helideck.

In this case, an ABL wind flow blow horizontally from right to left of the geometry with reference velocity of 8m/s. Turbulence models used for this phase of simulation are k- ϵ and RSM turbulence model to examine which one is going to present the real situation best, and also access if they give the same results. Moreover, application of non - isotropic turbulence models such as RSM is compulsory by ISO 15138 and NORSOK C-004 rev 2 as RSM turbulent models predict complex flows such as flow containing rotation and separation accurately (Bakker, 2002-2006a).

In addition, to observe the effect of mesh refinement on the results, the simulations are done for coarse and fine meshes. Therefore, a coarse base mesh is firstly defined with no layer, to observe how the model would work and examine the accuracy of initial settings. After the case was converged with acceptable results and physical interpretation, a finer mesh in conjunction with layer addition is applied, so totally there are 4 cases in this phase of project:

- Case 1: k- ϵ turbulence model with coarse mesh
- Case 2: LRR turbulence model with coarse mesh
- Case 3: k- ϵ turbulence model with fine mesh
- Case 4: LRR turbulence model with fine mesh

The details of simulation setups are as follow:

5.1 Pre- processing

5.1.1 Mesh generation

In this phase of project, the mesh is generated by `snappyHexMesh` utility which is an advanced automatic meshing tool available in OpenFOAM. `snappyHexMesh` generate 3D hexa dominated body aligned meshes with mesh quality guarantee (Hjertager, December 2009a).

This automatic mesh generation tool needs three inputs:

First, the surface geometry in Stereolithography (.stl) is required. In this phase, the STL surface of platform geometry is provided by Dr. Giljarhus as a starting point which is needed for mesh generation by `snappyHexMesh`. This file is located under constant directory, on trisurface folder. In this directory the command `surfaceCheck` executed to ensure about soundness of the stl geometry. This utility generates two output folders called `badFaces` and `problemFaces` (Hjertager, December 2009b). For this project, the first file contains one set of bad faces, and the second folder contains 2784 faces with edges connected to only one face.

The second step is creating the base mesh by `blockMesh` tool. The computational domain bounding box around the platform is (-200 -45 -0.0008) to (280 115 160) shown in figure 12 for the coarse mesh.

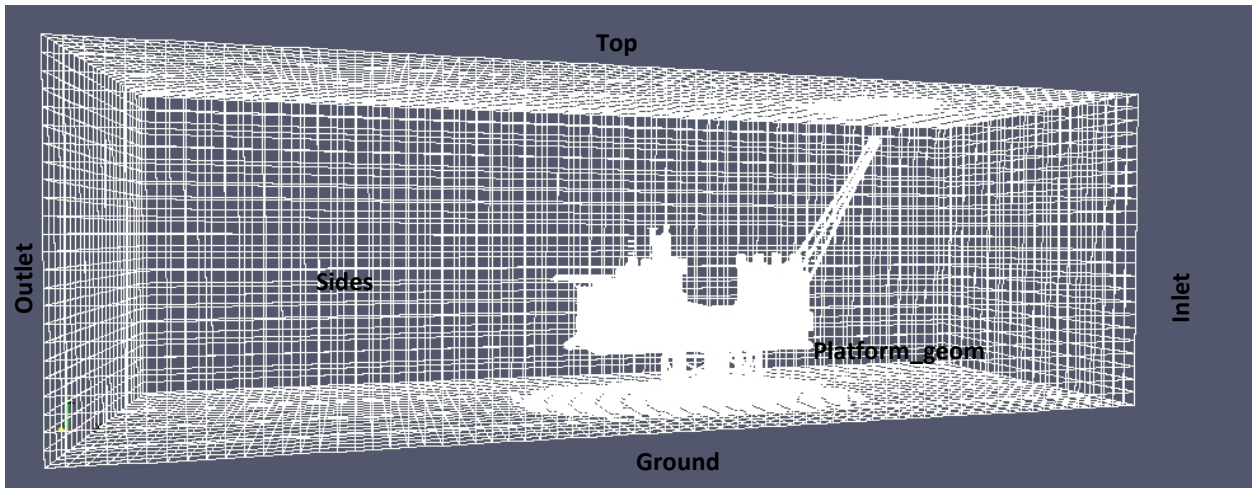


Figure 12: geometry and computational domain in OpenFOAM

Mesh and computational domain which is used for KFX simulation at this phase of report is shown in below figure:

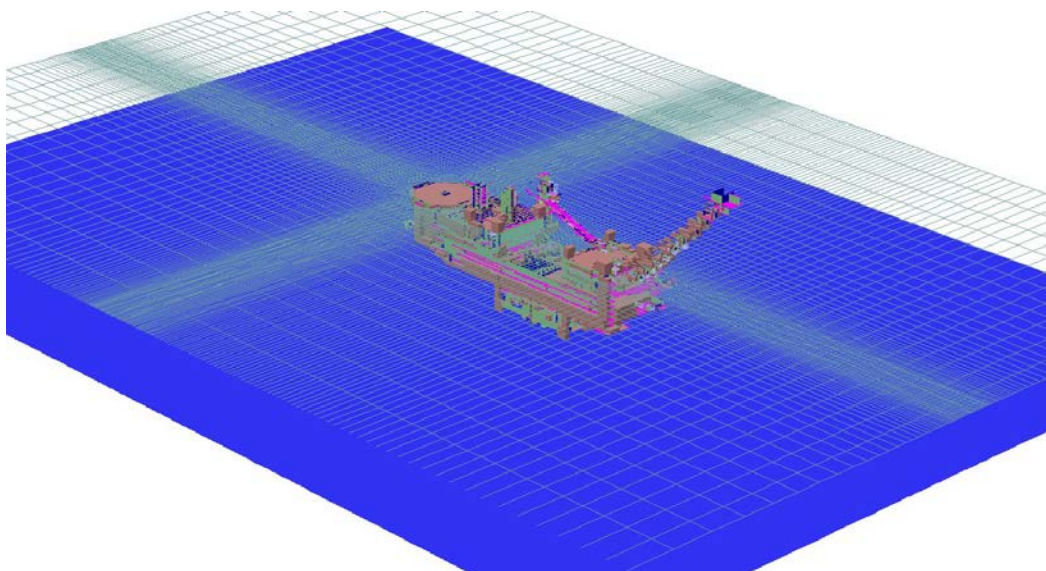


Figure 13: geometry and computational domain in KFX

Total numbers of cells for coarse and fine mesh are 609417 and 2390662 respectively, including hexahedra, tetrahedral and prisms cells. The detail of `blockMeshDict` is available on appendix C.1.

And finally the last step of mesh generation is to set up general controls for refinement, snapping, layers addition and mesh quality criteria on `snappyHexMeshDict` located under system directory. In this project, firstly `castellatedMesh` and `snap` switched true on `snappyHexMeshDict` while `addLayers` switched off, and `snappyHexMesh` utility executed for coarse mesh to make sure that automatic mesh generation is OK. In this step, two time dump added for refinement and snapping mesh phase. Then, to have finer mesh the `addLayers` switched true to add layers for the platform geometry and `refinementSurfaces` for the platform increased from levels (3 3) to (4 4). After these changes, the `snappyHexMesh` utility executed again for the finer mesh, and then the third time-dumps added because of layer addition. Detail of `snappyHexMeshDict` for coarse mesh is on appendices C.2.

To transfer mesh setting to 0 folder time dump `snappyHexMesh -overwrite` command executed. Because, for solving the case initial fields are needed and this command overwrite the last mesh setting to 0 folder which initials fields are located there.

To ensure about mesh quality, `checkMesh` utility is executed after above mentioned steps. For the coarse mesh, maximum and average mesh non-orthogonality was 49.17 and 7.12 respectively which it is OK, and Max skewness was about 4.17 which is acceptable as it has values less than 10. For the finer mesh, non-orthogonality was also OK with values of 52.79 and 6.73 for maximum and average respectively, and Max skewness was about 3.03 which is acceptable as it is less than 10. To see created snappy mesh better on paraFoam, extract cells by region filter is used. Below figure shows a cut generated by `ExtractCellsByRegion` filter normal to y plane.

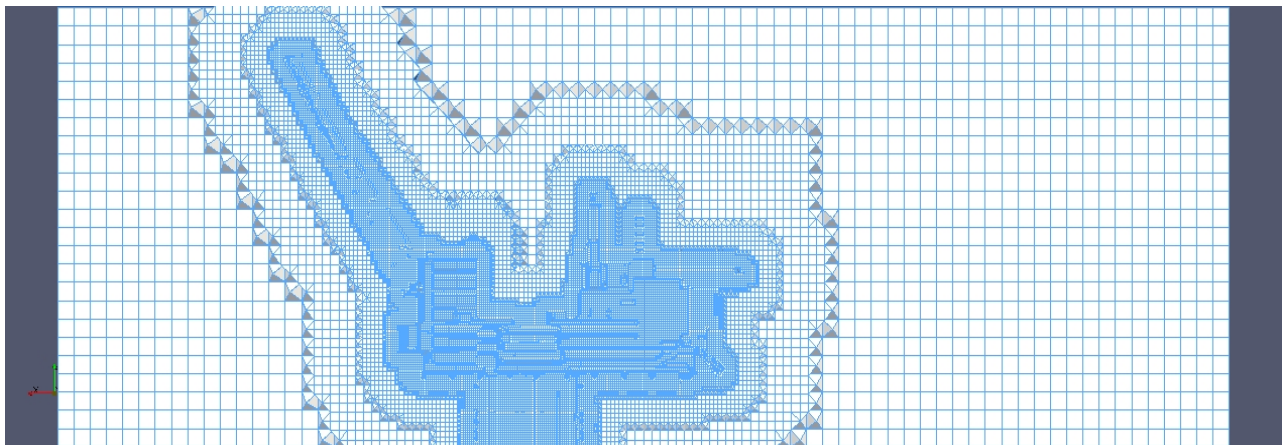


Figure 14: snappy mesh of coarse mesh

5.1.2 Initial fields

After mesh generation, a new boundary field added called `platform_geom`, this patch should be added to all initial fields, and the boundary type defined for it; is the same as boundary type defined for ground in each of the initial fields.

To simulate ABL airflow at inlet, `groovyBC` is used at inlet for U, k and epsilon fields. The detail definitions of each of these fields are mentioned in appendices. The reference velocity is assumed 8m/s at reference height of 10 meter the same as the first phase of simulation mentioned in chapter 3 of this report, and the other initial fields are the same as well. However, the wind blows from right side of domain to left side in this phase of report.

In addition, as for case 2 and 4 RSM turbulence model is chosen, there is a new unknown variable called R. R field contain Reynolds stress tensor. The values of Reynolds tensor are calculated by following formula. These formulas give values for standard deviation of the three wind components for flat uniform terrain (Franceschi et al., 2009) which can also applied for our cases as we have neutral boundary layer.

$$\frac{\sigma_x}{u^*}=2.39 \quad \text{eq(26)}$$

$$\frac{\sigma_y}{u^*}=1.92 \quad \text{eq(27)}$$

$$\frac{\sigma_z}{u^*}=1.25 \quad \text{eq(28)}$$

The u^* value is 0.3 as calculated in chapter 3.1.2. The values of Reynolds stress tensor for case 2 and 4 are calculated based on above formulas. The detail of R field is mentioned on appendices C.3 and C.9.

5.1.3 Fluid properties

In this case the fluid is air with the same property as is mentioned in chapter 3.1.3.

5.1.4 Turbulence properties

In this phase, we have a turbulent flow situation with high Reynolds number, consequently turbulence models should be chosen that appropriate for high Reynolds numbers.

Furthermore, fully turbulent regime is assumed which means that models that only valid for fully turbulent flows such as k- ϵ model can be applied.

In this report, `kEpsilon` turbulence model is selected for case 1 and 3, and for case 2 and 4 `LRR` turbulence model is chosen in `RASProperties`, as `LRR` is one of the most commonly used Reynolds Stress Models, and application of non-isotropic turbulence models for helideck studies is obligatory according to ISO 15138 and NORSOK C-004 rev2.

5.2 Solving

For all 4 cases, `simpleFoam` solver is chosen, due to fact that air flow is an incompressible fluid, and steady state regime is assumed.

5.2.1 Solution setups

In `fvSolution` file, `CAMG` and `smoothSolver` are selected to solve unknown variables of this phase of simulation, as they are recommended solvers due to their speed (Hjertager, December 2009a). In general, `CAMG` solver is used for pressure and `smooth` solver applied for other variables like `U`, `k`, `epsilon` and `R`. These `smoothSolver` needs a smoothing scheme and the recommended one is `GaussSeidel` (Hjertager, December 2009a) which is used in this phase.

Relaxation factors used for these steady state cases are quite typical, 0.3 for pressure and 0.6 for other variables.

Due to non-orthogonality value which is around 50 for the cases, corrections need to be defined via both `fvSolution` and `fvSchemes` files. Hence, 3 correction loops set for SIMPLE pressure-velocity coupling in `fvSolution` file. The detail of `fvSolution` file is in appendix C.11.

