THE IMPACTS OF AIRBORNE CLOUD MICROPHSYICAL INSTRUMENTATION MOUNTING LOCATION ON MEASUREMENTS MADE DURING THE OBSERVATIONS OF AEROSOLS AND CLOUDS AND THEIR INTERACTIONS (ORACLES) PROJECT

By

Joseph Robert O’Brien

Bachelor of Science, Millersville University, 2013

Master of Science, University of North Dakota, 2016

A Dissertation

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Grand Forks, North Dakota

December

2021

Copyright 2021 Joseph R. O’Brien

## This dissertation, submitted by Joseph R. O’Brien in partial fulfillment of the requirements for the Degree of Doctor of Philosophy from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

David J. Delene – Chairperson

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Michael R. Poellot

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Greg M. McFarquhar

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Andrew Detwiler

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Frank Bowman

This dissertation is being submitted by the appointed advisory committee as having met all of the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Chris Nelson

Dean of the School of Graduate Studies

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date

PERMISSION

## Title: The Impacts of Airborne Cloud Microphysical Instrumentation Mounting Location on Measurements Made During the ObseRvations of Aerosols above CLouds and their intEractionS (ORACLES) Project

## Department: Atmospheric Sciences

## Degree: Doctor of Philosophy

## In presenting this dissertation in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my dissertation work or, in his absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this dissertation or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my dissertation.

## Joseph R. O’Brien

## **{DATE**}

TABLE OF CONTENTS

LIST OF FIGURES vi

LIST OF TABLES vii

ACKNOWLEDGEMENTS viii

ABSTRACT x

CHAPTER

INTRODUCTION AND BACKGROUND 1

ORACLES 4

MOTIVATION 5

AIRFLOW INVESTIGATIONS 7

PROPOSED WORK 10

METHODOLOGY 12

OPENFOAM – PRE-PROCESSING 13

OPENFOAM- SOLVER 16

OPENFOAM – *rhoSimpleFoam –* SUB-MODELS 19

OPENFOAM – LAGRANGIAN PARTICLE TRACKING 20

OPENFOAM – CASE DIRECTORY 21

OPENFOAM – NASA P-3 CASES 23

RESULTS N/A

DISCUSSION N/A

SUMMARY AND CONCLUSIONS N/A

REFERENCES 25

APPENDIX A – *snappyHexMesh* CONFIGURATION FILES 29

APPENDIX B – *thermoPhysicalProperties* CONFIGURATION FILES 37

APPENDIX C – BOUNDARY CONDITIONS CONFIGURATION FILES 39

APPENDIX D – SYSTEM DIRECTORY CONFIGURATION FILES 44

LIST OF FIGURES

Figure Page

1. Cloud Droplet Probe (CDP) mounting locations with respect to the leading with Respect to the leading edge of the wings on-board the NASA P-3 Orion during ORACLES 2018 field experiment in Sao Tome. Figure 1-A (Figure 1-B) contains the Navy (Extended) pylon design with the Langley Airborne Research Center (Uni. Of Hawaii) CDP………………………………………. 8
2. Three-dimensional computer-aided design (CAD) models of the NASA P-3 Navy pylon configuration (Figure 2-A) and Extended pylon configuration (Figure 2-B). Each configuration contains designs for the Cloud Droplet Probe and “bullet” canister used during ORACLES 2018……………………………. 14
3. From Greenshields (2016), example of the mesh generation from the *snappyHexMesh* meshing tool (Fig. 5.8, Fig 5.10, Fig 5.11, and 5.13 in paper).. 16
4. Mach Number for the NASA P-3 Orion during in-situ cloud sampling throughout ORACLES. Shading for each violin plot represents data distribution. Black (red) lines indicate mean (median) of the distribution……... 17
5. Directory structure for the OpenFOAM solver, rhoSimpleFoam, for the NASA\_P3\_extendedPylon\_v2\_tas120\_aoa0\_900T33 case. Listed are all required files for processing. Names listed within brackets indicate a directory.. 22
6. ORACLES aircraft environmental parameters used to initialize the rhoSimpleFoam solver………………………………………………………….. 24

LIST OF TABLES

Table Page

1. Description of the configuration of the compressible flow simulations to investigate airflow around the NASA P-3………………………………. 23

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to the members of my advisory Committee for their guidance and support during my time in the Doctor of Philosophy program at the University of North Dakota.

To my mom and dad,

Thank you for the support and encouragement these many years.

ABSTRACT

In progress…

*The abstract should be a succinct summary of the thesis/dissertation in three parts: a presentation of the problem, account of the research and methods used, and the conclusions of the research. There is no word limit on your abstract. However, ProQuest publishes print indexes that include citations and abstracts of all dissertations and theses published by ProQuest/UMI. These print indexes require limits of 350 words for doctoral dissertations and 150 words for master’s theses. In the editorial process for these print publications, ProQuest/UMI will simply truncate your abstract if it exceeds these word limits and remove any non-text content. You may wish to limit the length of your abstract if this concerns you.*

CHAPTER I

INTRODUCTION AND BACKGROUND

One of the largest remaining uncertainties within anthropogenic climate forcing are the impacts of atmospheric aerosols. Aerosols are a colloidal system in which the dispersed phased is composed of either solid or liquid particles, and the dispersion medium is some gas, typically air (SOURCE). Common examples of aerosols are smoke, dust, pollen, and salt. Aerosols are typically classified by their location (e.g. urban, rural, desert, etc.), type (e.g. continental or marine), and origin (e.g. natural or anthropogenic). For each of these classifications, the size distribution and composition of atmospheric aerosols are diverse and can be described with three modes: Aitken, accumulation, and coarse. Within the Aitken mode, aerosol radii are smaller than 0.1 microns and are formed by chemical conversion from gas to liquid or solid particulates. Aitken mode aerosols typically grow rapidly and coagulate to form accumulation mode aerosols, or aerosols with radii between 0.1 microns and 1 microns. Within this accumulation mode, aerosols are too small to be removed from the atmosphere through sedimentation, and thus accumulate until scavenged, or removed, by cloud droplets and rainfall. To be scavenged by cloud droplets, hygroscopic aerosols can serve as nuclei on which water condenses at typical atmospheric supersaturations (AMS GLOSSARY). These aerosols are known as cloud condensation nuclei (CCN). Coarse mode aerosols, aerosols with radii greater than 1 micron, are dominated by mechanical production

(i.e. wind driven) of dust and sea spray, and also by combustion sources. Unlike accumulation mode aerosols, coarse mode aerosols are large enough to be removed by sedimentation (along with rainfall) and are thus mostly constrained to the lower troposphere. As the production of aerosols are dependent on the source of the aerosol mode, concentrations of aerosols within the atmosphere vary greatly from region to region, with higher concentrations found in urban environments due to anthropogenic production.

Within the atmosphere, aerosols can affect the radiation balance of the Earth-atmosphere system through the scattering and absorption of solar radiation, known as the aerosol direct effect (Liou 2002). An incident radiated energy is scattered by a particle in its path when the direction of propagation is altered due to reflection, refraction, or diffraction without absorption taking place (Jacob 1999). Scattering is most efficient for a particle radius in proximity to the wavelength of the corresponding radiation, and can be described by a physical term called *size parameter (x)*:

|  |
| --- |
|  |

where a is the radius of the particle and λ is the wavelength of radiation (Jacob 1999; Liou 2002). When the particle radius is much smaller than the wavelength of radiation (x << 1), the scattering is called Rayleigh scattering, where the efficiency of the scattering is proportional to the fourth power of the wavelength in the direction of propagation. This process is responsible for the appearance of a blue sky, as the blue wavelength is scattered more efficiently than the longer red wavelength. Mie scattering occurs when the radius of the particle is roughly the same size as the wavelength of radiation (x = ~1), where the efficiency of the scattering is proportional to the square power of the wavelength, resulting in a stronger signal than Rayleigh scattering that is also has a strong angular dependency. Absorption is the process where an incident radiated energy is retained by a particle, which then isotropically (i.e. not varying in magnitude with direction) emits a fraction of the energy at a longer wavelength.

The radiative properties of aerosols are dependent on their radiative indices as functions of the incident wavelength and on the size, shape and chemical composition of the aerosol. Direct radiative forcing of aerosols, or estimates of the change in energy flux within the atmosphere due to natural or anthropogenic factors of climate change, is dependent on the ratio of absorption to total extinction and also relies on the albedo of the underlaying surface-atmosphere layer (Coakley and Chylek 1975; Redemann et al. 2021). The Fifth Assessment Report by the Intergovernmental Panel on Climate Change (IPCC) assessed the direct radiative forcing of aerosols as -0.35 W m-2 [-0.85 to +0.15 W m-2], with a negative radiative forcing indicting a cooling effect at the atmosphere due to the backscattering of incident solar radiation (Boucher et al. 2013; Bellouin et al. 2020).

Among the uncertainties within the radiative forcing of aerosols are interactions between clouds and aerosols produced through biomass burning (BB), which refers to the burning of living and dead vegetation through human-induced burns as well as natural lightning induced fires. The capacity of BB aerosols to absorb and scatter incident solar radiation is dependent on the nature of the burned biomass and burning conditions (Brioude et al. 2009). One of the emissions of biomass burning is black carbon soot aerosol, the strongest absorbing aerosol within the atmosphere (Redemann et al. 2021). Previous investigations have found that the sign of the direct aerosol forcing is dependent on location of the BB aerosols with respect to underlaying clouds as the black carbon aerosol decrease the albedo of the underlaying cloud (Chand et al. 2009).

Two additional aerosol cloud interactions (ACIs) that contribute to the uncertainties within the radiative forcing of aerosols are the indirect and semi-direct aerosol effect. The aerosol indirect effect refers to the processes which involve aerosols acting as CCN, with the two most prominent theories being the Twomey and Albrecht effects (Lohmann and Feichter 2005). Twomey (1977) investigated the role of aerosols (through natural and anthropogenic causes) on droplet activation within fixed cloud water amounts, and found that with increasing CCN, smaller, more numerous cloud droplets were formed. This resulted in increased cloud albedo, enhancing the reflectance of solar radiation and producing a slight negative radiative forcing at the top of the atmosphere (TOA). Albrecht (1989) investigated the Twomey effect in varying cloud water amounts with marine stratocumulus (Sc), proposing a link between cloud lifetime and aerosols through precipitation. Albrecht (1989) argued that while the Twomey effect creates more numerous, smaller droplets (with increasing aerosol concentrations), the result of a decrease in droplet radius is the reduction in collision-coalescence and precipitation efficiency. Albrecht (1989) suggested that the decrease in precipitation efficiency would increase cloud liquid water and fractional cloudiness, resulting in brighter, longer lasting Sc.

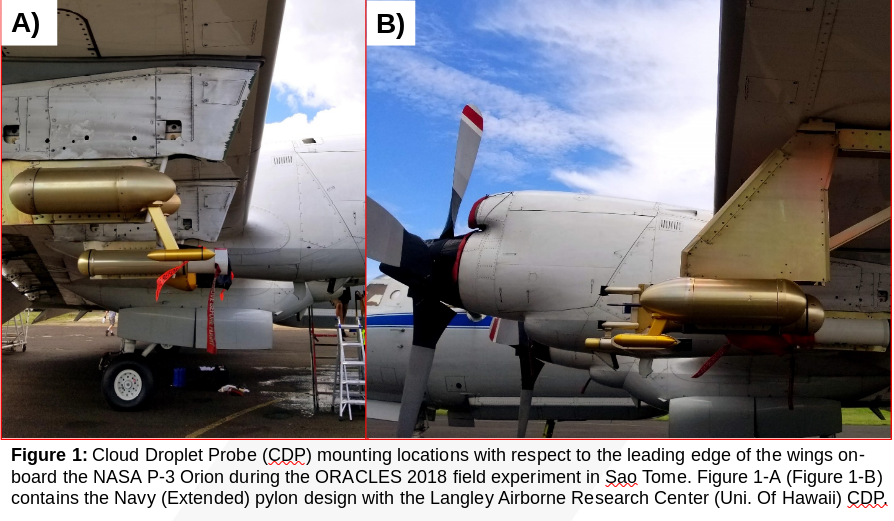
The semi-direct aerosol effect refers to the interactions between absorbing aerosols and clouds within the same vertical column of atmosphere, where the heating of mid-troposphere by aerosol solar absorption influences cloud dynamics (Hansen et al. 1997). Hansen et al. (1997) theorized that the presence of black carbon aerosols within the marine boundary layer (MBL) would lead to increased heating and reduction in relative humidity and large scale cloud cover. Reduction in cloud cover would result in the decrease of planetary albedo and a positive radiative forcing. Conversely, using large-eddy simulations of the marine stratocumulus off the Californian coast, Johnson et al. (2004) showed that the sign of the radiative forcing of the semi-direct effect depends on the vertical location of the aerosol plume with respect to the marine stratocumulus. Johnson et al. (2004) was also able to show that marine stratocumulus geometric thickness was influenced by the vertical distribution of the aerosol and cloud layers, with an increase in LWP when these are vertically separated due to the relaxation of the entrainment of dry mid-tropospheric air into the marine boundary layer. With the uncertainty within the sign and magnitude of the radiative forcing of BB aerosols, investigations into the interactions between BB aerosols and underlying cloud populations are needed.

ORACLES Field Project

ORACLES (ObseRvations of Aerosols above CLouds and their intEractionS) was a

five-year NASA investigation into the climate impacts of Southern Africa’s BB aerosols (Redemann et al. 2021). The southern African continent accounts for almost one third of the Earth’s BB emissions, with burned areas in southern Africa increasing even with the decreasing trend in global (Andela et al. 2017; Redemann et al. 2021). BB aerosol emissions from southern Africa are routinely transported west off the continent and over one of the world’s three semi-permanent marine stratocumulus (Sc) cloud decks. These regions, in which the annual mean Sc cloud cover is between 40-60%, reside in the subtropical eastern oceans, roughly 5-10° longitude to the west of North America, South America, and Africa (Wood 2012). The abundance of BB aerosols, Sc cloud coverage, and limited previous in-situ observational experiments made the southeast Atlantic region advantageous to investigate aerosol-cloud interactions. Three separate field experiments were conducted off the coast of Africa, utilizing the NASA P-3 aircraft in September 2016, August 2017, and October 2018. The University of North Dakota (UND), in coordination with the Cooperative Institute for Severe and High-Impact Weather Research and Operations (CIWRO), the University of Oklahoma (OU) and University of Illinois at Urbana-Champaign (UIUC), integrated and operated a suite of in-situ cloud microphysical instrumentation into the NASA P-3 aircraft. The research objectives of this group were to investigate the impact of aerosols on cloud properties and precipitation, primarily the aerosol indirect effect in precipitation suppression and cloud lifetime (Albrecht 1989; Twohy et al. 2005; Lohmann and Feichter 2005, Fan et al. 2016). To investigate these microphysical interactions on the macrophysical properties of Sc, accurate environmental and cloud in-situ measurements are needed. However, during the course of the ORACLES campaigns, the accuracy of the cloud probe measurements was uncertain due to the mounting location of instruments with respect to the aircraft wing. This probe measurement uncertainty made conducting analysis of the aerosol-cloud interaction process problematic.

MOTIVATION

In order to have accurate in-situ cloud microphysical measurements, the volume of air sampled at the instrument location must be known. Typically, to simplify these sample volume measurements, it is often assumed that the sampling locations are representative of freestream (i.e. undisturbed, unaltered, upstream) atmospheric flow. However, pressure perturbations ahead of the aircraft and the instruments themselves are known to alter particle trajectories at the sample volumes of the instruments. Additional deviations of airflow speed and direction occur as the aircraft undergoes maneuvers during flight. The overall concern is that uncertainty due to sampling location may be larger than the uncertainty due to determining the sample volume. To minimize uncertainty due to these perturbations, instruments are typically mounted on pylons that extend the sample volumes to a location some distance away from the aircraft. The P-3 Aircraft used underwing pylons for mounting probes. While the goal is to have instruments sample freestream conditions, there is no agreement on a pylon configuration that would accomplish for this for wing mounted instrumentation. Additionally, it is currently unknown if instruments should be mounted ahead or behind the leading edge of the aircraft wing to achieve freestream conditions. Historically, instruments have been mounted ahead of the leading edge of the aircraft wing, which is why the original pylon configuration (“Navy pylon”) on the NASA P-3 (Fig. 1-A) was concerning. To address these concerns, a new pylon configuration (“Extended pylon”) was developed by the NASA Wallops Island Flight Facility engineers to allow the instrumentation to be located as far below and ahead of the leading edge of the aircraft wing as possible in order to sample within the freestream. The extended pylon shown in Fig. 1-B, was manufactured for the ORACLES-2017 campaign and installed at the outboard pylon location of the left wing. To help determine which pylon configuration provided observations that were most representative of the Sc cloud environment, analysis of simultaneous observations from two Droplet Measurement Technologies (DMT) Cloud Droplet Probes (CDP) were used during ORACLES-2018. CDPs provide observations on the size and number of cloud droplets, 2-50 microns in size, through measurement of the forward scattering of cloud droplets. With an airspeed provided from the aircraft and determination of the CDP sample volume calculation, derived parameters of cloud droplet concentrations, liquid water content and effective radius can be analyzed. After eight research flights, pylon locations of the two CDPs were swapped in order to minimize the effect of each specific instrument on the analysis. Analysis of these simultaneous CDP observations found that for the vertical flight profiles through Sc, known as ‘sawtooths’, measured cloud droplet concentrations that were 4-6% greater on the Navy pylon design as compared to the Extended pylon design. While the relative difference in concentrations between pylons configurations is shown with this CDP comparison, there continue to be questions about which configuration would provide data that are more representative of the Sc cloud environment.