Mesh quality also influences the setting of discretization scheme file, so `snGradSchemes` and `laplacianSchemes` set to corrected in `fvSchemes` file to let the correction loops that defined in `fvSolution` file, work.

For case 2 and 4, with LRR turbulence model, appropriate discretization scheme is also added for new unknown variable `R`. The detail of `fvSchemes` file setting is on appendix C.10.

In this phase of simulation as mentioned before `snappyHexMesh` is used for mesh generation. This utility needs `decomposeParDict` to work. Due to fact that, for our cases it was not necessary to execute the computations in parallel as they were not too big, the `numberOfSubdomains` and other `coeff` sets to 1. Details of `decomposeParDict` is in appendix C.13.

To be able to extract values of `k`, `R`, `epsilon` and `U` against computational domain height after simulation, a set defined in `sampleDict` file under system directory. Details of `sampleDict` file is mentioned in appendix C.12.

5.2.2 Solving case 1 and 3, k- ϵ turbulence model

After above mentioned settings, the case 1 and 3 solved by executing below command:

```
simpleFoam | tee out
```

In case 1 with `kEpsilon` turbulence model and coarse mesh, the simple solution converged after 174 iterations, and for case 3 with finer mesh the simulation converged after 230 iterations.

5.2.3 Solving case 2 and 4, LRR turbulence model

After case 1 and 3 with `k-ε` turbulence model gets converged, their results would be used as starting point for case 2 and 4 with `LRR` turbulence model, as with this method RSM model forced to converge sooner. Therefore, entire of case 1 and case 3 folders with coarse and fine mesh are copied as starting point case 2 and case 4. As mentioned before `LRR` as a Reynolds stress models has new unknown variables called Reynolds stress or `R` field. The initial values of Reynolds tensor are calculated according to 26-28 equations. Then the turbulence model on `RAS` properties is changed to `LRR`, and `fvSolution` file and `fvScheme` are completed for case 2 And 4.

After these changes made, `simpleFoam` command executed to solve the cases.

5.3 Post- processing

After above mentioned 4 cases got converged, turbulent kinetic energy and vertical airflow fluctuations' patterns are shown in `paraFOAM` for further assessments. These figures and their interpretations are presented in the next chapter.

6 Results and discussion

It is known that CFD results depend on the mesh chosen for simulation (Deckers et al., 2012), so selecting the right mesh has a great importance. In this project, it is tried to achieve a balance between the mesh size and expected computational time due to limitation of the laptop that simulation executed on it, as the more mesh quality, the more computational time needed.

Furthermore, in this master project it is tried to consider real conditions in the simulations. However, there is some limitation in these modeling. The most important limitation in ABL and thermal gradient simulation is the geometry used, which is simplified a great deal for both of them. For simulation of ABL a simple 2D box is used as the main focus was simulation of wind profile against height, so the width had less importance. This simplification also decreases the computational time as used laptop had limited processing speed, and for hot plume simulation as mentioned before a simplified 3D box with a secondary inlet on bottom of box is used.

However, the geometry limitation is solved a great deal for ABL simulation, as in the third phase of this master thesis project, a real fixed platform geometry is used.

The results of each phase of project simulations are mentioned in the following paragraphs.

6.1 Simulation results of atmospheric boundary layer

The results of ABL simulation for a simple 2D box are presented in following diagrams. To observe how horizontal velocity, turbulent kinetic energy, dissipation rate and turbulent eddy viscosity profiles are developed from inlet to outlet, they are plotted for different lengths of geometry against height, they are measured at inlet $x=0$, at length of $x=150$, $x=300$, $x=450$ and at outlet which is $x=600$ by functioned defined in `controlDict` as mentioned before, and then the values are plotted.

The simulation results of OpenFOAM are plotted for U_x , k , ϵ , and ν_{nut} in the following diagrams against distance from wall or ground to the height of computational domain.

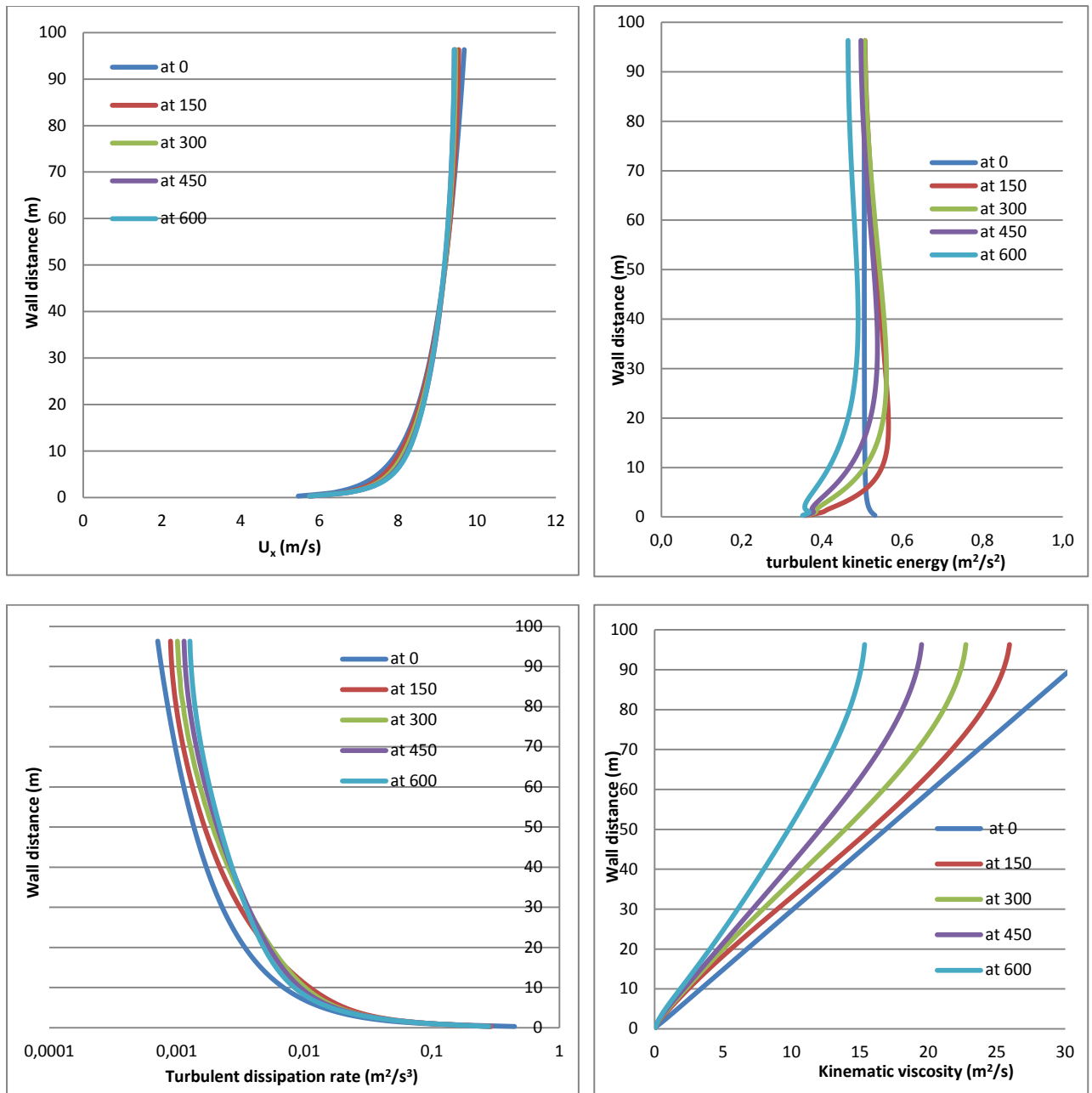


Figure 15: U_x, k, ϵ and ν_{t} are plotted against height at different length with OpenFOAM

The above diagrams show an acceptable degree of homogeneity between inlet and outlet profile especially for U_x, k and epsilon.

The roughness length in this case as mentioned before, has small value of $z_0=0.0002$. According to figure 5 for small value of roughness length, the wind profile is more vertical compare to bigger value of roughness length. As one can see from above velocity profile, the plotted horizontal velocity fall in a vertical line which is confirm the accuracy of assumption and results.

As mentioned before the results of OpenFOAM as an open source code needs validation. Therefore, to ensure about the accuracy of simulation results, ABL simulation is also done by

CFD simulation of wind condition and thermal gradient around an offshore helideck

Dr.Giljarhus by means of KFX as a commercial CFD code. The initial values used in inlet are the same as ones that used in OpenFOAM, and grading and mesh properties is also the same.

The comparison between results of OpenFOAM and KFX is presented in the following diagrams. The comparison is made for length of $x=450$ for U_x , k , epsilon, and nut.

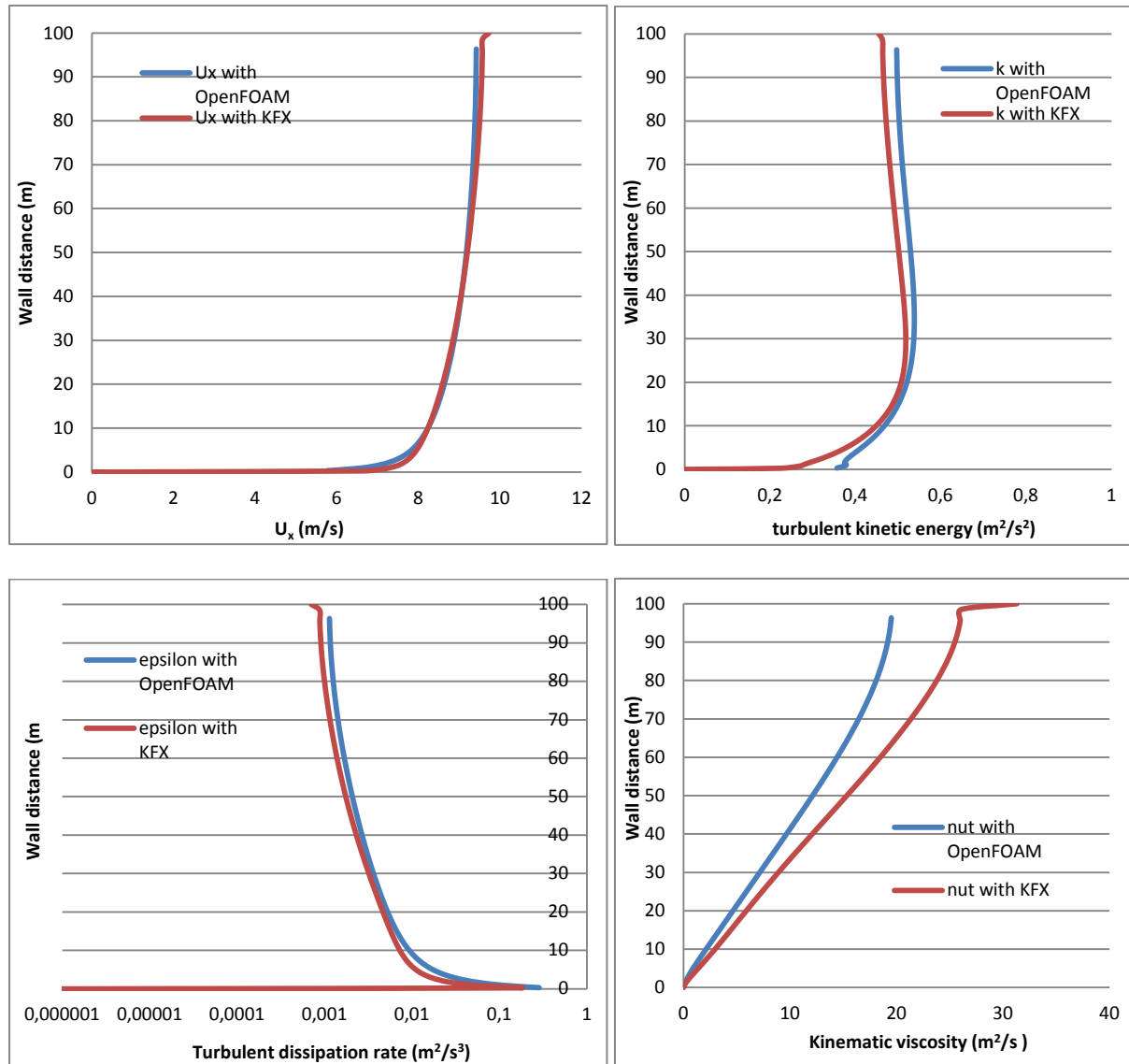


Figure 16: comparison of U_x , k , epsilon and nut results from OpenFOAM with KFX results

The above diagrams show an acceptable degree of homogeneity between OpenFOAM and KFX results, especially for velocity profile which they are fairly similar. These results also imply that OpenFOAM toolbox is a very useful tool for simulation of atmospheric boundary layer flows.