**Figure 1:** Cloud Droplet Probe (CDP) mounting locations with respect to the leading edge of the wings on-board the NASA P-3 Orion during the ORACLES 2018 field experiment in Sao Tome. Figure 1-A (Figure 1-B) contains the Navy (Extended) pylon design with the Langley Airborne Research Center (Uni. Of Hawaii) CDP.

AIRFLOW INVESTIGATIONS

Since the advent of reliable cloud microphysical instrumentation, considerable research has been done to investigate the air flow and particle trajectories around research aircraft in an effort to improve measurement performance. The concern of these studies is whether instruments are placed such that they sample free-stream conditions. Beard (1983) investigated the orientation of precipitation particle images within a Particle Measurement Systems (PMS) Inc optical array probe (OAP), noting the titling of the particle images from direction of flight (i.e. canting angle) was dependent on particle size and habit. From the canting angle and deformation of the particle images, Beard (1983) calculated the acceleration of airflow ahead of the PMS canisters (assuming incompressible flow ahead of a simple sphere), showing flow distortion up to 50 cm ahead of the instrument. King (1984) and King et al. (1984) were able to calculate the potential flow (i.e. incompressible, inviscid, and non-turbulent flow) and particle trajectories around Commonwealth Scientific and Industrial Research Organization (CSIRO) F-27 research aircraft to study instrument placement on the aircraft fuselage. King (1984) was able to show that the maximum width of the shadow zone (i.e. regions, often larger than the fuselage boundary layer, that do not contain particles of a certain size) occurs out to 0.2 fuselage radius (distanced normalized by the radius of the aircraft fuselage) and airflow departures from freestream velocities occur up to 3 fuselage radii ahead of the nose. Based on the equations of motion derived in King (1984), Drummond and MacPherson (1985) studied the air flow effects on droplets measured by four PMS OAPs on board the National Aeronautical Establishment of Canada’s Twin Otter aircraft. Drummond and MacPherson (1985) found that droplet concentrations are under-sampled by up to 25% for aircraft lift coefficients (i.e. dimensionless coefficient that relates lift to fluid flow around a body) of 0.79 (~4 degrees angle of attack). Aircraft lift coefficients can also be used to describe the relationship between a particular 2-D airfoil and the angle of attack, with the maximum lift coefficient for a given airfoil occurring at the stall angle. MacPherson and Baumgardner (1988) investigated airflow around wing-mounted PMS 2D-C instruments onboard the National Center for Atmospheric Research (NCAR) Beechcraft King Air. After noticing non-symmetrical icing on the outboard 2D-C, MacPherson and Baumgardner (1988) conducted extensive wind tunnel tests of the King Air pylon assembly with chalk and oil, calculating velocity ratios and side flow angles for multiple angles of attack at two sample locations (inboard and outboard). MacPherson and Baumgardner (1988) were able to show a deceleration and turning of streamlines within 0.75 meters of the 2D-C sampling plane, corresponding to droplet relaxation time of 60 micron diameter particles. MacPherson and Baumgardner (1988) suggest that droplets smaller than this size would follow streamline trajectories, but they do not calculate particle trajectories within their study.

Norment (1988) also studied the air flow effects on a PMS 2D-C and a PMS Forward Scattering Spectrometer Probe (FSSP) on board the NASA Lewis Research Center DHC6 Twin Otter aircraft. Similar to Drummond and MacPherson (1985), the PMS instruments used within the Norment (1988) study were mounted behind the leading edge of the aircraft wing. Norment (1988) modeled the potential flow around the Twin Otter wing, PMS canister and specific instruments for two aircraft true air speeds (49 and 128 m/s) and two angles of attack (0 and 4 degrees). While the study found that for wing-mounted instruments results are insensitive to freestream air speed, Norment (1988) found that the PMS FSSP under-sampled cloud droplets (5-50 microns diameter) by 10-13% at 0 degrees angle of attack, and 18-24% at 4 degrees angle of attack.

While these efforts have attempted to create guidelines for instrumentation placement or correction to data from instruments known to be poorly located, instruments are still integrated into research aircraft today on an ad hoc basis to suit structural integrity. Recently, there has been a renewed effort to re-evaluate the air flow and trajectory analysis for individual aircraft and pylon assemblies. O’Brien (2016) observed the preferred, vertical orientation of ice crystals within the sample volume of the Stratton Park Engineering Company (SPEC, Inc) Two-Dimensional Stereo Probe (2D-S) OAP on board the University of North Dakota Citation II Research Aircraft, suggesting an induced electric field on the aircraft (along with airflow deceleration) was responsible for the particle orientations. Following discussions about their unique under-wing (and behind the leading edge) pylons on the Facility for Airborne Atmospheric Measurement’s (FAAM) BAe-146-301 aircraft flown during ORACLES-2017, Bennett et al. (2019) performed a CFD investigation to understand the aerodynamics around their pylon assembly. Using ANSYS Fluent (version 16), Bennett et al. (2019) were able to produce solutions for compressible flows via an implicit, steady-state, density-based solver that was independently verified with observations from the Droplet Measurement Technologies (DMT) Aircraft-Integrated Meteorological Measurement System (AIMMS). While Bennett et al. (2019) only considered the effect of longitudinal and lateral velocity components on the sizing of hydrometeors within the DMT Cloud Imaging Probe (CIP-100), it nevertheless shows the ability of the current state of CFD to accurately produce the aerodynamics around aircraft instrumentation. Spanu et al. (2020) conducted numerical simulations for compressible flows with Reynolds-averaged Navier-Stokes equations (RANS), a Launder-ShamaKE turbulence model, and Lagrangian particle tracking within OpenFOAM (version 4.0x) to investigate flow around wing-mounted instruments on board the DSR Dassault Falcon 20E. Through these simulations, Spanu et al. (2020) was able to determine the sampling efficiency (i.e. concentration of particles within the sample volume compared to free steam concentrations) for the DMT Cloud and Aerosol Spectrometer (CAPS). At typical cloud sampling conditions (100 m/s true air speed, and static pressure 900 mb), cloud droplets less than 100 microns in diameter were shown to have a sampling efficiency of 77%, where droplets larger than 100 microns were shown to be minimally affected (Spanu et al. 2020).

OBJECTIVE

To complement the ORACLES-2018 CDP analysis, a computational fluid dynamical (CFD) study of the NASA P-3 pylon configurations would offer the opportunity to determine which configuration allows for the true representation of the cloud environment. Additionally, a CFD analysis would allow for increased confidence in the suitability of the NASA P-3 aircraft for future cloud microphysical missions. objective of this proposed study is to determine the effects of airflow on the measurements made by cloud microphysical instrumentation installed behind the leading edge of the aircraft wing, where the majority of the ORACLES in-situ cloud microphysical data were obtained. Through a computational fluid dynamic (CFD) analysis of the NASA P-3 pylon configurations used during ORACLES, we will address the following scientific questions:

1. What is the sampling efficiency of cloud droplets at the instrument mounting location for the Navy and Extended NASA P-3 pylon configurations?

2. For future cloud microphysical missions, should wing-mounted instruments sample ahead or behind of the leading edge of the aircraft wing?

CHAPTER II

METHODOLOGY

Computational Fluid Dynamics – OpenFOAM

Previous analysis into airflow around the DMT Cloud-Aerosol-Precipitation Spectrometer (CAPS) by Spanu et al. (2020) has shown the viability of the Open Source Field Operation and Manipulation (OpenFOAM) package for atmospheric research. OpenFOAM is open source CFD software owned by the OpenFOAM foundation, distributed under a General Public license and was created in the C++ programming language with an object-oriented coding approach (Jasak et al.). OpenFOAM is based on the finite volume method, which with the use of the C++ operator overloading feature, is able to represent and evaluate partial differential equations through a series of scalar-vector operations (Chen et al. 2014). A particular benefit of this approach is the ability for researchers to understand the syntactical language of the code due to its mimicking of mathematical formula. The open-source nature of OpenFOAM allows for the development of community code and tutorials for researchers starting out with CFD analysis, and removes the cost barrier of commercial software fees for many University researchers.

The core of the OpenFOAM distribution can be broken into two categories: *solvers* and *utilities* (Greenshields 2017). *Utilities* contain scripts and modules for pre-processing and post-processing each simulation. Pre-processing involves the generation and manipulation of simple or complex volumetric meshes. A mesh, or grid, divides a domain into many cells that contain nodes, faces, and edges (i.e. boundaries). Depending on the unique requirements of

the CFD solutions, meshes can be configured into structured or unstructured grids in two-dimensional or three-dimensional space. For structured grids, meshes consist of quadrilateral (two-dimensional) or hexahedrons (three-dimensional) cells and are always parallel to the coordinate axes, which can be expressed as Cartesians or polar coordinates. Additional structured meshes, such as block-structures, are domains decomposed into a small number of structured meshes, which each region potentially containing different mesh resolutions. Unstructured meshes consist of tetrahedral (two-dimensional) and hexahedral (three-dimensional) cells which can be configured to accommodate completely arbitrary geometries at increased computational costs compared to structured meshes. Volumetric meshes are meshes that represent both the surface and volume of a structure within a domain.

Post-processing contains an extensive graphical user interface (GUI), *paraview*, in order to visualize and plot each simulation, as well as tools to evaluate each simulation. *Solvers* are physical models designed to solve a problem within a range of physical categories, which include compressible and incompressible flow; electromagnetics; multiphase flow; Lagrangian particle dynamics; and stress analysis (Chen et al. 2014, Greenshields 2017). Each of these standard solvers dictate the boundary conditions, space and temporal dimensions, and physical properties (and their gradients) calculated during each simulation.

OPENFOAM – PRE-PROCESSING

To achieve quality results within OpenFOAM, realistic models of the desired pylon configurations are needed for numerical calculations. Three-dimensional computer-aided design (CAD) models of the NASA P-3 pylon configurations, with associated instrument canisters and instruments, were provided by the NASA Wallops Island Flight Facility and Diagram

Description automatically generatedshown in Figure 2. With the FreeCAD software package, additional CAD models of both pylon configurations attached to the P-3 wing hard-points at Station 9 (furthest outboard, port-side location) were created from the NASA Wallops Island Flight Facility CAD models. FreeCAD is an open-source, three-dimensional parametric CAD modeling application that allows users to produce models of objects for a variety of sizes and purposes (SOURCE). The primary use of FreeCAD is to generate meshes from solid geometries for three-dimensional printing. The mesh generation features of FreeCAD allow for the length between each node to be specified to dictate how the surface mesh conforms to the geometry surface. Once the NASA P-3 pylon configurations were assembled, surface meshes of the assembly were created within FreeCAD. Due to limited computational allowances, the length between surface nodes for the P-3 fuselage were set to ten millimeters (10 mm). The length between surface nodes for the P-3 pylon and OAP canisters were set to the maximum resolution of a hundred microns (0.1 mm). The NASA P-3 pylon configuration surface meshes were saved in the Wavefront OBJ file format, a human readable format that represents three-dimensional geometry by storing the position of each vertex (SOURCE). Due to the differences in the native units between FreeCAD and OpenFOAM, the entire P-3 pylon configuration surface mesh was scaled by one thousandth (1/1000) to convert saved vertex positions from millimeters to meters.

**Figure 2:** Three-dimensional computer-aided design (CAD) models of the NASA P-3 Navy pylon configuration (Figure 2-A) and Extended pylon configuration (Figure 2-B). Each configuration contains designs for the Cloud Droplet Probe and “bullet” canister used during ORACLES 2018.

To generate the volume mesh within OpenFOAM, a simplistic three-dimensional cubic domain is first created with the *blockMesh* utility. Similar to previous studies (King 1984; Norment 1988; Spanu et al. 2020), the domain is configured to be 10 times the length of the pylon assembly in order to minimize the effects of domain boundaries, resulting in a volume mesh three hundred meters in length in each dimension. To incorporate the complex surface mesh of the P-3 pylon configurations generated by FreeCAD into the volume mesh, the OpenFOAM utility *snappyHexMesh* is used to conform the volume mesh to the CAD model surface. As shown in Figure 3, *snappyHexMesh* chisels the volume mesh to the geometry surface by splitting each volume mesh cell overlapping the position of the geometry surface mesh and iteratively refining each cell to snap to the object surface. To do this, the edges of the surface mesh are first defined with the *surfaceFeatureExtract* utility (Figure 3-A). Cell splitting is performed on the edges specified by this utility and then across the surface of the geometry (Figure 3-B). After every cell on the parameter of the geometry has been split, if desired, boundary layer cells are added on the perimeter of the geometry (Figured 3-C). Finally, cells and edges within the geometry parameter are removed to “snap in” the surface mesh (Figured 3-D). All of these features are controlled by the *snappyHexMeshDict* configuration file within the openFOAM case. The *snappyHexMesh* configuration for the NASA P-3 pylon analysis is contained in Appendix A.

OPENFOAM - SOLVER

In each of their studies, Bennett et al. (2019) and Spanu et al. (2020) raise concerns on the validity of air speed calculations within King (1984) due to the assumption of an incompressible fluid. As aircraft speed approaches 0.3 Mach, air density can no longer Chart

Description automatically generatedconsidered independent of velocity, and using Benouli’s equation for incompressible flow leads to a 10% overestimation of air speed Spanu et al. (2020). As shown in Figure 4, the average mach number of the ORACLES cloud sampling profiles exceeds 0.3 Mach (0.35 average Mach for sawtooth profiles; 0.39 average Mach for level cloud profiles). Therefore, to accurately describe the flow around the P-3 pylon assembly, an OpenFOAM steady-state *solver, rhoSimpleFoam,* will be used to calculate compressible, turbulent flow, using a Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm. The SIMPLE algorithm follows a segregated solution strategy, where each of the governing equations are solved in turn and inserted into subsequent equation until the model solutions converge (Moukalled and Darwish 2016). For compressible flow, the conservation equations of momentum, energy (including the equation of state) and continuity for a Newtonian fluid acting as an ideal gas are:Chart, box and whisker chart

Description automatically generated

**Figure 3*:*** From Greenshields (2016), example of the mesh generation from the *snappyHexMesh* meshing tool (Fig. 5.8, Fig 5.10, Fig 5.11, and 5.13 in paper)

**Figure 4:** Mach Number for the NASA P-3 Orion during in-situ cloud sampling throughout ORACLES. Shading for each violin plot represents data distribution. Black (red) lines indicate mean (median) of the distribution.

|  |
| --- |
| Momentum: |
| Energy: |
|  |
|  |
| Equation of State: |
| Continuity: |

where ρ is the density of the fluid, **v** is the velocity vector in three dimensions, ρ**vv** is the dyadic product, μ is the molecular viscosity coefficient, **fB** is the body force term, cp is specific heat capacity, T is the temperature of the fluid, k is the thermal conductivity of the substance, p is the pressure, is the rate of heat source or sink within the material volume.

From MouKalled and Darwish (2016), the SIMPLE algorithm follows as such:

1. The solution is started at time *t* for pressure, velocity, density, temperature and mass flow rate fields as the initial guesses (i.e. boundary conditions).
2. Solutions of the momentum equation to obtain a new velocity field *v\** are found.
3. The equation of state is used to calculate a new density field *ρ\**.
4. The mass flow rates at the control volume faces are updated to a obtain a momentum satisfying mass flow rate, *m\**.
5. With the new mass flow rates, the coefficients of the pressure corrections are found and applied to obtain a pressure correction field *p’*.
6. The pressure, density and velocity fields at the control volume centers and the mass flow rate at the control volume faces are updated to obtain continuity-satisfying fields.
7. The energy equation is solved to obtain a new temperature field *T\**.
8. Set *v\**, *ρ\**, *T\**, *p\**, and *m\** as the initial guess for velocity, density, temperature, pressure, and mass flow.
9. If solution does not meet convergence criteria, algorithm goes to step two and repeats.