In figure 17, the velocity profile resulted by OpenFOAM and KFX would be compared with actual velocity profile. The values of actual velocity profile are calculated by ABL velocity equation proposed by Richards and Hoxey (eq 23):

$$u = \frac{u_\tau}{\kappa} \ln\left(\frac{z+z_0}{z_0}\right)$$

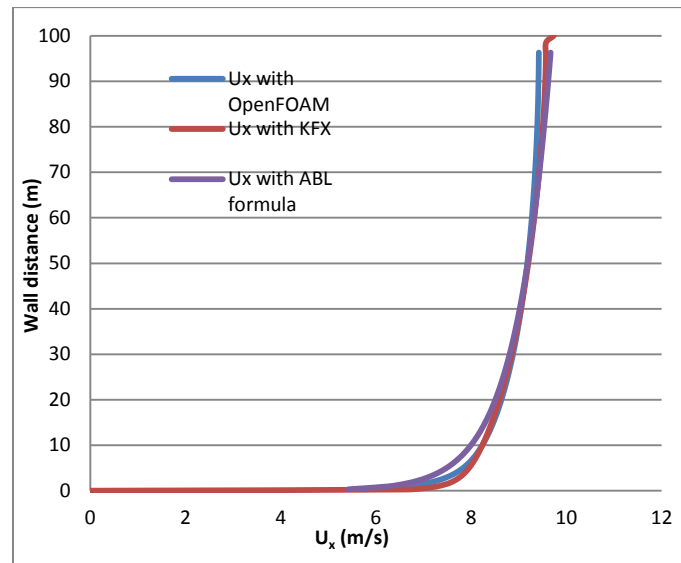


Figure 17: velocity profile of OpenFOAM, KFX and actual velocity profile

According to above diagram there is a great homogeneity between the simulations results of OpenFOAM with KFX velocity profile, and also actual velocity profile which confirms the accuracy of the simulations.

One of the most important missions of this report is to verify vertical velocity fluctuation above helideck, as explained in chapter 2.2.2.1.1 this value can be interpreted as follows due to the fact that in this case with the $k-\epsilon$ turbulence model, it is assumed that turbulence is isotropic and fully developed.

$$\sigma_z = \sqrt{\frac{2}{3}k} \quad \text{eq(11)}$$

The values of standard deviation of vertical flow velocity are plotted in the below diagram for different values of turbulent kinetic energy at $x=450$ against the wall distance. σ_z is plotted for both simulation results of OpenFOAM and KFX.

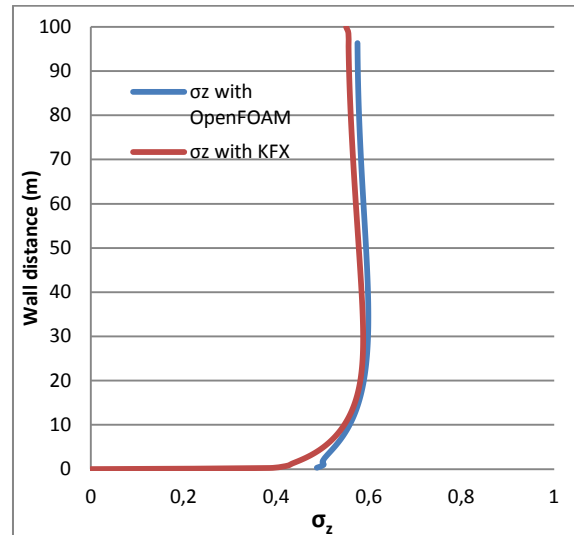


Figure 18: standard deviation of vertical velocity against height

As shown in above diagram, the values of vertical velocity fluctuation for both OpenFOAM and KFX code is around 0.5 at different height and this value is definitely acceptable for helicopter safe approach as it is less than 1.75m/s which defined by CAP 437.

In addition, comparison between results of σ_z with OpenFOAM and KFX shows acceptable homogeneity.

6.2 Simulation results of emission of hot plume

In this part, the plume path pattern resulted from simulation with OpenFOAM would be presented, and the results would be compared with the simulation results of KFX done by Dr.Giljarhus, to ensure about the accuracy of the results.

In below figures 285°K contour is applied to specify the location of 2°C isotherm as required according to NORSOK-004 and CAP437 standard.

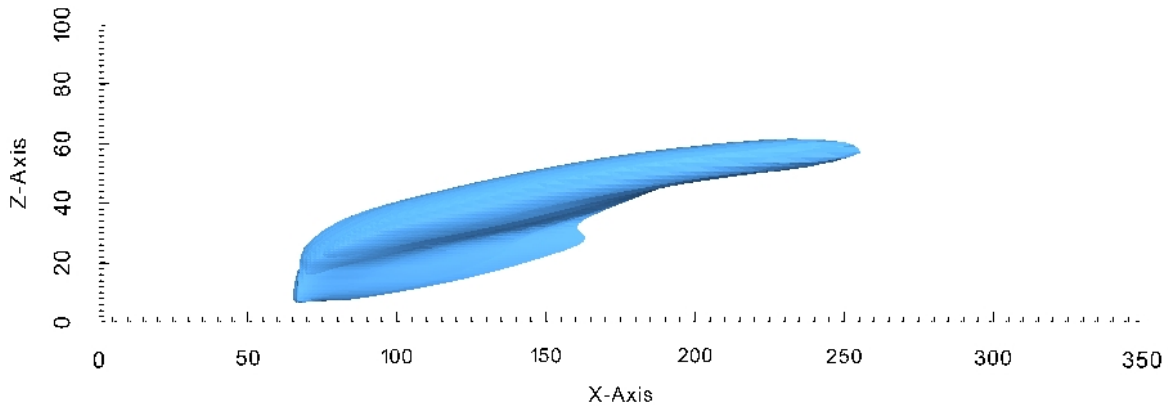


Figure 19: plume side with OpenFOAM

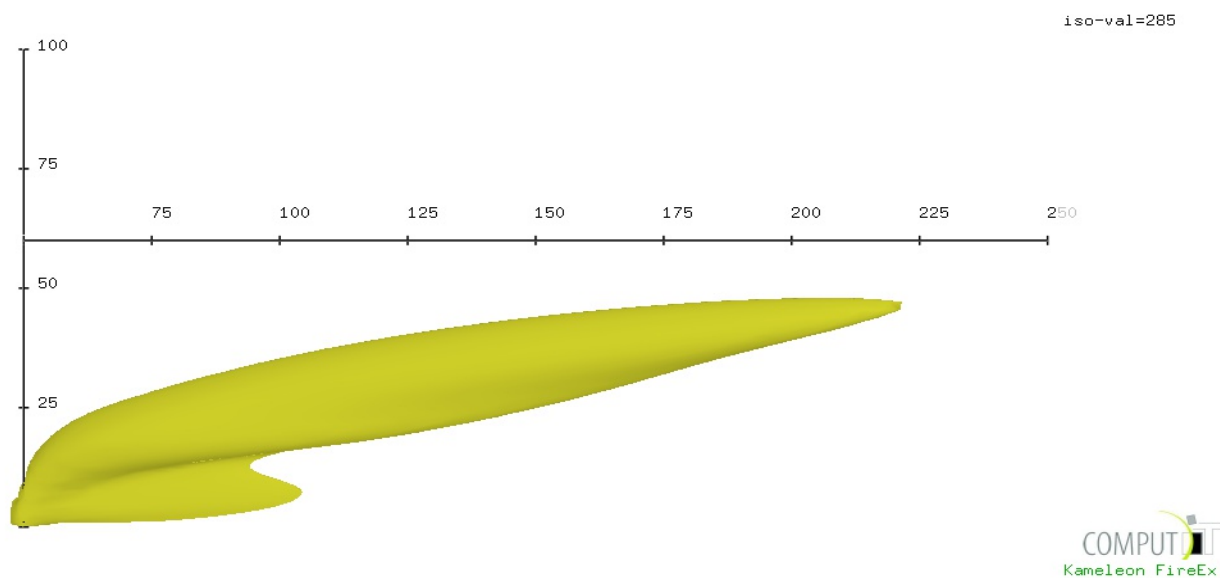


Figure 20: plume side with KFX

The plume length with OpenFOAM is about to 220m and with KFX is 225m. As shown in above figures the plume moves upward after it goes out of the secondary outlet and the maximum plume height at its end is around 60m with OpenFOAM, and 50m with KFX.

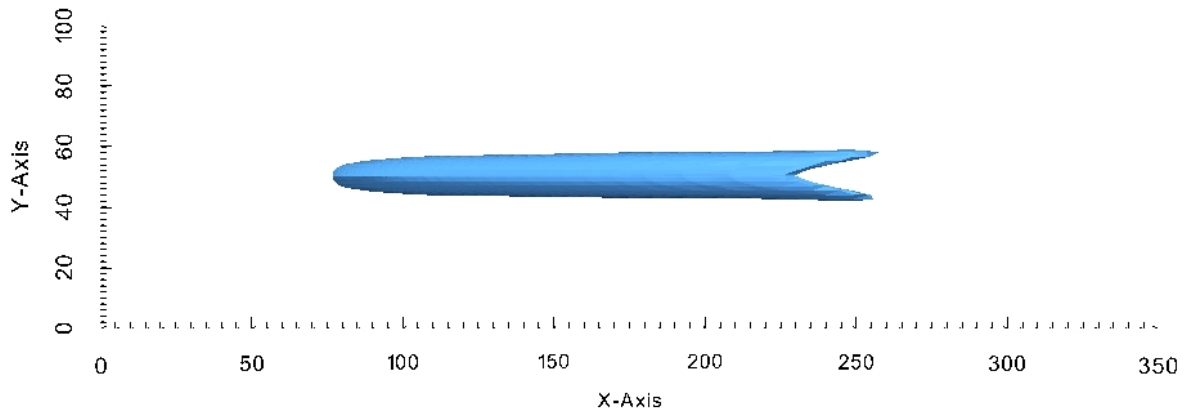


Figure 21: plume top with OpenFOAM

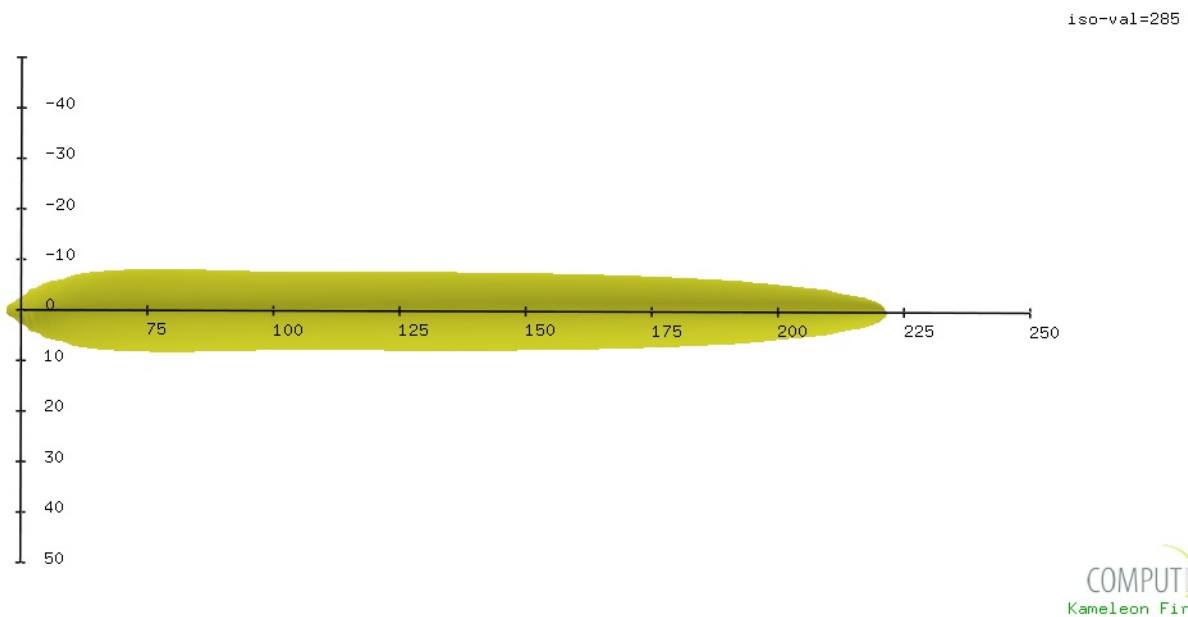


Figure 22: plume top with KFX

From above figures, maximum plume width is about 20m with both OpenFOAM and KFX.

To sum up, according to above mentioned plume pattern sizing, there is an acceptable homogeneity with the simulation results of OpenFOAM and KFX, and they confirm each other.

6.3 Simulation results of ABL simulation around the helideck

In this part, the simulation results regarding to vertical turbulence fluctuation would be verified as it is one the most important factors that affect helicopter safe operations in offshore platforms.

To achieve this goal, the final simulation results of turbulent kinetic energy (k) and vertical Reynolds tensor (R_{zz}) are needed above the helideck height up to height of computational domain. Therefore, the following command executed to extract data according to sets defined in `sampleDict`.

```
sample - time 174
```

This command generates a folder called `sets` which contains files for value of U, k and ϵ against height for the last iteration of case 1.

6.3.1 Comparison results of case 1 and 2 with coarse mesh to case 3 and 4 with finer mesh

In this part, the aim is to compare the results of simulation with k- ϵ turbulence model with the results of simulation with RSM turbulence model while both of them have the same coarse mesh, and when they have finer mesh.