OPENFOAM *rhoSimpleFoam* – SUB-MODELS

Similar to Spanu et al. (2020), *rhoSimpleFOAM* solver will use Reynolds-averaged Navier-Stokes equations (RANS) with a sheer stress transport (SST) k-ω turbulence model and a fixed composition thermophsyical model, *hePsiTherm*. The SST k-ω turbulence model was developed to better predict flows with adverse pressure gradients and allows for easier integration throughout a complex mesh, where k is the turbulent kinetic energy and ω is the specific turbulence dissipation rate (Wilcox 1998, MouKalled and Darwish 2016). Within this framework, the specific turbulence dissipation ω is defined as:

|  |
| --- |
|  |

where ϵ is the rate of dissipation of turbulent kinetic energy. From MouKalled and Darwish (2016), the two conservation equations to define the k-ω model are:

|  |
| --- |
|  |
|  |

with the assigned values to the model constraints:

|  |
| --- |
|  |

where

|  |
| --- |
|  |
|  |
|  |
|  |

The thermophysical model, *hePsiThermo*, is pressure-temperature system, based on compressibility, that is constructed to describe the energy, heat, and physical properties of a fixed composition, or non-reacting, fluid. Within hePsiThermo, a transport model used for internal energy and enthalpy calculations is configured to assume constant dynamic viscosity *μ*, and the thermal conductivity *Κ* is determined by the Prandtl number

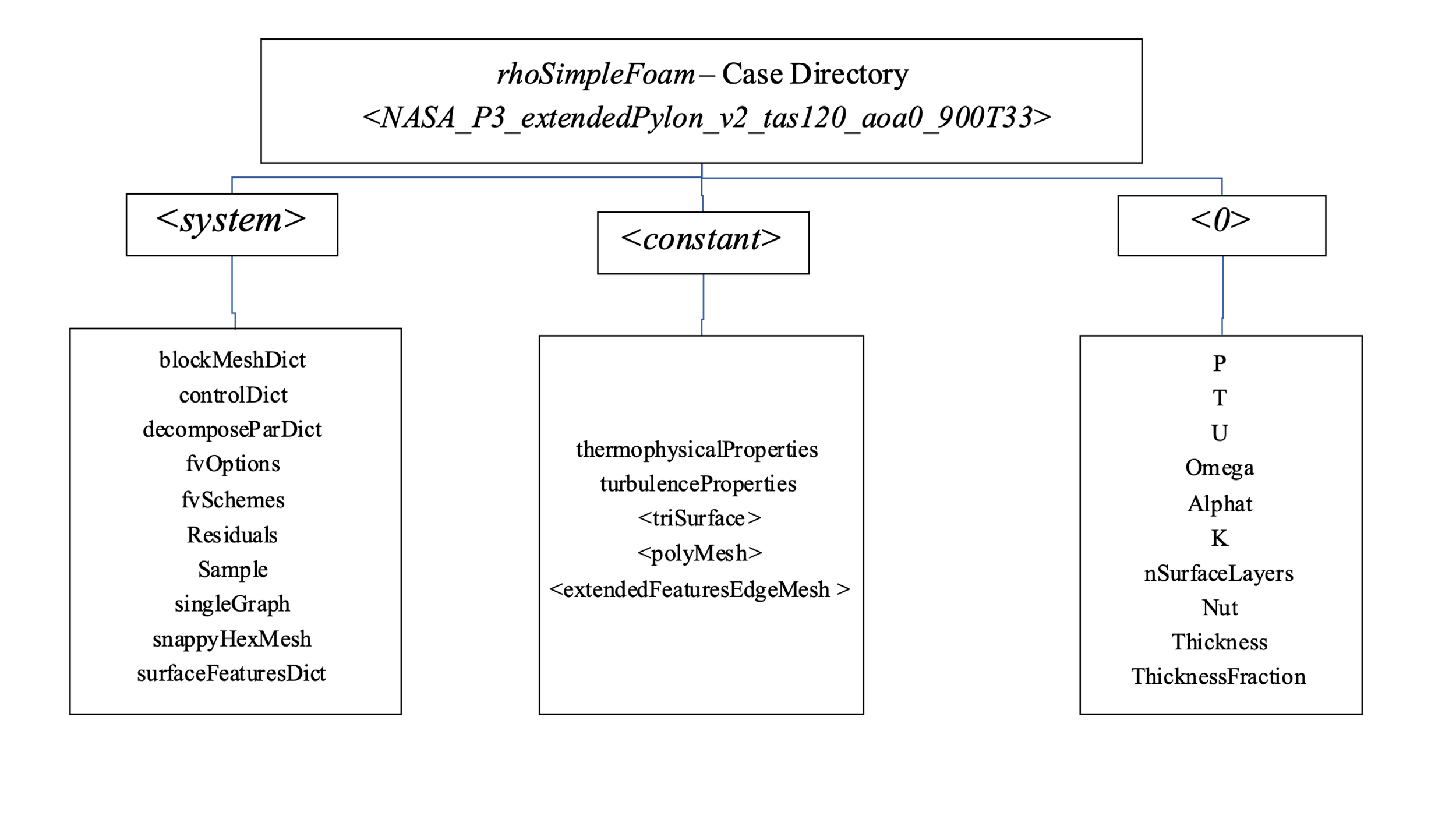
|  |
| --- |
|  |

The model is configured to assume a constant specific heat and heat of fusion, as well as, the equation of state for a perfect gas, defined in (4). The species of the fluid to be modeled is set by the molecular weight, which is configured to air at room temperature. Each parameter set is included within the *thermophysicalProperties* files within the case directory and can be found in Appendix B.

OPENFOAM – LAGRANGIAN PARTICLE TRACKING

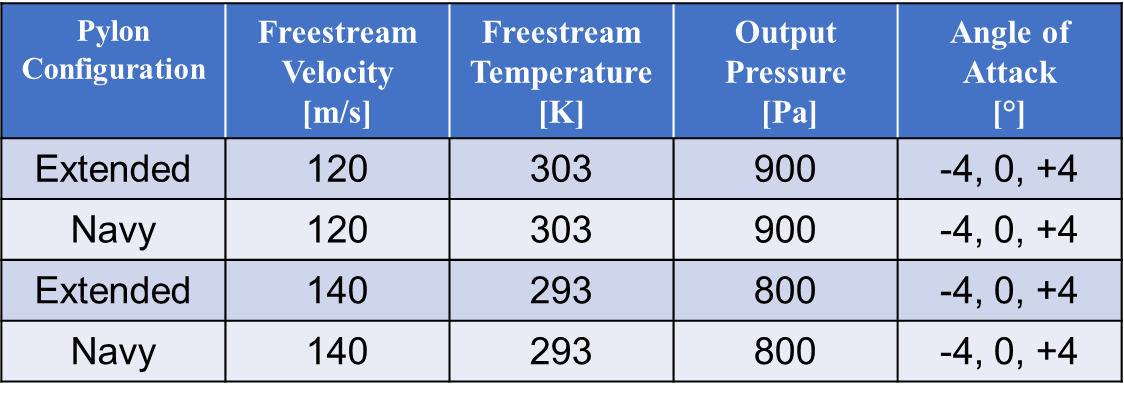
Along with analysis into the streamlines around the P-3 pylon assembly calculated from the *rhoSimpleFoam* solution, the sampling efficiency at the instrument sampling location is needed. As in Norment (1988) and Spanu et al. (2020), the sampling efficiency is calculated to be the concentration of droplets at the sampling location compared to the droplet concentration at the domain border. In order to accurately calculate sampling efficiency for the pylon assembly, solutions to describe particle motion within the compressible flow solutions are needed. As in Spanu et al. (2020), an Eulerian-Lagrangian approach will be undertaken to solve for one-way coupling of multi-phase flow. *RhoSimpleFoam* will be used to calculate a solution of the fluid phase (i.e. freesteam air), while a Lagrangian particle solver, *simpleReactingParcleFoam,* will be used to calculate the Newtonian forces upon a particulate phase (i.e. water hydrometers). We will assume a one-way coupling of the multi-phase flows, such that only flow-induced drag forces and gravity are acting on the particles. The influence of particle to particle interactions on the fluid will not be considered. To help verify these simulations, droplet concentrations for a range of aircraft speeds, flight levels, and angle of attacks will be compared with the measured CDP data from ORACLES-2018.

OPENFOAM – CASE DIRECTORY

The structure of the case directory for the compressible flow simulations of the NASA P-3 aircraft is described in Figure 5. To differentiate between cases, each case is structured to contain the airframe, specific pylon, version number, and selected airspeed, angle of attack, pressure and temperature within the name of the directory. The environmental parameters (e.g. pressure, temperature, velocity, turbulent kinetic energy *k* and specific turbulence dissipation *ω*) are each defined in their respective files underneath the initial time directory, *0*. For the velocity file *U*, the three dimensional velocity components of the inlet patch is set in meters per second, along with the boundary conditions for the volume and surface meshes. For the volume mesh, described by the variable *freestream,* the *freestreamVelocity* boundary condition is set to provide freestream velocity throughout the domain. For the surface mesh, described by the variable *wall*, a *noSlip* boundary condition fixes the velocity to zero on the cells in direct contact with the geometry. For the temperature file *T*, the temperature of the inlet patch is set in Kelvin and applied uniformly throughout the domain. The volume mesh is given boundary condition *inletOutlet*, which requires the input and output patches of the domain to have the same temperature. The surface mesh is given the boundary condition *zeroGradient*, which extrapolates the temperature of the geometry from the nearest cell value. For the pressure file *P*, the pressure of the outlet patch is set in pascals, with a *freestreamPressure* boundary condition applied to the volume mesh that provides freestream pressure to the domain. Additionally, the surface mesh boundary condition is set to *zeroGradient*, previously described. For the turbulent kinetic energy file *k,* the inlet patch value is set in squared meters per seconds squared, with the volume mesh boundary condition set to *inletOutlet*. For the surface mesh, the *kqRwallFunction* is applied, which provides a wrapper around the *zeroGradient* condition to add specific parameter weights for the sublayer assumptions of the turbulence model. The specific dissipation rate *ω* is setup similar to the *k* file, with the surface mesh boundary condition set to *omegaWallFunction*. This boundary conditions provides a constraint on the dissipation rate and production of turbulent kinetic energy as determined by the turbulence model. The rest of the files within the *0* directory are related to output from the *snappyHexMesh* algorithm, which defines the layers and thickness of the surface mesh. Each of these files can be found in Appendix C.