For case 2 and 4 with RSM turbulence model, the standard deviation of vertical air flow velocity is interpreted as follow:

$$\sigma_z = \sqrt{R_{zz}} \quad \text{eq(29)}$$

To verify σ_z quantity, it is plotted in below diagrams. In these diagrams the values of $\sqrt{R_{xx}}$ and $\sqrt{R_{yy}}$ are also plotted from height of helideck up to height of computational domain, to show that RSM model is non- isotropic as R_{xx} , R_{zz} and R_{yy} have different values. Figure 23 present R values for coarse mesh while figure 24 is for fine mesh.

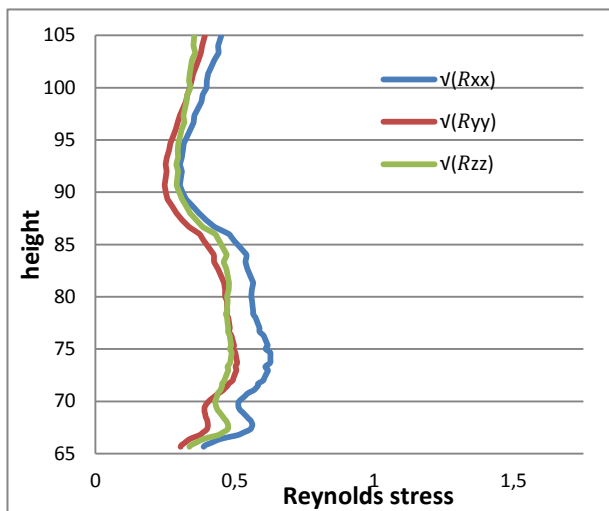


Figure 23: $\sqrt{R_{xx}}$, $\sqrt{R_{yy}}$ and $\sqrt{R_{zz}}$ for coarse mesh

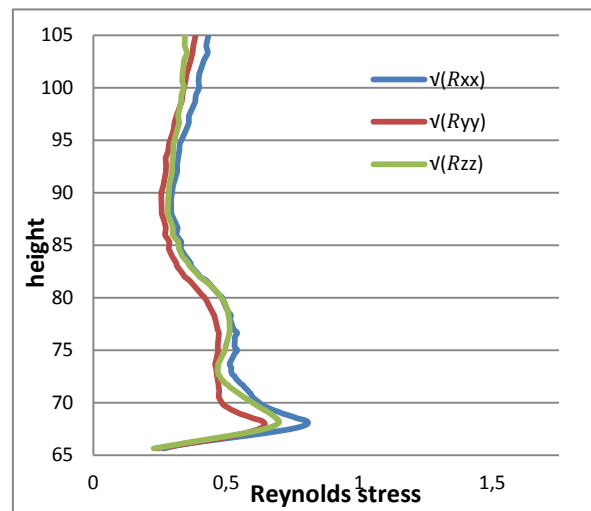


Figure 24 : $\sqrt{R_{xx}}$, $\sqrt{R_{yy}}$ and $\sqrt{R_{zz}}$ for fine mesh

The above figures diagrams shows acceptable values for vertical turbulence fluctuation, as in both diagrams σ_z have values less than 1.75m/s. However, finer mesh show more fluctuations in Reynolds tensor especially at heights above the helideck, but σ_z value are still less than CAP 437 criteria. Moreover, as expected, the value of $\sqrt{R_{xx}}$ is bigger compared to $\sqrt{R_{yy}}$ and $\sqrt{R_{zz}}$, as according to equation 26 to 28 mentioned on chapter 5.1.2 of this report, ratio of $\frac{\sigma_x}{u^*}$ is higher than $\frac{\sigma_y}{u^*}$, and ratio of $\frac{\sigma_y}{u^*}$ is higher than $\frac{\sigma_z}{u^*}$.

As explained in chapter 2.2.2.1.1, vertical air flow fluctuation can interpreted as follow for case 1 and 3 with k- ϵ turbulence model, as turbulence is assumed isotropic and fully developed.

$$\sigma_z = \sqrt{\frac{2}{3}k} \tag{eq(11)}$$

The comparison between σ_z for case 1 and 2 with coarse mesh, and case 3 and 4 with fine mesh are shown in following diagrams.

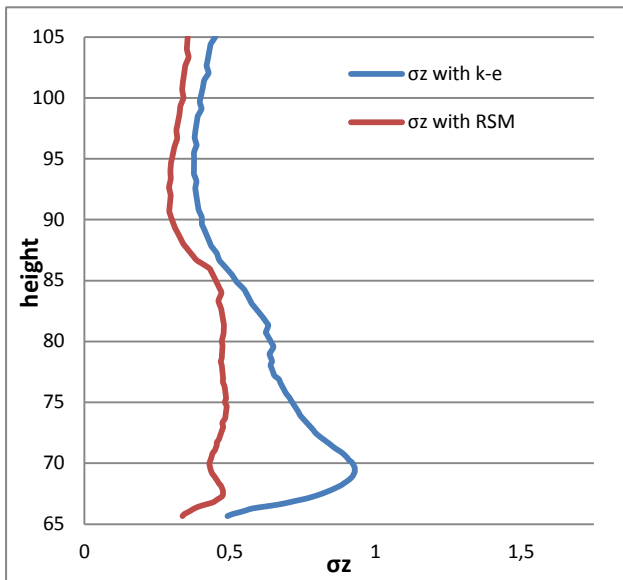


Figure 25: σ_z for case 1 & 2 with coarse mesh

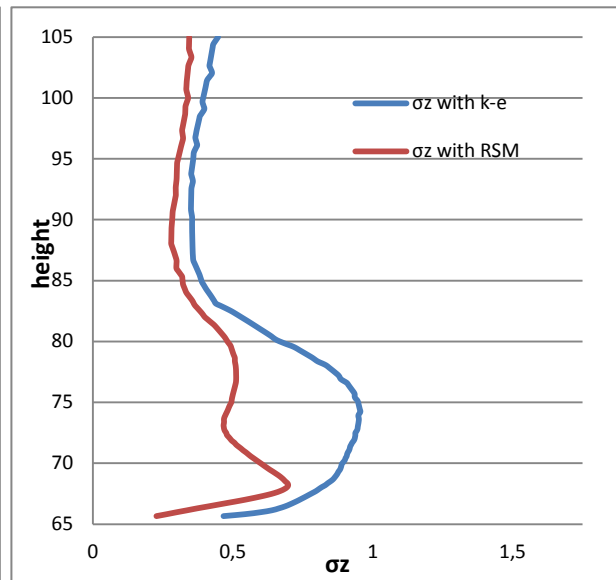


Figure 26: σ_z for case 3 & 4 with fine mesh

Above diagrams show higher values of σ_z at height of 65 to 85 especially in k - ϵ model which is expected due to fact that there is more turbulence at that area because of presence of modules on the platform and presence of helideck. Although, σ_z values are totally acceptable as they are less than 1.75m/s criteria in both coarse and fine mesh and with two different turbulence models.

In addition, comparison between coarse mesh and fine mesh shows more details and fluctuation in fine mesh as expected due to its higher resolution and mesh quality, but in general the results are fairly the same and fall in the same pattern.

Furthermore, the simulation results of k-ε turbulence model show more values of σ_z compare to RSM model, which is reasonable due to assumption made in this report for calculation of vertical velocity fluctuation component in Reynolds tensor and vertical air flow fluctuation for k- ε turbulence model. In this report as mentioned in chapter 5.1.2 the following ratio is considered for σ_z in RSM model:

$$\frac{\sigma_z}{u^*} = 1.25 \quad \text{eq(28)}$$

And for k-ε turbulence model, as mentioned in chapter 3.1.2 of this report, following formula used for turbulent kinetic energy calculation proposed by Richards and Hoxey:

$$k = \frac{u_\tau^2}{C_\mu^{0.5}} \quad \text{eq(8)}$$

In this formula the value of C_μ is considered 0.033, as this value is assumed in KFX calculation. On the other hand, due to isotropic assumption of k-ε turbulence model, following equation present the relation of σ_z and turbulent kinetic energy, as also explained on chapter 2.2.2.1.1 of this report:

$$\sigma_z = \sqrt{\frac{2}{3}k} \quad \text{eq(11)}$$

Replacing ABL turbulent kinetic energy equation to equation 11, is resulted to following ratio for σ_z in k-ε turbulence model:

$$\frac{\sigma_z}{u^*} = 1.92 \quad \text{eq(30)}$$

Comparing equation 28 with equation 30, show that according to above mentioned formula σ_z value resulted for k- ε turbulence model is higher than one calculated for RSM model. Therefore, according to these arguments and calculation and assumption used in this report, it is obvious that the simulation results of σ_z are expected to be higher for k-ε turbulence model compared to RSM model.

6.3.2 Turbulent kinetic energy and vertical air flow fluctuation plots

In this part, the turbulent kinetic energy side and top patterns are viewed in paraFOAM for both coarse and fine mesh and their patterns are compared with the simulation results of KFX. As KFX has only k-ε turbulence model, so for comparison of turbulent kinetic energy patterns, case 1 and case 3 with k-ε turbulence model are chosen. In addition, to have similar plot for KFX and OpenFOAM, KFX results are converted and viewed in paraFOAM by Dr.Giljarhus, to make comparison of results easier.

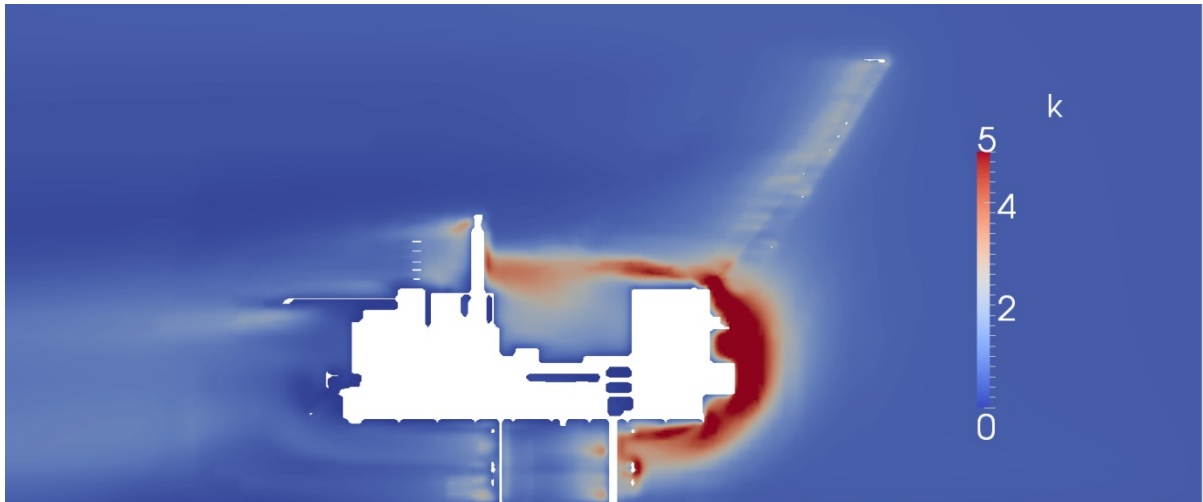


Figure 27: turbulent kinetic energy from side view with OpenFOAM for case 1, coarse mesh

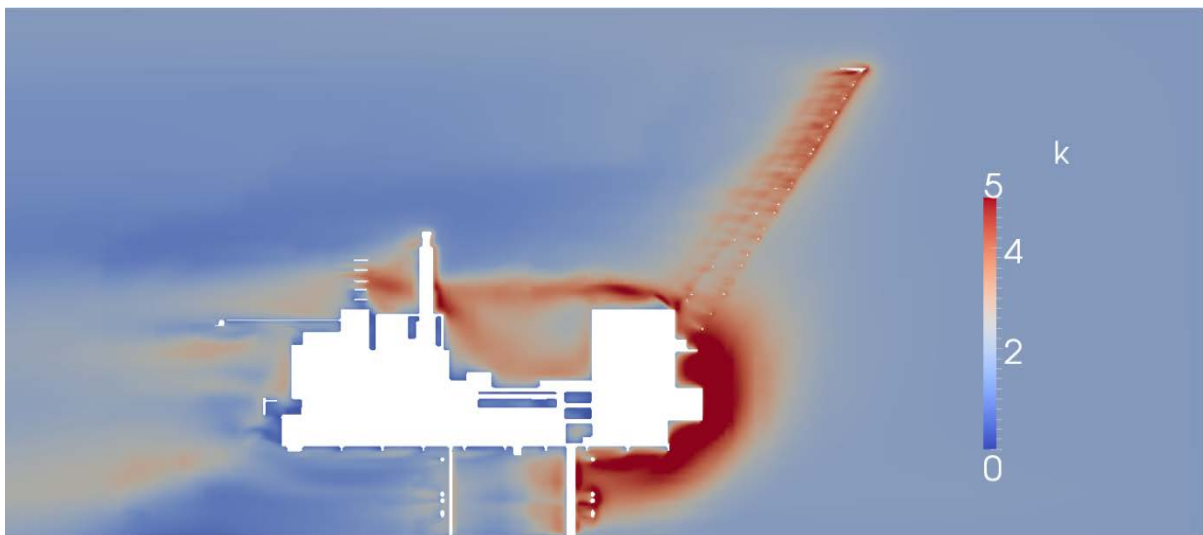


Figure 28: turbulent kinetic energy from side view with OpenFOAM for case 3, fine mesh