**Figure 5**: Directory structure for the OpenFOAM solver, rhoSimpleFoam, for the NASA\_P3\_extendedPylon\_v2\_tas120\_aoa0\_900T33 case. Listed are all required files for processing. Names listed within brackets indicate a directory.

Within the *constant* directory, the vertices of every cell within the domain are defined within the *triSurface* directory in a human-readable format. Every cell face and surface are contained within the *polyMesh* directory, with the edges of the surface mesh defined within the *extendedFeaturesEdgeMesh* directory. The parameters for the thermophysical and turbulence models are also included within this directory.

Within the *system* directory, control of the processing (i.e. number of integrations, write control, write style, etc) of the simulation are contained within the *controlDict* file. The *fvOptions* and *fvSolution* files contain the options and interpolation schemes for the simulation. The *residual*, *sample* and *singleGraph* files are used to output specific information from each simulation in human-readable formats for a given set of coordinates within the domain. All of these files are contained within Appendix D.

**Table 1:** Description of the configuration of the compressible flow simulations to investigate airflow around the NASA P-3.

OPENFOAM – NASA P-3 CASES

To investigate the airflow around the NASA P-3 for the Extended and Navy pylon configurations, simulations of both pylon configurations are performed for multiple airspeeds, pressure and temperature combinations and angle of attacks as described within Table I. The airspeed, pressure and temperature parameters were determined from the average environmental parameters observed during the ORACLES-2018 cloud profiles, shown in Figure 6. Multiple airspeeds, pressures, and temperatures were chosen to determine the sensitivity of the compressible flow solutions to airspeed and environmental temperature and pressure. As there was no available angle of attack measurement from the NASA P-3 during ORACLES, the maximum range of angle of attacks was chosen for this analysis to capture the full range of flow distortion by the movement of the aircraft. Since any change to the orientation of the aircraft within the domain would require reprocessing of the *snappyHexMesh* algorithm, the angle of attack was implemented within each simulation by Chart, box and whisker chart

Description automatically generatedaltering the three-dimensional velocity components of the input patch within the *U* initialization file.

**Figure 6:** ORACLES aircraft environmental parameters used to initialize the rhoSimpleFoam solver

REFERENCES

Albrecht, B. A., 1989: Aerosols, cloud microphysics, and fractional cloudiness. *Science*, **245**, 1227+.

Andela, N., and Coauthors, 2017: A human-driven decline in global burned area. *Science*, **356**, 1356–1362, https://doi.org/10.1126/science.aal4108.

Beard, K. V., 1983: Reorientation of Hydrometeors in Aircraft Accelerated Flow. *J. Appl. Meteorol. Climatol.*, **22**, 1961–1963, https://doi.org/10.1175/1520-0450(1983)022<1961:ROHIAA>2.0.CO;2.

Bellouin, N., and Coauthors, 2020: Bounding Global Aerosol Radiative Forcing of Climate Change. *Rev. Geophys.*, **58**, e2019RG000660, https://doi.org/10.1029/2019RG000660.

Bennett, C. J., G. J. Nott, A. Wellpott, N. Lawson, M. Delise, B. Woodcock, and G. B. Gratton, 2019: Characterizing Instrumentation Canister Aerodynamics of FAAM BAe-146-301 Atmospheric Research Aircraft. *J. Aerosp. Eng.*, **32**, 04019047, https://doi.org/10.1061/(ASCE)AS.1943-5525.0001044.

Boucher, O., and Coauthors, 2013: Clouds and Aerosols. *Clim. Change 2013 Phys. Sci. Basis Contrib. Work. Group Fifth Assess. Rep. Intergov. Panel Clim. Change*, 571–657.

Brioude, J., and Coauthors, 2009: Effect of biomass burning on marine stratocumulus clouds off the California coast. *Atmos Chem Phys*, **9**, 8841–8856, https://doi.org/10.5194/acp-9-8841-2009.

Chand, D., R. Wood, T. L. Anderson, S. K. Satheesh, and R. J. Charlson, 2009: Satellite-derived direct radiative effect of aerosols dependent on cloud cover. *Nat. Geosci.*, **2**, 181–184, https://doi.org/10.1038/ngeo437.

Coakley, J. A., and P. Chylek, 1975: The Two-Stream Approximation in Radiative Transfer: Including the Angle of the Incident Radiation. *J. Atmospheric Sci.*, **32**, 409–418, https://doi.org/10.1175/1520-0469(1975)032<0409:TTSAIR>2.0.CO;2.

Drummond, A. M., and J. I. MacPherson, 1985: Aircraft Flow Effects on Cloud Drop Images and Concentrations Measured by the NAE Twin Otter. *J. Atmospheric Ocean. Technol.*, **2**, 633–643, https://doi.org/10.1175/1520-0426(1985)002<0633:AFEOCD>2.0.CO;2.

Fan, J., Y. Wang, D. Rosenfeld, and X. Liu, 2016: Review of Aerosol–Cloud Interactions: Mechanisms, Significance, and Challenges. *J. Atmospheric Sci.*, **73**, 4221–4252, https://doi.org/10.1175/JAS-D-16-0037.1.

Hansen, J., M. Sato, A. Lacis, and R. Ruedy, 1997: The missing climate forcing. *Philos. Trans. R. Soc. B Biol. Sci.*, **352**, https://doi.org/10.1098/rstb.1997.0018.

Jacob, D. J., 1999: *Introduction to Atmospheric Chemistry*. Princeton University Press,.

Johnson, B. T., K. P. Shine, and P. M. Forster, 2004: The semi-direct aerosol effect: Impact of absorbing aerosols on marine stratocumulus. *Q. J. R. Meteorol. Soc.*, **130**, 1407–1422, https://doi.org/10.1256/qj.03.61.

King, W. D., 1984: Air Flow and Particle Trajectories around Aircraft Fuselages. I: Theory. *J. Atmospheric Ocean. Technol.*, **1**, 5–13, https://doi.org/10.1175/1520-0426(1984)001<0005:AFAPTA>2.0.CO;2.

——, D. E. Turvey, D. Williams, and D. J. Llewellyn, 1984: Air Flow and Particle Trajectories around Aircraft Fuselages. II: Measurements. *J. Atmospheric Ocean. Technol.*, **1**, 14–21, https://doi.org/10.1175/1520-0426(1984)001<0014:AFAPTA>2.0.CO;2.

Liou, K. N., 2002: *An Introduction to Atmospheric Radiation*. Elsevier, 598 pp.

Lohmann, U., and J. Feichter, 2005: Global indirect aerosol effects: a review. *Atmos Chem Phys*, **5**, 715–737, https://doi.org/10.5194/acp-5-715-2005.

MacPherson, J. I., and D. Baumgardner, 1988: Airflow about King Air Wingtip-Mounted Cloud Particle Measurement Probes. *J. Atmospheric Ocean. Technol.*, **5**, 259–273, https://doi.org/10.1175/1520-0426(1988)005<0259:AAKAWM>2.0.CO;2.

Moukalled, F., and M. Darwish, 2016: Fluid Flow Computation: Compressible Flows. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*, F. Moukalled, L. Mangani, and M. Darwish, Eds., *Fluid Mechanics and Its Applications*, Springer International Publishing, 655–690.

Norment, H. G., 1988: Three-Dimensional Trajectory Analysis of Two Drop Sizing instruments: PMS\* OAP and PMS\* FSSP. *J. Atmospheric Ocean. Technol.*, **5**, 743–756, https://doi.org/10.1175/1520-0426(1988)005<0743:TDTAOT>2.0.CO;2.

O’brien, J. R., Sensitivity Of Two-Dimensional Stereo (2DS) Probe Derived Parameters To Particle Orientation. 108.

Redemann, J., and Coauthors, 2021: An overview of the ORACLES (ObseRvations of Aerosols above CLouds and their intEractionS) project: aerosol–cloud–radiation interactions in the southeast Atlantic basin. *Atmospheric Chem. Phys.*, **21**, 1507–1563, https://doi.org/10.5194/acp-21-1507-2021.

Spanu, A., M. Dollner, J. Gasteiger, T. P. Bui, and B. Weinzierl, 2020: Flow-induced errors in airborne in situ measurements of aerosols and clouds. *Atmospheric Meas. Tech.*, **13**, 1963–1987, https://doi.org/10.5194/amt-13-1963-2020.

Twohy, C. H., M. D. Petters, J. R. Snider, B. Stevens, W. Tahnk, M. Wetzel, L. Russell, and F. Burnet, 2005: Evaluation of the aerosol indirect effect in marine stratocumulus clouds: Droplet number, size, liquid water path, and radiative impact. *J. Geophys. Res. Atmospheres*, **110**, D08203, https://doi.org/10.1029/2004JD005116.

Twomey, S., 1977: The Influence of Pollution on the Shortwave Albedo of Clouds. *J. Atmospheric Sci.*, **34**, 1149–1152, https://doi.org/10.1175/1520-0469(1977)034<1149:TIOPOT>2.0.CO;2.

Wood, R., 2012: Stratocumulus Clouds. *Mon. Weather Rev.*, **140**, 2373–2423, https://doi.org/10.1175/MWR-D-11-00121.1.

(Pending Citations)

Wilcox, David C. *Turbulence modeling for CFD*. Vol. 2. La Canada, CA: DCW industries, 1998.

APPENDIX A

*snappyHexMesh* CONFIGURATION FILE

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class dictionary;

object snappyHexMeshDict;

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

// Which of the steps to run

castellatedMesh true;

snap true;

addLayers false;

// Geometry. Definition of all surfaces. All surfaces are of class

// searchableSurface.

// Surfaces are used

// - to specify refinement for any mesh cell intersecting it

// - to specify refinement for any mesh cell inside/outside/near

// - to 'snap' the mesh boundary to the surface

geometry

{

pmsCanister

{

type triSurfaceMesh;

file "NASA\_P3\_extendedPylon\_CDP\_OAP\_v5.obj";

}

refinementBox

{

type searchableBox;

min (-40.0 -40.0 -40.0);

max ( 40.0 40.0 -40.0);

// original refinement box;

//min ( -1.0 -0.7 0.0);

//max ( 8.0 0.7 2.5);

}

refinementBox2

{

type searchableBox;

min (13.5 2.5 13.0);

max (15.5 4.0 15.0);

}

};

// Settings for the castellatedMesh generation.

castellatedMeshControls

{

// Refinement parameters

// ~~~~~~~~~~~~~~~~~~~~~

// If local number of cells is >= maxLocalCells on any processor

// switches from from refinement followed by balancing

// (current method) to (weighted) balancing before refinement.

//maxLocalCells 100000;

//maxLocalCells 2000000;

maxLocalCells 4000000;

// Overall cell limit (approximately). Refinement will stop immediately

// upon reaching this number so a refinement level might not complete.

// Note that this is the number of cells before removing the part which

// is not 'visible' from the keepPoint. The final number of cells might

// actually be a lot less.

//maxGlobalCells 2000000;

//maxGlobalCells 40000000;

maxGlobalCells 80000000;

// The surface refinement loop might spend lots of iterations refining just a

// few cells. This setting will cause refinement to stop if <= minimumRefine

// are selected for refinement. Note: it will at least do one iteration

// (unless the number of cells to refine is 0)

//minRefinementCells 20;

minRefinementCells 200;

// Allow a certain level of imbalance during refining

// (since balancing is quite expensive)

// Expressed as fraction of perfect balance (= overall number of cells /

// nProcs). 0=balance always.

maxLoadUnbalance 0.10;

// Number of buffer layers between different levels.

// 1 means normal 2:1 refinement restriction, larger means slower

// refinement.

nCellsBetweenLevels 3;

//nCellsBetweenLevels 15;

// Explicit feature edge refinement

// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// Specifies a level for any cell intersected by its edges.

// This is a featureEdgeMesh, read from constant/triSurface for now.

features

(

{

file "NASA\_P3\_extendedPylon\_CDP\_OAP\_v5.eMesh";

level 6;

}

);

// Surface based refinement

// ~~~~~~~~~~~~~~~~~~~~~~~~

// Specifies two levels for every surface. The first is the minimum level,

// every cell intersecting a surface gets refined up to the minimum level.

// The second level is the maximum level. Cells that 'see' multiple

// intersections where the intersections make an

// angle > resolveFeatureAngle get refined up to the maximum level.

refinementSurfaces

{

pmsCanister

{

// Surface-wise min and max refinement level

level (5 6);

// Optional specification of patch type (default is wall). No

// constraint types (cyclic, symmetry) etc. are allowed.

patchInfo

{

type wall;

inGroups (pmsCanisterGroup);

}

}

}

// Resolve sharp angles

//resolveFeatureAngle 30;

//resolveFeatureAngle 10;

resolveFeatureAngle 3;

// Region-wise refinement

// ~~~~~~~~~~~~~~~~~~~~~~

// Specifies refinement level for cells in relation to a surface. One of

// three modes

// - distance. 'levels' specifies per distance to the surface the

// wanted refinement level. The distances need to be specified in

// descending order.

// - inside. 'levels' is only one entry and only the level is used. All

// cells inside the surface get refined up to the level. The surface

// needs to be closed for this to be possible.

// - outside. Same but cells outside.

refinementRegions

{

refinementBox

{

mode inside;

levels ((1E15 4));

}

}

// Mesh selection

// ~~~~~~~~~~~~~~

// After refinement patches get added for all refinementSurfaces and

// all cells intersecting the surfaces get put into these patches. The

// section reachable from the locationInMesh is kept.

// NOTE: This point should never be on a face, always inside a cell, even

// after refinement.

locationInMesh (3.0001 3.0001 0.43);

//locationInMesh (0.0 0.0 4.5);

// Whether any faceZones (as specified in the refinementSurfaces)

// are only on the boundary of corresponding cellZones or also allow

// free-standing zone faces. Not used if there are no faceZones.

//allowFreeStandingZoneFaces true;

allowFreeStandingZoneFaces false;

}

// Settings for the snapping.

snapControls

{

//- Number of patch smoothing iterations before finding correspondence

// to surface

nSmoothPatch 3;

//- Relative distance for points to be attracted by surface feature point

// or edge. True distance is this factor times local

// maximum edge length.

tolerance 1.0;

//- Number of mesh displacement relaxation iterations.

//nSolveIter 30;

nSolveIter 300;

//- Maximum number of snapping relaxation iterations. Should stop

// before upon reaching a correct mesh.

nRelaxIter 5;

// Feature snapping

//- Number of feature edge snapping iterations.

// Leave out altogether to disable.

nFeatureSnapIter 50;

//- Detect (geometric only) features by sampling the surface

// (default=false).

implicitFeatureSnap false;

//- Use castellatedMeshControls::features (default = true)

explicitFeatureSnap true;

//- Detect points on multiple surfaces (only for explicitFeatureSnap)

//multiRegionFeatureSnap false;

multiRegionFeatureSnap false;

}

// Settings for the layer addition.

addLayersControls

{

// Are the thickness parameters below relative to the undistorted

// size of the refined cell outside layer (true) or absolute sizes (false).

relativeSizes true;

// Per final patch (so not geometry!) the layer information

layers

{

"pmsCanister.\*"

{

nSurfaceLayers 1;

}

}

// Expansion factor for layer mesh

expansionRatio 1.0;

// Wanted thickness of final added cell layer. If multiple layers

// is the thickness of the layer furthest away from the wall.

// Relative to undistorted size of cell outside layer.

// See relativeSizes parameter.

finalLayerThickness 0.3;

// Minimum thickness of cell layer. If for any reason layer

// cannot be above minThickness do not add layer.

// Relative to undistorted size of cell outside layer.

minThickness 0.1;

// If points get not extruded do nGrow layers of connected faces that are

// also not grown. This helps convergence of the layer addition process

// close to features.

// Note: changed(corrected) w.r.t 17x! (didn't do anything in 17x)

nGrow 0;

// Advanced settings

// When not to extrude surface. 0 is flat surface, 90 is when two faces

// are perpendicular

featureAngle 60;

// At non-patched sides allow mesh to slip if extrusion direction makes

// angle larger than slipFeatureAngle.

slipFeatureAngle 30;

// Maximum number of snapping relaxation iterations. Should stop

// before upon reaching a correct mesh.

nRelaxIter 3;

// Number of smoothing iterations of surface normals

nSmoothSurfaceNormals 1;

// Number of smoothing iterations of interior mesh movement direction

nSmoothNormals 3;

// Smooth layer thickness over surface patches

nSmoothThickness 10;

// Stop layer growth on highly warped cells

maxFaceThicknessRatio 0.5;

// Reduce layer growth where ratio thickness to medial

// distance is large

maxThicknessToMedialRatio 0.3;

// Angle used to pick up medial axis points

// Note: changed(corrected) w.r.t 17x! 90 degrees corresponds to 130 in 17x.

minMedianAxisAngle 90;

// Create buffer region for new layer terminations

nBufferCellsNoExtrude 0;

// Overall max number of layer addition iterations. The mesher will exit

// if it reaches this number of iterations; possibly with an illegal

// mesh.

nLayerIter 50;

}

// Generic mesh quality settings. At any undoable phase these determine

// where to undo.

meshQualityControls

{

#include "meshQualityDict"

}

// Advanced

// Write flags

writeFlags

(

scalarLevels

layerSets

layerFields // write volScalarField for layer coverage

);

// Merge tolerance. Is fraction of overall bounding box of initial mesh.