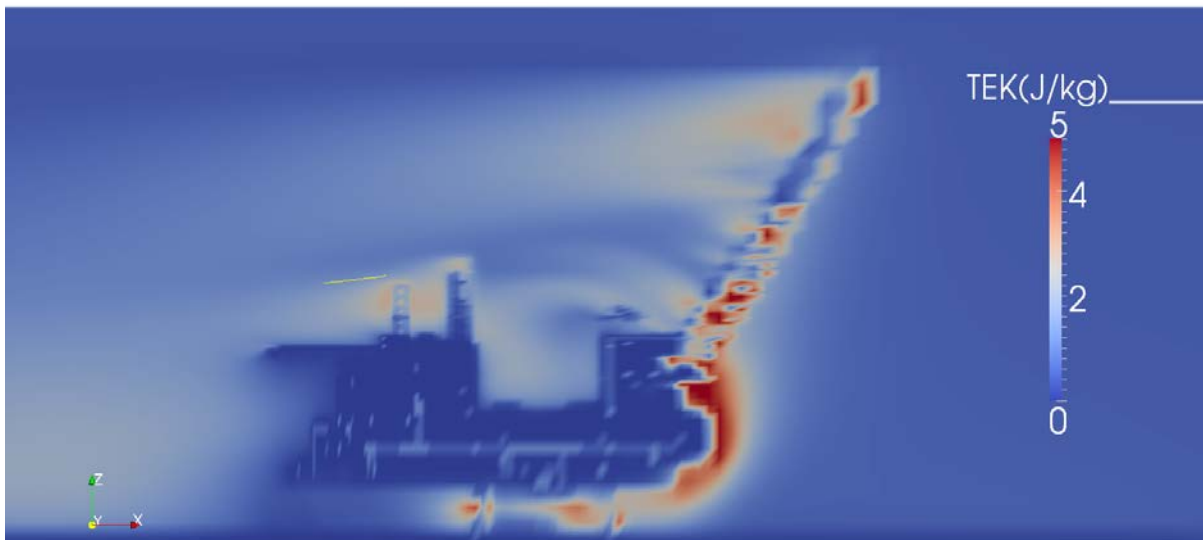


Figure 29: turbulent kinetic energy from side view with KFX

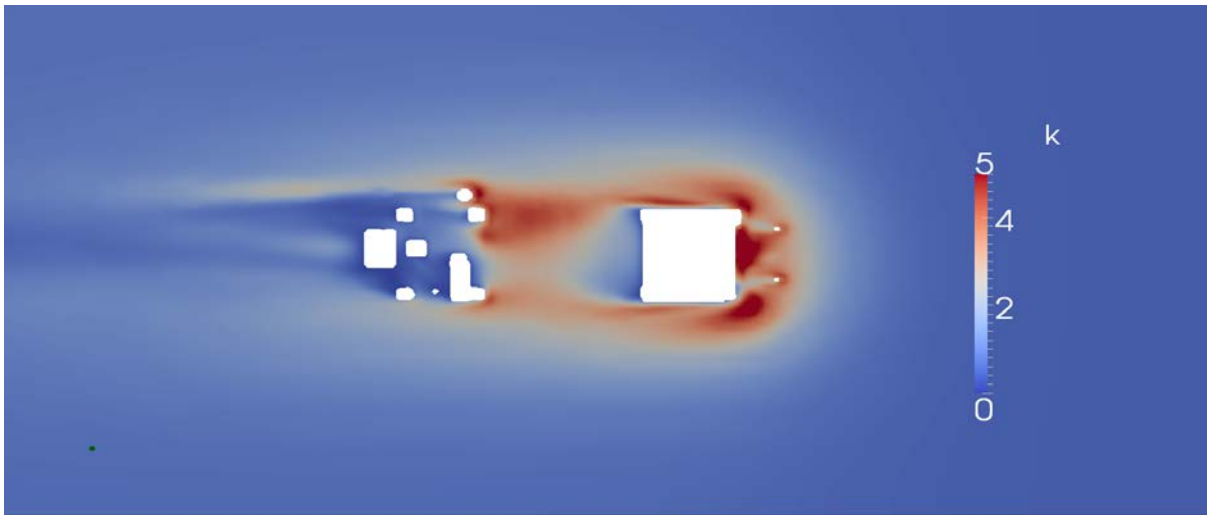


Figure 30: turbulent kinetic energy from top view with OpenFOAM for case 1, coarse mesh

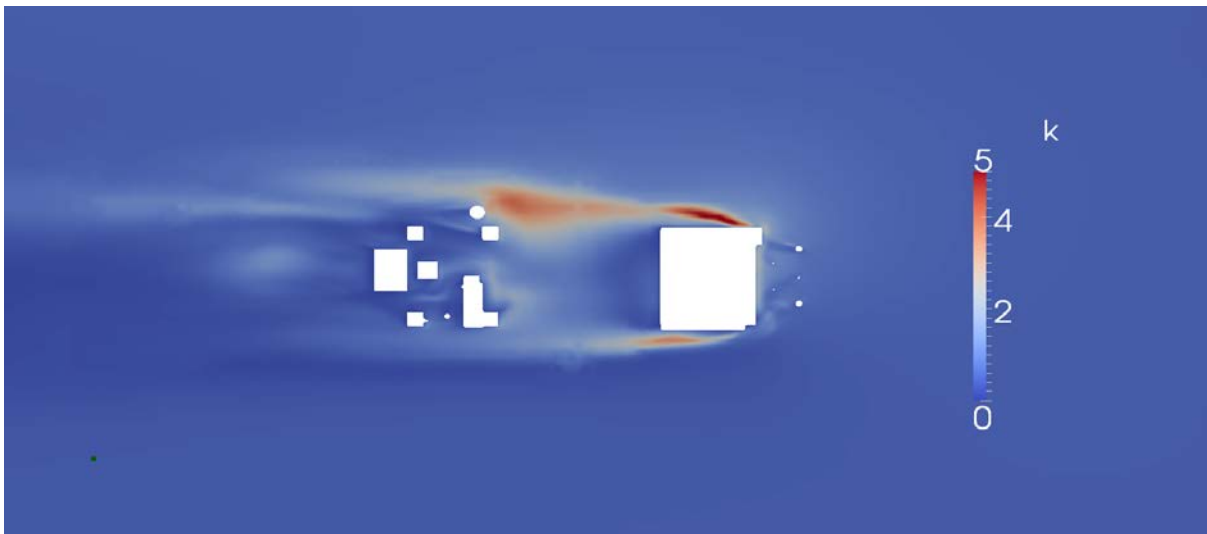


Figure 31: turbulent kinetic energy from top view with OpenFOAM for case 3, fine mesh

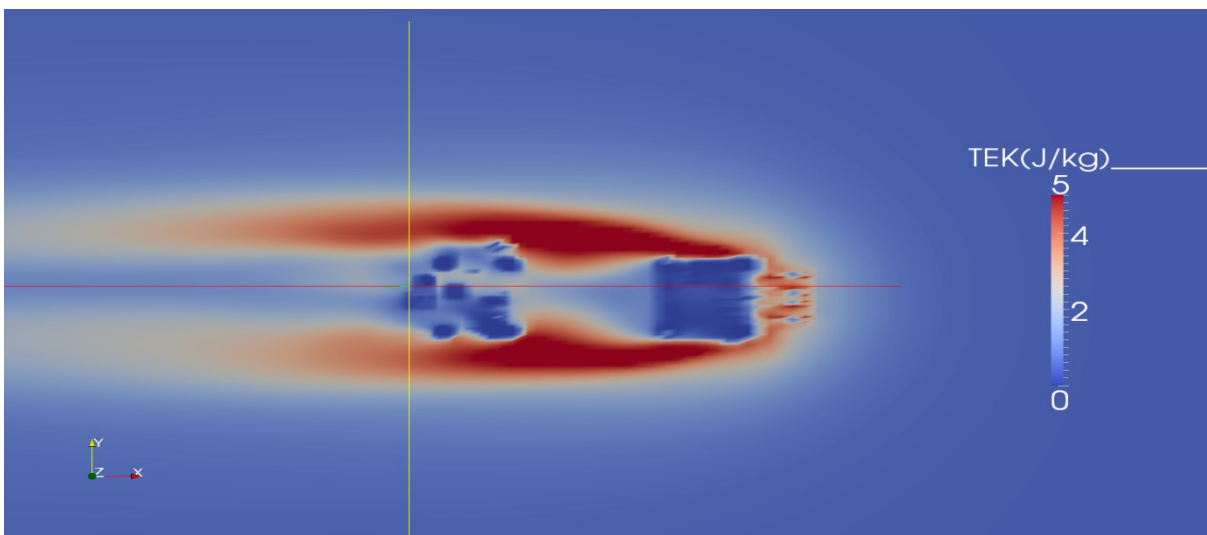


Figure 32: turbulent kinetic energy from top view with KFX

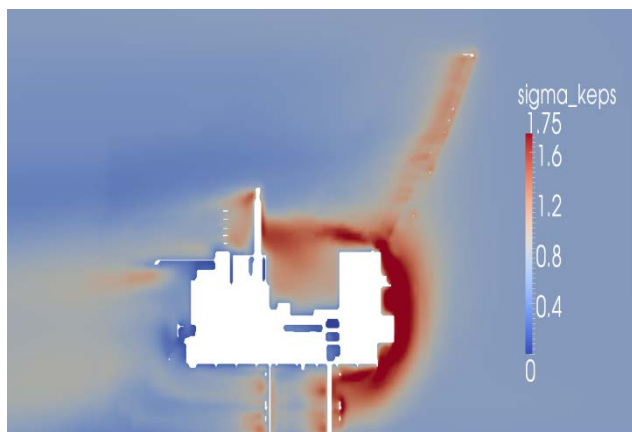
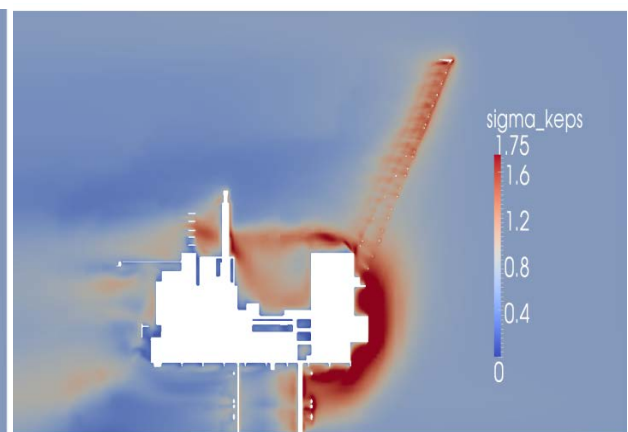
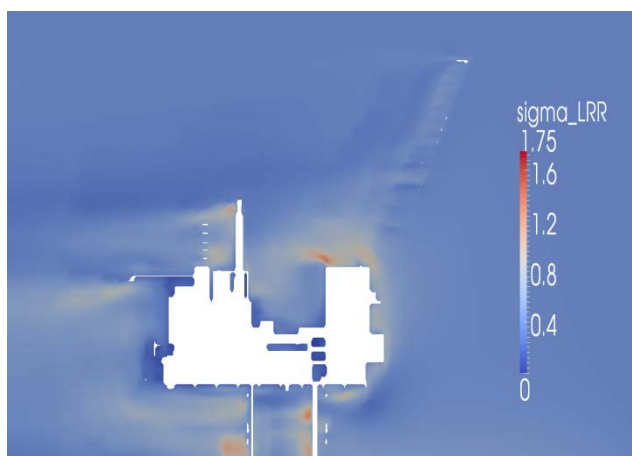
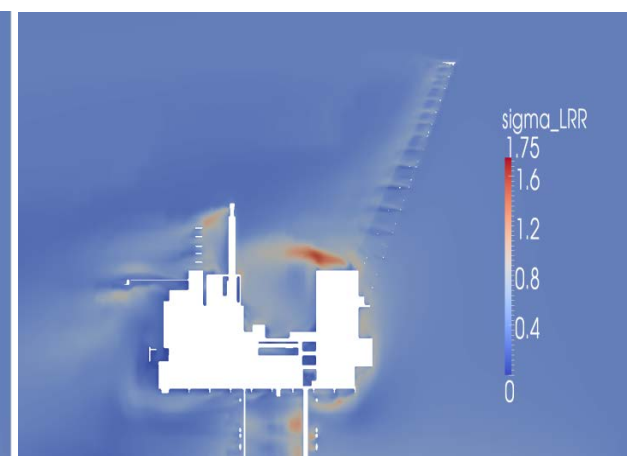
Interpreting of above figures is quite difficult as, there is not any distinct difference or similar trend between OpenFOAM fine and coarse mesh results and KFX.

It was expected that finer mesh, generate more turbulent kinetic energy compare to coarse mesh, as its resolution is higher and it resolve the objects better. However, in figure 28 it is shown that in finer mesh; more turbulent kinetic energy is generated compare to coarse one, but from top view as shown in figure 31 this is not happen compare to coarse mesh.

In addition, it was expected that KFX generate more turbulent kinetic energy as it has a sub-grid model which resolve the object. However, in top view comparison of KFX with OpenFOAM it shows that turbulent kinetic energy generated by KFX is higher, but in side view comparison it is not the case.

Although, there are some contradicts in the results, but in general the results are fairly the same and confirm each other.

In following figures, vertical air flow fluctuation would be presented for all 4 cases:

Figure 33: σ_z for case1 (k- ϵ and coarse mesh)Figure 34: σ_z for case3 (k- ϵ and fine mesh)Figure 35: σ_z for case2 (LRR and coarse mesh)Figure 36: σ_z for case4 (LRR and fine mesh)

From above figures it is obvious that σ_z values are higher for case 1 and 3 with k- ϵ turbulence model compare to case 2 and 4 with LRR turbulence model with the same argument explained before on chapter 6.3.1.

In addition, finer mesh shows more details in σ_z pattern as it is expected due to higher mesh quality compare to coarse mesh.

7 Conclusion

7.1 Project conclusions

In order to ensure helicopter safe operation on offshore platform, verticals airflow fluctuation is calculated by simulation of ABL around a simple 2D box, and then ABL simulated around an offshore fixed platform and helideck. In addition, hot air plume coming out of a secondary inlet is simulated for simple 3D box geometry.

ABL simulation for simple 2D geometry with OpenFOAM, and its results comparison with KFX results show an acceptable degree of homogeneity between OpenFOAM and KFX results; especially for velocity profile, which validate the OpenFOAM results as an open source code, and prove that OpenFOAM toolbox is a useful tool for ABL flow simulation. In addition, the OpenFOAM results show a great degree of homogeneity of velocity, turbulent kinetic energy and epsilon profiles in inlet, specified points at length of computational domain, and outlet.

In case of hot exhaust plume simulation, hot plume 2°C isotherm location, sizing and pattern simulated by OpenFOAM was fairly similar to simulation result of KFX which confirms its accuracy, and provide a basis for further improvement for hot plume simulation with OpenFOAM.

Finally, vertical air flow fluctuation that calculated at heights above helideck, show that these values are acceptable for the defined wind condition as σ_z results for all 4 cases are definitely less than 1.75 m/s criterion defined by CAP437. Moreover, comparison of simulation results with k- ϵ and RSM turbulence models for both coarse and fine mesh, show that the vertical air flow fluctuation values are higher for k- ϵ turbulence model compare to RSM model, due to assumption made and formula used for calculation of σ_z value in these two turbulence models. In addition, turbulent kinetic energy patterns resulted from OpenFOAM was in general similar to the KFX results. However, there is not obvious trend between OpenFOAM coarse and fine mesh and KFX results.

7.2 Future work

Although, in this thesis it is tried to simulate the main aerodynamic hazards around an offshore platform to ensure helicopter safe operation, there is still room for further development specially regarding to simulation of hot plume exhausted from an offshore platform, so following items could develop in future studies:

- Simulation of hot plume exhausted from an offshore platform with real geometry of a platform.
- Running hot plume simulation by considering other gas like as methane exhausted instead of air, and consequently using other solver than `buoyantSimpleFoam`, like as `reactingFoam`.
- Further studies regarding to `groovyBC`, since applying `groovyBC` for epsilon shown some divergence problem when `buoyantSimpleFoam` solver is used, so further studies could be done to examine this problem.
- Real experiments could be carried out regarding exhausted plume to prepare basis for comparison of CFD simulation results with real experiment results.
- Influence of grid parameters should be studied in more details.

In addition, future studies could be developed for ABL simulation in following area:

- Regarding to vertical airflow fluctuation criteria define by CAP 437, some questions were raised by the research done by a Brazilian group (Silva et al., 2010). First, according the wind tunnel turbulence measurement they have done on the side of ship model in a non-disturbed flow condition, standard deviation of vertical air flow found 1.45m/s, so how it is possible to have standard deviation of 1.75 m/s in a real condition with turbulent flow? Second, how is it possible to assess only vertical turbulence component since frontal velocity gradient has significant effect on helicopter performance? (Silva et al., 2010). Therefore, additional research might be required to validate this criterion.

8 References

- BAKKER, A. 2002-2006a. Applied Computational Fluid Dynamics, Lecture 10 - Turbulence Models.
- BAKKER, A. 2002-2006b. Lecture 13 - Heat Transfer, Applied Computational Fluid Dynamics.
- BALOGH, M., PARENTE, A. & BENOCCI, C. 2012. RANS simulation of ABL flow over complex terrain applying an Enhanced k-e model and wall function formulation: Implementation and comparison for fluent and OpenFOAM. *Journal of Wind Engineering and Industrial Aerodynamics*.
- BLOCKEN, B., STATHOPOULOS, T. & CARMELIET, J. 2007. CFD simulation of the atmospheric boundary layer: wall function problems. *Atmospheric Environment* 41 (2007) 238–252.
- CAP 437, C. A. A. May 2012. CAP 437, Standards for Offshore Helicopter Landing Areas. UK: CAA publications.
- CHEN, Q., GU, Z., SUN, T. & SONG, S. 1995. Wind environment over the helideck of an offshore platform. *Journal of Wind Engineering and Industrial Aerodynamics* 54/55 (1995) 621-631.
- CHURCHFIELD, M. J., VIJAYAKUMAR, G., BRASSEUR, J. G. & MORIARTY, P. J. 2010. Wind energy-related atmospheric boundary layer large-eddy simulation using OpenFOAM COMPUTIT. 2013. *ComputIT – KFX®* [Online]. Available: <http://www.computit.no/?module=Articles;action=Article.publicShow;ID=347>.
- DECKERS, X., HAGA, S., TILLEY, N. & MERCI, B. 2012. Smoke control in case of fire in a large car park: CFD simulations of full-scale configurations. *Fire Safety Journal*.
- ENGINEERINGTOOLBOX. 2013. *Air - Absolute and Kinematic Viscosity* [Online]. Available: http://www.engineeringtoolbox.com/air-absolute-kinematic-viscosity-d_601.html.
- FRANCESCHI, M. D., ZARDI, D., TAGLIAZUCCA, M. & TAMPIERI, F. 2009. Analysis of second-order moments in surface layer turbulence in an Alpine valley. *QUARTERLY JOURNAL OF THE ROYAL METEOROLOGICAL SOCIETY*.
- GILJARHUS, K. E. 2013. Helideck turbulence criterion.
- GONG, J. & LI, Y. 2010. CFD modelling of the effect of fire source geometry and location on smoke flow multiplicity. *Tsinghua University Press and Springer-Verlag Berlin Heidelberg 2010*.
- HELIDECKINSPECT. 2013. *HELIDECK CERTIFICATION SERVICES* [Online]. Available: <http://www.helideckinspect.com/>.
- HJERTAGER, B. H. December 2009a. Lecture notes in OpenFOAM. In: STAVANGER, U. O. (ed.).
- HJERTAGER, B. H. December 2009b. OpenFOAM Tutorials. In: STAVANGER, U. O. (ed.) *Based on a Short Course given by Icon Ltd at Baumont Hotel, Old Windsor in November 2007, London, UK* Stavanger, Norway
- HJERTAGER, B. H. January 2002. Computational Analysis of Fluid Flow Processes. In: STAVANGER, U. O. (ed.) March 2009 ed. Stavanger, Norway.
- HU, C. & FUKUCHI, N. 2003. A field modeling approach to prediction of hot gas movement induced by marine compartment fires. *International journal of offshore and polar engineering*, 13.
- ISO 15138, I. S. 2007. Petroleum and natural gas industries — Offshore production installations — Heating, ventilation and air-conditioning.

- KANO, M. 1999. Reynolds- Stress Model taking into Account Strong Anisotropy of Wall Turbulence (1st report) New Proposal for Reynolds-Stress Model.
- KULKARNI, P. R., SINGH, S. N. & SESHADRI, V. 2006. Parametric studies of exhaust smoke–superstructure interaction on a naval ship using CFD. *Computers & Fluids* 36 (2007) 794–816.
- LYSENKO, D., ERTESVÅG, I. S. & RIAN, K. E. 2012. Modeling of turbulent separated flows using OpenFOAM. *Computers & Fluids* xxx (2012) xxx–xxx.
- MAP, A. P. A. 2009. Governmental Acquisition of New Rescue Helicopters – Technical Consultancy Support for JD. *Review and assessment of helicopter deck and landing places in the Norwegian AOR*. MAP Aircraft Projects AS.
- MORRISON, G. 2001. OFFSHORE TECHNOLOGY REPORT 2000/089. *Helicopter safety offshore*. United Kingdom: Health and Safety Executive.
- MURAKAMI, S. 1993. Comparison of various turbulence models applied to a bluff body. *J Wind Eng Ind Aerodyn* 1993;46–47:21–36.
- NORSOK , S. Rev. 1, 2004. NORSOK STANDARD C-004, Helicopter deck on offshore installations. Strandveien 18, P.O. Box 242,N-1326 Lysaker,NORWAY: Standards Norway.
- NORSOK , S. Rev. 2, Feb. 2013. NORSOK STANDARD C-004, Helicopter deck on offshore installations. Strandveien 18, P.O. Box 242,N-1326 Lysaker,NORWAY: Standards Norway.
- OPENFOAM. 2011- 2013. *OpenFOAM Foundation releases OpenFOAM® 2.1.1* [Online]. Available: <http://www.openfoam.org/version2.1.1/>.
- OSENBROCH, J. July 2006. *CFD study of gas dispersion and jet fires in complex geometries*. PHD, Aalborg University Esbjerg Niels Bohrs Vej 8.
- RAMLI, N. I., ALI, M. I., SAAD, M. S. H. & MAJID, T. A. 2009. Estimation of the Roughness Length (zo) in Malaysia using Satellite Image. *The Seventh Asia-Pacific Conference on Wind Engineering, November 8-12, 2009, Taipei, Taiwan*.
- RICHARDS, P. J. & HOXEY, R. 1993. Appropriate boundary conditions for computational wind engineering models using the k–e turbulence model. *Journal of Wind Engineering and Industrial Aerodynamics* 47.
- ROWE, S. J., HOWSON, D. & DAVID, R. THE RESPONSE OF HELICOPTERS TO AERODYNAMIC DISTURBANCES AROUND OFFSHORE HELIDECKS.
- ROWE, S. J., HOWSON, D. & SPARKES, P. 2002. Research Improving the Safety of Offshore Helicopter Operations. *Offshore Technology Conference*.
- SAFETY REGULATION GROUP May 2009. CAA paper 2008/02 Offshore helideck environmental research. UK.
- SILVA, D. F. D. C. E., PAGOT, P. R., NADER, G. & JABARDO, P. J. S. 2010. CFD SIMULATION AND WIND TUNNEL INVESTIGATION OF A FPSO OFFSHORE HELIDECK TURBULENT FLOW. *Proceedings of the ASME 2010 29th International Conference on Ocean, Offshore and Arctic Engineering OMAE2010*.
- STERN, F., XING, T., SHAO, J. & GHOSH, S. 2005. *Introduction to Fluid Mechanics* [Online]. Available: <http://www.docstoc.com/docs/83223736/Introduction-to-Fluid-Mechanics-Fred-Stern-Tao-Xing-Jun-Shao-Surajeet-Ghosh-Shanti-Bhushan-8-27-2009-AFD>.
- UNHALE, S. A. 2004. *Application and analysis of RANS based turbulence models for bluff body aerodynamics*. Master of Science, Texas Tech University.

- VERSTEEG, H. K. & MALALASEKERA, W. 2007. *An introduction to computational fluid dynamics: the finite volume method*, Harlow, Pearson/Prentice Hall.
- WILCOX, D. C. 2006. *Turbulence modeling for CFD*, [La Cãnada, Calif.], DCW Industries.
- ZIENKIEWICZ, O. C., TAYLOR, R. L. & NITHIARASU, P. 2005. *The finite element method for fluid dynamics*, Amsterdam, Elsevier.

Appendices

Appendix A: ABL simulations of simple geometry

Appendix A.1: blockMeshDict

```

convertToMeters 1;
vertices
(
    ( 0    -1    0)
    ( 600  -1    0)
    ( 600   1    0)
    ( 0    1    0)
    ( 0    -1   100)
    ( 600  -1   100)
    ( 600   1   100)
    ( 0    1   100)
);
blocks
(
    hex (0 1 2 3 4 5 6 7) (300 1 75) simpleGrading (1 1 4)
);
edges
(
);
boundary
(
    outlet
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }
    sides
    {
        type empty;
        faces
        (
            (1 5 4 0)
            (3 7 6 2)
        );
    }
    inlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }
);

```

```

    }
    ground
    {
        type wall;
        faces
        (
            (0 3 2 1)
        );
    }
    top
    {
        type patch;
        faces
        (
            (4 5 6 7)
        );
    }
};
mergePatchPairs
(
);

```

Appendix A.2: ABL conditions

```

Uref                8;
Href                10;
z0                  uniform 0.0002;
turbulentKE         0.505;
windDirection       (1 0 0);
zDirection          (0 0 1);
zGround             uniform 0;

```

Appendix A.3: initial conditions

```

flowVelocity        (8 0 0);
pressure            0;
turbulentKE         0.505;
turbulentEpsilon    0.007;

```

Appendix A.4: side and top patches

```

top
{
    type slip;
}
sides
{
    type empty;
}

```

Appendix A.5: U field

```
#include      "include/initialConditions"

dimensions   [0 1 -1 0 0 0 0];

internalField  uniform $flowVelocity;

boundaryField
{
    #include "include/ABLConditions"

    outlet
    {
        type          zeroGradient;
    }

    inlet
    {
        type          atmBoundaryLayerInletVelocity;
        Uref          $Uref;
        Href          $Href;
        n             $windDirection;
        z             $zDirection;
        z0            $z0;
        value         $internalField;
        zGround       $zGround;
    }

    ground
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }

    #include "include/sideAndTopPatches"
}
```

Appendix A.6: k field

```
#include      "include/initialConditions"

dimensions   [0 2 -2 0 0 0 0];

internalField  uniform $turbulentKE;

boundaryField
{
    #include "include/ABLConditions"

    outlet
    {
        type          zeroGradient;
    }

    inlet
    {
        type          fixedValue;
        value         $internalField;
    }

    ground
    {
        type          kqRWallFunction;
        value         $internalField;
    }

    #include "include/sideAndTopPatches"
}
```

Appendix A.7: ϵ field

```
dimensions      [0 2 -3 0 0 0 0];

#include         "include/initialConditions"

internalField   uniform $turbulentEpsilon;

boundaryField
{
    #include "include/ABLConditions"

    outlet
    {
        type          zeroGradient;
    }
    inlet
    {
        type          atmBoundaryLayerInletEpsilon;
        Uref          $Uref;
        Href          $Href;
        z             $zDirection;
        z0            $z0;
        value         $internalField;
        zGround      $zGround;
    }

    ground
    {
        type          epsilonWallFunction;
        Cmu          0.09;
        kappa        0.4;
        E            9.8;
        value         $internalField;
    }

    #include "include/sideAndTopPatches"
}
```

Appendix A.8: P field

```
#include      "include/initialConditions"

dimensions   [0 2 -2 0 0 0 0];

internalField uniform $pressure;

boundaryField
{
    inlet
    {
        type          zeroGradient;
    }

    outlet
    {
        type          fixedValue;
        value         uniform $pressure;
    }

    ground
    {
        type          zeroGradient;
    }

    #include "include/sideAndTopPatches"
}
```

Appendix A.9: nut field

```
dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    #include "include/ABLConditions"
    inlet
    {
        type          calculated;
        value         uniform 0;
    }

    outlet
    {
        type          calculated;
        value         uniform 0;
    }

    ground
    {
        type          nutkAtmRoughWallFunction;
        z0            $z0;
        value         uniform 0.0;
    }

    #include "include/sideAndTopPatches"
}
}
```

Appendix A.10: controlDict

```

application      simpleFoam;
startFrom        latestTime;
startTime        0;
stopAt           endTime;
endTime          1000;
deltaT           1;
writeControl     timeStep;
writeInterval    50;
purgeWrite       0;
writeFormat      ascii;
writePrecision   12;
writeCompression uncompressed;
timeFormat       general;
timePrecision    6;
runTimeModifiable yes;
functions
{
    testSample
    {
        type sets;
        functionObjectLibs ("libsampling.so");

        setFormat raw;
        interpolationScheme cell;
        fields
        (
            U
            k
            epsilon
            nut
        );
        sets
        (
            line1
            {
                type          midPoint;
                axis          z;

                start        (0 0 0);
                end          (0 0 100);
            }

            line2
            {
                type          midPoint;
                axis          z;

                start        (150 0 0);
                end          (150 0 100);
            }
        )
    }
}

```


CFD simulation of wind condition and thermal gradient around an offshore helideck

```
    }  
  
    line3  
    {  
        type          midPoint;  
        axis          z;  
  
        start        (300 0 0);  
        end          (300 0 100);  
    }  
  
    line4  
    {  
        type          midPoint;  
        axis          z;  
  
        start        (450 0 0);  
        end          (450 0 100);  
    }  
  
    line5  
    {  
        type          midPoint;  
        axis          z;  
  
        start        (600 0 0);  
        end          (600 0 100);  
    }  
);  
outputControl    timeStep;  
outputInterval  1;  
  
    }  
}
```

Appendix A.11: fvSchemes

```

ddtSchemes
{
    default          steadyState;
}
gradSchemes
{
    default          Gauss linear;
    grad(p)          Gauss linear;
    grad(U)          Gauss linear;
}
divSchemes
{
    default          none;
    div(phi,U)       Gauss linearUpwindV grad(U);
    div((nuEff*dev(T(grad(U)))) Gauss linear;
    div(phi,epsilon) Gauss upwind;
    div(phi,k)       Gauss upwind;
}
laplacianSchemes
{
    default          Gauss linear limited 0.333;
}
interpolationSchemes
{
    default          linear;
}
snGradSchemes
{
    default          limited 0.333;
}
fluxRequired
{
    default          no;
    p;
}

```

Appendix A.12: fvSolution

```
solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-7;
        relTol          0.1;
        smoother        GaussSeidel;
        nPreSweeps      0;
        nPostSweeps     2;
        cacheAgglomeration on;
        agglomerator     faceAreaPair;
        nCellsInCoarsestLevel 10;
        mergeLevels     1;
    }

    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-8;
        relTol          0.1;
        nSweeps         1;
    }

    k
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-8;
        relTol          0.1;
        nSweeps         1;
    }

    epsilon
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-8;
        relTol          0.1;
        nSweeps         1;
    }
}

SIMPLE
{
    nNonOrthogonalCorrectors 0;
}
```

```
    residualControl
    {
        p          1e-2;
        U          1e-3;
        "(k|epsilon)" 1e-3;
    }
}

relaxationFactors
{
    fields
    {
        p          0.3;
    }
    equations
    {
        U          0.7;
        k          0.7;
        epsilon    0.7;
    }
}

cache
{
    grad(U);
}
```

Appendix B: hot plume simulation

Appendix B.1: blockMeshDict

```

convertToMeters 1;
vertices
(
    (0 0 0) // vertex nb 00
    (50 0 0) // vertex nb 01
    (51 0 0) // vertex nb 02
    (350 0 0) // vertex nb 03
    (0 50 0) // vertex nb 04
    (50 50 0) // vertex nb 05
    (51 50 0) // vertex nb 06
    (350 50 0) // vertex nb 07
    (0 51 0) // vertex nb 08
    (50 51 0) // vertex nb 09
    (51 51 0) // vertex nb 10
    (350 51 0) // vertex nb 11
    (0 100 0) // vertex nb 12
    (50 100 0) // vertex nb 13
    (51 100 0) // vertex nb 14
    (350 100 0) // vertex nb 15
    (0 0 100) // vertex nb 16
    (50 0 100) // vertex nb 17
    (51 0 100) // vertex nb 18
    (350 0 100) // vertex nb 19
    (0 50 100) // vertex nb 20
    (50 50 100) // vertex nb 21
    (51 50 100) // vertex nb 22
    (350 50 100) // vertex nb 23
    (0 51 100) // vertex nb 24
    (50 51 100) // vertex nb 25
    (51 51 100) // vertex nb 26
    (350 51 100) // vertex nb 27
    (0 100 100) // vertex nb 28
    (50 100 100) // vertex nb 29
    (51 100 100) // vertex nb 30
    (350 100 100) // vertex nb 31
);
blocks
(
    hex (0 1 5 4 16 17 21 20) (20 30 70) simpleGrading (0.25 1 4)
    hex (4 5 9 8 20 21 25 24 ) (20 2 70) simpleGrading (0.25 1 4)
    hex ( 8 9 13 12 24 25 29 28) (20 30 70) simpleGrading (0.25 1 4)
    hex (1 2 6 5 17 18 22 21 ) (2 30 70) simpleGrading (1 1 4)
    hex (5 6 10 9 21 22 26 25 ) (2 2 70) simpleGrading (1 1 4)
    hex (9 10 14 13 25 26 30 29 ) (2 30 70) simpleGrading (1 1 4)
    hex (2 3 7 6 18 19 23 22) (80 30 70) simpleGrading (4 1 4)
    hex (6 7 11 10 22 23 27 26 ) (80 2 70) simpleGrading (4 1 4)
    hex (10 11 15 14 26 27 31 30 ) (80 30 70) simpleGrading (4 1 4)

```

```

);
edges
(
);
boundary
(
    outlet
    {
        type patch;
        faces
        (
            (3 19 23 7)
            (7 23 27 11)
            (11 27 31 15)
        );
    }
    mainInlet
    {
        type patch;
        faces
        (
            (0 16 20 4)
            (4 20 24 8)
            (8 24 28 12)
        );
    }
    secondaryInlet
    {
        type patch;
        faces
        (
            (5 6 10 9)
        );
    }
    sides
    {
        type symmetryPlane;
        faces
        (
            (0 1 17 16)
            (1 2 18 17)
            (2 3 19 18)
            (12 13 29 28)
            (13 14 30 29)
            (14 15 31 30)
        );
    }
    ground
    {
        type wall;
        faces

```

CFD simulation of wind condition and thermal gradient around an offshore helideck

```
(
  (0 1 5 4)
  (4 5 9 8)
  (8 9 13 12)
  (1 2 6 5)
  (9 10 14 13)
  (2 3 7 6)
  (6 7 11 10)
  (10 11 15 14)
    );
}
top
{
  type patch;
  faces
  (
    (16 17 21 20)
    (20 21 25 24)
    (24 25 29 28)
    (17 18 22 21)
    (21 22 26 25)
    (25 26 30 29)
    (18 19 23 22)
    (22 23 27 26)
    (26 27 31 30)
      );
}
);
mergePatchPairs
(
);
```

Appendix B.2: initial conditions

```
flowVelocity      (8 0 0);  
pressure          1e5;  
turbulentKE      37.5;  
turbulentEpsilon 1887;
```

Appendix B.3: side and top patches

```
top  
{  
    type slip;  
}  
  
sides  
{  
    type symmetryPlane;  
}
```


Appendix B.4: \mathbf{u} field

```

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (15 0 0);

boundaryField
{
    outlet
    {
        type          zeroGradient;
    }

    mainInlet
    {
        type          groovyBC;
        value         uniform (0 0 0);
        variables
        "Uref=8.0;Href=10.0;z0=0.0002;kappa=0.41;Cmu=0.033;ustar=Uref*
kappa/log((Href+z0)/z0);";
        fractionExpression  "1";
        valueExpression
        "vector((ustar/kappa)*log((pos().z+z0)/z0), 0, 0)";
    }

    secondaryInlet
    {
        type          fixedValue;
        value         uniform (0 0 100);
    }

    ground
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }

    #include "include/sideAndTopPatches"
}

```

Appendix B.5: k field

```
#include      "include/initialConditions"

dimensions   [0 2 -2 0 0 0 0];

internalField  uniform $turbulentKE;

boundaryField
{
    outlet
    {
        type          zeroGradient;
    }

    mainInlet
    {
        type          fixedValue;
        value         $internalField;
    }

    secondaryInlet
    {
        type          fixedValue;
        value         $internalField;
    }

    ground
    {
        type          compressible::kqRWallFunction;
        value         $internalField;
    }

    #include "include/sideAndTopPatches"
}
```

Appendix B.6: ϵ field

```

dimensions      [0 2 -3 0 0 0 0];

#include        "include/initialConditions"

internalField   uniform $turbulentEpsilon;

boundaryField
{
    outlet
    {
        type          zeroGradient;
    }

    mainInlet
    {
        type          groovyBC;
        value         $internalField;
        variables
        "Uref=8.0;Href=10.0;z0=0.0002;kappa=0.41;Cmu=0.033;ustar=Uref*
        kappa/log((Href+z0)/z0);";
        fractionExpression "1";
        valueExpression"(((pow (ustar,3))/(kappa*(pos().z+ z0))))";
    }

    secondaryInlet
    {
        type          fixedValue;
        value         $internalField;
    }

    ground
    {
        type          compressible::epsilonWallFunction;
        //type        epsilonWallFunction;
        Cmu           0.09;
        kappa         0.4;
        E             9.8;
        value         $internalField;
    }
}

#include "include/sideAndTopPatches"
}

```

Appendix B.7: P field

```
#include      "include/initialConditions"

dimensions   [1 -1 -2 0 0 0 0];

internalField  uniform 1e5;

boundaryField
{
    mainInlet
    {
        type          zeroGradient;
    }

    secondaryInlet
    {
        type          zeroGradient;
    }

    outlet
    {
        type          fixedValue;
        value         $internalField;
    }

    ground
    {
        type          zeroGradient;
    }

    #include "include/sideAndTopPatches"
}
```

Appendix B.8: T field

```
dimensions      [0 0 0 1 0 0 0];

internalField   uniform 283;

boundaryField
{
    mainInlet
    {
        type          fixedValue;
        value          uniform 283;
    }
    secondaryInlet
    {
        type          fixedValue;
        value          uniform 483;
    }

    ground
    {
        type          fixedValue;
        value          uniform 283;
    }

    outlet
    {
        type          inletOutlet;
        inletValue     uniform 283;
        value          uniform 283;
    }
    #include "include/sideAndTopPatches"
}
}
```

Appendix B.9: alphas field

```
dimensions      [1 -1 -1 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    ground
    {
        type      alphasWallFunction;
        Prt       0.85;
        value     uniform 0;
    }
    mainInlet
    {
        type      calculated;
        value     uniform 0;
    }

    secondaryInlet
    {
        type      calculated;
        value     uniform 0;
    }

    outlet
    {
        type      calculated;
        value     uniform 0;
    }

    #include "include/sideAndTopPatches"
}
```

Appendix B.10: thermophysicalProperties

```
thermoType
hPsiThermo<pureMixture<constTransport<specieThermo<hConstTherm
o<perfectGas>>>>>;

pRef          100000;

mixture
{
    specie
    {
        nMoles          1;
        molWeight        28.9;
    }
    thermodynamics
    {
        Cp              1000;
        Hf              0;
    }
    transport
    {
        mu              1.8e-05;
        Pr              0.7;
    }
}
```

Appendix B.11: fvSchemes

```

ddtSchemes
{
    default          steadyState;
}
gradSchemes
{
    default          Gauss linear;
}
divSchemes
{
    default          none;
    div(phi,U)      Gauss linearUpwindV grad(U);
    div(phi,h)      Gauss upwind;
    div(phi,K)      Gauss upwind;
    div(phi,k)      Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div(phi,R)      Gauss upwind;
    div(R)          Gauss linear;
    div((muEff*dev2(T(grad(U)))) Gauss linear;
}
laplacianSchemes
{
    default          none;
    laplacian(muEff,U) Gauss linear uncorrected;
    laplacian((rho*(1|A(U))),p_rgh) Gauss linear uncorrected;
    laplacian(alphaEff,h) Gauss linear uncorrected;
    laplacian(DkEff,k) Gauss linear uncorrected;
    laplacian(DepsilonEff,epsilon) Gauss linear uncorrected;
    laplacian(DREff,R) Gauss linear uncorrected;
}
interpolationSchemes
{
    default          linear;
}
snGradSchemes
{
    default          corrected;
}
fluxRequired
{
    default          no;
    p_rgh;
}

```


Appendix B.12: fvSolution

```

solvers
{
    p
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-08;
        relTol          0.01;
    }
    p_rgh
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-08;
        relTol          0.01;
    }
    "(U|h|k|epsilon|R)"
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0.1;
    }
}
SIMPLE
{
    nNonOrthogonalCorrectors 0;
    pRefCell          0;
    pRefValue         0;
    residualControl
    {
        p_rgh          1e-2;
        U              1e-3;
        h              1e-3;
        // possibly check turbulence fields
        "(k|epsilon|omega)" 1e-3;
    }
}
relaxationFactors
{
    fields
    {
        rho            1.0;
        p_rgh          0.7;
    }
    equations
    {
        U              0.2;
        h              0.2;
        "(k|epsilon|R)" 0.5;
    }
}

```

}

Appendix C: ABL simulations around helideck

Appendix C.1: blockMeshDict

```

convertToMeters 1;

vertices
(
    (-200 -45 0)
    (280 -45 0)
    (280 115 0)
    (-200 115 0)
    (-200 -45 160)
    (280 -45 160)
    (280 115 160)
    (-200 115 160)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (72 24 24) simpleGrading (1 1 1)
);

edges
(
);

boundary
(
    sides
    {
        type patch;
        faces
        (
            (3 7 6 2)
            (1 5 4 0)
        );
    }

    outlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }

    inlet
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }

    ground

```

```

    {
        type wall;
        faces
        (
            (0 3 2 1)
        );
    }

    top
    {
        type patch;
        faces
        (
            (4 5 6 7)
        );
    }
};

```

Appendix C.2: snappyHexMeshDict

```

castellatedMesh true;
snap true;
addLayers false;
geometry
{
    platform.stl
    {
        type triSurfaceMesh;
        name platform;
    }
    refinementPlatform
    {
        type searchableCylinder;
        point1 (16.75 36.5 60);
        point2 (16.75 36.5 110);
        radius 45;
    }
};
castellatedMeshControls
{
    maxLocalCells 100000;
    maxGlobalCells 7000000;
    minRefinementCells 10;
    maxLoadUnbalance 0.10;
    nCellsBetweenLevels 6;
    features
    (
        );
    refinementSurfaces

```

```
{
  platform
  {
    level (3 3);
  }
}
resolveFeatureAngle -1;
refinementRegions
{
  refinementPlatform
  {
    mode inside;
    levels ((1E15 1));
  }
}
locationInMesh (-45 -25.7 30);
allowFreeStandingZoneFaces true;
}
snapControls
{
  nSmoothPatch 3;
  tolerance 4.0;
  nSolveIter 1;
  nRelaxIter 5;
}
addLayersControls
{
  relativeSizes true;
  layers
  {
    platform_geom
    {
      nSurfaceLayers 3;
    }
  }
  expansionRatio 1.0;

  finalLayerThickness 0.1;
  minThickness 0.01;
  nGrow 3;
  featureAngle 90;
  nRelaxIter 3;
  nSmoothSurfaceNormals 1;
```

```
nSmoothNormals 3;
nSmoothThickness 10;
  maxFaceThicknessRatio 0.5;
  maxThicknessToMedialRatio 0.3;
  minMedianAxisAngle 90;
  nBufferCellsNoExtrude 0;
  nLayerIter 50;
}
meshQualityControls
{
  maxNonOrtho 65;
  maxBoundarySkewness 1;
  maxInternalSkewness 1;
  maxConcave 80;
  minVol 1e-13;
  minTetQuality 1e-30;
  minArea -1;
  minTwist 0.02;
  minDeterminant 0.001;
  minFaceWeight 0.02;
  minVolRatio 0.01;
  minTriangleTwist -1;
  nSmoothScale 4;
  errorReduction 0.75;
}
debug 0;
mergeTolerance 1e-6;
```

Appendix C.3: ABL conditions

```
Uref          8.0;
Href          10;
z0            uniform 0.002;
windDirection (-1 0 0);
zDirection    (0 0 1);
zGround       uniform 0.0;
ABLvariables
"Uref=8.0;Href=10.0;z0=0.0002;kappa=0.4;Cmu_b=0.033;ustar=Uref
*kappa/log((Href+z0)/z0);";
Rtensor       uniform (0.4997 0.4014 0.2613 0.3224 0.2099 0.1367);
```

Appendix C.4: initial conditions

```
flowVelocity  (-8 0 0);
pressure      0;
turbulentKE   0.5;
turbulentEpsilon 0.01;
```

Appendix C.5: side and top patches

```
top
{
    type slip;
}
sides
{
    type slip;
}
```

Appendix C.6: u field

```

#include          "include/initialConditions"
dimensions       [0 1 -1 0 0 0 0];
internalField    uniform $flowVelocity;
boundaryField
{
    #include "include/ABLConditions"
    outlet
    {
        type          zeroGradient;
    }
    inlet
    {
        type          groovyBC;
        value         uniform (0 0 0);
        variables     $ABLvariables;
        fractionExpression "1";
        valueExpression "vector(-
(ustar/kappa)*log((pos().z+z0)/z0), 0, 0)";
    }
    ground
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }
    platform_geom
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }

    #include "include/sideAndTopPatches"
}

```


Appendix C.7: k field

```

#include          "include/initialConditions"

dimensions       [0 2 -2 0 0 0 0];

internalField    uniform $turbulentKE;

boundaryField
{
    #include "include/ABLConditions"

    outlet
    {
        type          zeroGradient;
    }

    inlet
    {
        type          groovyBC;
        value         uniform 0.5;
        variables     $ABLvariables;
        fractionExpression "1";
        valueExpression "pow(ustar,2)/sqrt(Cmu_b)";
    }

    ground
    {
        type          kqRWallFunction;
        value         $internalField;
    }

    platform_geom
    {
        type          kqRWallFunction;
        value         $internalField;
    }

    #include "include/sideAndTopPatches"
}

```

Appendix C.8: ϵ field

```

dimensions      [0 2 -3 0 0 0 0];
#include         "include/initialConditions"
internalField   uniform $turbulentEpsilon;
boundaryField
{
    #include "include/ABLConditions"
    ground
    {
        type          epsilonWallFunction;
        Cmu           0.09;
        kappa         0.4;
        E             9.8;
        value         $internalField;
    }
    platform_geom
    {
        type          epsilonWallFunction;
        Cmu           0.09;
        kappa         0.4;
        E             9.8;
        value         $internalField;
    }

    outlet
    {
        type          zeroGradient;
    }

    inlet
    {
        type          groovyBC;
        value         uniform 0.1;
        variables     $ABLvariables;
        fractionExpression "1";
        valueExpression
"pow(ustar,3)/(kappa*(pos().z+z0))";
    }

    #include "include/sideAndTopPatches"
}

```

Appendix C.9: R field

```
dimensions      [0 2 -2 0 0 0 0];

#include "include/ABLConditions";

internalField $Rtensor;

boundaryField
{
    sides
    {
        type      slip;
    }
    outlet
    {
        type      zeroGradient;
    }
    inlet
    {
        type      fixedValue;
        value     $Rtensor;
    }
    ground
    {
        type      kqRWallFunction;
        value     $Rtensor;
    }
    top
    {
        type      slip;
    }
    platform_geom
    {
        type      kqRWallFunction;
        value     $Rtensor;
    }
}
```

Appendix C.10: fvSchemes

```
ddtSchemes
{
    default          steadyState;
}

gradSchemes
{
    default          Gauss linear;
}

divSchemes
{
    default          none;
    div(phi,U)      Gauss upwind;
    div(phi,k)      Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div(phi,R)      Gauss upwind;
    div(R)          Gauss linear;
    div((nuEff*dev(T(grad(U)))) Gauss linear upwind;
}

laplacianSchemes
{
    default          Gauss linear corrected;
}

interpolationSchemes
{
    default          linear;
}

snGradSchemes
{
    default          corrected;
}

fluxRequired
{
    default          no;
    p;
}
```

Appendix C.11: fvSolution

```

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-5;
        relTol          0.1;
        smoother        GaussSeidel;
        nPreSweeps      0;
        nPostSweeps     2;
        cacheAgglomeration on;
        agglomerator    faceAreaPair;
        nCellsInCoarsestLevel 10;
        mergeLevels     1;
    }

    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-5;
        relTol          0.1;
        nSweeps         1;
    }

    k
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-5;
        relTol          0.1;
        nSweeps         1;
    }

    epsilon
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-5;
        relTol          0.1;
        nSweeps         1;
    }
}
R

```

```
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-5;
        relTol          0.1;
        nSweeps         1;
    }
}
SIMPLE
{
    nNonOrthogonalCorrectors 3;
    residualControl
    {
        p 1e-2;
        U 1e-3;
        k 1e-3;
        epsilon 1e-3;
        R 1e-3;
    }
}
potentialFlow
{
    nNonOrthogonalCorrectors 10;
}

relaxationFactors
{
    fields
    {
        p          0.3;
    }
    equations
    {
        U          0.6;
        k          0.6;
        epsilon    0.6;
        R          0.6;
    }
}

cache
{
    grad(U);
}
```

Appendix C.12:sampleDict

```

interpolationScheme cellPoint;
setFormat          raw;
sets
(
    heliProfile
    {
        type      uniform;
        axis      z;
        start     ( 16 37 65.5 );
        end       ( 16 37 110 );
        nPoints   250;
    }

);
fields              ( U k epsilon nut R );

```

Appendix C.13:decomposeParDict

```

numberOfSubdomains 6;
method              hierarchical;
simpleCoeffs
{
    n                (4 1 1);
    delta            0.001;
}

hierarchicalCoeffs
{
    n                (3 2 1);
    delta            0.001;
    order            xyz;
}

manualCoeffs
{
    dataFile         "cellDecomposition";
}

```

Appendix D: content of enclosed CD

The enclosed CD contains the master thesis file and OpenFOAM® simulations main folders (0, constant and system folders) in a ZIP format. The content of CD is as follow:

- Thesis report file in pdf format
- Thesis report file in docs format
- ABL simulation files for simple geometry
- Hot plume simulation files
- ABL simulations files for coarse mesh and k- ϵ turbulence model
- ABL simulations files for fine mesh and k- ϵ turbulence model
- ABL simulations files for coarse mesh and LRR turbulence model