// Note: the write tolerance needs to be higher than this.

mergeTolerance 1e-6;

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

APPENDIX B

*thermoPhysicalProperties* CONFIGURATION FILE

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class dictionary;

location "constant";

object thermophysicalProperties;

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

thermoType

{

type hePsiThermo;

mixture pureMixture;

transport const;

thermo hConst;

equationOfState perfectGas;

specie specie;

energy sensibleInternalEnergy;

}

mixture // air at room temperature (293 K)

{

specie

{

molWeight 28.9;

}

thermodynamics

{

Cp 1005;

Hf 0;

}

transport

{

mu 1.82e-05;

Pr 0.71;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

APPENDIX C

BOUNDARY CONDITIONS CONFIGURATION FILES

(0 DIRECTORY)

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class volVectorField;

**object U;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

Uinlet (120 0 0);

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

internalField uniform $Uinlet;

boundaryField

{

freestream

{

type freestreamVelocity;

freestreamValue uniform $Uinlet;

value uniform $Uinlet;

}

wall

{

type noSlip;

}

#includeEtc "caseDicts/setConstraintTypes"

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class volScalarField;

**object p;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

pOut 9e4;

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

internalField uniform $pOut;

boundaryField

{

freestream

{

type freestreamPressure;

freestreamValue uniform $pOut;

}

wall

{

type zeroGradient;

}

#includeEtc "caseDicts/setConstraintTypes"

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class volScalarField;

**object T;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

Tinlet 306;

dimensions [0 0 0 1 0 0 0];

internalField uniform $Tinlet;

boundaryField

{

freestream

{

type inletOutlet;

inletValue uniform $Tinlet;

value $inletValue;

}

wall

{

type zeroGradient;

//type fixedValue;

//wallValue uniform $Tinlet;

//value $wallValue;

}

#includeEtc "caseDicts/setConstraintTypes"

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class volScalarField;

**object omega;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

omegaInlet 400000;

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

internalField uniform $omegaInlet;

boundaryField

{

freestream

{

type inletOutlet;

inletValue uniform $omegaInlet;

value uniform $omegaInlet;

}

wall

{

type omegaWallFunction;

value uniform $omegaInlet;

}

#includeEtc "caseDicts/setConstraintTypes"

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class volScalarField;

**object k;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

kInlet 10.00;

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

internalField uniform $kInlet;

//internalField uniform 100;

boundaryField

{

freestream

{

type inletOutlet;

inletValue uniform $kInlet;

value uniform $kInlet;

}

wall

{

type kqRWallFunction;

value uniform $kInlet;

}

#includeEtc "caseDicts/setConstraintTypes"

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

APPENDIX D

SYSTEM DIRECTORY CONFIGURATION FILES

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class dictionary;

**object controlDict;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

application rhoSimpleFoam;

startFrom startTime;

startTime 0;

stopAt endTime;

endTime 10000;

deltaT 1;

writeControl timeStep;

writeInterval 2000;

purgeWrite 0;

//writeFormat ascii;

writeFormat binary;

writePrecision 6;

writeCompression off;

timeFormat general;

timePrecision 6;

runTimeModifiable true;

functions

{

#includeFunc MachNo

#includeFunc residuals

#includeFunc cuttingPlane

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class dictionary;

**object blockMeshDict**;

Does anyone actually read dissertations?

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

domain

{

// Hex A

xMin -150;

xMax 50;

yMin -100;

yMax 100;

zMin -100;

zMax 100;

// Number of cells

zCells 200; // aerofoil to far field

xCells 200; // sum of previous spots

yCells 200; // adding the dimension

}

vertices

(

// Hex A

($domain.xMin $domain.yMin $domain.zMin) // vertice 0

($domain.xMax $domain.yMin $domain.zMin) // vertice 1

($domain.xMax $domain.yMax $domain.zMin) // vertice 2

($domain.xMin $domain.yMax $domain.zMin) // vertice 3

($domain.xMin $domain.yMin $domain.zMax) // vertice 4

($domain.xMax $domain.yMin $domain.zMax) // vertice 5

($domain.xMax $domain.yMax $domain.zMax) // vertice 6

($domain.xMin $domain.yMax $domain.zMax) // vertice 7

);

blocks

(

hex (0 1 2 3 4 5 6 7) ($domain.xCells $domain.yCells $domain.zCells) simpleGrading (1 1 1)

);

edges

(

);

boundary

(

inlet

{

type patch;

inGroups (freestream);

faces

(

// Hex A

(0 4 7 3)

);

}

outlet

{

type patch;

inGroups (freestream);

faces

(

// Hex B

(1 5 6 2)

);

}

frontAndBack

{

type patch;

inGroups (freestream);

faces

(

// front

(0 4 5 1)

// back

(3 7 6 2)

);

}

topAndBottom

{

type patch;

inGroups (freestream);

faces

(

// top

(4 7 6 5)

// bottom

(0 3 2 1)

);

}

);

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class dictionary;

**object fvSolution;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

solvers

{

p

{

solver GAMG;

smoother GaussSeidel;

tolerance 1e-6;

relTol 0.01;

}

"(U|k|omega|e)"

{

solver PBiCGStab;

preconditioner DILU;

tolerance 1e-6;

relTol 0.1;

}

}

SIMPLE

{

residualControl

{

p 1e-4;

U 1e-4;

"(k|omega|e)" 1e-4;

}

nNonOrthogonalCorrectors 0;

pMinFactor 0.1;

pMaxFactor 2;

}

relaxationFactors

{

fields

{

p 0.7;

rho 0.01;

}

equations

{

U 0.3;

e 0.7;

"(k|omega)" 0.7;

}

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class dictionary;

**object fvSchemes;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

ddtSchemes

{

default steadyState;

}

gradSchemes

{

default Gauss linear;

limited cellLimited Gauss linear 1;

grad(U) $limited;

grad(k) $limited;

grad(omega) $limited;

}

divSchemes

{

default none;

div(phi,U) bounded Gauss linearUpwind limited;

turbulence bounded Gauss upwind;

energy bounded Gauss linearUpwind limited;

div(phi,k) $turbulence;

div(phi,omega) $turbulence;

div(phi,e) $energy;

div(phi,K) $energy;

div(phi,Ekp) $energy;

div(phid,p) Gauss upwind;

div((phi|interpolate(rho)),p) bounded Gauss upwind;

div(((rho\*nuEff)\*dev2(T(grad(U))))) Gauss linear;

}

laplacianSchemes

{

default Gauss linear corrected;

}

interpolationSchemes

{

default linear;

}

snGradSchemes

{

default corrected;

}

wallDist

{

method meshWave;

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

\\*---------------------------------------------------------------------------\*/

FoamFile

{

version 2.0;

format ascii;

class dictionary;

**object surfaceFeaturesDict;**

}

// \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* //

surfaces ("NASA\_P3\_extendedPylon\_CDP\_OAP\_v5.obj");

// Identify a feature when angle between faces < includedAngle

includedAngle 15;

subsetFeatures

{

// Keep nonManifold edges (edges with >2 connected faces)

nonManifoldEdges no;

// Keep open edges (edges with 1 connected face)

openEdges yes;

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //

/\*--------------------------------\*- C++ -\*----------------------------------\*\

========= |

\\ / F ield | OpenFOAM: The Open Source CFD Toolbox

\\ / O peration | Website: https://openfoam.org

\\ / A nd | Version: 7

\\/ M anipulation |

-------------------------------------------------------------------------------

**File: singleGraph**

Description

Writes graph data for specified fields along a line, specified by start

and end points.

\\*---------------------------------------------------------------------------\*/

start (10 2.72 13.75);

end (20.0 2.72 13.75);

fields (U p T rho Ma total(p));

// Sampling and I/O settings

#includeEtc "caseDicts/postProcessing/graphs/sampleDict.cfg"

// Override settings here, e.g.

setConfig

{

type lineCell;

axis xyz; // x, y, z, xyz

}

// Must be last entry

#includeEtc "caseDicts/postProcessing/graphs/graph.cfg"

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //