The Pennsylvania State University
The Graduate School

VERIFICATION AND VALIDATION OF SINGLE PHASE AND CAVITATING FLOWS USING AN OPEN SOURCE CFD TOOL

A Thesis in
Mechanical Engineering
by
Robert W. Erney

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2008

The thesis of Robert W. Erney was reviewed and approved* by the following:

Dr. Eric Paterson
Professor of Mechanical Engineering
Thesis Advisor, Chair of Committee

Dr. John Cimbala
Professor of Mechanical Engineering
Faculty Reader

Dr. Karen A. Thole
Professor of Mechanical Engineering
Department Head of Mechanical and Nuclear Engineering

*Signatures are on file in the Graduate School.

Abstract

This thesis presents an open source implementation of a cavitation model and assesses the effectiveness of the solver with previous work on several validation cases. The free computational fluid dynamics (CFD) package OpenFOAM is used to first validate single phase solutions on a flat plate, hemispherical headform, and a NACA0012 airfoil. Motivated by the need for a simulation tool that is free to the public yet still reliable, verification and validation are essential if a mainstream shift is to be made from the costly licensing fees of commercial codes to free open source software. After analyzing forces, convergence history, various finite volume schemes, grid studies, and turbulence models for simple, single-phase cases, the results are used as a baseline for more complicated cases such as two-phase cavitating hemispherical headforms and NACA hydrofoils. The CFD results show good agreement with the experimental data for both single phase and cavitating flows. For the single phase solutions, both the steady and unsteady solution algorithms were shown to produce accurate, converged results. Cavitating flow cases demonstrated that both the compressible and incompressible cavitation solvers are capable of providing accurate and realistic results when compared to proven experimental data. Verifying and validating these hydrodynamic benchmarks has provided insight into open source capabilities of OpenFOAM and allows future work to further investigate the phenomenon of cavitation on more sophisticated hydrodynamic equipment such as propellers and water jets.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The author would like to thank the Naval Sea Systems Command, specifically the SEA 073R3 program and Ms. Diane Segelhorst, for financial support on this thesis. I would like to thank my advisor, Dr. Eric Paterson for not only chosing me for this project, but also providing his guidance, support, and patience along the way. Last but certainly not least, I would like to thank Robert and Alicia Erney, for being the best parents a son could ask for.

Chapter 1

Introduction

Cavitation, an unfavorable effect in many systems, is defined as a rapid decrease of pressure resulting in a phase change to vapor from an initially homogeneous liquid. In most cases, cavitation arises when the liquid is exposed to high velocities, such as in marine propellers, pumps, hydraulic turbines, etc. When the vapor bubbles - which are formed from a rapid pressure drop - burst, they cause adverse effects to these types of mechanical systems. Cavitation can occur in many forms, including bubble, sheet, and vortex cavitation depending on the flow parameters. In particular for marine propellers, the violent collapse of vapor-filled cavitation bubbles over time eventually leads to noise and material damage on the blades. This type of process leads to torque and thrust breakdown that obviously degrades the performance of the propellers. Cavitation has been and continues to be a significant concern of propulsor designers for the Navy, not to mention the commercial marine industry. A non-dimensional parameter called the cavitation number is a way of characterizing the level of cavitation and is defined by:

$$\sigma_v = \frac{p - p_v(T)}{\triangle p} \tag{1.1}$$

where p is the local pressure, p_v is the vapor pressure of the fluid, and $\triangle p = \frac{1}{2}\rho U^2$ for the case of a propeller. The cavitation number cannot be considered a scaling parameter for non-cavitating flow and only becomes a similarity parameter at cavitation inception [1]. Cavitation inception is the point at which cavitation first appears and is defined as σ_{vi}. For this investigation however, fully developed cavitation is the interest of study, not the initial cavitation inception. Successfully predicting cavitation and its negative effects on thrust breakdown in propellers is increasingly important as fuel prices continue to rise.

In order to prevent such occurrences of cavitation (Figure 1.1), it is crucial to be able to predict and model cavitation in pump jets and propellers. This is where computational fluid dynamics (CFD) becomes a powerful tool. Compared to the costs of running a large water tunnel, the preparation time, and building the complex scale model, the computational solution provides

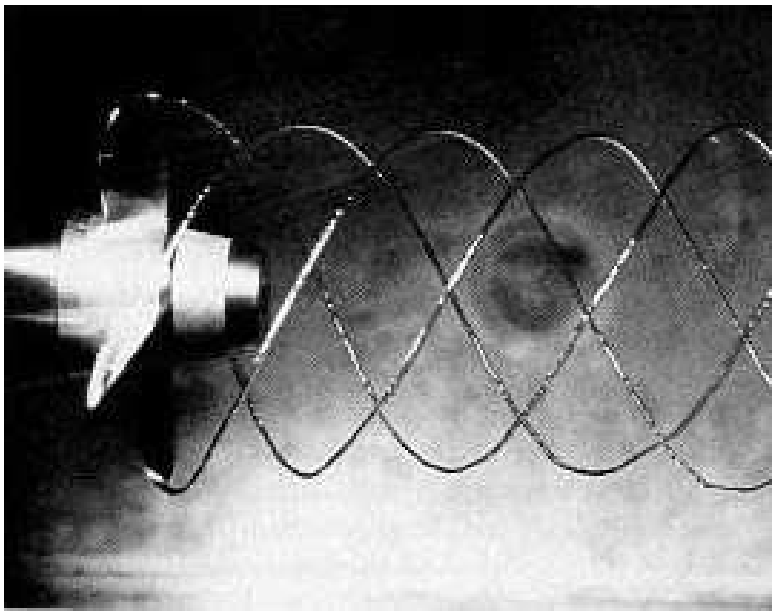

Figure 1.1. Vortex cavitation from the blade tips of a marine propeller.

a cheap and efficient solution. This trend will only continue over the years as more powerful computers emerge at lowering costs. However, the drawback to computational solutions is that they are only approximations of the real flow phenomena. While CFD provides fairly accurate solutions to real life problems, it still must be validated by experimental results. These two fields, experimentation and computation, go hand in hand and will both be used to validate flow phenomena of simple test cases as well as marine propellers. This type of validation is crucial for development in both fields. For example, by comparing the computational results to the experimental results, better codes can be developed to enhance solution accuracy of the simulations. On the other hand, the results generated by the computational results can provide insight into more fully understanding the experiments.

Some important experimental studies have been performed over the years that, to this day, are still used as validation cases. One such example in the field of cavitation is the work performed by Rouse and McNown at the University of Iowa in the 1940's [2]. Cavitation and pressure distributions over various head forms are included in the report, including the hemispherical headform under investigation in this paper. The head form provides a simple geometry to validate both single phase and two-phase CFD solutions. In addition to the head forms, NACA airfoils have been extensively tested and have well documented properties such as lift and drag [3]. Both the NACA0012 and the NACA66(MOD) airfoils will be studied in this paper. The NACA0012 experimental data will provide a comparison of lift and drag at various angles of attack for single phase flow, whereas the NACA66(MOD) airfoil data will be used to asses the accuracy of two phase flow using various cavitation models by comparing the coefficient of pressure over the foil.

Validating these types of solutions with confirmed experimental data allows future users to

use open source code such as OpenFOAM for more complicated and practical applications. For example, propellers in marine applications experience a drop in thrust when cavitation occurs. Accurately modeling the bubbles that are introduced into the flow and subtract from the normal water flow are important for naval designers of the propeller blades. This type of phenomenon has been documented in the past for propellers [4], as well as water jets [5]. In terms of CFD, in 2005, Lindau et al. [6] used a three-dimensional, multiphase Reynolds-averaged, Navier-Stokes (RANS) computational tool UNCLE-M to investigate the cavitation effects on an open propeller, designated P4381. They were successful in modeling the critical cavitation number leading to thrust and torque breakdown that matched experimental data from Boswell [4]. In 2002, Athavale et al. [7] showed it was possible to model full cavitation on a two-stage axial pump, centrifugal water pump, and a rocket pump inducer using reduced Rayleigh-Plesset equations to account for bubble formation. The cavitation model was implemented into a general purpose commercial CFD code, CFD-ACE+. These publications show how different ways of modeling cavitation on rotating machinery can still produce meaningful results.

This present work will demonstrate how an open source CFD tool called OpenFOAM (OF) can be used as a platform for model development and collaboration. Due to the restrictions of most commercial CFD codes, the potential for an open source code that can be developed by the users to suit specific needs is appealing at multiple levels, from industry to academia. The code used for this thesis, OpenFOAM, has grown rapidly over the past few years. Due to the high cost of commercial CFD codes' limited licenses, small business and those in academia have realized the benefit of a free CFD tool that has no license limitations. Additionally, each new version of OF has increased its standard libraries of solvers and utilities to meet the demands of CFD users. However, since the code is open source, free, and relatively new, rigorous testing needs to be performed to ensure that the CFD code is on par with competing commercial codes. Ultimately, the aim of this report is to use an open source code to model cavitation over several benchmarks, including headforms and NACA airfoils, and compare the results to published experimental data. Different cavitation models such as a compressible barotropic model and incompressible mass transfer models will be compared for performance and accuracy.

In order to successfully demonstrate the effectiveness of the open source CFD tool OpenFOAM, several topics will be covered throughout this thesis. First, a literature review will be presented that covers cavitation modeling in the field of CFD along with some selected publications that have featured OpenFOAM as the CFD tool. Then a general overview of the governing equations follows with additional discussion of how the numerics are implemented into OpenFOAM. After that, the solvers used for the computational validation cases are discussed and documented. Finally, the results of several hydrodynamic benchmarks are presented that include a turbulent flat plate, NACA0012 and NACA66(mod) hydrofoils, and a hemispherical headform. The results are compared to experimental results as well as other solvers to assess the accuracy of this open source CFD tool.

Chapter 2

Background and Literature Review

There have been many important and diverse techniques published in the field of cavitation modeling. Both incompressible and compressible models have been used successfully in various forms over the years to simulate complex cavitating flows. The following section serves to introduce several publications that have demonstrated successful results using various cavitation models as well as a few OpenFOAM publications of significance.

2.1 Incompressible Cavitation Models

2.1.1 "Multi-Phase CFD Analysis of Natural and Ventilated Cavitation about Submerged Bodies" by Kunz *et al.* (1999)

Building from earlier viscous CFD methods that relied on solving a single continuity equation that abruptly switched from vapor and liquid densities [8], Kunz *et al.* [9, 10] solved separate continuity equations for the liquid and vapor phases with the objective of modeling submerged bodies subjected to natural and ventilated cavitation. This allows for proper modeling of dynamics and thermodynamics of each of the phases involved with cavitation - the liquid, condensible vapor, and the non-condensible gas fields. A similar approach was taken a year earlier by Merkle *et al.* [11] to model natural sheet cavitation, with the main difference between the two models being that Kunz implemented a three-species formulation to account for two gaseous fields. This, however, does not account for compressible effects in the two-phase mixture region, instead favoring a method consisting of constant densities for each of the liquid and vapor phases. The following is an overview of the equations used in Kunz's model. The governing differential system employed is cast in Cartesian coordinates as:

$$\left(\frac{1}{\rho_m \beta^2}\right)\frac{\partial p}{\partial \tau} + \frac{\partial u_j}{\partial x_j} = (\dot{m^+} + \dot{m^-})\left(\frac{1}{\rho_l} - \frac{1}{\rho_v}\right) \tag{2.1}$$

$$\frac{\partial}{\partial t}(\rho_m u_i) + \frac{\partial}{\partial \tau}(\rho_m u_i) + \frac{\partial}{\partial x_j}(\rho_m u_i u_j) = \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu_{m,t}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + \rho_m g_i \tag{2.2}$$

$$\frac{\partial \alpha_l}{\partial t} + \left(\frac{\alpha_l}{\rho_m \beta^2}\right)\frac{\partial p}{\partial \tau} + \frac{\partial \alpha_l}{\partial \tau} + \frac{\partial}{\partial x_j}(\alpha_l u_j) = (\dot{m^+} + \dot{m^-})\left(\frac{1}{\rho_l}\right) \tag{2.3}$$

$$\frac{\partial \alpha_{ng}}{\partial t} + \left(\frac{\alpha_{an}}{\rho_m \beta^2}\right)\frac{\partial p}{\partial \tau} + \frac{\partial \alpha_{an}}{\partial \tau} + \frac{\partial}{\partial x_j}(\alpha_{an} u_j) = 0 \tag{2.4}$$

where α_l and α_{ng} represent the liquid phase and non-condensible gas (NCG) volume fractions, and mixture density and mixture turbulent viscosity are defined as

$$\rho_m \equiv \rho_l \alpha_l + \rho_v \alpha_v + \rho_{ng} \alpha_{ng} \tag{2.5}$$

$$\mu_{m,t} = \frac{\rho_m C_\mu k^2}{\epsilon} \tag{2.6}$$

Note that each of the equations contains two sets of time-derivatives – those written in terms of the variable t corresponding to physical time terms, and those written in terms of τ corresponding to pseudo-time terms that are employed in the time-iterative solution procedure. Corresponding artificial time-derivative terms are also introduced in the component phasic continuity equations, which ensures that the proper differential equation is satisfied. After several steps, the following proper non-conservative form of the liquid volume fraction equation is reached.

$$\frac{\partial \alpha_l}{\partial t} + \frac{\partial \alpha_l}{\partial \tau} + u_j \frac{\partial \alpha_l}{\partial x_j} \equiv (\dot{m}^+ + \dot{m}^-)\left(\frac{\rho_m}{\rho_l \rho_v}\right) \tag{2.7}$$

In terms of physical modeling, the terms $\dot{m}^-$ and $\dot{m}^+$ are used respectively to represent the mass transfer rates from liquid to vapor and vice versa.

$$\dot{m}^- = \frac{C_{dest} \rho_v \alpha_l \mathrm{MIN}[0, p - p_v]}{1/2 \rho_l U_\infty^2 t_\infty} \tag{2.8}$$

$$\dot{m}^+ = \frac{C_{prod} \rho_v {\alpha_l}^2 (1 - \alpha_l)}{t_\infty} \tag{2.9}$$

where C_{prod} and C_{dest} are empirical constants set to varying values, for example C_{dest} and $C_{prod} = 0.2$ in earlier simulations[9] and are both set to 100 in a later publication[10]. For turbulence closure, the $k - \epsilon$ turbulence model with wall functions was utilized. Three sets of results were presented that include natural cavitation on axisymmetric bodies, ventilated cavities, and three-dimensional natural cavitation. The pressure results for the hemispherical headform were in good agreement with experimental data for both the 2D and 3D cases. The cone and blunt fore-body headforms were less successfully, due in part to the single phase turbulence modeling as well as the inability to capture more complex off-body cavitation.

2.1.2 "Mathematical Basis and Validation of the Full Cavitation Model" by Singhal *et al.* (2002)

In 2002, Singhal *et al.* [7, 12] described results from the "Full Cavitation" model. Used in commercial codes, this model boasts satisfying two downfalls of previous cavitation models, including the lack of robustness of numerical algorithms and a lack of generality of the correlations or approach used. The full cavitation model uses the standard Navier-Stokes equations for variable fluid density and the k-ϵ turbulence model. The density of the fluid is a function of the vapor mass fraction, f, which is solved from a transport equation combined with conservative forms of the mass and momentum equations. The $\rho - f$ relationship is:

$$\frac{1}{\rho} = \frac{f}{\rho_v} + \frac{1-f}{\rho_l} \tag{2.10}$$

and the vapor volume fraction α is formulated from f as:

$$\alpha \equiv f\frac{\rho}{\rho_v} \tag{2.11}$$

The transport equation governs the vapor mass fraction and is represented as:

$$\frac{\partial}{\partial t}(\rho f) + \nabla \cdot (\rho \vec{V} f) = \nabla \cdot (\Gamma \nabla f) + R_e - R_c \tag{2.12}$$

The source terms R_e and R_c are the vapor generation and condensation rates. They are functions of fluid properties such as liquid and vapor phase densities, saturation pressure, and liquid vapor surface tension, as well as flow parameters such as pressure and velocity. This model uses a homogeneous simplification approach to cavitation to speed up computations, rather than the two-fluid approach that requires significantly more computational effort. In order to account for bubble dynamics, this model utilizes the Rayleigh-Plesset equation to arrive at a simplified equation for vapor transport, the details of which will be neglected for this discussion. The turbulence effects are represented by using a probability density function (PDF) approach. The final form of the full cavitation model that accounts for a mixture of liquid, liquid vapor, and non-condensible gas is represented as:

$$\frac{1}{\rho} = \frac{f_v}{\rho_v} + \frac{f_g}{\rho_g} + \frac{1-f_v-f_g}{\rho_l} \tag{2.13}$$

where the non-condensible gas density ρ_g is calculated as:

$$\alpha_g = f_g\frac{\rho}{\rho_g} \tag{2.14}$$

$$\alpha_l = 1 - \alpha_v - \alpha_g \tag{2.15}$$

The following expressions are representations for vapor generation/condensation rates in terms

of the vapor mass fractions:

$$R_e = C_e \frac{\sqrt{k}}{\sigma} \rho_l \rho_v \left[\frac{2}{3} \frac{P_v - P}{\rho_l} \right]^{1/2} (1 - f_v - f_g) \tag{2.16}$$

$$R_c = C_c \frac{\sqrt{k}}{\sigma} \rho_l \rho_v \left[\frac{2}{3} \frac{P - P_v}{\rho} \right]^{1/2} f_v \tag{2.17}$$

where C_e and C_c are 0.02 and 0.01, respectively. Validation of this model was performed on a NACA66(MOD) hydrofoil, a submerged cylindrical body, and flow in a sharp-edged orifice. Although the limitations of the model include the isothermal flow assumption and a fixed, uniform mass concentration of NCG, good agreement has been reached in all cases without adjusting any coefficients.

2.1.3 "A Pressure-Based Method for Turbulent Cavitating Flow Computations" by Senocak and Shyy (2002)

Senocak and Shyy [13] account for changes in density by using a pressure-based algorithm, a volume fraction transport equation, and a pressure-velocity-density coupling scheme to solve turbulent cavitating flow computations. While no Mach number or temperature effects are factored into the equations, the pressure correction equation shares common features with that of high-speed flows - a convective-diffusive type instead of a strictly diffusive type. The governing equations used are:

$$\frac{\partial \rho_m}{\partial t} \nabla \cdot (\rho_m \vec{u}) = 0, \tag{2.18}$$

$$\frac{\partial}{\partial t}(\rho_m \vec{u}) + \nabla \cdot (\rho_m \vec{u} \vec{u}) = -\nabla P + \nabla \cdot [(\mu + \mu_t) \nabla \vec{u}] + 1/3 \nabla [(\mu + \mu_t) \nabla \cdot \vec{u}] \tag{2.19}$$

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \vec{u}) = (\dot{m}^- + \dot{m}^+), \tag{2.20}$$

with the mixture density and the turbulent viscosity defined as:

$$\rho_m = \rho_l \alpha l + \rho_v (1 - \alpha_l) \qquad \mu_t = \frac{\rho_m C_\mu k^2}{\epsilon} \tag{2.21}$$

The physical phenomenon of cavitation is modeled through $\dot{m}^+$ and $\dot{m}^-$ terms in equation (2.20), which represent the condensation and evaporation of the phases resulting in a variable density field. In this paper, $\dot{m}^+$ and $\dot{m}^-$ are identical to equations (2.24) and (2.25) with the minor variable changes $\gamma = \alpha$, $C^+ = C_{prod}$, and $C^- = C_{dest}$. The time scale is taken as the ratio of the characteristic length scale to the reference velocity scale (l/U).

In terms of turbulence modeling, this paper utilized the original k-ϵ model with wall functions. In order to compensate for recirculation zones, rotation, and large streamline curvatures, the non equilibrium k-ϵ constants were also tested in this study. Table (2.1) shows the variation in the two types of models.

MODEL	C_μ	$C_{\epsilon 1}$	$C_{\epsilon 2}$	σ_k	σ_ϵ
Original k-ϵ	0.09	1.44	1.92	1.0	1.3
Nonequilibrium k-ϵ	0.09	$1.15 + 0.25(P/\epsilon)$	$1.45 + 0.45(P/\epsilon)$	0.8927	1.15

Table 2.1. Empirical Constants used in k-ϵ Model

In order to deal with highly compressible flows, Senocak and Shyy revised the pressure-based SIMPLE algorithm. Through some derivation, the pressure-correction equation is as follows:

$$-\nabla_d \cdot (\rho^* \mathbf{D} \nabla_d P^{'}) + \nabla_d \cdot (C_\rho \vec{u}^* P^{'}) = -\nabla_d \cdot (\rho^* \vec{u}^*) + \nabla_d \cdot (C_\rho P^{'} \mathbf{D} \nabla_d P^{'}) \tag{2.22}$$

For the cavitation model, the following relation between density correction and pressure correction is introduced to establish the pressure-density coupling:

$$\rho^{'} = C(1 - \alpha_l) P^{'}, \tag{2.23}$$

where C is an arbitrary constant.

In order to validate their model, Senocak and Shyy compared the computational cavitation results over both hemispherical and blunt headforms. With varying density ratios, they were able to achieve accurate results that matched with experimental findings. They also successfully compared the reattachment points to experimental data for further accuracy of their cavitation solver. One downfall of the model was that the steady-state solution procedure did not account for the inherent unsteadiness of cavitating flow.

2.1.4 "Modeling of Cavitating Flow Around a Stationary/Moving Wing Profile" by Wikström (2005)

In 2005, Wikström [14] describes an LES approach to solving cavitating flows for incompressible fluids utilizing a volume fraction interface capturing technique. Using OpenFOAM, Wikström incorporated source terms to model cavitating flow over a NACA0012 airfoil at varying angles of attack. The mass transfer term, $\dot{m}$, was based off of the work of Kunz [10]. The cavitation results from Wikström show that using a Large Eddy Simulation (LES) over a 2D NACA0012 airfoil at varying angles of attack agrees qualitatively with experimental data. Although lacking in quantitative experimental comparison, this paper shows that development of a cavitation model is feasible in OF and gives a basis of comparison for the RANS model developed in this thesis. The follow equations describe the model used by Wikström where $\dot{m} = \dot{m}^+ + \dot{m}^-$ and:

$$\dot{m}^- = \frac{C^- \rho_v \gamma_l \mathrm{MIN}[0, \bar{p} - p_v]}{1/2 \rho_l U_\infty^2 t_\infty} \tag{2.24}$$

$$\dot{m}^+ = \frac{C^+ \rho_v \gamma_l^2 (1 - \gamma_l)}{t_\infty} \tag{2.25}$$

C^+ and C^- are constant parameters that in this paper are set to 1000, t_∞ (1 s for the full scale model) and U_∞ (10 m/s relate to the largest velocity and time scales of the flow, and p_v is the vapor pressure. The constant parameters C^+ and C^- are set to as high as possible in order to simulate almost instant vaporisation as pressure drops below p_v. The phase change source term, S_γ, is defined as:

$$S_\gamma = \dot{m}/\rho_l \tag{2.26}$$

The other source term, the volumetric source, S_ρ, is defined as:

$$S_\rho \equiv \dot{m}(\frac{1}{\rho_l} - \frac{1}{\rho_v}) \tag{2.27}$$

and primarily turns up in the pressure equation. The right hand side includes the specific mass transfer, $\dot{m}$, as well as the densities that handle the bulk volume change when mass is transfered from one phase to the other. Although accurate comparison was made to experimental cavitation data from a NACA0012 airfoil, the results were mostly quantitative. Both the 2D and 3D computational results were realistic, and even though more validation data would have been beneficial, this paper successfully demonstrated an open-source implementation of a cavitation model in OF.

2.2 Compressible Cavitation Models

A variety of work has been done with compressibility effects in cavitation modeling including Venkateswaran [15], Schmidt [16], Delale *et. al* [17] and Brajdic-Mitidier [18]. The basis of these compressible models can be traced back to the work of Kubota [19], who in 1992 originally proposed a new model of cavitation that explained the interaction between viscous effects including vortices and cavitation bubbles.

2.2.1 "A new modeling of cavitating flows: a numerical study of unsteady cavitation on a hydrofoil section" by Kubota *et al.* (1992)

The equations of this bubble two-phase (BTF) model can be seen below. The model he developed treats the inside and outside of a cavity as one continuum by regarding the cavity as a compressible viscous fluid whose density changes greatly. The Navier-Stokes equations including cavitation bubble clusters are solved in finite-difference form by a time-marching scheme, where the growth and collapse of a bubble cluster is given by a modified Rayleigh equation.

2.2.1.1 Macroscopic Modeling

The governing equations of the macroscopic flow field are as shown below. The equation of continuity is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{2.28}$$

where t, $\rho\mathbf{v}$, $\mathbf{v}$, and ρ are time, mass flux vector, velocity vector, and density of the mixture, respectively.

The conservation equation for momentum: Navier-Stokes equation is

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot(\rho\mathbf{v}\mathbf{v}) = -\nabla P + \frac{1}{Re}\mu\left\{\nabla^2\mathbf{v} + \frac{1}{3}\nabla(\nabla\cdot\mathbf{v})\right\}, \tag{2.29}$$

where P is the pressure in the mixture, μ is the viscosity of the mixture, and Re is the Reynolds number. The liquid is treated as incompressible. Only the change of void fraction causes the compressibility of the mixture. It is assumed that there is no slip between the two phases. In order to account for variable density, the density of the water containing bubbles is defined as follows:

$$\rho = (1 - f_g)\rho_L, \tag{2.30}$$

where ρ_L is the water density and f_g is the local void fraction. The mass and momentum of the vapor are ignored, since they are very small compared with those of the liquid. The change of liquid mass owing to the phase change is also ignored. The phase change only affects the change of void fraction (density) in the mixture.

The viscosity of the mixture is assumed to be,

$$\mu = (1 - f_g)\mu_L + f_g\mu_G, \tag{2.31}$$

where μ_L is the liquid viscosity and μ_G is the vapor viscosity.

2.2.1.2 Microscopic Modeling - Local Homogeneous Model

This following details how cavitation is treated microscopically as bubble clusters using Rayleigh's equation to track the size of bubbles. The final form of the Local Homogeneous Model is as follows:

$$(1 + 2\pi\Delta r^2 nR)R\frac{D^2R}{Dt^2} + (\frac{3}{2} + 4\pi\Delta r^2 nR)(\frac{DR}{Dt})^2 + 2\pi\Delta r^2\frac{Dn}{Dt}R^2\frac{DR}{Dt} = \frac{P_v - P}{\rho_L}. \tag{2.32}$$

where Δr is the grid interval and R is the typical bubble radius.

2.2.1.3 Quasi-Poisson Equation for Pressure

By taking the divergence of the Navier-Stokes equation (2.29) and adding in the bubble dynamics described in equation (2.32), the quasi-Poisson equation for pressure including the motion of the cluster bubbles is as follows:

$$\nabla^2 P + P(P) = R'\left(\rho\mathbf{v}, \mathbf{v}, \frac{\partial R}{\partial t}, R\right) + D(\rho\mathbf{v}, \mathbf{v}), \tag{2.33}$$

where

$$P'(P) = -\rho_L 4n\pi R^2 P(P), \tag{2.34}$$

$$R'\left(\rho\mathbf{v},\mathbf{v},\frac{\partial R}{\partial t},R\right)=-\rho_L 4n\pi R\left[RR\left(\rho\mathbf{v},\mathbf{v},\frac{\partial R}{\partial t},R\right)+2\left(\frac{\partial R}{\partial t}\right)^2\right] \tag{2.35}$$

Equations (2.29) and (2.32) are time-integrated with the Euler explicit scheme using the value of pressure P obtained by solving (2.33). Kubota goes on to detail the computational results on a NACA0015 hydrofoil at angles of attack of 8 ° and 20 °. When the angle of attack is 8 °, the BTF cavity model expresses the could-type feature of cavitation shed from the trailing edge of the attached cavities. The 20 ° angle of attack produced, as expected, fully separated flow from the leading edge of the hydrofoil and vortex cavitation occurred in the separated region. Although these early results were promising and laid the foundation for future compressibility models, more development of the microscopic cavity model was needed which would have improved the quantitative agreement with the experimental results.

2.2.2 "A Fully Compressible Two-Dimensional Model of Small, High-Speed, Cavitating Nozzles" by Schmidt *et al.* (1999)

Schmidt *et al.* [16] dealt with the construction of a numerical model that treats liquid and vapor as a continuum for predicting high speed nozzle flow. The compressibility of both phases was included to account for high pressure values, and a third-order shock-capturing technique was applied to the continuity equation to capture sudden jumps in density. The solver was able to handle liquid-to-vapor density ratios of up to 10,000:1. Schmidt's model differs from Kubata by including the vapor mass and abandoning the constitutive relation that treated the flow as a uniform mixture of liquid and tiny spherical bubbles, which would not be appropriate for small, high-speed nozzles. Although this model accounts for compressibility of both pure phases, it lacks a turbulence model.

2.2.2.1 The Simplified Equation of State

For an isentropic model accounting for phase change ($dp = a^2 d\rho$), the speed of sound, a, is defined as:

$$\sqrt{\frac{1}{\alpha\cdot\rho_g+(1-\alpha)\rho_l\cdot\left(\frac{\alpha}{\rho_g\cdot a_g^2}+\frac{(1-\alpha)}{\rho_l\cdot a_l^2}\right)}} \tag{2.36}$$

The values of a_g and a_l are the sound speeds of the pure phases. In this expression, α is the void fraction and is given by

$$\alpha=\frac{\rho-\rho_l}{\rho_g-\rho_l} \tag{2.37}$$

By assuming constant sound speeds for the saturated vapor and liquid, the pressure may be analytically integrated as a function of void fraction. Integrating from the saturated liquid state gives:

$$p=p_l^{sat}+p_{gl}\cdot\log\left(\frac{\rho_g\cdot a_g^2\cdot[\rho_l+\alpha\cdot(\rho_g-\rho_l)]}{\rho_l\cdot[\rho_g\cdot a_g^2-\alpha\cdot(\rho_g\cdot a_g^2-\rho_l\cdot a_l^2)]}\right) \tag{2.38}$$

where p_{gl} is not a function of void fraction, but rather a parameter of the fluid properties:

$$p_{gl} = \frac{\rho_g \cdot a_g^2 \cdot \rho_l \cdot a_l^2 \cdot (p_g - \rho_l)}{\rho_g^2 \cdot a_g^2 - \rho_l^2 \cdot a_l^2} \tag{2.39}$$

Therefore, pressure is given by an analytic function of density that is linear for pure liquid or vapor and is given by equation (2.38) in the two-phase region, where $\rho_g < \rho < \rho_l$. Pressure according to this relationship has been shown as a function of density. No partial differential equation is required for pressure. The pressure can be found analytically from the cell density, which dramatically reduces the computational cost of a time step.

2.2.2.2 Numerical Solution

This model considers the compressibility of both pure phases. In order to do that, the model is written as three partial differential equations and one algebraic equation of state:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{V} + \omega H_\rho = 0 \tag{2.40}$$

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot \rho u \mathbf{V} + \omega H_x = -\frac{\partial p}{\partial x} + \nabla \mu \nabla u + \omega H_{xv} \tag{2.41}$$

$$\frac{\partial \rho v}{\partial t} + \nabla \cdot \rho v \mathbf{V} + \omega H_y = -\frac{\partial p}{\partial y} + \nabla \mu \nabla v + \omega H_{yv} \tag{2.42}$$

$$p = p(\rho) \tag{2.43}$$

Equation 2.43 refers to Eq. 2.38 or to the integral of the isentropic model of the phase change, depending on the cell density. In Eqs. 2.40, 2.41, and 2.42, ω is zero for calculations in polar coordinates. The model was used for several different test cases: a collapse of a single bubble, simple nozzle flow, and an axisymmetric fuel injector tip. Each of the results compared favorabley to experimental data, although the authors noted that an improvement of numerical treatment of the interface region is needed to prevent smearing of the collapse region.

2.2.3 "Physical and Numerical Modeling of Unsteady Cavitation Dynamics" by Schnerr and Sauer (2001)

Schnerr and Sauer [17] detailed the physics of their cavitation model that relies on the creation and destruction of vapor bubbles. This process is modeled using the surrounding conditions (pressure and temperature) and the slip between the vapor bubbles and liquid is neglected. Using the VOF method along with the homogeneous mixture assumption, the authors were able to validate the solver by comparing computational results of flow over a NACA0015 hydrofoil to experimental data. The void fraction and velocity filed are solved from:

$$\frac{\partial \alpha}{\partial t} + \frac{\partial (\alpha u)}{\partial x} + \frac{\partial (\alpha v)}{\partial y} = \left(\frac{n_0}{1 + n_0 \cdot \frac{4}{3}\pi R^3} \right) \frac{d}{dt} \left(\frac{4}{3} \pi R^3 \right) \tag{2.44}$$

$$\nabla \cdot \mathbf{u} = -\frac{\rho_v - \rho_l}{\alpha\rho_v + (1-\alpha)\rho_l}\frac{d\alpha}{dt} \tag{2.45}$$

where α has been derived from a modified VOF method that accounts for phase transition as:

$$\alpha = \frac{V_v}{V_{cell}} = \frac{N_{bubbles} \cdot \frac{4}{3}\pi R^3}{V_v + V_l} = \frac{n_0 V_l \cdot \frac{4}{3}\pi R^3}{n_0 V_l \cdot \frac{4}{3}\pi R^3 + V_l} = \frac{n_0 \cdot \frac{4}{3}\pi R^3}{1 + n_0 \cdot \frac{4}{3}\pi R^3} \tag{2.46}$$

Under the assumptions that bubble-bubble interactions as well as bubble coalescence can be neglected, and that the bubbles remain spherical, the energy equation combined with the Rayleigh-Plesset equation becomes:

$$R\frac{d^2R}{dt^2} + \frac{3}{2}\left(\frac{dR}{dt}\right)^2 = \frac{p(R) - p_\infty}{\rho_l} - \frac{2\sigma}{\rho_l R} - 4\frac{\mu}{\rho_l R}\frac{dR}{dt} \tag{2.47}$$

To control the bubble growth, the following Rayleigh relation is used:

$$\dot{R} = \sqrt{\frac{2}{3}\frac{p(R) - p_\infty}{\rho_l}} \tag{2.48}$$

where $p(R)$ is the pressure in the liquid at the bubble boundary and p_∞ is the free stream pressure away from the bubbles. A simple nozzle case was used to identify parameters such as grid resolution, CFL number, scaling effects, and variations in σ. In terms of results, both the standard NACA0015 channel grid and the free surface grid yielded lift and drag results in agreement with experimental data.

2.2.4 "Computation of Multiphase Mixture Flows with Compressibility Effects" by Venkateswaran *et al.* (2002)

Building from the same multiphase code as Kunz [9], Venkateswaran *et al.* [15] was able to develop and incorporate compressibility effects into the model. This revised model accounts for finite acoustic speeds in the constituent phases, which can lead to supersonic flow and the development of shock waves in the mixture region. Perturbation theory was used to develop the preconditioning or artificial compressibility methods in order to ensure that both the compressible and incompressible phases retained accuracy and efficiency. Mathematical descriptions of the implementation of compressibility are detailed in both mass and volume fraction form, with the final conclusion being that both methods arrive at similar solutions. The time-marching multi-phase system formulation for compressibility effects is:

$$\Gamma_p^\alpha \frac{\partial Q_\alpha}{\partial \tau} + \frac{\partial Q}{\partial t} + \frac{\partial E_j}{\partial x_j} = H \tag{2.49}$$

where τ is the pseudo-time and t is the physical time. The physical time-derivative, flux vectors, and source terms are represented by:

$$Q = \begin{bmatrix} \bar{\rho_v}\alpha_v \\ \bar{\rho_l}\alpha_l \\ \rho_v u_i \end{bmatrix}, \quad E_j = \begin{bmatrix} \bar{\rho_v}\alpha_v u_j \\ \bar{\rho_l}\alpha_l u_j \\ \rho_v u_i u_j + p\delta_{ij} - \mu_{m,t}\sigma_{ij} \end{bmatrix}, \quad H = \begin{bmatrix} -\left(\dot{m}^+ + \dot{m}^-\right) \\ \left(\dot{m}^+ + \dot{m}^-\right) \\ (0)_i \end{bmatrix} \tag{2.50}$$

The results of the time-marching, compressible, multiphase, homogeneous mixture flows are presented for an array of cases including a one-dimensional shock tube problem, natural cavitation on submerged bodies, and supersonic flow on an underwater projectile as well as an underwater rocket plume. The shock tube problem compared favorably to the analytical solution, and the supersonic test cases showed promise that the algorithm could handle such flow phenomena.

When Venkateswaran investigated the differences between the compressible and incompressible forms of the code on two hemispherical headforms, he concluded that the compressible model captures the dynamics of the cavity more correctly than the incompressible model. This observation was based on pressure graphs, time averaged pressure and volume fraction fields, as well drag histories for both the incompressible and compressible cases.

2.3 OpenFOAM Publications

Over the last decade, OpenFOAM has provided graduate students with the ability to develop their own sophisticated models to solve real world problems. In 1996, Hrvoje Jasak, one of the original creators of OpenFOAM, used it to present his PhD thesis [20] which describes in detail the inner workings of the Finite Volume method associated with OpenFOAM. Since then, many students from the Imperial College in London have published PhD thesis using OpenFOAM. In terms of application of OpenFOAM to solve multiphase flow, one of the earlier papers published was by Ubbink [21] in 1997 which detailed the process for implementing a phase change VOF model that handled sharp interfaces on a wide range of interactions including merged or ruptured interfaces. Additionally, this paper demonstrated the effect of using different finite volume schemes for the convective scalar transport equation and seeing how that affects the amount of numerical dissipation. A year later, Hill [22] published his PhD thesis on the prediction of dispersed multiphase flows that included solid-liquid, liquid-liquid, and gas-liquid.

In 2001, Brennan's [23] PhD thesis documented results of numerical simulations of two-phase flows in settling tanks. Using OpenFOAM, Brennan was successful in creating a two-phase solver that worked in conjunction with a modified pressure interpolation scheme to properly model the interaction and flow found in settling tanks. A screen shot of the settling tank used in Brennan's work can be seen in Figure 2.1.

The following year, Rusche [24] published his PhD thesis entitled "Computaitonal Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions." This work continued to develop the OpenFOAM code that had been used by the previous graduates of Imperial College

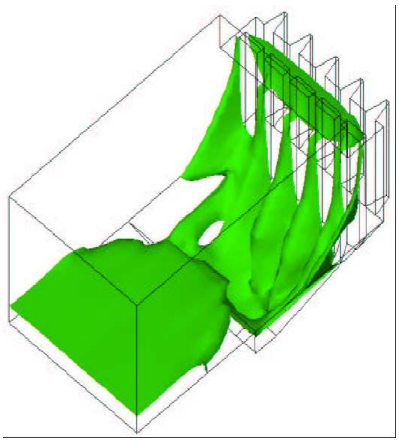

Figure 2.1. A full scale 3-D iso-surface view of a settling tank simulation

and provided accurate solutions to such problems as a rising air bubble, phase separation in a tank, and sudden enlargement in a circular pipe with direct numerical simulations. Figure 2.2 demonstrates the CFD results of a rising bubble.

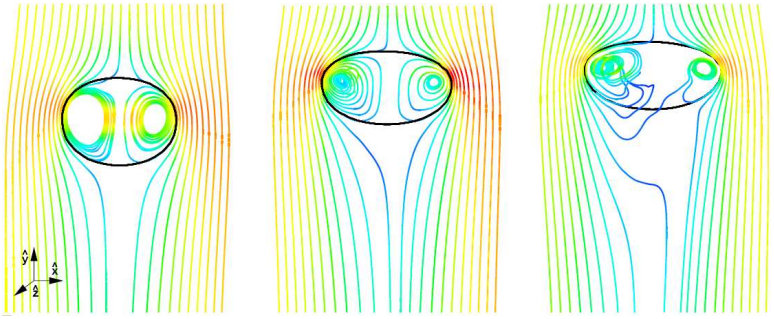

Figure 2.2. Increasing time steps of stream lines colored by velocity magnitude around a rising bubble

Karac [25] later published his PhD thesis on droplet impacts on water filled polyethylene containers. After performing drop tests to obtain material properties, OpenFOAM was used by model the containers pressure and strain histories. Both results agreed well with the experiments.

In terms of hydrodynamic CFD work, Nilisson [26] has showed that turbine analysis using OF is both possible and accurate when compared to experimental results as well as other CFD software tools such as CFX. Four main cases were investigated for a draft tube and a runner, both steady and unsteady. Qualitative results of pressure and velocity distribution matched both experimental and the Fluent CFX computational results. This paper demonstrates OF's ability to accurately predict turbo-machinery results (Figure 2.3).

Early in 2008, Huuva published his PhD thesis from Chalmers that detailed cavitation modeling using OF [27]. In his paper, Huuva developed a cavitation model using OpenFOAM and analyzed several cases including flow over a circular cylinder, an axisymmetric hill, a generic submarine hull, a NACA0015 airfoil, and a Twist11 hydrofoil. His code was developed from Kunz's mass transfer model described earlier. In addition to this model, Huuva also developed a

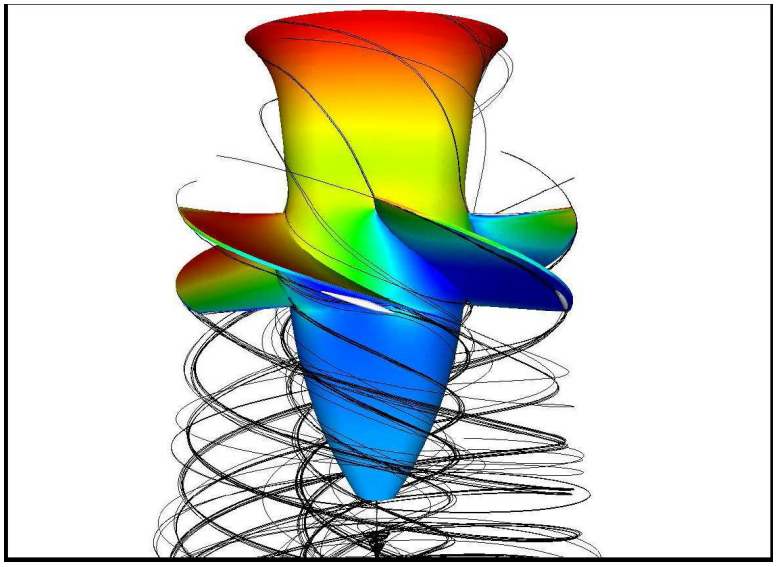

Figure 2.3. The overall flow features from a Hölleforsen runner

mass transfer model for cavitating liquids based on the Fluent cavitation code, which he refers to as Singhal's mass transfer model. This type of model is based on a bubble radius, in a similar fashion to that of Kubota [19]. This work was the basis for later publications involving 3-D hydrofoils and marine propellers operating in an artificial wake [28]. Figures 2.4 and 2.5 show results from Huuva's work.

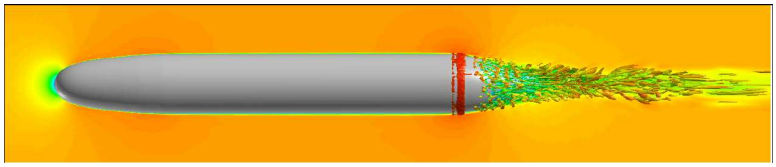

Figure 2.4. Non-cavitating flow over a generic submarine hull

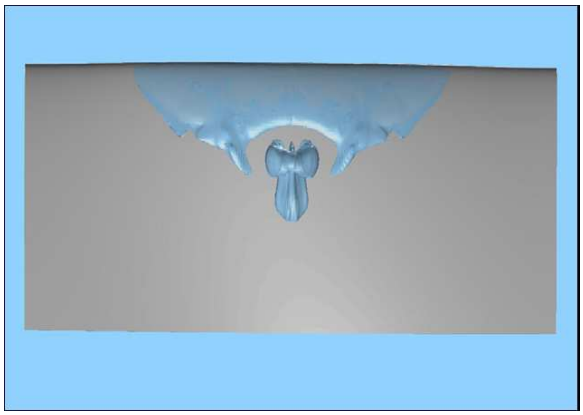

Figure 2.5. Cavitating flow over a NACA0015 airfoil using LES

Chapter 3

Governing Equations & Numerical Methods

3.1 Single Phase Compressible Form

Using vector notation, the conservation of mass, momentum, and energy equations can be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{3.1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = \rho \mathbf{g} + \nabla \cdot \sigma \tag{3.2}$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) = \rho \mathbf{g}\mathbf{u} + \nabla \cdot (\sigma \mathbf{u}) - \nabla \cdot \mathbf{q} + \rho Q \tag{3.3}$$

where ρ is the fluid density, $\mathbf{u}$ is the velocity vector, σ is the stress tensor, e is the total specific energy, Q is the volume energy source, $\mathbf{q}$ is the heat flux, and t is time.

In order to close the system (there are more unknowns than equations), constitutive relations are required. For Newtonian fluids, the following set of equations is used. The equation of state relates density to pressure and temperature

$$\rho = \rho (p, T) \tag{3.4}$$

The stress tensor σ is related to the strain-rate of the velocity field

$$\sigma = -p\mathbf{I} + \mu \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right] + \lambda (\nabla \cdot \mathbf{u}) \mathbf{I} \tag{3.5}$$

Fourier's law of heat conduction relates the heat flux to the gradient of the temperature field

$$\mathbf{q} = k \nabla T \tag{3.6}$$

The total specific energy is the sum of mechanical energy e_M (or kinetic energy) and internal energy u, and can be expressed as

$$e = e_M + u(p,T) = \frac{1}{2}\mathbf{u}\mathbf{u} + u(p,T) \tag{3.7}$$

Inserting Eq. (3.5) into Eq. (3.2), and assuming that $\lambda = \frac{2}{3}\mu$ gives the compressible Navier-Stokes equations.

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot(\rho\mathbf{u}\mathbf{u}) = \rho\mathbf{g} - \nabla\left(p + \frac{2}{3}\mu\nabla\cdot\mathbf{u}\right) + \nabla\cdot\left[\mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right] \tag{3.8}$$

Inserting Eqs. (3.5) and (3.6) into Eq. (3.3) gives the compressible energy equation

$$\begin{aligned}\frac{\partial(\rho e)}{\partial t} + \nabla\cdot(\rho e\mathbf{u}) &= \rho\mathbf{g}\mathbf{u} - \nabla\cdot(p\mathbf{u}) - \nabla\cdot\left(\frac{2}{3}\mu\left(\nabla\cdot\mathbf{u}\right)\mathbf{u}\right) \\ &\quad + \nabla\cdot\left[\mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right] + \nabla\cdot(k\nabla T) + \rho Q\end{aligned} \tag{3.9}$$

Finally, the transport coefficients of viscosity and conduction are functions of the thermodynamic state variables, i.e., $\mu = \mu(p,T)$ and $k = k(p,T)$.

3.2 Single Phase Incompressible Form

For problems where the density is constant, the governing equations can be simplified

$$\nabla\cdot\mathbf{u} = 0 \tag{3.10}$$

$$\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\mathbf{u}) = -\frac{1}{\rho}\nabla p + \mathbf{g} + \nabla\cdot(\nu\nabla\mathbf{u}) \tag{3.11}$$

$$\frac{\partial T}{\partial t} + (\mathbf{u}\cdot\nabla)\,T = \Phi + \frac{1}{\rho C_p}\nabla\cdot(k\nabla T) + Q \tag{3.12}$$

where Φ is the rate of dissipation of mechanical energy per unit mass.

3.3 Turbulence Modeling

Turbulence in OpenFOAM can be modeling using either the Reynolds-averaged Navier Stokes (RANS) equations, or large-eddy simulation (LES) schemes with sub-grid-scale models. Primary focus in this thesis is the Reynolds averaging approach where it is assumed that the flow variables can be decomposed into mean and fluctuating components, e.g., for the velocity field

$$\mathbf{u}(\mathbf{x},t) = \mathbf{U}(\mathbf{x},t) + \mathbf{u}'(\mathbf{x},t) \tag{3.13}$$

where $\mathbf{u}(\mathbf{x},t)$, $\mathbf{U}(\mathbf{x},t)$ and $\mathbf{u}'(\mathbf{x},t)$ are the instantaneous velocity, time-averaged velocity, and turbulent fluctuations of velocity, respectively.

Substituting Eq. (3.13) into Eqs. (3.10) and (3.11) and taking the time-average of the resultant equations gives the unsteady RANS equations

$$\nabla \cdot \mathbf{U} = 0 \tag{3.14}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) = -\frac{1}{\rho}\nabla p + \mathbf{g} + \nabla \cdot (\nu \nabla \mathbf{U}) - \nabla \cdot \overline{\mathbf{u}'\mathbf{u}'} \tag{3.15}$$

For the standard linear closure, the Reynolds-stress tensor is modeled as a linear function of the mean strain-rate tensor

$$-\overline{\mathbf{u}'\mathbf{u}'} = \nu_t \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^T \right) + \frac{2}{3} k \mathbf{I} \tag{3.16}$$

where $k = \frac{1}{2}\overline{\mathbf{u}' \cdot \mathbf{u}'}$ is the turbulent kinetic energy and ν_t is the turbulent eddy viscosity, which in OpenFOAM can be computed using one of the many pre-coded models (e.g., Spalart Allmaras, $k - \omega$ SST, and various other versions of $k - \epsilon$ with either wall functions or near-wall models).

3.4 Finite Volume Discretization

The governing equations for conservation of mass, conservation of momentum, and transport of scalars can be represented by the generic transport equation for ϕ

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \mathbf{U} \phi) - \nabla \cdot (\rho \Gamma_\phi \nabla \phi) = S_\phi (\phi) \tag{3.17}$$

which is comprised of 4 basic terms: temporal acceleration; convective acceleration; diffusion; and a source term. The finite volume method requires that each term in Eq. (3.17) be satisfied over the control volume V_p around the point P in the integral form

$$\int_t^{t+\Delta t} \left[\frac{\partial}{\partial t} \int_{V_p} \rho \phi \, dV + \int_{V_p} \nabla \cdot (\rho \mathbf{U} \phi) \, dV - \int_{V_p} \nabla \cdot (\rho \Gamma_\phi \nabla \phi) \, dV \right] dt = \int_t^{t+\Delta t} \left[\int_{V_p} S_\phi (\phi) \, dV \right] dt \tag{3.18}$$

The discretization of each term will be discussed in the following sections.

3.4.1 Spatial Discretization

Discretization of the solution domain can fundamentally be broken down to that of time and space. The time discretization is performed by simply breaking it down into a set of time steps δt that may or may not change during the numerical simulation. The space domain is more detailed as it requires the subdivision of the domain into a number of cells, referred to as control volumes. An OpenFOAM example of general polyhedral cells can be seen in Figure 3.1. These cells must not overlap one another and must completely fill the domain. If these two requirements are met, the cells are referred to as contiguous. In OpenFOAM, there is no limitation on the

number of faces bounding each cell, nor is there a restriction on the orientation of each face. These types of meshes are in direct contrast with meshes that contain cells with a prescribed alignment, and hence is described as an "arbitrarily unstructured" mesh. This type of readability allows users to solve more intricate meshes as well as handle more complex domain geometries.

The first step in spatial discretization is to transform the volume integrals in Eq. (3.18) to surface integrals by using Gauss's Theorem (also known as the Divergence theorem). In it's most general form, Gauss's Theorem can be written as

$$\int_V \nabla \star \phi \, dV = \int_S d\mathbf{S} \star \phi \tag{3.19}$$

where $\mathbf{S}$ is the surface area vector, ϕ represents any tensor field, and the star operator $\star$ is used to represent any tensor product, i.e., inner, outer, cross and the respective derivatives (div, grad, curl). Volume and surface integrals are then linearized using appropriate schemes which are described for each term.

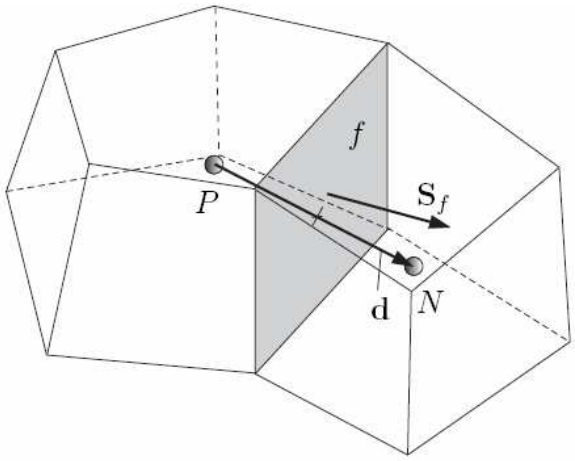

Figure 3.1. Two typical neighboring cells, P and N, and the parameters in finite volume discretization.

3.4.1.1 Convection Term

The convection term is integrated over a control volume and linearized as follows:

$$\int_V \nabla \cdot (\rho \mathbf{U} \phi) \, dV = \int_S d\mathbf{S} \cdot (\rho \mathbf{U} \phi) = \sum_f \mathbf{S}_f \cdot (\rho \mathbf{U})_f \, \phi_f = \sum_f F \phi_f \tag{3.20}$$

where the face field ϕ_f can be evaluated using a variety of schemes including central, upwind, and blended differencing. Central (or linear) differencing, which is second-order accurate, can be written as:

$$\phi_f \quad = \quad f_x \phi_P + (1 - f_x) \, \phi \tag{3.21}$$

$$f_x = \frac{\overline{fN}}{\overline{PN}} \tag{3.22}$$

OpenFOAM also includes two other centered schemes: cubicCorrection and midPoint. Upwind differencing for ϕ_f, which guarantees boundedness but is first-order accurate, can be written as

$$\phi_f = \begin{cases} \phi_P & \text{for} \quad F \geq 0 \\ \phi_N & \text{for} \quad F < 0 \end{cases} \tag{3.23}$$

In addition to the standard upwind scheme, linearUpwind, skewLinear and Quick schemes are variants of upwinded schemes which are available in OpenFOAM. Finally, there are a number of blended schemes that attempt to preserve both boundedness and accuracy of the solution. The gamma scheme can be written as

$$\phi_f = (1-\gamma)\left(\phi_f\right)_{UD} + \gamma\left(\phi_f\right)_{CD} \tag{3.24}$$

where the blending factor $0 \leq \gamma \leq 1$ determines how much dissipation is introduced.

The mass flux F in Eq. (3.20) is calculated from interpolated values of ρ and $\mathbf{U}$. Similar to interpolation of ϕ_f, F can be evaluated using a variety of schemes including centered, upwinded, and blended schemes, the latter of which includes a number of TVD/NVD schemes (such as limitedLinear, vanLeer, MUSCL, limitedCubic, SFCD, and Gamma).

3.4.1.2 Laplacian Term

The Laplacian term is integrated over a control volume and linearized as follows:

$$\int_V \nabla \cdot (\Gamma \nabla \phi)\, dV = \int_S d\mathbf{S} \cdot (\Gamma \nabla \phi) = \sum_f \Gamma_f \mathbf{S}_f \cdot (\nabla \phi)_f \tag{3.25}$$

The face gradient discretization is implicit when the length vector d between the center of the cell of interest P and the center of the neighboring cell N is orthogonal to the face plane, i.e., parallel to $\mathbf{S}_f$:

$$\mathbf{S}_f \cdot (\nabla \phi)_f = |S_f| \, \frac{\phi_N - \phi_P}{|d|} \tag{3.26}$$

In the case of non-orthogonal meshes, an additional explicit term is introduced which is evaluated by interpolating cell center gradients, themselves calculated by central differencing cell center values.

3.4.1.3 Source Terms

Source terms can be specified in three ways: explicit, implicit, and implicit/explicit. Explicit source terms are incorporated into an equation simply as a field of values. For example, to solve

Poisson's equation $\nabla^2\phi = f$, ϕ and f would be defined as `volScalarField` and then do

```
solve(fvm::laplacian(phi) == f)
```

In contrast, an implicit source term is integrated over a control volume and linearized by

$$\int_V S_\phi(\phi)\, dV = SpV_P\phi_P$$

The Implicit/Explicit approach changes between the two based upon the sign of the source term. If the source is positive, it is treated as an implicit source term so that it increases the diagonal dominance of the matrix. If the source is negative, it is treated as an explicit source term. In mathematical terms, the mixed source approach can be written as

$$\int_V S_\phi\left(\phi\right)\, dV = SuV_P + SpV_P\phi_P \tag{3.27}$$

3.4.2 Temporal Discretization

3.4.2.1 Temporal Derivatives

The first derivative $\partial/\partial t$ is integrated over a control volume with one of two schemes: 1st-order Euler implicit, or 2nd-order backward difference

$$\frac{\partial}{\partial t}\int_V \rho\phi\, dV = \frac{\left(\rho_P\phi_P V_P\right)^n - \left(\rho_P\phi_P V_P\right)^0}{\Delta t} \tag{3.28}$$

$$\frac{\partial}{\partial t}\int_V \rho\phi\, dV = \frac{3\left(\rho_P\phi_P V_P\right)^n - 4\left(\rho_P\phi_P V_P\right)^0 + \left(\rho_P\phi_P V_P\right)^{00}}{2\Delta t} \tag{3.29}$$

where the new values are $\phi^n = \phi(t+\Delta t)$, the old values are $\phi^0 = \phi(t)$, and the old-old values are $\phi^{00} = \phi(t-\Delta t)$.

3.4.2.2 Treatment of Spatial Derivatives in Transient Problems

Reconsider the integral form of the transport equation Eq. (3.18).

$$\int_t^{t+\Delta t}\left[\frac{\partial}{\partial t}\int_{V_p}\rho\phi\, dV + \int_{V_p}\nabla\cdot\left(\rho\mathbf{U}\phi\right)\, dV - \int_{V_p}\nabla\cdot\left(\rho\Gamma_\phi\nabla\phi\right)\, dV\right] dt = \int_t^{t+\Delta t}\left[\int_{V_p} S_\phi\left(\phi\right)\, dV\right] dt$$

Using Eqs. (3.20), (3.25), and (3.27), Eq. (3.18) can be written in a semi-discretized form

$$\int_t^{t+\Delta t}\left[\frac{\partial}{\partial t}\int_{V_p}\rho\phi\, dV + \sum_f F\phi_f - \sum_f\left(\rho\Gamma_\phi\right)\mathbf{S}\cdot\left(\nabla\phi\right)_f\right] dt$$

$$= \int_t^{t+\Delta t} (SuV_P + SpV_P\phi_P)\, dt \tag{3.30}$$

For an Euler implicit approach, this equation can be reduced to

$$\frac{(\rho_P\phi_P V_P)^n - (\rho_P\phi_P V_P)^0}{\Delta t} + \sum_f F\phi_f^n - \sum_f (\rho\Gamma_\phi)\,\mathbf{S}\cdot(\nabla\phi)_f^n = (SuV_P + SpV_P\phi_P^n) \tag{3.31}$$

In contrast, Eq. (3.30) can also be evaluted with a Crank-Nicholson scheme. The result is

$$\frac{(\rho_P\phi_P V_P)^n - (\rho_P\phi_P V_P)^0}{\Delta t} + \sum_f F\left(\frac{\phi_f^n + \phi_f^0}{2}\right) - \sum_f (\rho\Gamma_\phi)\,\mathbf{S}\cdot\left(\frac{(\nabla\phi)_f^n + (\nabla\phi)_f^0}{2}\right)$$
$$= SuV_P + SpV_P\left(\frac{\phi_P^n + \phi_P^0}{2}\right) \tag{3.32}$$

Regardless of the approach, the equations can be reduced to the algebraic system for every control volume

$$a_P\phi_P^n + \sum_N a_N\phi_N^n = R_P \tag{3.33}$$

where the coefficients a_P and a_N are the diagonal and off-diagonal coefficients, respectively, and R_P is the source-term vector.

3.5 Solution Algorithm for the Navier-Stokes Equations

Solution of the incompressible Navier-Stokes equations requires that three items be addressed: derivation of an equation for pressure, linearization of the momentum equations, and implementation of a pressure-velocity coupling algorithm. These items will be discussed in the following sections.

3.5.1 Linearization

The nonlinear convection terms in Eqs. (3.8) and (3.11) are reduced to $\sum_f F\phi_f^n$ where $F = \mathbf{S}\cdot(\mathbf{U})_f$. The challenge is that F, and a_P and a_N from Eq. (3.33), are functions of (U). The important issue is that the fluxes F should satisfy the continuity equation, Eq. (3.1) or (3.10). Linearization of the convection term means that the existing velocity (or flux) field that satisfies continuity will be used to calculate a_P and a_N. For strongly nonlinear phenomenon, two approaches can be used to capture the nonlinearity: sub-iteration over the entire algorithm such that the lagged velocities, and therefore the fluxes and coefficients, are iteratively updated; or use of small time-steps such that the error in not updating the fluxes and coefficients remains small. Both impose a computational cost.

3.5.2 Derviation of the Pressure Equation

A semi-discrete form of the momentum equations is used to derive the pressure equation:

$$a_P \mathbf{U}_P = \mathbf{H}(\mathbf{U}) - \nabla p \tag{3.34}$$

This equation is clearly an extension of Eq. (3.33); however, the pressure term has been broken out, and $\mathbf{H}(\mathbf{U})$ consists of two parts, the "transport part" which includes the matrix of coefficients for all neighbors multiplied by corresponding velocities, and the "source-term part" which includes part of the transient term and all other source terms (apart from the pressure gradient). For example, $\mathbf{H}(\mathbf{U})$ for the incompressible Navier-Stokes equations (excluding source terms due to gravity and turbulence) using Euler implicit temporal differencing is

$$\mathbf{H}(\mathbf{U}) = -\sum_N a_N \mathbf{U}_N + \frac{\mathbf{U}^0}{\Delta t} \tag{3.35}$$

Equation (3.34) can also be solved for the velocity at the cell center by dividing by a_P

$$\mathbf{U}_P = \frac{\mathbf{H}(\mathbf{U})}{a_P} - \frac{1}{a_P}\nabla p \tag{3.36}$$

From this, the velocity at the cell face can be found through interpolation, i.e.,

$$\mathbf{U}_f = \left(\frac{\mathbf{H}(\mathbf{U})}{a_P}\right)_f - \left(\frac{1}{a_P}\right)_p (\nabla p)_f \tag{3.37}$$

To derive the pressure equation, the discrete incompressible continuity equation is written

$$\sum_f \mathbf{S} \cdot \mathbf{U}_f = 0 \tag{3.38}$$

and Eq. (3.37) is inserted,

$$\nabla \cdot \left(\frac{1}{a_P}\nabla p\right) = \sum_f \mathbf{S} \cdot \left(\frac{\mathbf{H}(\mathbf{U})}{a_p}\right)_f \tag{3.39}$$

The Laplacian operator on the left-hand side can be discretized using the method described in Section 3.4.1.2, which results in the final form of the discretized incompressible Navier-Stokes equations

$$a_P \mathbf{U}_P = \mathbf{H}(\mathbf{U}) - \sum_f \mathbf{S}(p)_f \tag{3.40}$$

$$\sum_f \mathbf{S} \cdot \left[\left(\frac{1}{a_P}\right)_f (\nabla p)_f\right] = \sum_f \mathbf{S} \cdot \left(\frac{\mathbf{H}(\mathbf{U})}{a_P}\right)_f \tag{3.41}$$

Finally, if the face fluxes F are computed using U_f from Eq. (3.37),

$$F = \mathbf{S} \cdot \mathbf{U}_f = \mathbf{S} \cdot \left[\left(\frac{\mathbf{H}(\mathbf{U})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_p (\nabla p)_f \right] \tag{3.42}$$

then the fluxes are guaranteed to be conservative.

3.5.3 Pressure-Velocity Coupling

The discretized form of the Navier-Stokes system in Eqs. (3.40) and (3.41) are coupled in that each contains velocity and pressure. While there are algorithms for solving the fully-coupled set of equations, this remains computationally expensive in comparison to the segregated methods for coupling the pressure and velocity fields. The most common segregated methods are the PISO [29] and SIMPLE [30] algorithms and their derivatives (e.g., SIMPLE-C, SIMPLER). Both are commonly used in OpenFOAM; however, SIMPLE and PISO are typically used for steady and transient problems, respectively. Each is briefly discussed in the following sections.

3.5.3.1 PISO

The pressure-implicit split-operator (PISO) algorithm is a predictor-corrector approach for solving transient flow problems. For the PISO algorithm, the momentum equation is solved first; however, since the exact pressure gradient source term is not known at this stage, the pressure field from the previous time-step is used instead. This stage is called the momentum predictor and gives an approximation of the new velocity field. Using the predicted velocities, the $\mathbf{H}(\mathbf{U})$ operator can be assembled and the pressure equation can be formulated. The solution of the pressure equation gives the first estimate of the new pressure field. This step is called the pressure solution.

Next, conservative fluxes consistent with the new pressure field are computed using Eq. (3.42). The velocity field should also be corrected as a consequence of the new pressure distribution. Velocity correction is done in an explicit manner, using Eq. (3.36). This is the explicit velocity correction stage.

A closer look at Eq. (3.36) reveals that the velocity correction actually consists of two parts: a correction due to the change in the pressure gradient $\left(\frac{1}{a_P}\nabla p\right)$ and the transported influence of corrections of neighboring velocities $\left(\frac{\mathbf{H}(\mathbf{U})}{a_P}\right)$. The fact that the velocity correction is explicit means that the latter part is neglected. It is therefore necessary to correct the $\mathbf{H}(\mathbf{U})$ term, formulate the new pressure equation and repeat the procedure. In other words, the PISO loop consists of an implicit momentum predictor followed by a series of pressure solutions and explicit velocity corrections. The loop is repeated until a pre-determined tolerance is reached.

Another issue is the dependence of $\mathbf{H}(\mathbf{U})$ coefficients on the flux field. After each pressure solution, a new set of conservative fluxes is available. Therefore, it would be possible to recalculate the coefficients in $\mathbf{H}(\mathbf{U})$. This, however, is not done: it is assumed that the non-linear coupling

is less important than the pressure-velocity coupling, consistent with the linearization of the momentum equation. The coefficients in $\mathbf{H}(\mathbf{U})$ are therefore kept constant through the whole correction sequence and will be changed only in the next momentum predictor.

The overall PISO algorithm can be summarized as follows:

1. Set the initial conditions.
2. Begin the time-marching loop.
3. Assemble and solve the momentum predictor equation with the available face fluxes (and pressure field).
4. Solve the equation, and explicitly correct the velocity field. Iterate until the tolerance for pressure-velocity system is reached. At this stage, pressure and velocity fields for the current time-step are obtained, as well as the new set of conservative fluxes.
5. Using the conservative fluxes, solve all other equations in the system. If the flow is turbulent, calculate the eddy viscosity from the turbulence variables.
6. Go to the next time step, unless the final time has been reached.

3.5.3.2 SIMPLE

The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm provides an iterative method for solving steady-flow problems. The algorithm is comprised of several steps: 1) an approximation of the velocity field is obtained by solving the momentum equation; 2) the pressure equation is solved in order to obtain an updated pressure field; 3) conservative fluxes are computed using Eq. (3.42); 4) the velocity field is explicitly corrected using Eq. (3.36); and 5) all secondary equations (e.g., turbulent kinetic energy and specific dissipation) are solved. At each step, under-relaxation is used to stabilize the nonlinearity of the solution algorithm.

The overall SIMPLE algorithm can be summarized as follows:

1. Set all field values to some initial values.
2. Begin the global iteration loop.
3. Assemble and solve the under-relaxed momentum predictor equation.
4. Solve the pressure equation and calculate the conservative fluxes. Update the pressure field with an appropriate under-relaxation. Perform the explicit velocity correction using Eq. (3.36).
5. Solve the other equations in the system using the available fluxes, and the updated pressure and velocity fields.
6. Check the convergence criterion for all equations. If the system is not converged, start a new global iteration.

3.6 Homogeneous Multiphase

Multiphase flow occurs when more than one fluid is present. In general, the fluids consist of different chemical species such as air and water; however, in some applications, different thermodynamic phases of the same species may occur, e.g., when the fluid cavitates or boils.

A disperse multiphase flow may be modeled using either a Lagrangian multiphase model (e.g., [31]) or an Eulerian-Eulerian multiphase model. While the Lagrangian methods are of central importance to problems at the Applied Research Laboratory Penn State (e.g., cavitation inception, bubble dynamics, and spray), the current focus is on the Eulerian-Eulerian model. It is important to distinguish between multicomponent and multiphase flow. A multicomponent fluid is assumed to consist of a mixture of chemical species which are mixed at the molecular level, e.g., trace-chemical transport problems of explosives detection [32] and canine olfaction [33, 34]. In this case, single mean velocity, pressure, scalar, and temperature fields are solved for the fluid. In contrast, the fluids in a multiphase flow are assumed to be mixed at macroscopic length scales. Examples are free-surface flows, cavitation, gas bubbles in a liquid, and liquid droplets or solid particles in a gas. Two distinct methods are often used for Eulerian-Eulerian multiphase flow: inhomogeneous and homogeneous models.

Inhomogeneous multiphase flow refers to the case where separate velocity fields and other relevant fields exist for each fluid. The pressure field is shared by all fluids. The fluids interact via interphase transfer terms. NPHASE [35] is an example of an inhomogeneous multiphase flow code. An example of an inhomogeneous solver in OpenFOAM is the bubbleFoam solver.

Of primary focus here is homogeneous multiphase flow, which can be considered to be a limiting case of Eulerian-Eulerian multiphase flow, where the conservation equations are formulated for the mixture, and an additional transport equation is solved for the local volume-fraction. Due to the homogeneous assumption, all fluids have the same velocity (i.e., zero slip between phases), temperature, turbulence, and pressure field. Cavitation modeling at ARL Penn State [36, 37, 38, 39, 40, 41, 42, 43] has relied on the homogeneous assumption.

3.6.1 Interface Capturing: Volume of Fluid

Many first-generation free-surface RANS codes relied upon interface tracking, which required separate solution of the kinematic free-surface boundary condition for the wave elevation, and a methodology for dynamically adapting the grid to both the free-surface and the ship hull. Unfortunately, interface tracking is unstable, and cannot resolve complex wave/geometry intersections or wave breaking.

Several of the multiphase solvers in OpenFOAM are formulated as Volume-of-Fluid (VOF) methods, including the interFoam family (interFoam, rasInterFoam, and lesInterFoam) of solvers. A simple tutorial case of a dam instantly breaking is used to show the effectiveness of the interFoam solvers. The water and atmospheric phases are shown separated below in Figure 3.2 as $\gamma = 1$ and $\gamma = 0$, respectively. As the solution proceeds in time, gravity forces the water over the bump while the VOF method maintains and tracks the points of fluid-fluid interaction.

The waves slosh around and droplets can be seen projecting through the air in Figure 3.3, demonstrating the effectiveness of the VOF method.

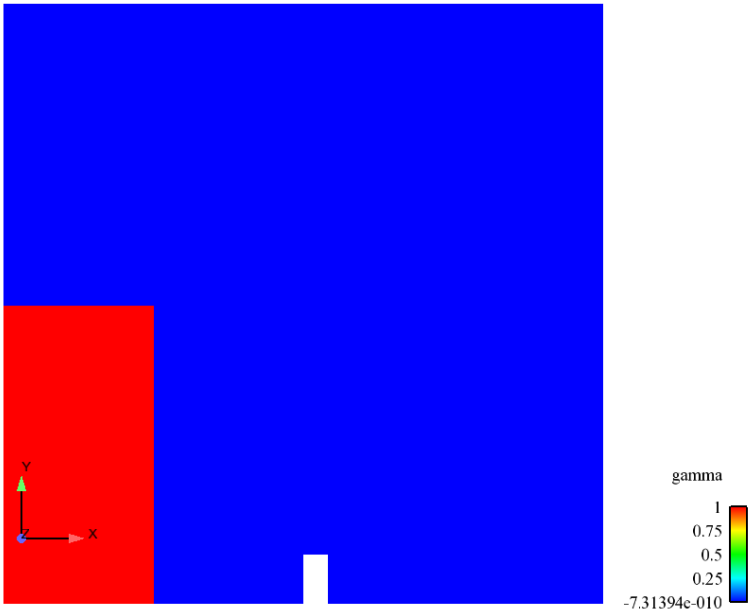

Figure 3.2. Initial setup of the damBreak case separating the water from the air.

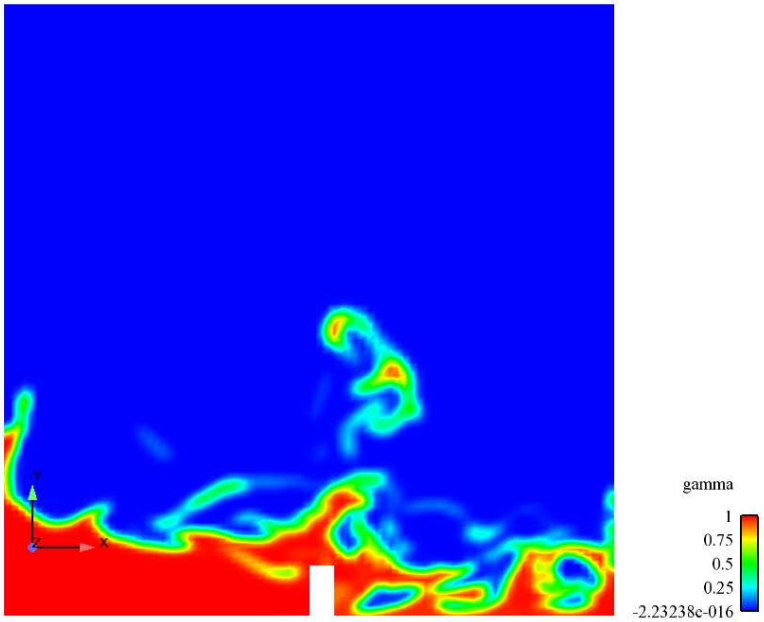

Figure 3.3. The VOF method maintains the fluid-fluid interaction one second into the simulation.

3.6.2 Compressible Cavitation Model

The most general formulation of a homogeneous cavitation model is to use the compressible form of the equations [37, 44].

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{u}) = 0 \tag{3.43}$$

$$\frac{\partial (\rho_m \mathbf{u})}{\partial t} + \nabla \cdot (\rho_m \mathbf{u}\mathbf{u}) = \rho_m \mathbf{g} + \nabla \cdot \sigma \tag{3.44}$$

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\gamma \mathbf{u}) = \frac{1}{\rho_\ell} \left(\dot{m}^- + \dot{m}^+ \right) \tag{3.45}$$

where ρ_ℓ is the liquid density, ρ_v is the vapor density, γ is the liquid volume fraction, $\rho_m = \rho_\ell \gamma + \rho_v (1 - \gamma)$ is the mixture density, and the mass transfer from liquid to vapor is defined as

$$\dot{m}^- = \frac{C_{dest} \rho_v \gamma \, \mathtt{min} \left[0, p - p_v \right]}{\left(\frac{1}{2} \rho_\ell U_\infty^2 \right) t_\infty} \tag{3.46}$$

and the mass transfer from vapor to liquid is

$$\dot{m}^+ = \frac{C_{prod} \rho_v \gamma^2 (1 - \gamma)}{t_\infty} \tag{3.47}$$

In these empirical relations, p_v is the vapor pressure, t_∞ is a time scale defined as the ratio of the characteristic length scale of the flow to the reference velocity scale U_∞. Following [36], the coefficients C_{dest} and C_{prod} have been set to 100.

For the turbulence model equations, they are formulated in terms of the mixture properties, i.e., ρ_m and μ_m.

3.6.3 Incompressible Cavitation Model

As shown in the previous chapter, the incompressible cavitation model governing equations are similar to the compressible cavitation equations with the exception of the density derivatives. The following equations are the basis for previous work done in the field of incompressible cavitation modeling [36, 45].

$$\nabla \cdot \mathbf{u} = \left(\frac{1}{\rho_\ell} - \frac{1}{\rho_v} \right) \left(\dot{m}^- + \dot{m}^+ \right) = S_p \tag{3.48}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho} \nabla p + \mathbf{g} + \nabla \cdot (\nu \nabla \mathbf{u}) \tag{3.49}$$

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\gamma \mathbf{u}) = \frac{1}{\rho_\ell} \left(\dot{m}^- + \dot{m}^+ \right) = S_\gamma \tag{3.50}$$

Chapter 4

OpenFOAM Implementation

4.1 OpenFOAM and the C++ Language

Written in the programming language C++, OpenFOAM is based on the Finite Volume method (FVM) and operates on arbitrary unstructured (more on this term later) 3D meshes. The ability for users to develop their own code makes OF a particularly powerful research tool in the field of CFD. There are several topics of interest to be covered prior to describing the mass transfer model itself.

OpenFOAM (Open Field Operation and Manipulation) is a C++ toolbox that was originally developed in the late 1980's at Imperial College, London, to solve continuum mechanics problems, primarily computational fluid dynamics. OpenFOAM was released free to the public in 2004 and since then has been developed by OpenCFD Ltd. Additionally, since OpenFOAM is an open source code, there is an active and growing community of users that create their own solvers and utilities to solve various CFD problems. OpenFOAM was the first and continues to be the leader as a general purpose CFD package to be distributed freely as an open-source license.

In addition to the ability for users to create their own applications, OpenFOAM is supplied with its own pre- and post-processing environments. The overall structure of OpenFOAM is shown in Figure 4.1.

4.1.1 The Structure of an OpenFOAM case

OpenFOAM requires three directories to be present in any case file: *system*, *constant*, and "time" directories (Figure 4.2). These three files contain all the information needed to run a simulation.

A *constant* directory contains two important subdirectories - the *polyMesh* folder, which contains the mesh information, and at least one file containing the physical properties of the system, typically the *transportProperties* file. Additional files may be present that contain thermodynamic properties, rotational parameters, or any initial constants the user wishes to specify.

A *system* directory is composed of files that control the solution procedure. At a minimum,

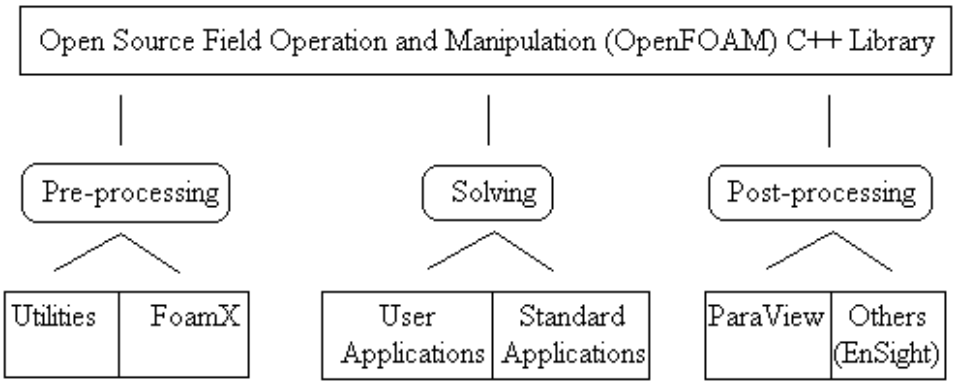

Figure 4.1. The structure of OpenFOAM.

this directory includes the files: *controlDict* which is where the start and end time, time steps, and data output parameters are set; *fvSchemes* that contains the discretization schemes; and, *fvSolution* which stores the equation solvers, tolerances and other algorithm controls.

The 'time' directories contain the data files for particular fields of interest. These data can either be initial or boundary conditions that the user has specified (in time step 0 file), or written data as the solution progresses.

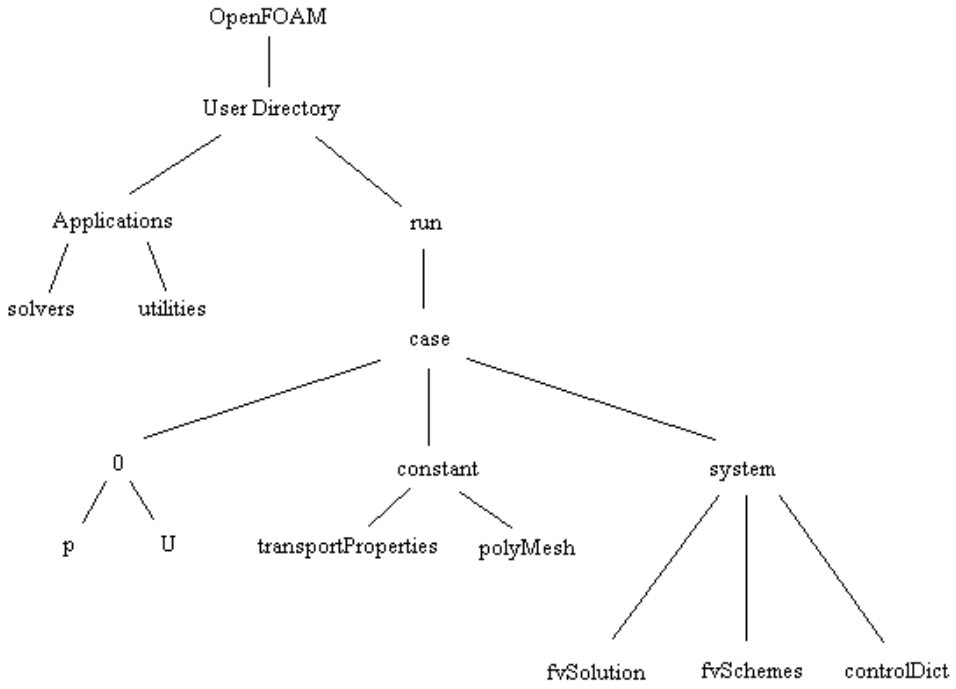

Figure 4.2. Directory tree of a generic OpenFOAM case.

4.1.2 Pre-Processing

FoamX is the pre-processing GUI software that has been historically included with OpenFOAM. This JAVA based program is useful for first time or beginning users to understand the structure and layout of OpenFOAM cases. Once the structure of OF is understood, setting up cases is usually faster from the command line rather than through the GUI, which is why FoamX has been phased out over time. In addition to FoamX, OpenFOAM offers users a simple mesh generation utility called "blockMesh". This mesh generation program quickly and easily allows users to generate geometrically simple meshes. For more complicated meshes, such as propellers, OpenFOAM provides utilities to convert meshes from software such as Fluent, Star-CD, Gambit, CFX, and others into the OpenFOAM format. Additionally, mesh modification can be performed through OF which includes, but is not limited to, mesh refinement, setting values for fields such as gamma, mesh merging, splitting of meshes, and mesh rotation. The most recent versions of OpenFOAM have also included an unstructured mesh generator.

4.1.3 Post-Processing

OpenFOAM comes standard with a version of ParaView that can read the output files of a case. According to the website www.paraview.org, ParaView creators describe it as an open-source, multi-platform application designed to visualize data sets of size varying from small to very large. It was originally created in 2000 as part of a collaborative effort between Kitware Inc. and Los Alamos National Laboratory and is still being developed by several additional sources including Sandia National Labs, CSimSoft, and Army Research Lab. Although ParaView is an adequate post-processing program, OpenFOAM provides post-processing utilities that allow users to convert the results of simulations into data recognizable by third party software such as EnSight, TecPlot, FieldView, or Fluent.

Sampling of data can also be specified through a *sampleDict* dictionary in the case *system* directory. These data can be written in various outputs including a raw tabular output or as gnuplot package. A log that records each time step, along with residuals for each solution variable, can also be plotted to analyze convergence history, number of iterations, forces, or any other parameter of interest.

4.1.4 Operators and Dimensions

In order to handle algebraic operations of tensor classes, OpenFOAM provides a syntax which closely models that of the mathematical description. For example, some functions are expressed by descriptive functions, *e.g.*, magnitude of $z$would be $\mathbf{mag(z)}$. Alternatively, simple mathematical operators retain there form in OpenFOAM. For example, scalar multiplication of A and B is simply represented as A * B. A complete list is provided in the User's Guide as a reference.

A useful feature of OpenFOAM is the automatic dimension checking performed on all specified equations. Using the SI base units, OF enables users to specify the dimensions of tensors in seven properties: mass, length, time, temperature, quantity, current, and luminous intensity.

This ability of OF acts as a safeguard to prevent performing meaningless calculations of tensors that do not have or produce consistent units of measurement.

4.2 Applications: Solvers and Utilities

Numerous solvers are included with OpenFOAM that can all be modified to suit the user's specific requirements. They range from basic CFD solvers, to compressible, incompressible, multiphase, LES, DNS, combustion, heat transfer, stresses, electromagnetics, and even a financial solver. Many pre- and post-processing utilities are available ranging from mesh manipulation, velocity field analysis, to surface data sampling. Since OF is an open-source CFD program, users are allowed and encouraged to create their own customized solvers and utilities that meet their specific demands. Generally, one of the pre-packaged solvers or codes can be modified to suit the user's needs, which is a convenient time saver over the alternative of creating the entire code from scratch. This technique of modifying certain solvers and utilities will be explored in the following sections.

4.2.1 Solver: simpleFoam

SimpleFoam is a steady-state Navier-Stokes solver which is based upon the SIMPLE algorithm. Incorporating all the turbulence models turbFoam does, the codes are virtually identical except for the use of SIMPLE over the PISO algorithm. This solver is ideal for rapidly converging to steady state solutions of the test cases covered later in this thesis such as the flat plate, airfoils at low angles of attack, and the hemispherical headform.

4.2.2 Solver: interFoam

InterFoam is a transient interface-capturing Navier-Stokes solver which is based upon the VOF and PISO methods. As discussed in the previous chapter, the class of interFoam solvers, including lesInterFoam, multiphaseInterFoam, and rasInterFoam are ideal for unsteady, multiphase flows. This code demonstrates one of the benefits of C++ - by utilizing header files that contain separate code, the main .C file is much more compact. When called using the **# include** command, the main code goes to the specified .H file, *e.g.*, UEqn.H, pEqn.H or gammaEqnSubCycle.H, and performs the operations contained within. This solver is the base for more complicated VOF models including the cavitation solvers.

4.2.3 Utility: Wall Shear Stress

The wall shear stress post-processing utility enables the shear stress values to be extracted from any surface of a simulation. This tool was useful for analyzing the turbulent flat-plate data, for example. Recall Eqs. (3.5)and (3.16) for the fluid stress tensor and the Reynolds stress.

$$\sigma = -p\mathbf{I} + \mu\left[\nabla\mathbf{U} + (\nabla\mathbf{U})^T\right] + \lambda\left(\nabla\cdot\mathbf{U}\right)\mathbf{I} \tag{4.1}$$

$$-\overline{\mathbf{u'u'}} = \nu_t \left(\nabla \mathbf{U} + (\nabla \mathbf{U})^T\right) + \frac{2}{3} k \mathbf{I} \tag{4.2}$$

The normal stresses correspond to the hydrostatic tensor (i.e., the diagonal), and the shear stresses correspond to the deviatoric tensor, the latter of which can written as

$$\frac{\tau_{eff}}{\rho} = \nu \left[\nabla \mathbf{U} + (\nabla \mathbf{U})^T\right] + \nu_t \left[\nabla \mathbf{U} + (\nabla \mathbf{U})^T\right] = \nu_{eff} \left[\nabla \mathbf{U} + (\nabla \mathbf{U})^T\right] \tag{4.3}$$

Using OpenFOAM functions, this can be written as

$$\frac{\tau_{eff}}{\rho} = 2\nu_{eff} \, \mathtt{dev}\left(\mathtt{symm}\left(\nabla \mathbf{U}\right)\right) \tag{4.4}$$

where, for a generic tensor $\mathbf{T}$

$$\mathtt{symm}\left(\mathbf{T}\right) = \frac{1}{2}\left(\mathbf{T} + \mathbf{T}^T\right) \tag{4.5}$$

$$\mathtt{dev}\left(\mathbf{T}\right) = \mathbf{T} - \frac{1}{3}\left(\mathtt{tr}\left(\mathbf{T}\right)\right) \tag{4.6}$$

$$\mathtt{tr}\left(\mathbf{T}\right) = T_{11} + T_{22} + T_{33} \tag{4.7}$$

Finally, to calculate the shear-stress vector parallel to the wall, the inner-product of the wall-normal vector and the shear-stress tensor is used,

$$\tau_w = \mathbf{n}_f \cdot \tau_{eff} \tag{4.8}$$

which can be evaluated either by directly using Eq. (4.3) on the boundary face, or by using the surface normal gradient function in OpenFOAM, i.e.,

$$\tau_w = \nu_{eff} * \mathtt{sngrad}(U) \tag{4.9}$$

4.2.4 Utility: Forces and Moments

The fluid dynamics forces and moments can be computed as the integral of the stress tensor acting on the no-slip boundaries. This utility was used to calculate the drag on surfaces throughout the simulations presented in this paper. The C++ code for the force and moment utilty is rather short, and therefore is shown below Equation (4.11) to demonstrate the structure and layout of a simple OpenFOAM utility.

$$\mathbf{F} = \int_{S_f} (p + \sigma) \, dS \tag{4.10}$$

$$\mathbf{M} = \int_{S_f} (x - x_0) \times (p + \sigma) \, dS_f \tag{4.11}$$

```
// Pressure component
pressureForces[patchI] =
    gSum
    (
        p.boundaryField()[curPatch]*
        mesh.Sf().boundaryField()[curPatch]
    );

pressureMoments[patchI] =
    gSum
    (
        (mesh.Cf().boundaryField()[curPatch] - origin_)
          ^ (
          p.boundaryField()[curPatch]*
          mesh.Sf().boundaryField()[curPatch]
        )
    );

// Viscous component
viscousForces[patchI] =
    gSum
    (
        -nu.value()*U.boundaryField()[curPatch].snGrad()*
          mesh.magSf().boundaryField()[curPatch]
    );

viscousMoments[patchI] =
    gSum
    (
        (mesh.Cf().boundaryField()[curPatch] - origin_)
          ^ (-nu.value()*U.boundaryField()[curPatch].snGrad()*
          mesh.magSf().boundaryField()[curPatch])
    );
```

4.2.5 Cavitation Solver: cavitatingFoam

CavitatingFoam, a fully compressible two-phase flow solver, is a recent addition to OF which is modeled by barotropic equations of state. It allows the user to select one of three different compressibility models: the Wallis, Chung, and linear models. The Wallis speed of sound model (Eqn. 4.12) is well documented [46] and is therefore the preferred choice for running the cavitatingFoam solver. However, the other models (Eqn. 4.13-4.15) were tested for comparison and

the results are included in the next chapter. A speed of sound graph produced by each of the compressibility models can be seen in Figure 4.3. Wallis Speed of Sound:

$$a = \frac{1}{\sqrt{(\gamma\rho_v sat + (1-\gamma)\rho_l sat)(\gamma\frac{\psi_v}{\rho_v sat} + (1-\gamma)(\frac{\psi}{\rho_l sat})}} \tag{4.12}$$

Chung Speed of Sound:

$$a = \frac{1}{[\frac{(1-\alpha)}{\sqrt{\psi_v}} + \gamma\frac{sfa}{\sqrt{\psi_l}}]\frac{\sqrt{\psi_v\psi_l}}{sfa}} \tag{4.13}$$

where sf_a is represented as:

$$sfa = \sqrt{\frac{\frac{\rho_v sat}{\psi_v}}{(1-\alpha)\frac{\rho_v sat}{\psi_v} + \gamma\frac{\rho_l sat}{\psi_l}}} \tag{4.14}$$

Linear Speed of Sound:

$$a = \frac{1}{\sqrt{\gamma\psi_v + (1-\gamma)\psi_l}} \tag{4.15}$$

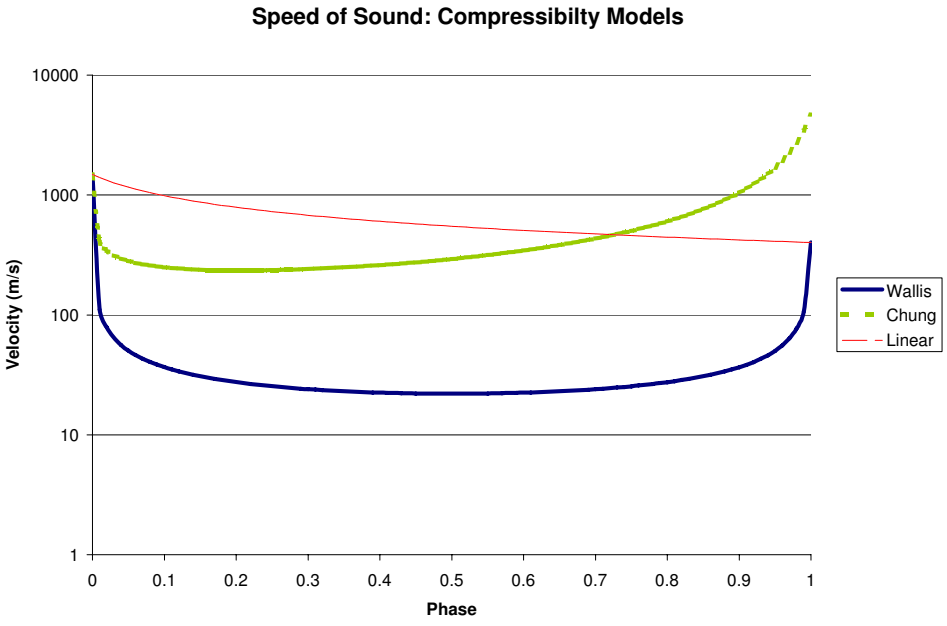

Figure 4.3. Speed of sound curve for a range of volume fractions using each compressibility model.

One feature this solver lacks is the implementation of a turbulence model. OF version 1.5 provides a solver similar to cavitatingFoam called rasCavitatingFoam which contains the option to

include a variety of RAS turbulence models. In addition, there is a cavitatingFoam version called lesCavitatingFoam which incorporates LES turbulence models. In each solver, the equations of interest are virtually the same, with minor variations in the velocity equations depending on which turbulence model is being used. A summary of the equations solved by cavitatingFoam - density, volume fraction, velocity, and pressure - are included below.

$$\rho = \frac{\partial \rho}{\partial t} + \nabla \cdot (\psi_v \rho) \tag{4.16}$$

$$\gamma = max[min[\frac{\rho - \rho_{lsat}}{\rho_{vsat} - \rho_{lsat}}, 1], 0] \tag{4.17}$$

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\phi U) - \nabla \cdot \mu_f \nabla U - \nabla(\mu_f (dev(U) \cdot A)) \tag{4.18}$$

$$p = \frac{\partial \psi N}{\partial t} - (\rho_l + (\psi_l - \psi_v) p_s at) \frac{\partial \gamma}{\partial t} - p_s at \frac{\partial \psi}{\partial t} + \nabla(\phi_v \rho) + \nabla(rp) - \nabla \cdot rUAf \nabla N \tag{4.19}$$

$$N = \frac{\rho - (1-\gamma)\rho_l - ((\gamma \psi_v + (1-\gamma)\psi_l) - \psi) p_{sat}}{\psi} \tag{4.20}$$

where ϕ is equal to $1/a^2$, dev is the deviatoric component, A is the cell face area, rp is $rUAf$ multiplied by the surface area of each face of each cell and the pressure gradient, and $rUAf$ is the density multiplied by the inverse of momentum matrix diagonal interpolated to the face.

4.2.6 Cavitation Solver: interPhaseChangeFoam

As previously mentioned, interPhaseChangeFoam is a solver for two incompressible, isothermal immiscible fluids with phase change that incorporates the VOF phase-fraction based interface capturing approach. One momentum equation is solved which incorporates the fluid properties of the mixture. The Kunz, Merkle, and Schnerr-Sauer models are included to simulate phase change, primarily for cavitation. The velocity and pressure equations of cavitatingFoam can be seen below, respectively. The volume fraction equation, which also contains source terms to account for the mass transfer, is solved using an implicit version of a multidimensional universal limiter with explicit solution (MULES) which ensures boundedness.

$$U = \frac{\partial(\rho U)}{\partial t} + \nabla(\rho \phi U) - S_p(\frac{\partial \rho}{\partial t} + \nabla \rho \phi) - \nabla \cdot \mu_f \nabla U - \nabla U \cdot \nabla \mu_f \tag{4.21}$$

$$P = \nabla \phi - \nabla \cdot U \nabla p + (\dot{V}_{vp} - \dot{V}_{cp})(\rho g h - p_{sat}) + S_p(\dot{V}_{vp} - \dot{V}_{cp}) \tag{4.22}$$

where S_p represents a source term and $\dot{V_{vp}}$ and $\dot{V_{cp}}$ are the volumetric vaporization and condenstion rates. The exact implementation of the pressure and volume fraction source terms will be discussed for each of the phase change models, Kunz, Merkle, and SchnerrSauer.

4.2.6.1 The Kunz Model

The OF implementation of the Kunz model has been slightly modified in order to handle the mass transfer terms in the pressure equation. The other equations found in OpenFOAM remain

the same as described in the original model. The modified OpenFOAM mass transfer term is as follows:

$$\dot{m}^{+} = \frac{C_{prod}\gamma^2(1-\gamma)\rho_v max[p - p_s at, 0]}{t_\infty max[p - p_s at, 0.01 p_{sat}]} \tag{4.23}$$

and the $\dot{m}^{-}$ term remains unchanged from Kunz's formulation (Eq. 2.3). Additionally, the mass transfer terms in the volume fraction equation are:

$$\dot{m}^{+} = \frac{C_{prod}\gamma^2(1-\gamma)\rho_v max[p - p_{sat}, 0]}{max[p - p_{sat}, 0.01 p_{sat}]} \tag{4.24}$$

$$\dot{m}^{-} = C_{prod}\gamma(p - p_{sat}, 0) \tag{4.25}$$

4.2.6.2 The Merkle Model

Based off the work of Merkle[11], the following coefficients are used in this phase change model for the volume fraction equation:

$$\dot{m}^{+} = \frac{C_{prod} max[p - p_{sat}, 0](1-\gamma)}{0.5 {U_\infty}^2 t_\infty} \tag{4.26}$$

$$\dot{m}^{-} = \frac{C_{dest} min[p - p_{sat}, 0]\gamma\rho_l}{0.5 {U_{\inf}}^2 t_\infty \rho_v} \tag{4.27}$$

In the same manner as the Kunz implementation, the Merkle model also contains explicit mass transfer terms for use in the pressure equation:

$$\dot{m}^{+} = (1-\gamma) pos(p - p_{sat}) \tag{4.28}$$

$$-\dot{m}^{-} = \gamma neg(p - p_{sat}) \tag{4.29}$$

where the boolean *pos* in OF represents a function which is greater than or equal to zero, and the *neg* boolean represents a function that is less than or equal to zero.

4.2.6.3 The Schnerr & Sauer Model

The Schnerr and Sauer model is a homogeneous bubbly liquid flow model that describes cavitation by a modified Rayleigh-Plesset equation[17] as described in Chapter 2. The OpenFOAM implementation is as follows:

$$rRb = \left[\frac{4\pi n}{3}\frac{\gamma}{1 + \gamma_{nuc} - \gamma}\right]^{(1/3)} \tag{4.30}$$

$$V_{nuc} = n\pi {d_{nuc}}^3/6 \tag{4.31}$$

$$\gamma_{nuc} = V_{nuc}/(1 + V_{nuc}) \tag{4.32}$$

$$\rho = \gamma\rho_l + (1-\gamma)\rho_v \tag{4.33}$$

$$P_{coeff} = \frac{3\rho_l\rho_v\sqrt{2/3\rho_l}rRb}{\rho\sqrt{p - p_{sat} + 0.01p_{sat}}} \tag{4.34}$$

$$\dot{m}^+ = C_{prod}\gamma P_{coeff} max[p - p_s at, 0] \tag{4.35}$$

$$\dot{m}^- = C_{dest}(1 - \gamma_{nuc} - \gamma)P_{coeff} min[p - p_{sat}, 0] \tag{4.36}$$

where n is the bubble number density, $dNuc$ is the nucleation site diameter, γ_{nuc} is the nucleation site volume fraction, and rRb is the reciprocal bubble radius. This model contains the mass transfer coefficient terms in an explicit manner for the pressure equation as:

$$\dot{m}^+ = C_{prod}(1-\gamma)\gamma P_{coeff} pos[p - p_s at] \tag{4.37}$$

$$\dot{m}^- = -C_{dest}(1 + \gamma_{nuc} - \gamma)\gamma P_{coeff} neg[p - p_{sat}] \tag{4.38}$$

Chapter 5

Cases and Results

5.1 Single Phase Non-Cavitating Flow

In order to confirm the accuracy of the solver, several test cases were run and compared to experimental results.

5.1.1 Validation of a Turbulent Boundary Layer on a Flat Plate

Perhaps one of the most fundamental test cases to perform, this study gave insight into the accuracy of OpenFOAM in handling such parameters as wall shear stress, boundary layer profiles, and local skin friction coefficients. Low Reynolds number turbulence models will be evaluated against one another as well as to the wall function turbulence model.

5.1.1.1 Validation Data and Flow Parameters

In order to validate the results of the turbulent flat plate case, comparison to turbulent flow equations are presented. Boundary layer profiles will be compared to the law of the wall and law of the wake, while local and total drag coefficients are compared to proven skin friction equations of White [47]. Both sublayer resolved grids and wall-function grids will be tested for accuracy with the Spalart-Allmaras and $k-\epsilon$ turbulence models respectively. The simulations were run at $Re_L = 9 \times 10^6$, with the free stream velocity set to 1 m/s and the kinematic viscosity, ν, equal to 1×10^{-6} m^2/s. The pressure was set to 0 m^2/s^2 (the units of pressure in OF are divided by density) and the free stream turbulent kinetic energy, k, and turbulent dissipation rate, ϵ, were set as follows:

$$k = \frac{1}{2}\overline{\mathbf{U}' \bullet \mathbf{U}'} \tag{5.1}$$

$$\epsilon = \frac{C_\mu^{0.75} k^{1.5}}{l} \tag{5.2}$$

where C_μ is a constant of the $k-\epsilon$ model equal to 0.09, and l is assumed to be 1% of the characteristic length, L. In a Cartesian coordinate system, k can be represented by:

$$k = \frac{1}{2}(U_x'^2 + U_y'^2 + U_z'^2) \tag{5.3}$$

where $U_x'^2$, $U_y'^2$, and $U_z'^2$ are the fluctuating components of velocity in x, y, and z directions, respectively. The fluctuating components are assumed to be 1% of the free stream velocity (1 m/s), so for these cases, assuming turbulence is isotropic, $U_x' = U_y' = U_z' = U' = 0.01$ m/s. Therefore, k is equal to the following:

$$k = \frac{3}{2}(\frac{1}{0.01})^2 \mathrm{m^2 s^{-2}} = 0.00015\,\mathrm{m^2 s^{-2}} \tag{5.4}$$

and ϵ follows as:

$$\epsilon = \frac{0.09^{0.75} \times 0.00015^{1.5}}{9 * 0.01} = 3.35 \times 10^{-6}\,\mathrm{m^2 s^{-3}} \tag{5.5}$$

As for the Spalart-Allmaras turbulence model cases, the value of $\tilde{\nu}$ was set to $\frac{\nu}{2} = 5 \times 10^{-7}\mathrm{m^2/s}$.

5.1.1.2 Grid Generation

For the wall-function grids that used the $k-\epsilon$ turbulence model, a 2D study was performed using varying y^+ values to determine the sensitivity of OF. Six grids were generated based on y^+ values corresponding to 25, 50, 100, 150, 200, and 500. Two sublayer resolved grids were created, one for $y^+ = 0.1$ and another for $y^+ = 1$. All of the grids contained 8,532 cells with a total length of 10 m and a 2.25 m height. The total length, L, of the flat plate was 9 m.

OpenFOAM has many different options available for boundary conditions. In this flat plate case, the left side of the grid is considered the "velocity inlet", the right side the "pressure outlet", the top was set to a "symmetry" plane, the front 1m of the bottom of the grid was set to a "slip" plane, and the wall was set to "wall". Since OF requires a 3D grid, the front and back planes were simply set to "empty" boundary condition to simulate a 2D case.

5.1.1.3 Numerical Parameters

The control dictionary in OpenFOAM cases is stored in the system directory and is where the user controls the running of the code. For these cases, the code was run for 5000 time steps to ensure complete convergence in all steady-state solutions. Also contained in the system directory is the *fvSchemes* and the *fvSolution* files, both of which were set to the same parameters for each of the flat plate cases. In the *fvSolution* file, the convergence criteria was set to 1×10^{-8} and the relaxation factors were set to 0.7 for all variables except pressure, which was set to 0.3. Table 5.1 lists the typical components of the *fvSchemes* file, while Table 5.2 shows the breakdown of finite volume schemes used to solve each term of the flat plate cases.

It should be noted that these finite volume schemes will be used for the remaining cases

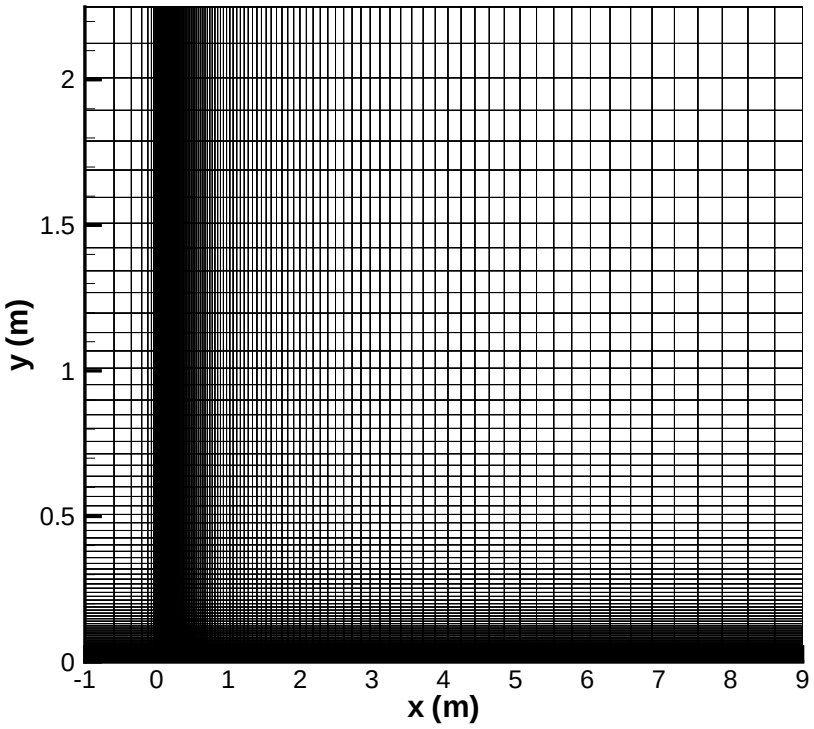

Figure 5.1. Mesh used in the y^+ 100 $k-\epsilon$ wall functions case.

Table 5.1. Finite Volume Schemes

OF Keyword	Category of mathematical terms
timeScheme	$\partial/\partial t$, $\partial^2/\partial t^2$
interpolateionSchemes	Interpolation (point-to-point)
snGradSchemes	Component of gradient normal to a cell face
gradSchemes	Gradient ∇
divSchemes	Divergence $\nabla\cdot$
laplacianSchemes	Laplacian ∇^2

Table 5.2. Flat Plate Cases: Finite Volume Schemes

OF Keyword	Description of Discretization Scheme
timeScheme	Steady State
interpolateionSchemes	Central Difference
snGradSchemes	Explicit non-orthogonal correction
gradSchemes	Second-Order, Gaussian integration
divSchemes	Combination of first-order, bounded and second-order, unbounded
laplacianSchemes	Unbounded, second-order, conservative

throughout this paper unless otherwise stated.

5.1.1.4 Simulation Results

5.1.1.4.1 Wall Functions - $k-\epsilon$ Model The first metric investigated was the convergence of the solution. The residuals for pressure and x-component of velocity were plotted and compared to one another. As expected, the coarser grids converged faster than the grids with more near wall clustering. Figure 5.2 shows one of the typical convergence histories of the wall function cases. All of the six grids followed the same trend and were set to identical convergence tolerances. This shows clearly that all cases reach a converged solution rather quickly - within 1000 time steps. Additionally, the total skin friction (C_F) over the plate was monitored as the solution progressed, and reached a stable value around the same point the solution converged and ceased iterating. The total skin friction coefficient was then compared to the empirical calculations of White [47]. The values for total skin friction along with a comparison to theory can be seen in Table 5.3. This type of verification indicates that a near wall spacing between 100 and 150 is the best range to accurately model the skin friction of a flat plate. White's equation for total skin friction has an error of $\pm$ 2%. These residuals and total drag coefficient combined prove each case reaches a converged solution.

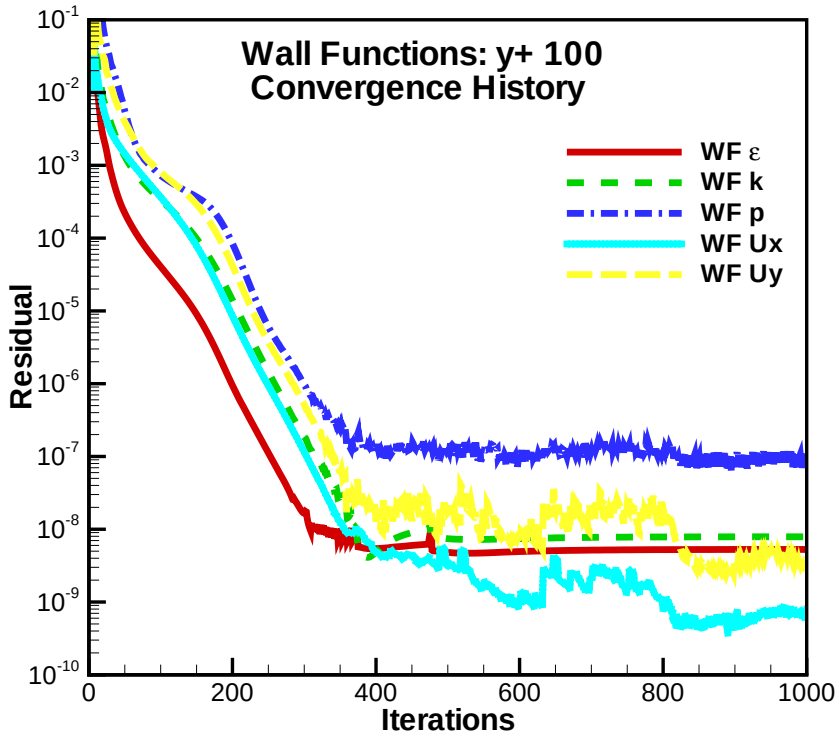

Figure 5.2. Residuals for one of the six wall function cases.

Another metric to gage the accuracy of each case is to compare the local skin friction coef-

Table 5.3. Wall Function Grids: Total Drag Comparison

White 3.056×10^{-3}	y^+ 25	y^+ 50	y^+ 100	y^+ 150	y^+ 200	y^+ 500
C_F $(\times10^3)$	2.982	2.957	3.036	3.030	3.021	2.940
% Error	2.42	3.24	0.64	0.85	1.15	3.79

ficient, C_f. Figure 5.3 shows how the skin friction varies over the flat plate for each of the wall function cases and how they relate to the empirical formulas from White. The results show that all the cases lie between the expected results from White. The y^+ 25 and the y^+ 500 have more pronounced fluctuations at the beginning of the flat plate compared to the other models. The y^+25 grid is most likely too close to the laminar sublayer region for wall functions, which require a first cell in the log-layer, and the y^+500 grid is too coarse for truly accurate results. These results, along with the accuracy of the total skin friction coefficient, indicate that OF performs more accurately with wall functions ranging between y^+ values of 50 and 200.

For completeness, the boundary layer profiles of each wall function case were also studied. Figure 5.4 shows there is no appreciable difference in solution from the range of y^+ values studied using wall functions other than the y^+ 500 grid, which shows its coarseness as the velocity profile approaches the wall.

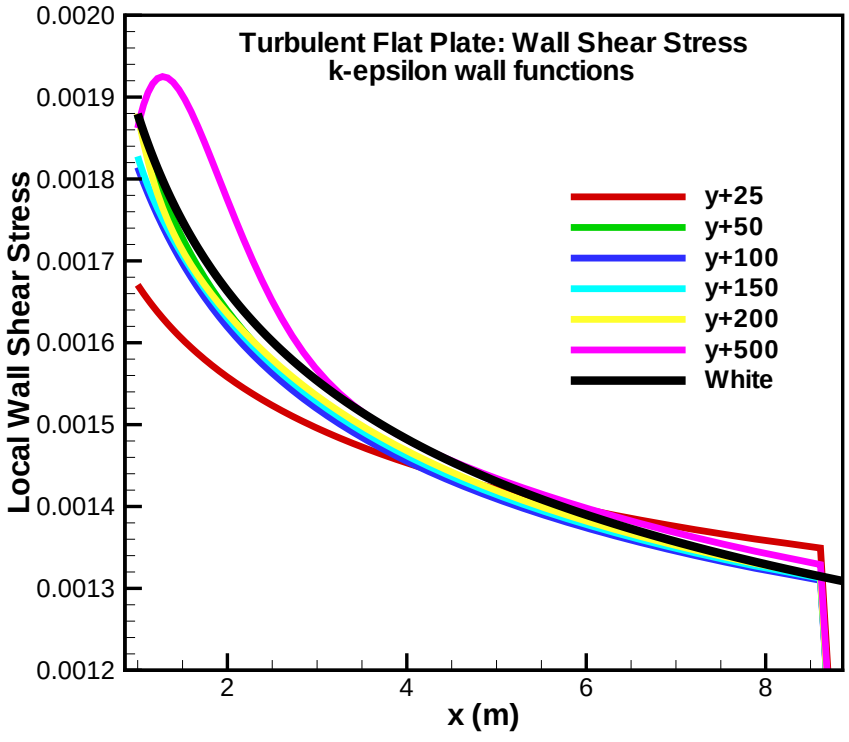

Figure 5.3. Local skin friction coefficient over the flat plate compared to empirical formula of White.

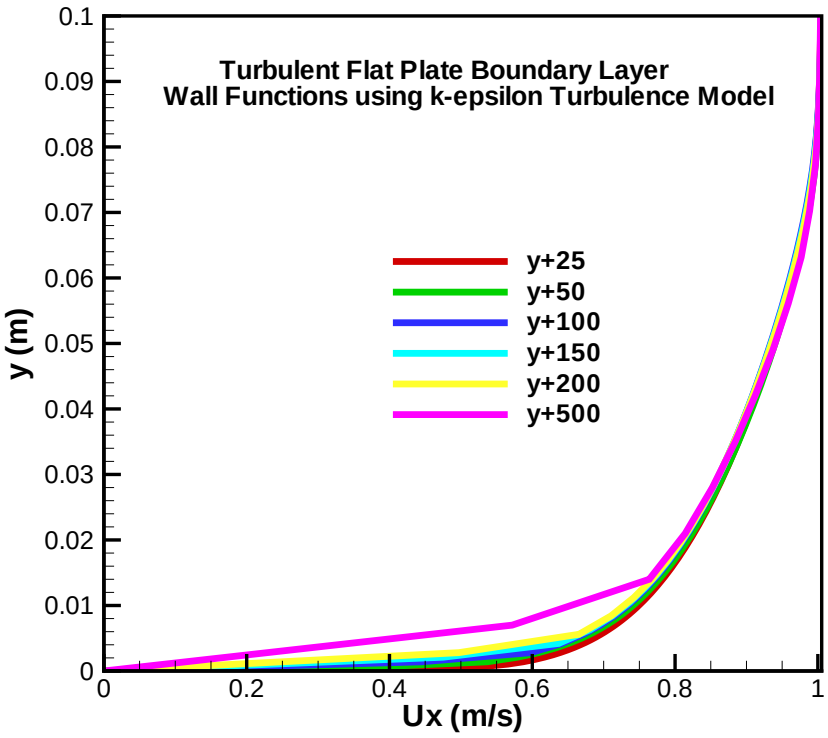

Figure 5.4. Boundary layer profiles of each of the wall function grids.

In terms of convergence time, the coarse grid converged the fastest, followed by the medium grid, and the fine grid required the longest computational time. The summary and exact totals of each grid's solution time is shown in Table 5.4. It can be expected that as grids using wall functions increase in complexity over the flat plate case, the overall computational time will naturally increase. However, the trend demonstrated in these cases of smaller dimensionless near wall spacing values resulting in larger computational times is also to be expected. Regardless of overall solution time, the y^+ 100 grid produced the best results in terms of total drag coefficient as well as local skin friction coefficient. The computational results shown that to produce the most accurate results without sacrificing CPU time with wall functions, a grid spacing of $100 < y^+ < 200$ is appropriate.

Table 5.4. Wall Function Cases: Solution Times

Grid	CPU Time (s)
y^+ 25	110.1 s
y^+ 100	42.5 s
y^+ 500	21.1 s

5.1.1.4.2 Low-Re Turbulence Models After validating the wall function turbulence models, the next step is to compare the various sublayer-resolved turbulence models provided by OpenFOAM. The four low-Re turbulence models under investigation are the LienCubicKELowRe (LC), LienLeschzinerLowRe (LL), LaunderSharmaKE (LS), and SpalartAllmaras (SA). While investigating the different low-Re turbulence models, it was found that the Spalart-Allmaras model that was included with the standard OpenFOAM release inaccurately predicted boundary layer profiles and coefficients of drag by as much as 20%. The SA version included in OF version 1.4.1 contains an additional term f_{v3} that is multiplied by the source term. This extra f_{v3} term tends to delay boundary-layer separated compared to the original SA model at lower Reynolds numbers between 1 and 10 million [48]. The differences between Spalart and Allmaras' original implementation and the f_{v3} model can be seen below:
Original SA model:

$$\hat{S} = \Omega + \frac{\hat{\nu} f_{v2}}{\kappa^2 d^2} \tag{5.6}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{5.7}$$

OF SA+f_{v3} implementation:

$$\hat{S} = f_{v3}\Omega + \frac{\hat{\nu} f_{v2}}{\kappa^2 d^2} \tag{5.8}$$

$$f_{v2} = \frac{1}{(1 + \chi / C_{v2})^3} \tag{5.9}$$

$$f_{v3} = \frac{(1 + \chi f_{v1})(1 - f_{v2})}{\chi} \tag{5.10}$$

After failing to obtain significant accuracy with the Spalart-Allmaras turbulence model, the code was reprogrammed without the f_{v3} term. This lead to a drastic improvement in accuracy and the results are shown in Figure 5.5. The percent error of the SA turbulence model from White's total C_f equation dropped from 17.4% with the f_{v3} term to 0.34% without it. Throughout the rest of the paper, when referring to the SA turbulence model, the SA model without the f_{v3} term is intended.

For the actual low-Re overall turbulence model study, a grid with a near wall spacing of $y^+ = 0.1$ was used to ensure sublayer resolution was used. Each case was run under the same parameters (k and ϵ) as the wall-functions case, with the exception of the SA model that requires a specified value for $\tilde{\nu}$. Figure 5.6 shows how the boundary layer profiles for each turbulence model vary at a point 5 m from the leading edge of the flat plate. The SA model almost directly matches the theoretical values of the law of the wall and the log region. While the LS model is fairly accurate, the LL model severely over predicts the boundary layer. The LC model gives completely inaccurate results, as it indicates a complete laminar boundary layer profile, $u^+ = y^+$, well past the expected range of approximately $u^+ = 10$.

Just as in the wall-functions study, the low-Re turbulence models were tested for total (Fig. 5.8) and local drag (Fig. 5.10) coefficients. The results show that the SA model is superior in both the local and total skin friction coefficient values, with the LS model performing adequately,

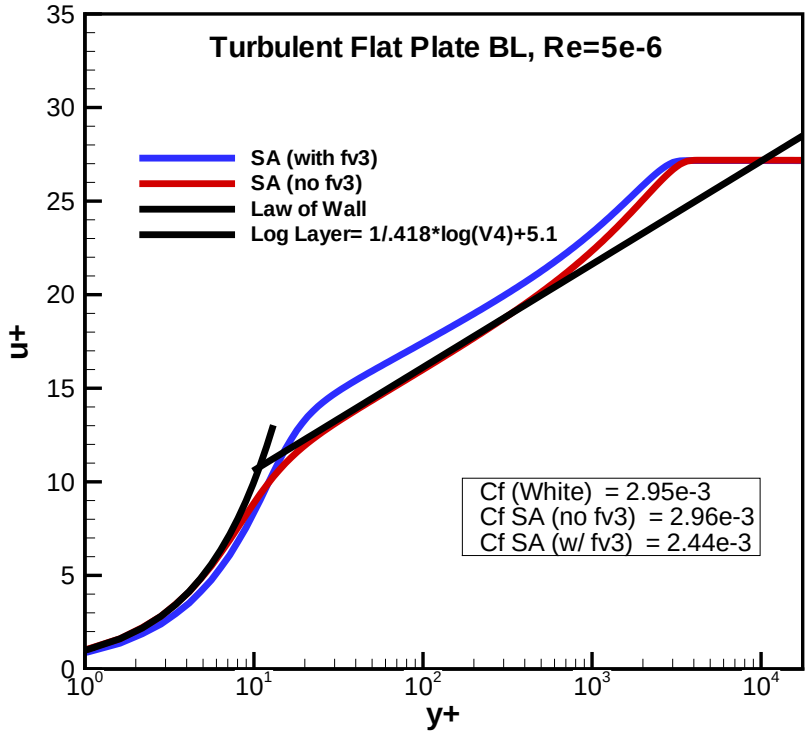

Figure 5.5. SA turbulence model comparison with and without the f_{v3} contribution.

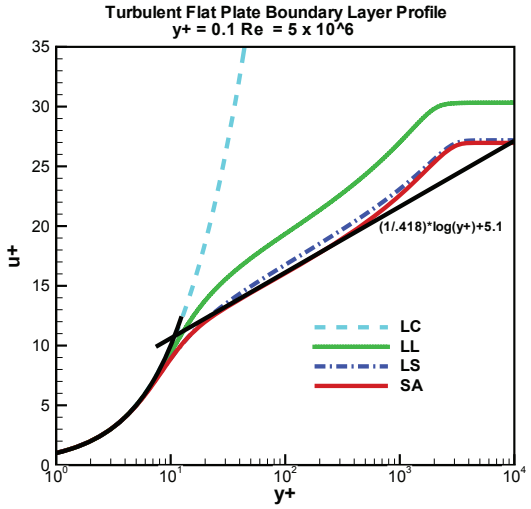

Figure 5.6. Dimensionless velocity plotted against the dimensionless near wall spacing for $y^+ = 0.1$.

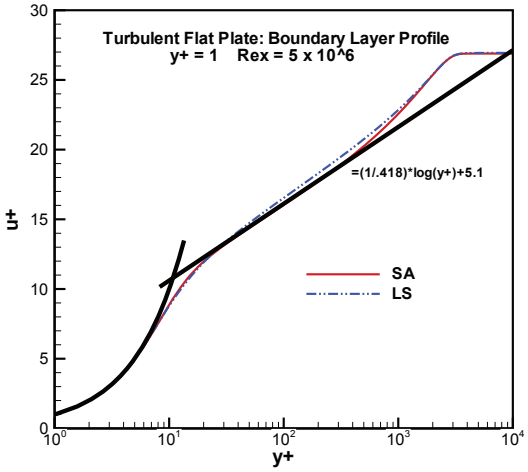

Figure 5.7. Dimensionless boundary layer profile for the $y^+ = 1$ test case.

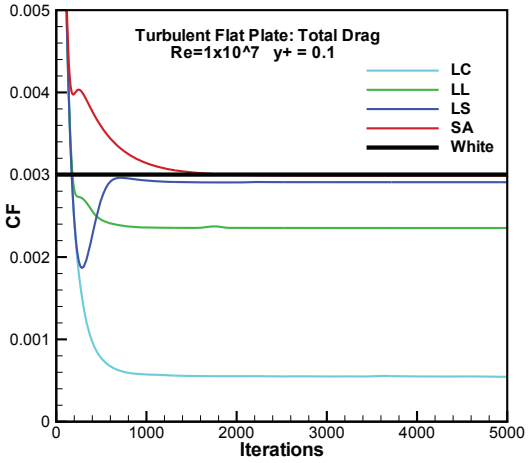

Figure 5.8. Total skin friction coefficient over the flat plate.

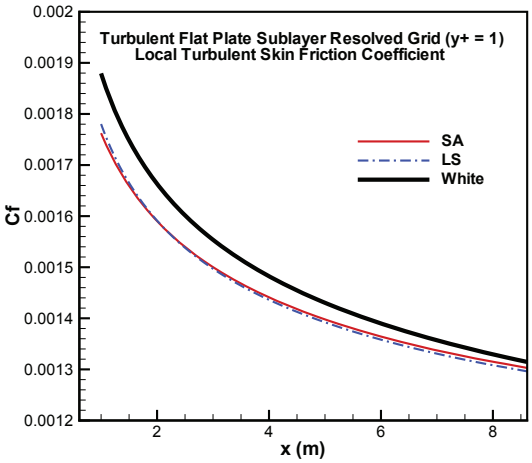

Figure 5.9. SA and LS models local skin friction coefficient over the flat plate. Both models under predict the expected results from White.

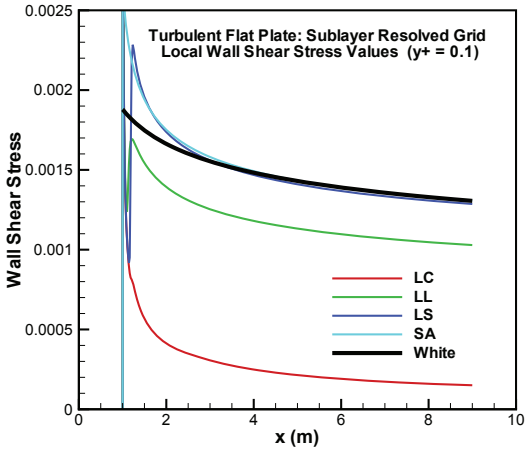

Figure 5.10. Local skin friction coefficient over the flat plate.

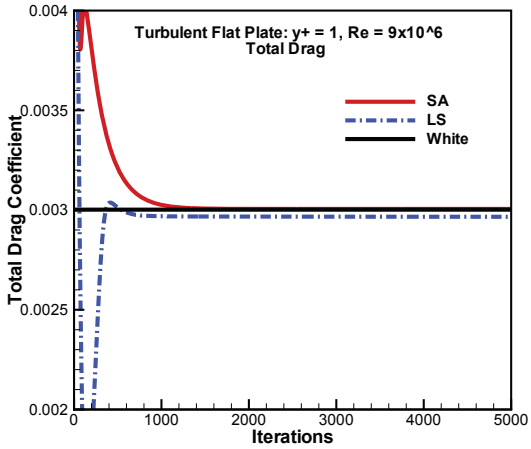

Figure 5.11. Each case converges faster than in Fig 5.8 and again the SA model is more accurate.

the LL model performing poorly, and the LC model giving false results. The percent error from White's equation for total drag are 0.42%, 3.08%, 21.63%, and 81.82%, respectively, for each model. It can be concluded the SA and LS models are the only turbulence models accurate enough to warrant further study at a higher y^+ value.

After validating the SA and LS models at an extremely low dimensionless wall value, the next few cases investigate SA versus LS turbulence models at a more reasonable y^+ value of 1. Solving a computation with a near wall spacing of $y^+ = 1$ rather than $y^+ = 0.1$ saves a significant amount of computation time when the number of cells becomes much larger on more complex grids. However, in both cases, OF should be able to obtain accurate results in the sublayer region of the boundary layer profile and this study will aim to confirm this. Again, the same three metrics are investigated, BL profiles, local wall shear stress, and total skin coefficient plots

for both cases.

Upon inspection of figures 5.7 - 5.9, the SA turbulence model outperforms the LS turbulence model for $y^+ = 1$. Furthermore, there is no significant change in results between the $y^+ = 1$ and the $y^+ = 0.1$ case. It can safely be concluded that, for future cases, an acceptable range of y^+ values for OF is between 0.1 and 1.

Table 5.5. Wall Functions and Sublayer Resolved Total Drag Comparison

White 3.00191×10^{-3}	WF y^+100	WF y^+150	SA y^+0.1	SA y^+1	LS y^+0.1	LS y^+1
C_F (×10^{-3})	3.036	3.030	2.989	3.005	2.909	2.966
% Error	1.14	0.94	0.42	0.12	3.08	1.19

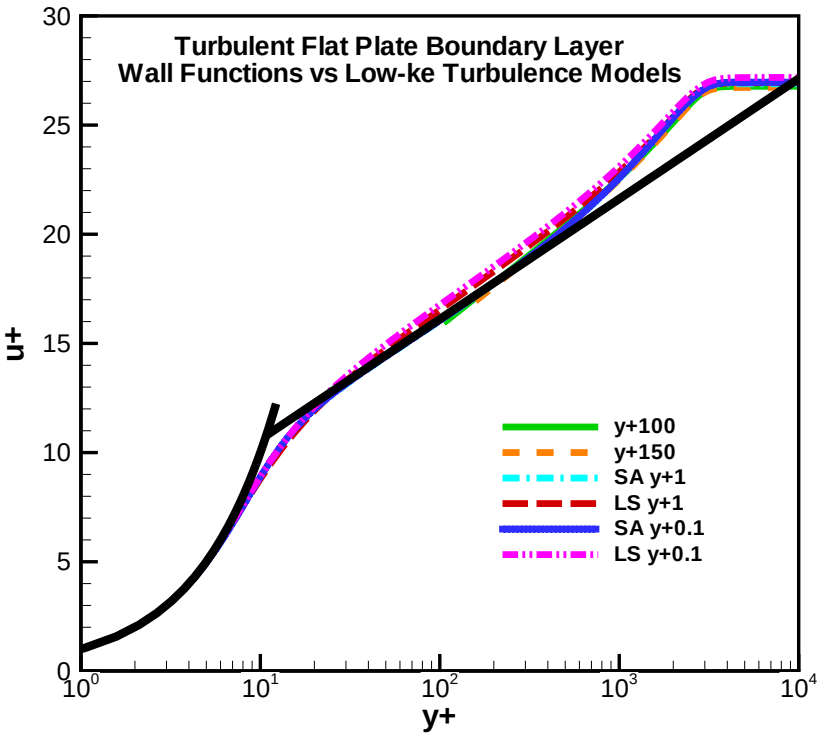

Figure 5.12. Dimensionless boundary layer profile of the best performing turbulence models.

With comparable boundary layer profiles (Figure 5.12) and similar results for local wall shear stress along the flat plate, a definitive measurement of computational accuracy between the different cases can be found in the difference from the expected total drag and actual total drag. Table 5.5 indicates that for the sublayer resolved grids, a near wall spacing of $y^+ = 1$ provides more accurate results than the $y^+ = 0.1$ grids. Additionally, the wall-functions grids performed almost identically, with the $y^+ = 150$ case having a slightly lower percent error of 0.94

% compared to 1.14 % with a near wall spacing of 100. Regardless, when accounting for the ± 2% margin of error inherent in the equation from White , all of these models performed within an acceptable accuracy range. Future cases can be run with confidence that these models can accurately predict the flow field characteristics.

5.1.2 Hemispherical HeadForm

Using the results from the flat plate study, the next test case to be investigated is the hemispherical headform. This simple test case has been studied in depth over the years, and has excellent experimental data with which to compare [2]. The hemispherical headform will be studied in two different cases, single phase for now and multiphase later using the cavitation model. This first investigation into the single phase performance of OF is twofold:

- To study the effectiveness of the 2D axisymmetric boundary condition "wedge" and compare it to the full 3D case.
- To asses the accuracy of the solver to match experimental coefficient of pressure (C_p) values.

5.1.2.1 Validation Data and Flow Parameters

An inlet velocity was set to 8.27 m/s to yield a Reynolds number of $Re = 2.1 \times 10^5$. The turbulence variables k and ϵ were set to 0.01025 m^2s^{-2} and 0.0672 m^2s^{-3} respectively. For the SA turbulence model cases, the value of $\tilde{\nu}$ was set to $5 \times 10^{-7} m^2/s$. The pressure field was set to a fixed value of zero at the exit, and zero gradient on the inlet and walls. Although Rouse and McNown performed extensive experimental studies on a wide variety of headform shapes (ogival, blunt, conical, and ellipsoidal), the hemispherical headform was chosen for comparison to the data as well as other computational results [13, 15].

5.1.2.2 Grid Generation

Two grid styles were used in this study - a wedge headform and a full headform. Both cases were solved using the $k-\epsilon$ turbulence model and were designed with a $y^+ = 125$. The wedge case was designed with 74,900 cells whereas the full hemispherical headform contained ten times as many cells, 749,000. The wedge boundary condition in OF is used for 2 dimensional axi-symmetric cases with a small wedge angle (typically 5° or less) and contain a thickness of 1 cell. A sublayer resolved wedge headform was also generated for comparison to the wall function grid. This type of comparison may have implications in terms of solution accuracy when solving for two phase flow later on using the cavitation solvers. The Spalart-Allmaras turbulence model grid contained 89,880 cells. The plane of symmetry must also lie along one of the coordinate planes. Eventually, the other periodic boundary condition, cyclic, can be used in modeling more advanced cases, such as propellers. The cyclic boundary condition treats two patches as if they are physically connected and does not require they be of the same orientation, just that the areas are the same.

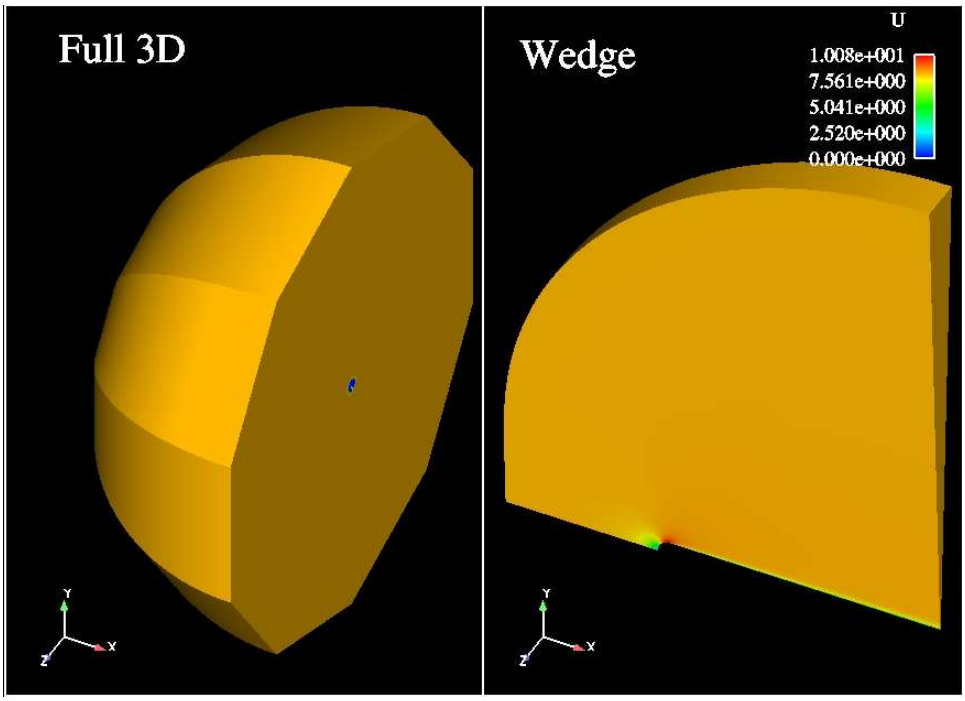

Figure 5.13. Varying grid geometries for both the wedge headform and the full 3D headform.

5.1.2.3 Numerical Parameters

These initial tests for the hemispherical headform were run using the same finite volume schemes listed in the flat plate section. The steady-state solver simpleFoam was used as the baseline for the headform results and is compared to the unsteady turbulent solver, rasInterFoam. To ensure a converged solution, the solution variables in each case were set to a solution tolerance of 1×10^{-8}. Figure 5.14 shows an example of the convergence history of one case.

These headform benchmark cases do not reach lower pressure convergence levels, as they did in the flat plate case, is because their relative tolerance values are set to 0.1, rather than 0. Solver relative tolerance sets the required reduction in the residuals within each iteration. The lower the relative tolerance, the more explicit terms need to be updated as part of the segregated iterative procedure. Lowering the relative tolerance would have in turn lowered the final residuals, but at a computational cost. The flat plate cases maintained a relative tolerance of 0, essentially forcing the pressure iterations to converge to the specified tolerance level.

5.1.2.4 Simulation Results

The first test for this headform case is to determine if OF handles periodic boundary conditions properly. In order to test the effectiveness of such boundary conditions, a wedge (5°) shaped headform was compared to a full 3-D headform geometry.

After both cases reached convergence (Figure 5.14), the first step in confirming the accuracy of the wedge boundary conditions was comparing the two cases side by side. Figure 5.15 shows

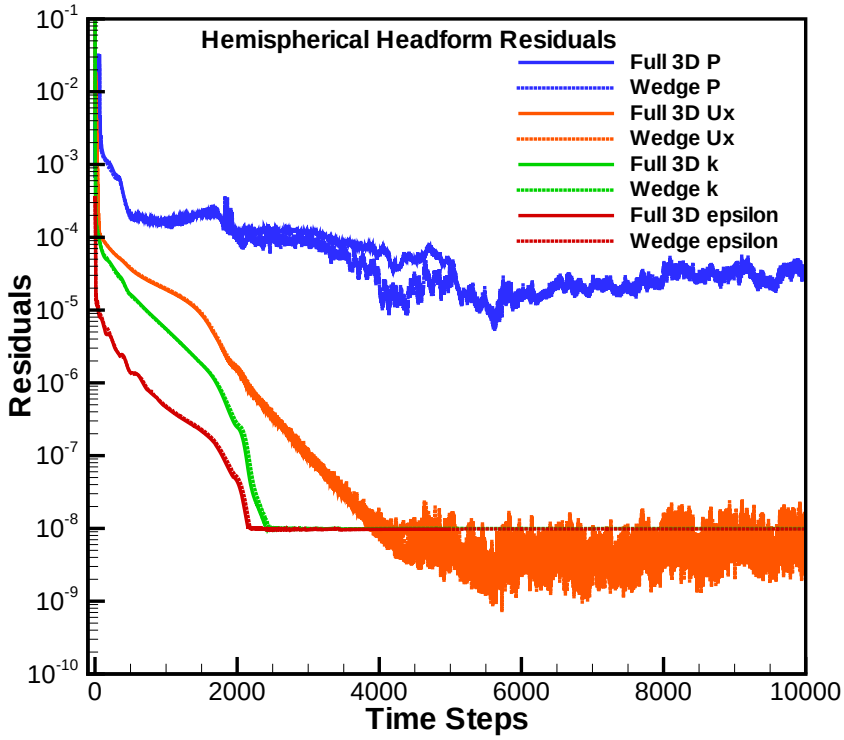

Figure 5.14. Convergence of several flow field properties.

the velocity field of both cases which are virtually indistinguishable. Additionally, the turbulence properties, k and ϵ along with the pressure field, were equally as matched for both cases. This indicates that the wedge/periodic boundary conditions provide accurate results at a significantly cheaper computational cost than the full 3D case.

5.1.2.4.1 Coefficient of Pressure The next step in confirming the accuracy of the solutions was to compare the results to experimental data. Rouse and McNown [2] published the definitive test case data for headforms in 1945 which will be used to compare the single, and later on, multi-phase C_p results. For this section, the wedge case will be additionally tested with the Spalart-Allmaras model. That grid will be set to a near wall spacing of $y^+ = 1$ and contain 89,880 cells.

Figure 5.17 shows that the wedge headform performs closer to the experimental results of Rouse and McNown. The results show that the 3D case is less accurate in predicting the minimum pressure over the hemispherical headform. Both the SA model and the $k - \epsilon$ wedge cases do a decent job of accurately predicting the pressure over the headform without using as much computational resources as the full 3D case.

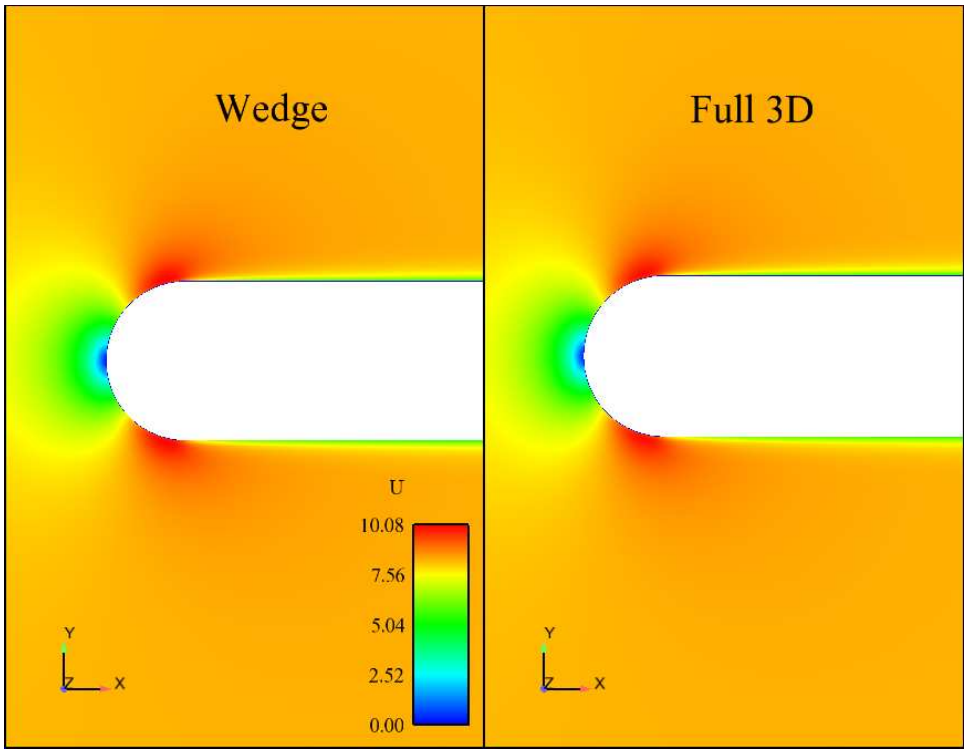

Figure 5.15. Velocity field comparison of both cases.

Figure 5.18 shows coefficient of pressure results from the "rasInterFoam" class of solvers which is generally used for unsteady, multiphase cases. This solver was used and compared to the steady state solvers is that the code in rasInterFoam is the basis of the cavitation solvers. The accuracy of this solver at predicting the flow field, in this case pressure distribution over the headform, will increase confidence in the solution accuracy of the cavitation solvers. Again, the main difference between this unsteady solver and the steady state solver is the solution algorithm that are implemented. The steady state solver relies on the SIMPLE algorithm with relaxation factors while the unsteady solver relies on the PISO algorithm with a Courant number.

This graph also shows results from testing of the $k-\omega$ turbulence model, as well as the Spalart-Allmaras turbulence model. The SST $k-\omega$ turbulence model is of interest because it provides a nice blend of wall functions along with sublayer resolution. This two-equation model uses the $k-\omega$ formulation in the inner parts of the boundary layer, acting like a low-Re turbulence model without the addition of any damping functions. The model then has the ability to switch over to a $k-\epsilon$ behavior in the free-stream, acting like a wall-function model. All of the cases run and shown in Figure 5.18 closely match the experimental data. Although these results show that the $k-\omega$ turbulence model performs accurately and is well suited for more complex geometries, for these simple benchmark cases, the extra near wall resolution is not needed. Therefore, future cases will continue to utilize the $k-\epsilon$ turbulence model.

As an additional check to verify that both the wedge geometry and the full 3D grid pro-

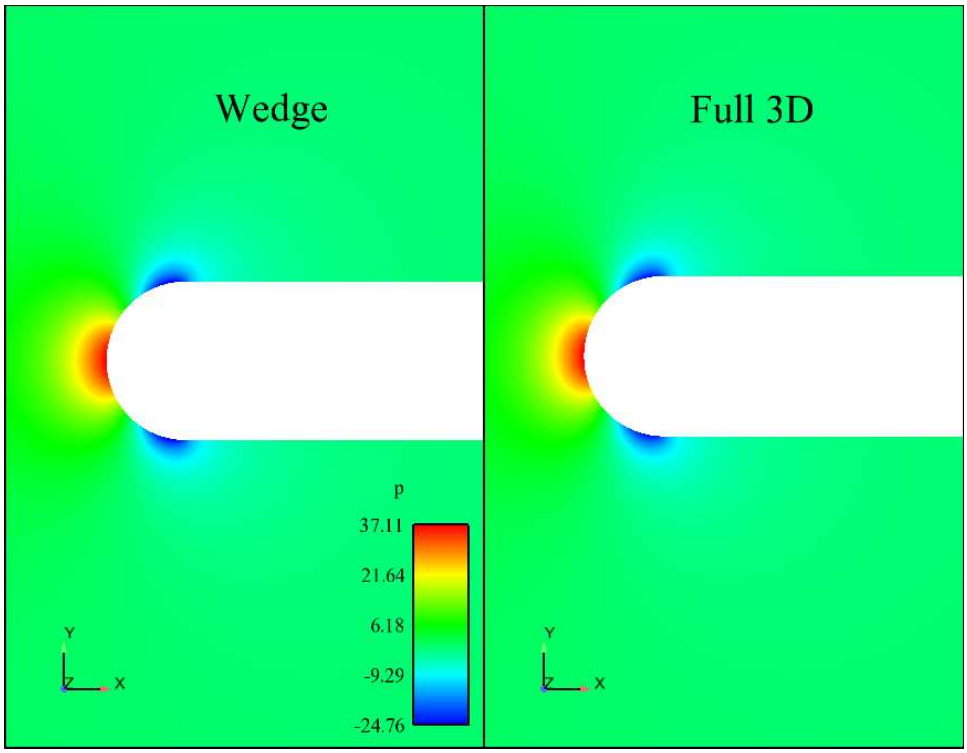

Figure 5.16. Pressure distribution over both cases.

duced the same results, the x component of velocity 5 radii downstream of the headform was investigated. Figure 5.19 shows how the two cases share a similar boundary layer profile at 5r downstream.

5.1.3 NACA0012 Airfoil

A simple and extensively documented airfoil, the NACA0012, provided an opportunity to compare single phase OpenFOAM results with experimental data. A small grid study was performed on a coarse, medium, and fine grid with both wall functions and sublayer resolution. The results were compared to the theoretical drag coefficients as well as the expected coefficient of pressures. In addition, lift and drag curves were generated for multiple angles of attack and compared to experimental data.

5.1.3.1 Validation Data and Flow Parameters

The first study on this airfoil was to determine which effect grid resolution over the suction and pressure side of the foil had in determining the final flow field results. In order to do this, three grids of increasing cell counts were created and tested for both the $k-\epsilon$ and Spalart-Allmaras turbulence models. In addition to the grid study, starting from zero angle of attack, the NACA0012 airfoil was tested for lift and drag coefficients up to 28 degrees angle of attack.

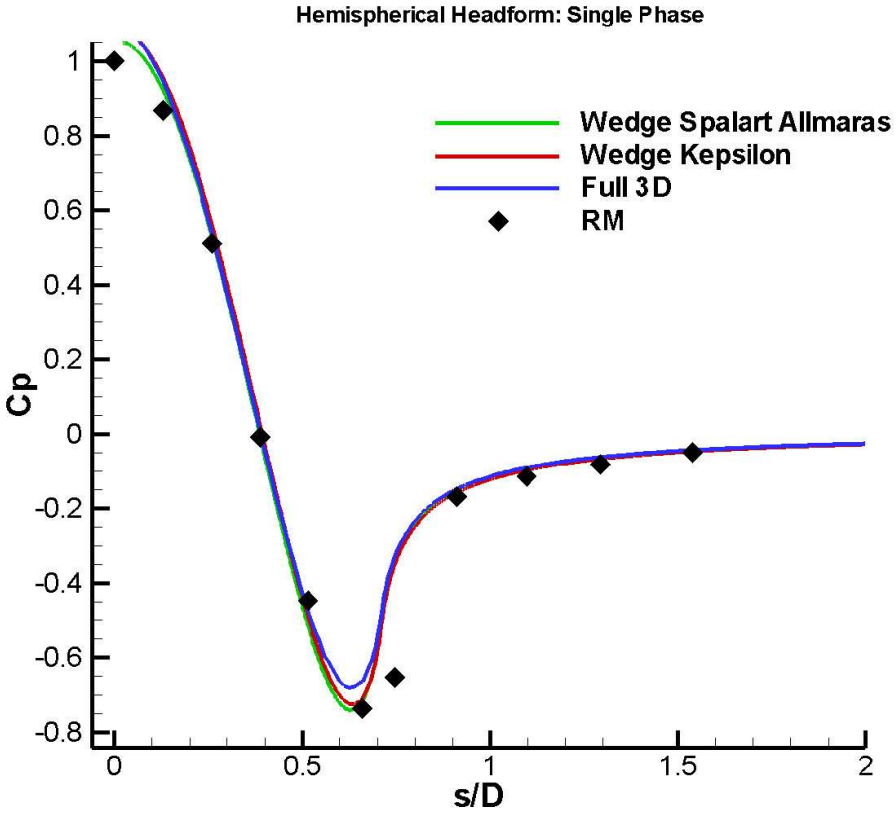

Figure 5.17. C_p values for the full 3D case ($k-\epsilon$) and the wedge versions of the SA and $k-\epsilon$ models. 's' represents the distance along the surface of the headform, and 'D' is the diameter.

The results were compared to an original NACA document published by Jacobs *et al.* [49] in 1933. These were the data later used in Theory of Wing Sections [3] in 1959. Additionally, the C_p values were compared to experimental data from a 1950 NACA technical report [50]. Two cases were set up based upon the previous test case results: one for the Spalart-Allmaras model with $y^+ = 1$ and the other a $k-\epsilon$ model with $y^+ = 100$. The initial conditions for both cases can be seen in Table 5.6. For the cases with angles of attack greater than zero, the x and y components of velocity were adjusted accordingly.

Table 5.6. NACA0012 Flow Parameters

Parameter	Value
U_{inf}	1 m/s
L	1 m
Re	9×10^6
k	0.00015 m^2/s^2
ϵ	3.02×10^{-5} m^2/s^3
$\tilde{\nu}$	3.1×10^{-8} m^2/s

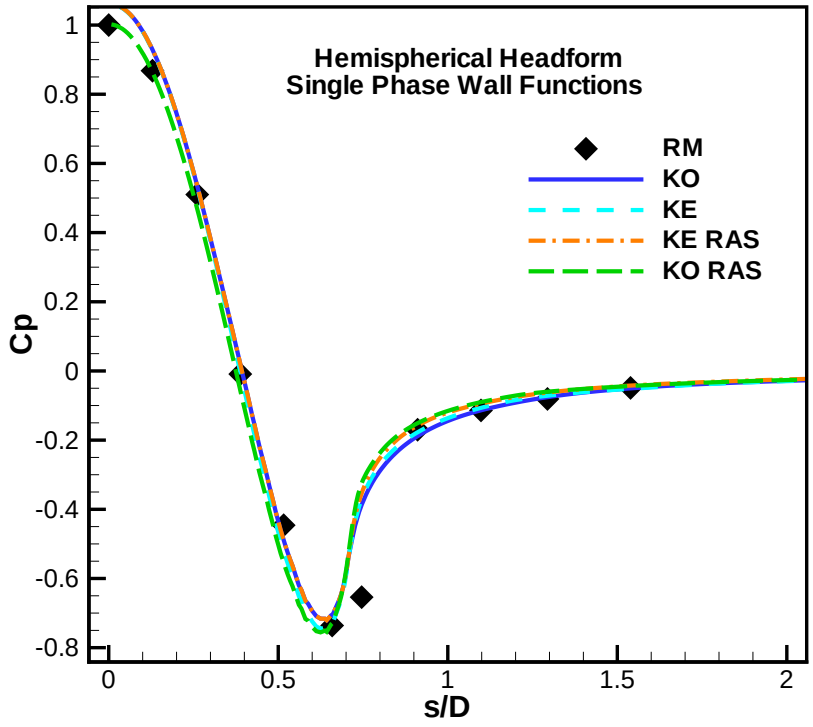

Figure 5.18. Hemispherical headform Cp results for both the $k-\epsilon$ and $k-\omega$ turbulence models. "RAS" indicates the unsteady solver.

5.1.3.2 Grid Generation

Eighteen different grids were generated for the NACA0012 cases. There were three different wall-function grids for each type of resolution (coarse, medium, and fine) from y^+ values of 100, 250, and 500. In a similar fashion, three different sublayer resolved grids were created with y^+ values of 0.5, 1, and 2; again each coarse, medium, and fine grid had each of these y^+ values included. The coarse grid contains 25 cells on both the pressure and suction sides of the airfoil, the medium grid contains 50, and the fine grid has 100 cells on the pressure side and suction sides. Table 5.7 summarizes the cell counts for each of the cases run. For the independent lift and drag cases, the sublayer resolved grid contained 24,354 cells while the wall functions grid had 14,652 cells.

5.1.3.3 Numerical Parameters

For the initial cases run on the NACA0012, the same numerical schemes were used as in the previous two test cases. However, in order to increase accuracy on the lift and drag results, an attempt was made to switch all the divergence terms to second-order accurate schemes instead

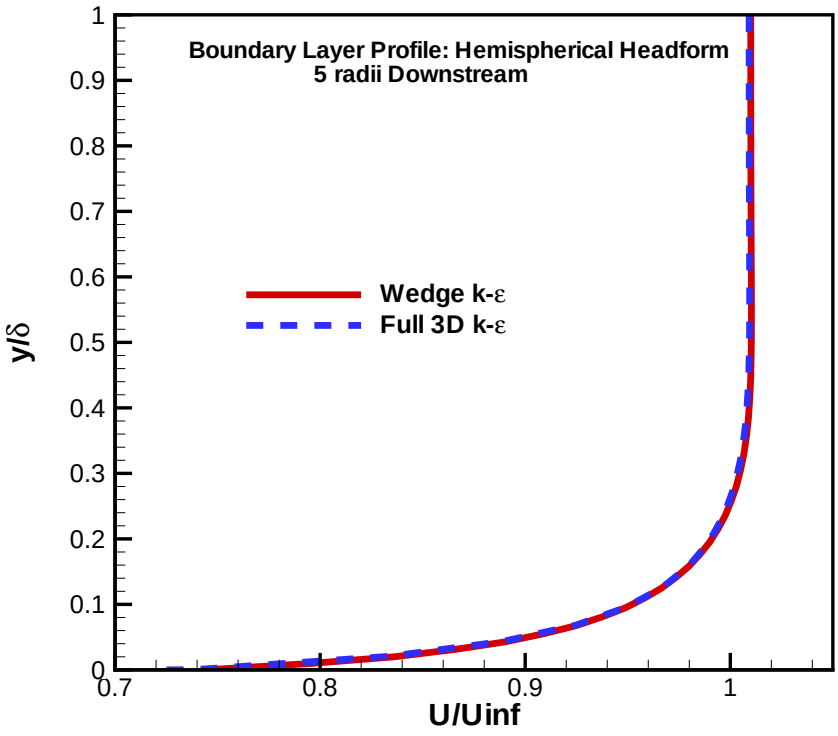

Figure 5.19. Hemispherical headform velocity profile 5 radii downstream.

of the mixed upwind and central differencing methods. The relative tolerance on the solvers for each variable was set to 0.1 with a tolerance value of 1×10^{-8}. This leads to the same limiting convergence criteria as the headform, but is more economical than setting the relative tolerance to zero. The relaxation factors varied from case to case, but stayed between 0.5 and 0.8 for all variables, with the exception of pressure, which was held at 0.3.

5.1.3.4 Simulation Results

Several computational results were analyzed for each grid resolution for all y^+ ranges: coefficient of pressure, total drag coefficient, and residuals for pressure. The total drag coefficient was also used as a tool for gaging convergence as once this value stabilizes, this is a good indication that the flow characteristics are not changing with time. Next, selected y^+ values were chosen for the coarse, medium, and fine grids and compared directly to one another. Lift and drag at various angles of attack for sublayer-resolved and a wall function grid were then compared to experimental data to determine what level of near wall spacing is required to accurately obtain lift and drag values.

Table 5.7. NACA0012 Single Phase Grids

Resolution	y^+ Range	Cells
Coarse	0.5	6,240
	1.0	5,904
	2.0	5,520
	100	3,552
	250	3,120
	500	2,736
Medium	0.5	12,740
	1.0	12,054
	2.0	11,270
	100	7,252
	250	6,370
	500	5,586
Fine	0.5	25,740
	1.0	24,354
	2.0	22,770
	100	14,652
	250	12,870
	500	11,286

5.1.3.4.1 Coarse Grid The coarsest grid, as expected, converged the fastest, but lacked the accuracy of the higher resolution grids. For example, the total drag over the airfoil was over predicted for each case (roughly 14% for the wall function grids and up to 39% for the $y^+ = 0.5$ grid). This can be seen in Figure 5.20.

The coefficient of pressure resemble the profile from experiments [50], but lack of grid resolution near the leading edge resulted in under-prediction of the maximum negative pressure coefficient. This error led to further miscalculation downstream, as the trailing edge pressure was slightly lower than that of the experimental data (Figure 5.21).

The pressure residuals, along with the convergence of total drag, show how each case reached convergence after their respective iterations. The wall functions grid converged faster than the SA grids, although the extreme grid spacing of the $y^+ = 500$ grid resulted in lack of acceptable convergence.

5.1.3.4.2 Medium Grid By doubling the number of cells around the airfoil, the medium grids produced noticeable improvement in results over the coarse grids. First, Figure 5.23 shows the total drag coefficient to be much closer to the theoretical value of 0.003053 for the sublayer resolved cases. The error for those cases was less than 5%. The wall function cases were comparable in accuracy to the coarse grid, with a percent error from theory in the 14% range.

In a similar fashion to the total drag coefficient, the coefficient of pressure showed marked improvement in correlating with experimental data, as shown in Figure 5.24. The minimum

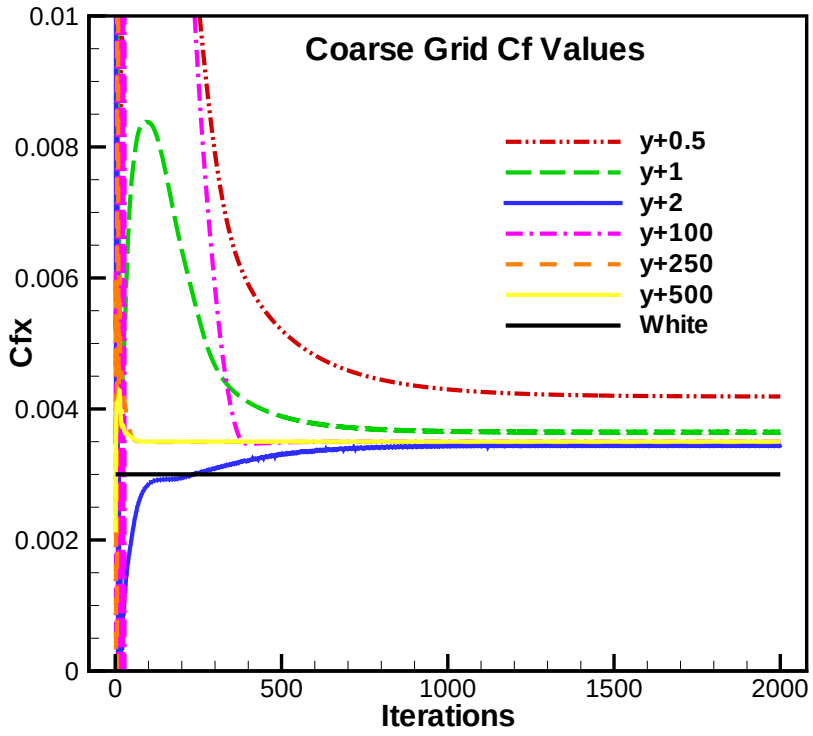

Figure 5.20. Comparison of total skin friction to theory over the hydrofoil for the coarse grid study.

pressure was closer to the data than the coarse grid, and for all ranges of y^+ the medium grids were comparable in accurately resolving the pressure field. No noticeable difference was observed in terms of pressure coefficient for the different near wall spacings.

The pressure residuals (Figure 5.25) show the same results as the coarse grid cases did: the wall function grids converge faster and to lower values than do the sublayer-resolved grids. Again, although the pressure field for the $y^+ = 500$ grid is consistent with the other simulations, the pressure residuals fail to converge to the same levels.

5.1.3.4.3 Fine Grid Total drag coefficient results for the fine cases show the same results produced by that of the medium grids - the sublayer-resolved grids produced more accurate results than the wall function model grids. Additionally, the percent error from theory was around 15% for the wall function grids and approximately 5% for the sublayer-resolved grids. These results in figures 5.23 and 5.26 indicate that increasing the grid resolution two-fold from the medium grids had no effect on solution accuracy in terms of total drag.

Although the total drag showed no change from the medium grid resolution, the pressure field was slightly improved over the coarser grids. All ranges of y^+ values produced results closer to

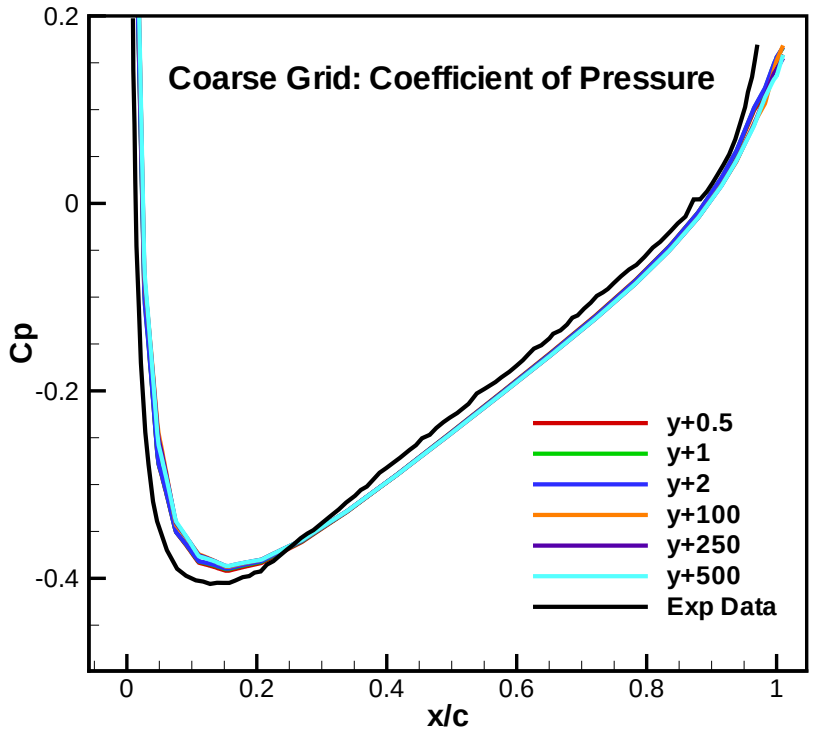

Figure 5.21. Comparison of C_p to experimental data over the hydrofoil for the coarse grid study.

experimental data. The results for each grid are are virtually indistinguishable in Figure 5.27.

With the additional axial grid points, the wall resolved grids were able to converge faster than the previous coarser grids. The wall modeled grids, however, actually took longer to converge than the sublayer-resolved grids. In fact, only the $y^+ = 100$ grid was able to reach a reasonable residual value.

5.1.3.4.4 Comparison of Grids With the exception of the coarse grids, the sublayer results provided more accurate results than the wall-function models (Figure 5.29). This type of result was expected, but also came at a computational price.

The next step in validating the solver was to identify how much sublayer resolution was needed for each turbulence model. For the wall-resolved cases, the medium spacing of 250 was sufficient in resolving accurate properties, while the sublayer cases showed that a y^+ value of 1 was suitable to obtain decent results. These two wall models provide a nice mix of solution accuracy without sacrificing excessive computational time.

Figure 5.30 shows how the steady-state incompressible solver using the Spalart-Allmaras turbulence model is able to better predict the pressure field over the foil by successfully doubling

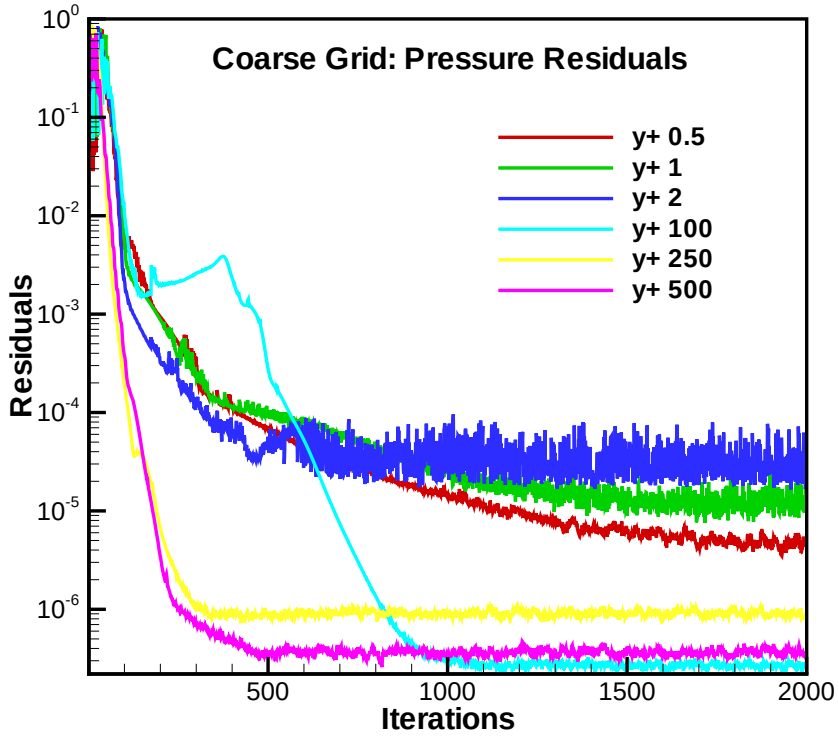

Figure 5.22. Convergence history of pressure residuals for each case using the coarse grid.

the grid points. In fact, no further resolution is needed for the foil, as the fine grid containing 100 cells on each side of the foil matched the experimental data. The same assessment applies for the wall function $y^+ = 250$ grid - the fine grid produced the best results out of the three.

5.1.3.4.5 Lift and Drag After validating the zero-degree NACA0012 cases for wall functions and sublayer-resolved grids, the next test was to verify the results with proven experimental data. Using the fine grids (200 total cells around the foil) for both the $k - \epsilon$ ($y^+ = 100$) and Spalart-Allmaras ($y^+ = 1$) turbulence models, various angles of attack were studied up to 28 degrees. For the first set of NACA0012 angle of attack simulations, each case was run using a first-order upwind divergence scheme. Once run to convergence, each case was rerun using a second-order divergence scheme. As expected, the results for drag (Figures 5.32 and 5.33) and lift (Figures 5.34 and 5.35) both show an improvement when switching to the second-order scheme. It should be noted that both cases were run using a steady-state solver, which has difficulties solving for the flow field at higher angles of attack once the flow has separated. A more ideal solver would be an unsteady solver, but the extra computational time was not warranted to obtain these simple test results.

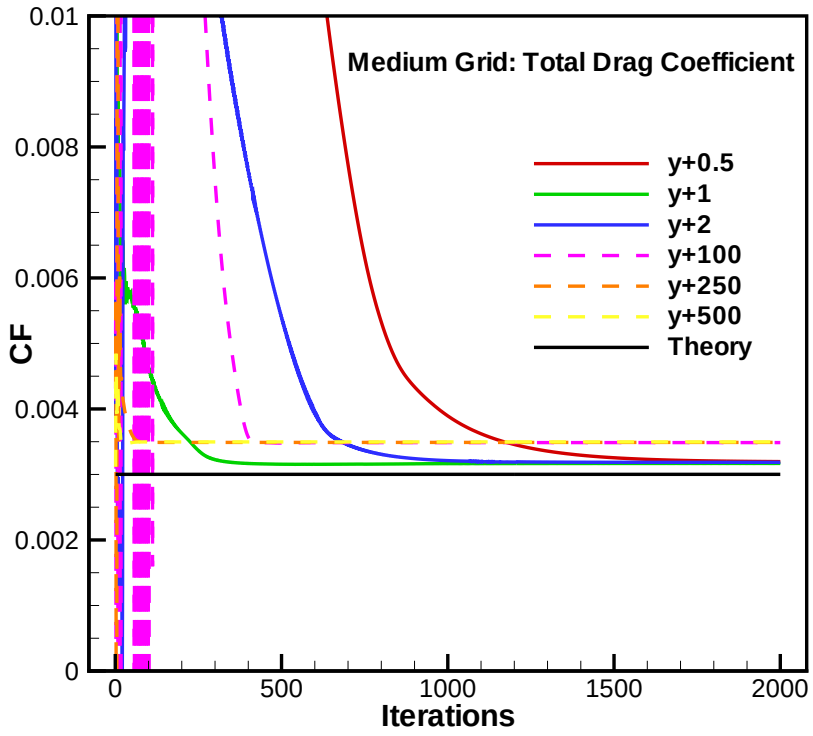

Figure 5.23. Comparison of total skin friction to theory over the hydrofoil for the medium grid study.

The biggest increase in accuracy when switching to the second-order divergence scheme was seen in the SA case. While the wall function grids with the $k-\epsilon$ model did have slight improvement, the SA case closely matched the peak lift data when using the higher order scheme. This is a promising result and will be useful for analyzing more complicated geometries in future cases.

5.1.4 NACA66(mod)

The NACA66(MOD) has been extensively documented [51] and will be used primarily to validate the cavitation solver. This section will be used to validate the single phase solutions used to start up the 2-phase computations.

5.1.4.1 Validation Data and Flow Parameters

The data used to compare the single phase solutions are compared to the original paper of Shen and Dimotakis [51], who originally set out to study the effect of surface cavitation on hydrodynamic forces. The experiment was carried out in the water tunnel of the California Institute of Technology in the late 80's. The hydrofoil was tested for lift, drag, and pressure distribution. Two different types of cavitation were studied: leading edge sheet cavitation and

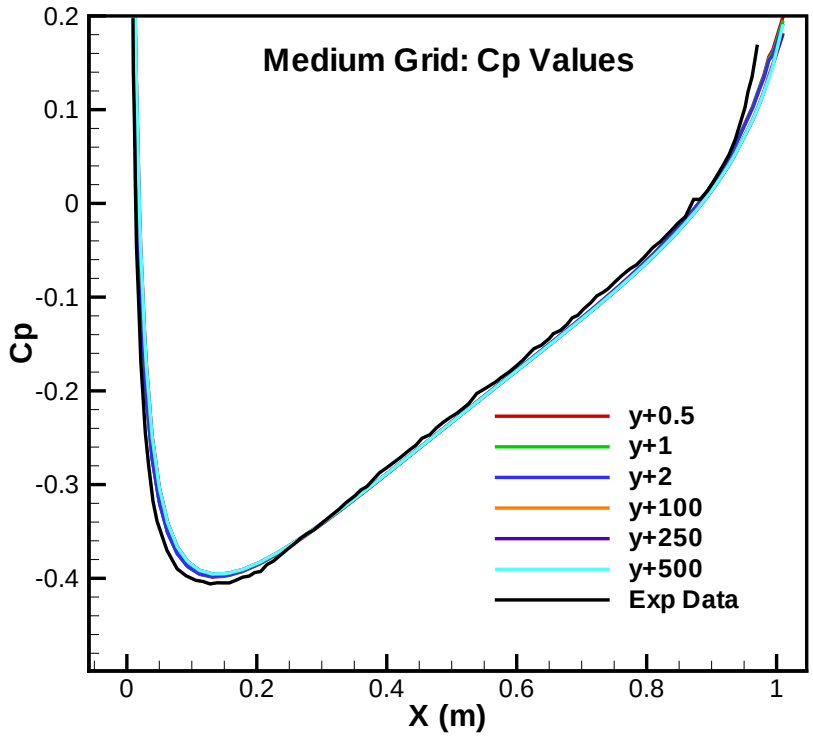

Figure 5.24. Comparison of C_p to experimental data over the hydrofoil for the medium grid study.

midchord cavitation. Each presents its own flow parameters and angle of attack. For these single phase cases, the leading edge fully whetted (single-phase) solution was run at 4° and 6°, while the midchord cavitation testing was performed at 1° angle of attack. The leading edge sheet cavitation cases were performed at Reynolds number based on chord of 2×10^6. Due to the lower angle of attack, the midchord cavitation cases were run at a slightly higher Reynolds number of 3×10^6. It should be noted that the cavity lengths and pressure coefficients taken in the experimental setup were the time-averaged values. The experimental model was fitted with 12 pressure taps on the suction side and one pressure tap on the pressure side. The taps were positioned at 3, 6, 11, 16, 26, 33, 45, 56, 68, 81, 90, and 95 percent of the chord from the leading edge on the suction side. The pressure side pressure tap was located at 9% of the chord. The percent error for the pressure readings was ±0.3%.

In order to match the experimental setup, the same experimental Reynolds numbers were used for the respective computational studies. The chord length was set to 1 m, the kinematic viscosity to $1 \times 10^{-6}\text{m}^2/\text{s}$, and the free stream velocity to 3 m/s for the 1° case and 2 m/s for the 4° and 6° cases. The single phase results were generated using the simpleFoam steady-state solver rather and an unsteady solver for efficiency in computational time. For this type of solver,

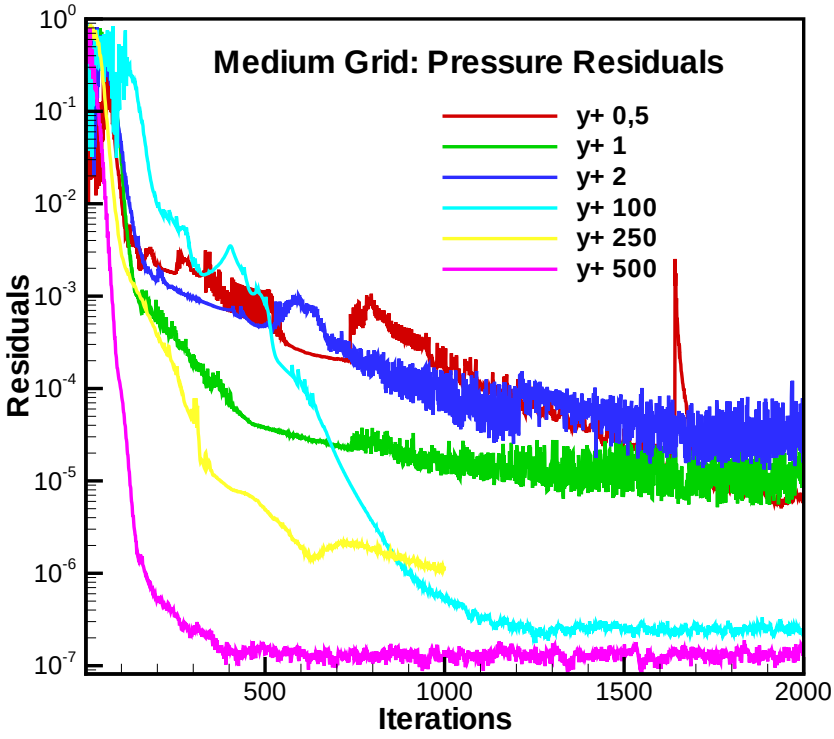

Figure 5.25. Convergence history of pressure residuals for each case using the medium grid.

the pressure field is computed per density; therefore the pressure field on the outlet was set to $101.325Pa$.

Although the $k-\epsilon$ turbulence model showed satisfactory results compared to the wall modeled Spalart-Allmaras model, both models were run on the NACA66(MOD) in single phase to ascertain if there was a marked difference between the two. Ideally, the wall function model would be used to save computational time if the solutions are comparable. The constants for free stream turbulent kinetic energy were set at 0.00135 m^2s^{-2} and 0.0006 $\mathrm{m^2s^{-2}}$ for the 1° and 4/6° cases respectively. The free stream turbulent dissipation was set at $4.0752e^{-5}$ $\mathrm{m^2s^{-3}}$ for the one degree case and $1.2075e^{-5}$ $\mathrm{m^2s^{-3}}$ for the other two. For the SA cases, $\tilde{\nu}$ was set to one half of the kinematic viscosity, or $5e^{-7}\mathrm{m^2}s$.

5.1.4.2 Grid Generation

As mentioned above, two different grids were used for this study - a sublayer-resolved grid and a wall functions grid. The wall function grid was designed for a y^+ value of around 55, whereas the other grid was designed with a much smaller near wall spacing that yielded a y^+ value around 0.55. Lower values were chosen to ensure the cavity was modeled properly in the near wall region.

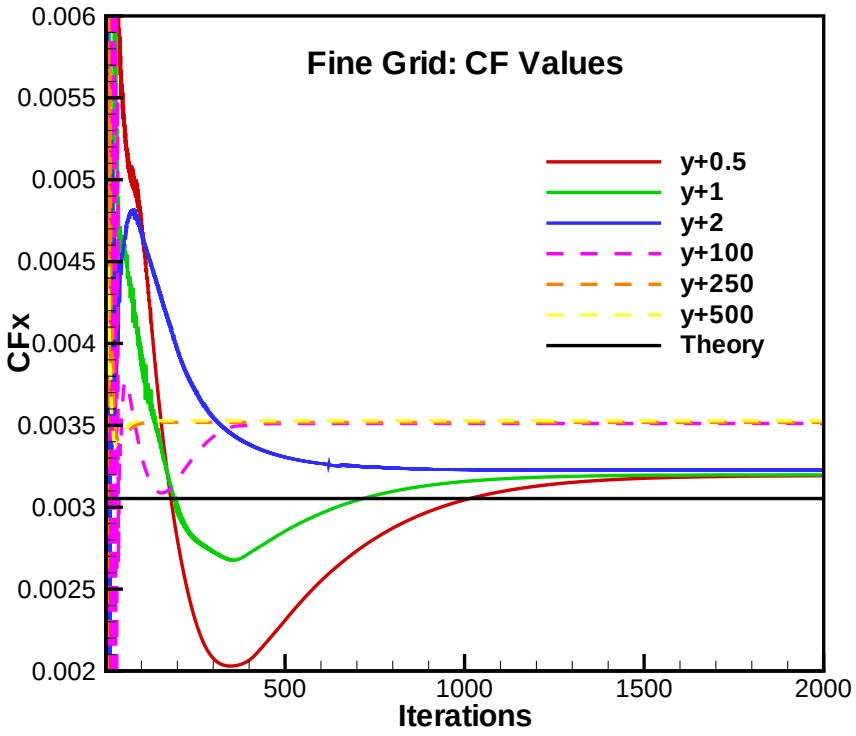

Figure 5.26. Comparison of total skin friction to theory over the hydrofoil for the fine grid study.

The sublayer-resolved grid contains 51,708 cells while the wall function grid has 31,275 cells, and can both be seen in Figures 5.36 through 5.39. The actual geometry of the airfoil was based on the specifications given in the Shen and Dimotakis paper and the original 1966 NACA66(MOD) paper by Brockett [52]. The experimental data used as comparison were based on a camber ratio of $f/c = 0.020$, a NACA meanline of $a = 0.8$, and a thickness ratio of $T = 0.09$.

5.1.4.3 Numerical Parameters

The single phase solutions were run using the same parameters as the NACA0012 airfoil. To recap, second-order discretization schemes were used after first converging with a mixed divergence scheme of linear and upwind. Also, two non-orthogonal correctors were used. The relaxation factor for pressure was set to 0.3, while all other variables were set to 0.7. Force coefficients were monitored along with residuals to ensure convergence, which took around 4000 iterations for the wall functions case and 8000 iterations for the sublayer-resolved case.

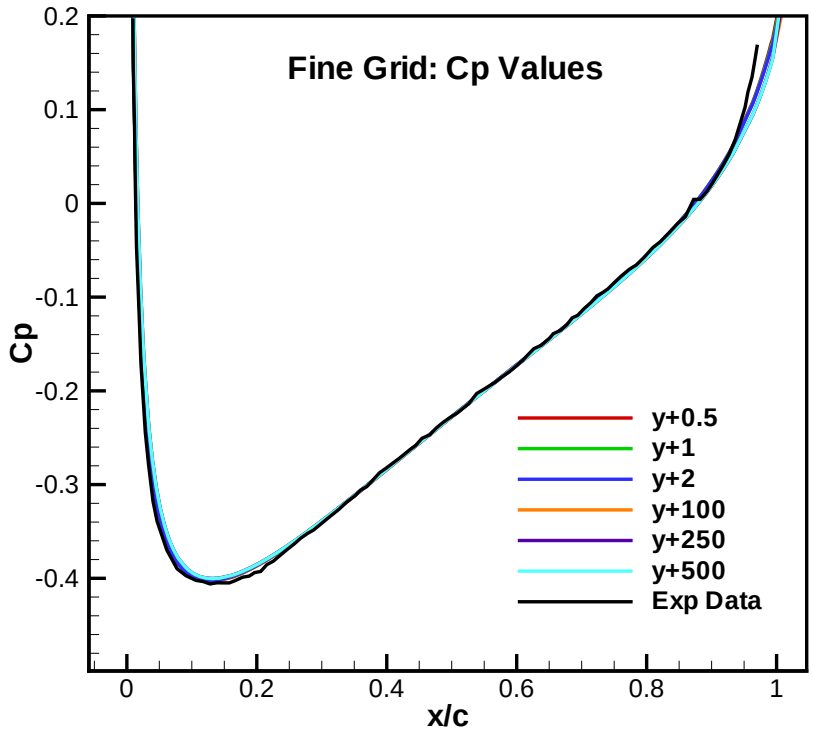

Figure 5.27. Comparison of C_p to experimental data over the hydrofoil for the fine grid study.

5.1.4.4 Simulation Results

In order to determine which grid would be used for the cavitation cases, a baseline comparison of the sublayer-resolved and wall function grid was performed. For each angle of attack, the results for both cases were comparable. The graphs in Figure 5.40 show that the pressure around the suction side of the airfoil is consistent with the experimental data. The pressure side of the foil, however, is slightly better predicted using the Spalart-Allmaras turbulence model. This over prediction of $-C_p$ by the $k-\epsilon$ turbulence model on the pressure side of the hydrofoil is not enough to warrant the extra computational time (over 10x more computational time on average) - especially since cavitation will be occurring on the suction side.

The graphical results of Figure 5.40 can be seen as the pressure field in Figure 5.41. Notice how the areas of low pressure are distributed over the suction side of the foil. The one-degree angle of attack case has an even low pressure distribution over the top surface whereas the other two cases have the lowest pressure at the leading edge of the suction side. This pressure field is the basis for onset of cavitation because in those regions of low pressure; once the values drop below the vapor pressure, cavitation will occur.

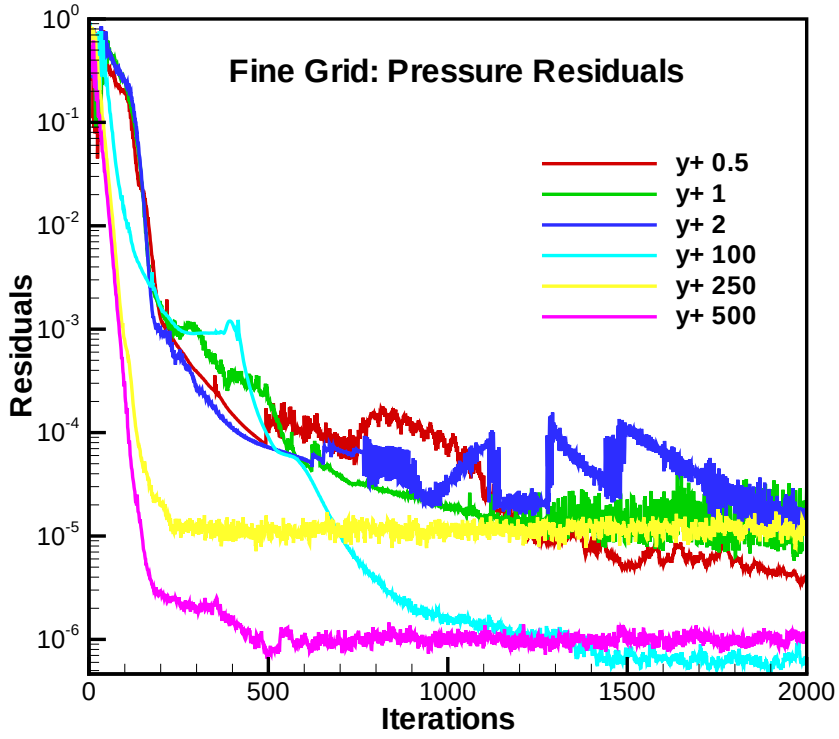

Figure 5.28. Convergence history of pressure residuals for each case using the fine grid.

5.2 Cavitating Flow

After validating and verifying the single phase test cases, the cavitating solvers described in the previous chapter will now be applied to two-phase flows. The hemispherical headform is the first case under investigation followed by the NACA0012 airfoil and the NACA66 hydrofoil.

5.2.1 Hemispherical HeadForm

The following cavitation cases are run using the hemispherical headform wedge grid that was analyzed in the single phase results section. Again, the experimental data used for comparison are from Rouse and McNown [2]. Cavitation numbers of 0.5, 0.4, 0.3, and 0.2 were tested using the different cavitation solvers in OpenFOAM. CavitatingFoam and rasCavitatingFoam are the compressible cavitation solvers used for the following cases. The difference between the two is that the latter contains a turbulence model whereas the former does not. In addition, both of these cavitation solvers contain three different compressibility models; the Wallis model, Chung model, and a linear model. All three of these were tested against one another to determine performance characteristics.

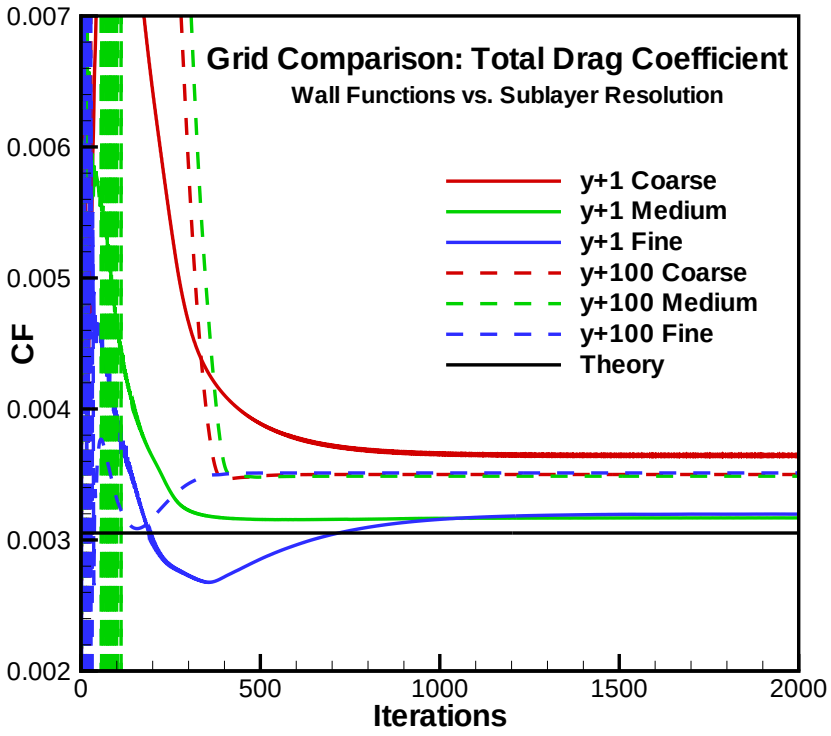

Figure 5.29. Total Drag Coefficient comparison for each grid type for both a $k-\epsilon$ and a Spalart-Allmaras turbulence model cases.

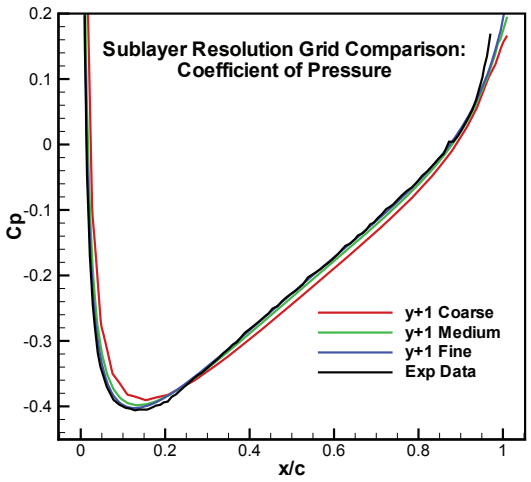

Figure 5.30. Coefficient of pressure comparison for the sublayer resolved Spalart-Allmaras turbulence model cases on each grid resolution.

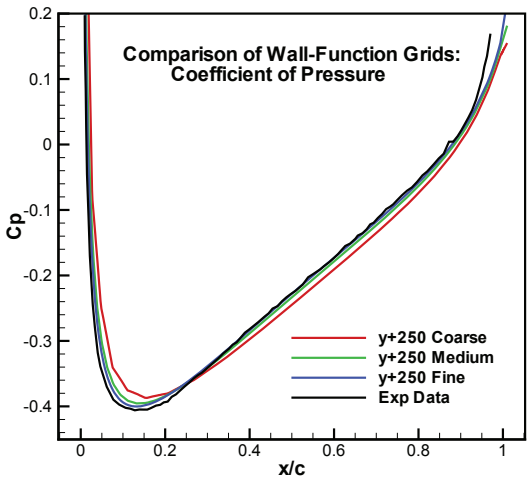

Figure 5.31. Coefficient of pressure comparison for the wall-functions $k-\epsilon$ turbulence model cases on each grid resolution.

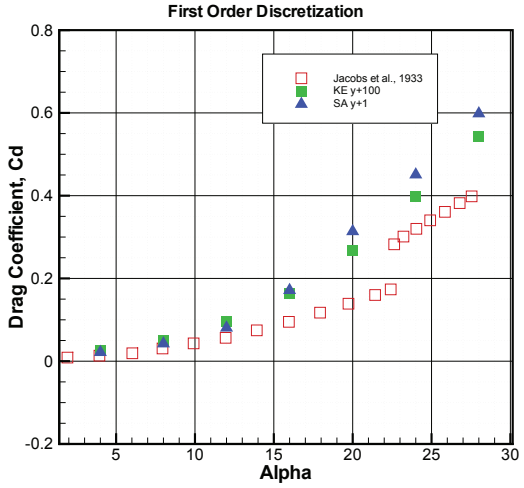

Figure 5.32. Drag Coefficient for an upwind scheme.

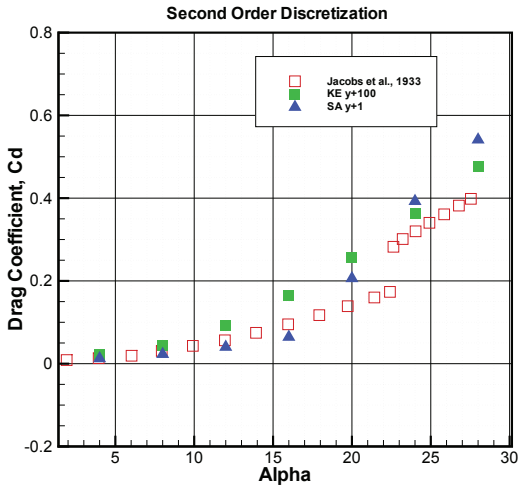

Figure 5.33. Drag Coefficient for a linear scheme.

Figure 5.34. Lift Coefficient for an upwind scheme.

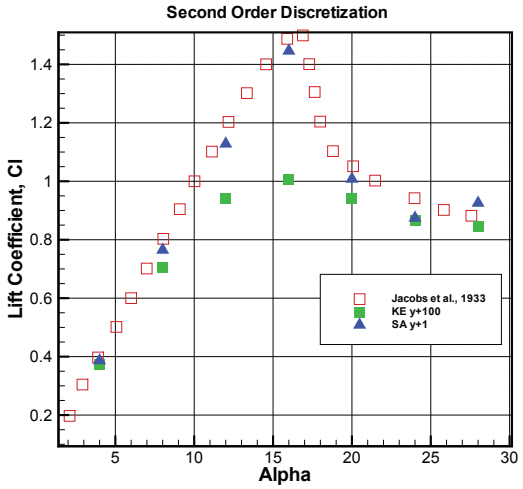

Figure 5.35. Lift Coefficient for a linear scheme.

Another cavitation solver was also tested using OpenFOAM called interPhaseChangeFoam. As described in Chapter 4, interPhaseChangeFoam is an incompressible solver that contains three phase change models: Kunz, Merkle, and Schnerr-Sauer. All of these phase change models are compared to one another in the following section. The best performing model will be used on the more complicated NACA66(MOD) case.

5.2.1.1 Validation Data and Flow Parameters

This two phase analysis again utilizes the experimental data from Rouse and McNown. Although the original published report contains numerous headform shapes, this thesis only investigates the hemispherical headform case. Based upon the single phase flow solution, the cavitation cases

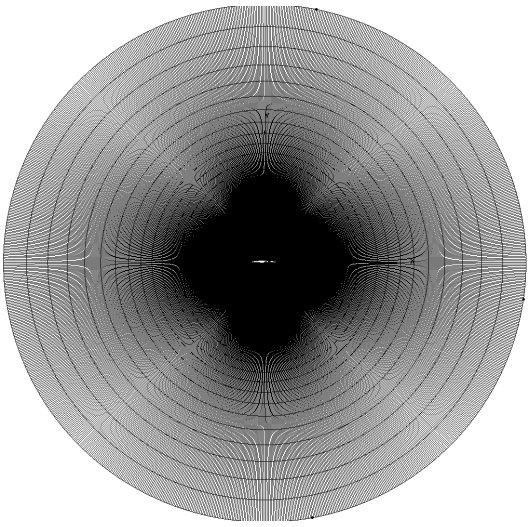

Figure 5.36. $k-\epsilon$ wall function NACA66(MOD) grid.

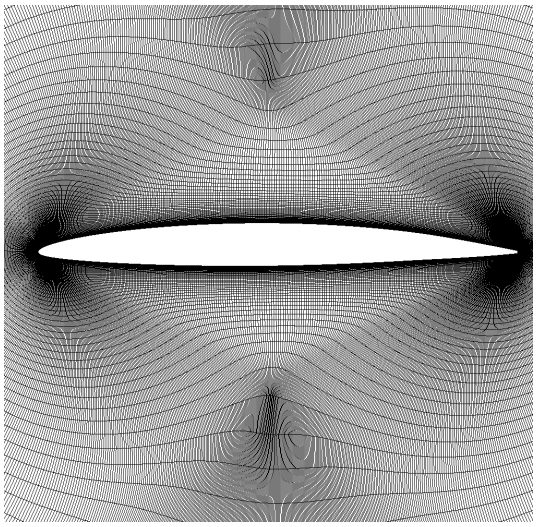

Figure 5.37. Closer view of the NACA66(MOD) hydrofoil.

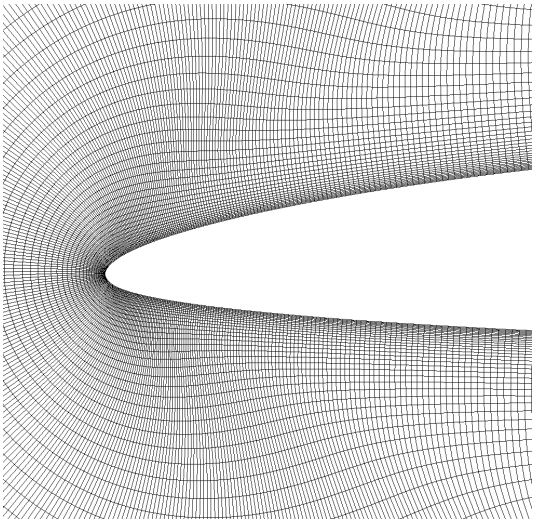

Figure 5.38. Close up of the $k-\epsilon$ wall function NACA66(MOD) grid.

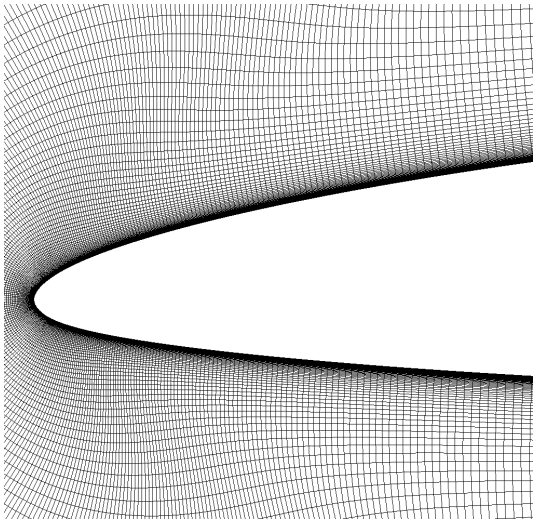

Figure 5.39. Close up of the Spalart-Allmaras sublayer-resolved NACA66(MOD) grid.

used the converged single phase results to start the simulation in order to aid in convergence of the two-phase solution. This includes pressure, velocity, and turbulence (either $\tilde{\nu}$ or k and ϵ fields.

For the compressible cavitation solvers, ρ (density), γ (volume fraction), and T (temperature) are required in addition to the standard initial conditions such as pressure, velocity, and turbulence variables. The incompressible solver does not account for thermodynamic effects and therefore does not require a specified initial condition for temperature. The values for density were supplied as 1000 kg/m^3 for water and 1 kg/m^3 for air. The volume fraction for the com-

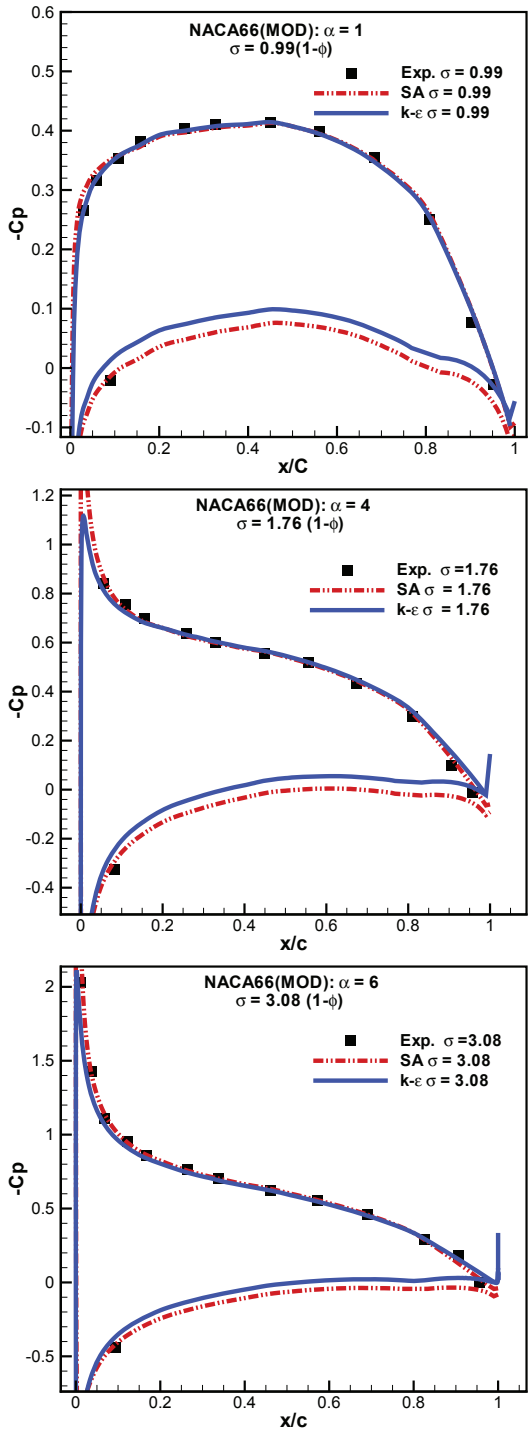

Figure 5.40. C_p values for both turbulence models at 1°, 4°, and 6°.

pressible cases was programmed as liquid water having a value of 1 and vapor a value of 0; the opposite holds true for the incompressible solver. Temperature was set to 300 K for the

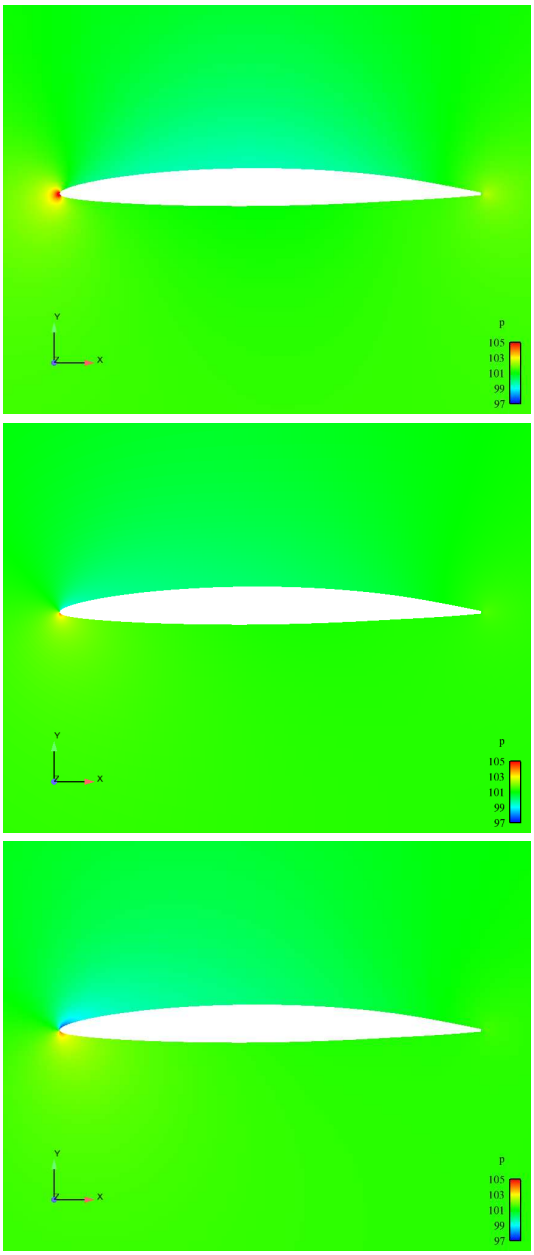

Figure 5.41. Pressure field for the 1° (top), 4° (middle), and 6° (bottom) angle of attacks.

compressible cases.

The compressible cavitation solvers also required the speed of sound (a) in both the liquid and vapor, which is used in determining the values of ψ_l and ψ_g. These constants are calculated by computing $1/a^2$. For the first cases run, this constant was set to actual values to mimic the real life flow 4.55×10^{-7} for ψ_l and 6.19×10^{-6} for ψ_g. These values will be used to control the acoustic Courant number to account for pressure waves. In terms of boundary conditions, the same parameters were used for the cavitation cases as the single phase cases. The only exception

is that the Reynolds number has been set to 1.36×10^5 to match the flow field data from Rouse and McNown's cavitation cases. In order to control the cavitation number, the saturation pressure was set accordingly while the exit pressure was held constant at 1 atm.

5.2.1.2 Grid Generation

The same grids from the single phase caseswere used. Although the wall function cases were shown to be accurate enough in the single phase results, a comparative study was performed using the Spalart-Allmaras turbulence model with sublayer resolution to determine if the effects of cavitation could be better modeled.

5.2.1.3 Numerical Parameters

Second-order schemes were used for all discretization schemes in the following cases, with limited linear differencing used for the divergence schemes to ensure boundedness. Some attempts were made at using first-order schemes on the volume fraction field discretization to produce smoother transition zones from the two phases, as well a blended scheme like Van Leer.

Unlike the single phase cases, these unsteady calculations utilize the PISO method rather than the SIMPLE algorithm. This means that rather than specifying relaxation factors, a Courant number is the parameter that determines convergence speed. For all of the unsteady cases run using OpenFOAM, the author has chosen to implement a fixed Courant number and an adjustable time step. For these cases, the Courant number varied depending on the solver, but generally the incompressible cases were run at a Courant number of 0.5 or lower, while the compressible cases, limited by the acoustic Courant number, were generally forced to run at much lower values. The acoustic Courant number was raised as high as 200 for some of the low cavitating flows, but for lower values of σ, and hence flows that contained more vapor, the acoustic Courant number was held to as low as 10 to maintain stability. This was one main drawback of the compressible cavitation solvers.

5.2.1.4 Simulation Results

First, the results of the compressible cavitation solver are presented, followed by the incompressible cavitation solver. Again, the compressibility models, along with the phase change models are evaluated and compared to experimental data.

5.2.1.4.1 cavitatingFoam This solver was used to compute the C_p of flows with $\sigma = 0.5$, 0.4, 0.3, and 0.2 over the hemispherical headform. Each case was run using the Wallis compressibility model. The results in Figure 5.42 correspond with the experimental results, with slightly higher peaks in C_p. This peak in pressure is due the existence of flow toward the boundary in this vapor region. Still farther downstream the pressure returns to the undisturbed free stream value. Since the solver is unsteady, the pressure field has been averaged over the time after the cavity has converged.

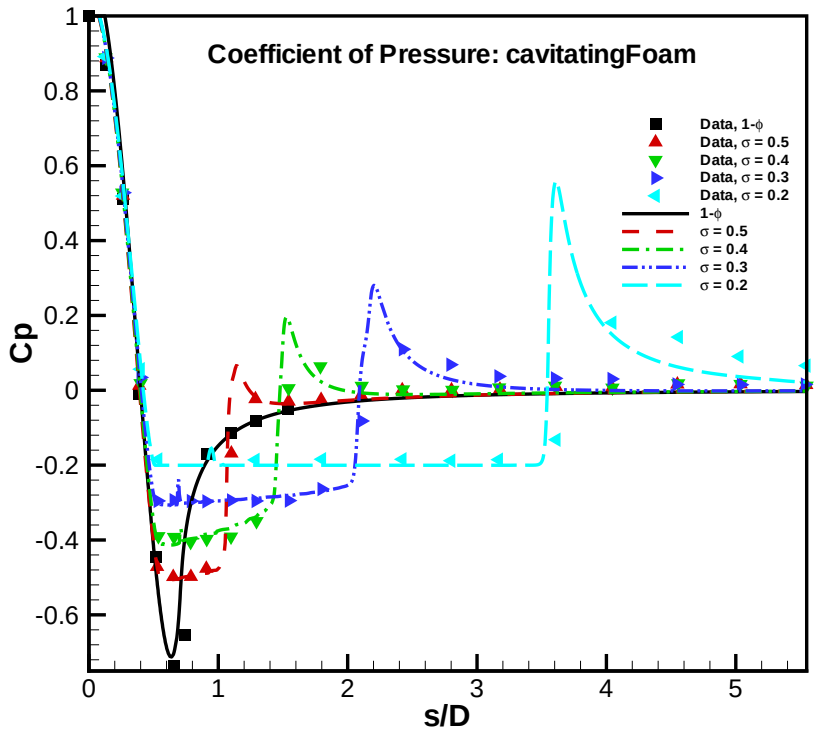

Figure 5.42. Coefficient of pressure for varying values of cavitation number using the cavitatingFoam solver without turbulence modeling.

Figure 5.43 shows the converged liquid volume fraction, surface pressure contours, selected velocity streamlines, and the computational grid for each cavitation number. As expected, the cavities grow in length down the headform as the cavitation number decreases. The pockets of vapor maintain an essentially constant pressure equal to the vapor pressure, which corresponds to reported experimental results. The streamlines show the recirculation zones at the ends of each cavity. Additionally, these flow field results are similar to what has been documented in previous CFD studies [9], [44], [53], [54]. Since cavitatingFoam is lacking a turbulence model, the next step was to try the simulations again using rasCavitatingFoam which incorporates the same compressibility models, but also accounts for turbulence.

There were some issues noted while running cavitatingFoam. First, since the solver accounts for an acoustic Courant number as well as a regular Courant number, both need to be limited and monitored to ensure convergence. The acoustic Courant number is allowed to exceed the normally restrictive value of 1, but remains the limiting factor for time step size. For example, to ensure a stable acoustic Courant number of around 50, the standard Courant number was limited to around 0.011. The maximum value of the acoustic Courant number that still provided an accurate converged solution was with the lowest cavitating case ($\sigma = 0.5$) in which the acoustic

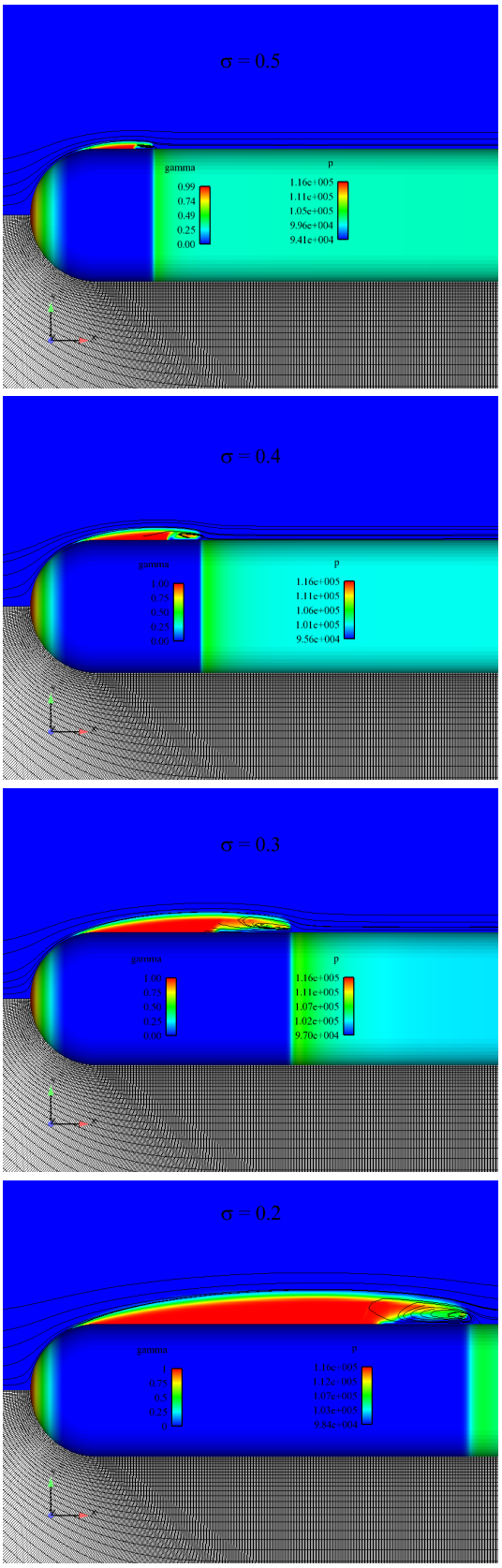

Figure 5.43. Increasing cavitation number and its effect on pressure and liquid volume fraction. Each frame was captured at the peak of the first shedding cycle.

Courant number was able to be raised to 200. In most cases, if the Courant numbers were not monitored closely, the case would fail to converge or produce inaccurate results. The main effect of this issue is the simulation is forced to run at extremely small time step values. These headform cases were limited to time steps as small 1×10^{-7}, resulting in excessively long computation times to reach a converged solution.

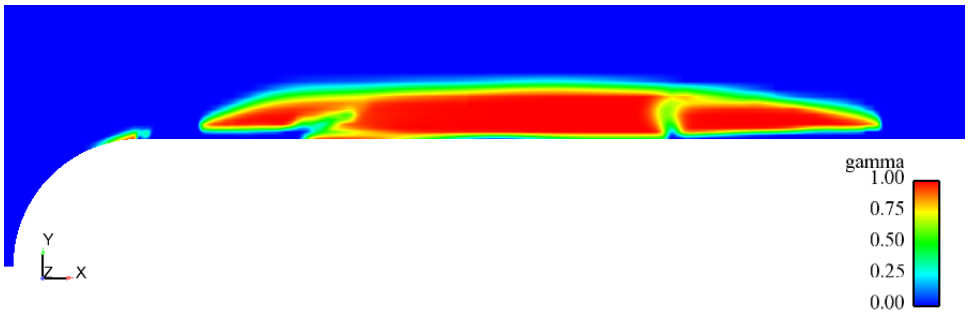

Figure 5.44. The discontinuous vapor cavity after several shedding cycles.

Another notable issue that arose is that after the vapor cavities had reached full development, there were pressure fluctuations that skewed the liquid volume fraction field. This type of phenomenon is shown in Figure 5.44 of the $\sigma = 0.2$ case. The pulsations cause the cavity to become broken and discontinuous. As for the cavities themselves, the shedding is expected in some cases, but the discontinuities in the cavity are disconcerting. The work of Rouse and McNown has shown that continuous cavities form, so this type of discontinuity is assumed to be computational error.

Overall, the compressible solver without turbulence modeling provided fairly accurate agreement with data. However, the shortcomings of a discontinuous cavity after fully developing the cavity, coupled with the lengthy required computational effort make this solver a poor overall choice for more complicated CFD problems.

5.2.1.4.2 rasCavitatingFoam With the addition of the $k-\epsilon$ turbulence model, rasCavitatingFoam was used to solve the same axisymmetric hemispherical headform cases. The first test was to see if with turbulence the solver would be able to better predict the coefficient of pressure. The entire range of cavitation numbers was run using rasCavitatingFoam and compared directly to the results of the cavitatingFoam solver. One particular case shown in Figure 5.45 where the instantaneous volume fraction field and surface pressure distribution shown for both solvers. The cavitatingFoam case has begun its first shedding cycle whereas the rasCavitatingFoam case has reached a stable cavity. From this point on, the cavity is for the most part stable with minor fluctuations, but no shedding occurs like in the non-turbulent case. This can be attributed to the turbulence modeling providing closure at the end of the cavity.

Figure 5.46 graphically shows what Figure 5.45 visualizes - that while the maximum predicted value of pressure over the headform at the rear of the cavitation bubble is more accurate than the non-turbulent calculation, the cavity length, and hence the surface pressure, fall slightly short of

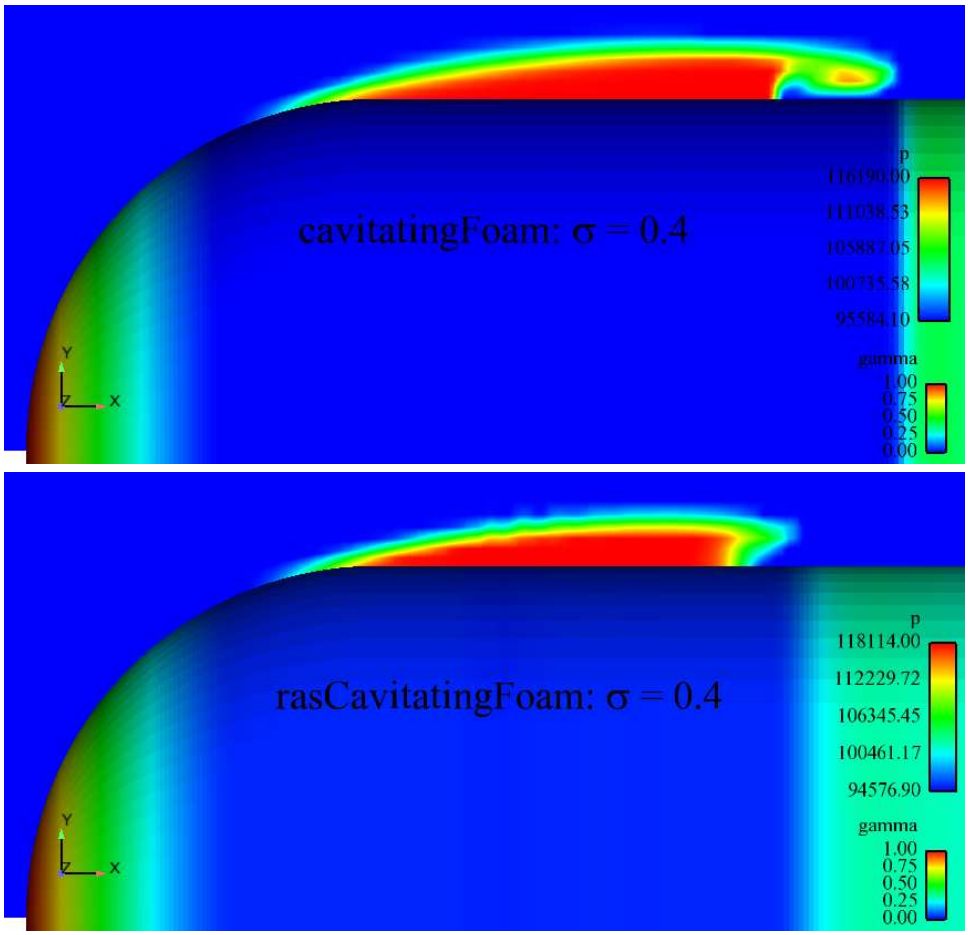

Figure 5.45. Cavity length for both the turbulent (bottom) and the non-turbulent (top) cavitation simulations.

the experimental data.

Although the turbulent cavitation solver under-predicts the cavity length, the solver is able to reach a stable converged solution unlike the cavitatingFoam case where, as shown before, the cavity fluctuates and eventually becomes disconnected. The pressure field remains intact, but the liquid volume fraction is non-physical. This added stability of the turbulent cavitation solver makes it a more suitable solver for physically modeling cavitating flow. It should be noted that all these cases so far have been run using the Wallis compressibility model. Since the option to use the Chung and Linear models exists in OpenFOAM, a comparison study was performed at a cavitation number of 0.4 on the hemispherical headform. RasCavitatingFoam was used with the $k-\epsilon$ turbulence model and the results in Figure 5.47 show that each model performs rather similarly. There is no noticeable visual difference between the models as seen in Figure 5.48.

In an attempt to lower the computational time of the rasCavitatingFoam solver, changes were made in the speed of sound of both the liquid and vapor phases. By doing this, theoretically, the time step size should increase without having to change the acoustic Courant number. The first

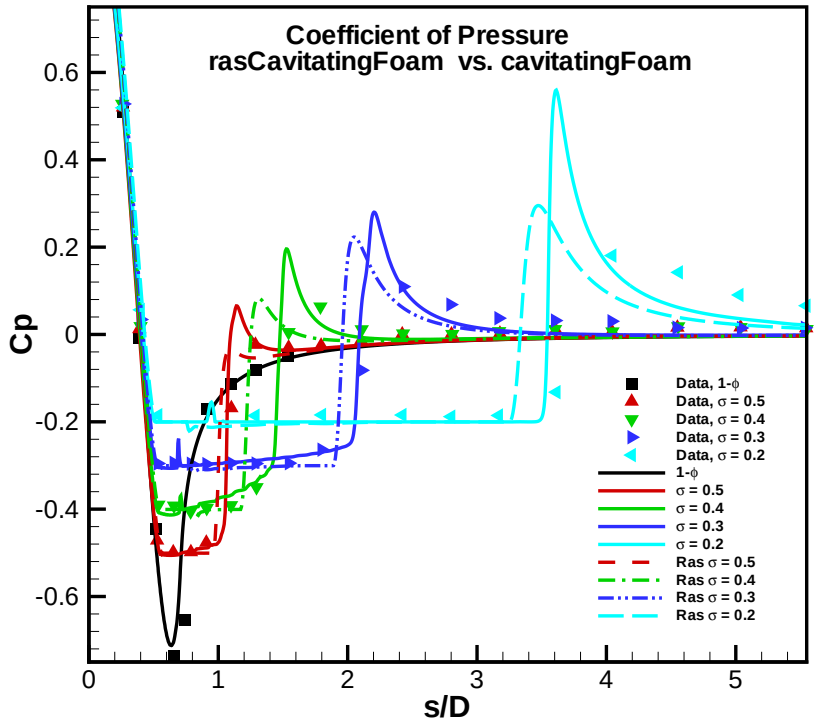

Figure 5.46. Coefficient of pressure for both the cavitatingFoam and the rasCavitatingFoam with turbulence modeling.

test was to decrease the speed of sound for both the liquid and vapor phase by a factor of 2. This meant the ψ_l and ψ_v values in the thermodynamic dictionary file changed from 4.55×10^{-7} and 6.19×10^{-6} to 1.82×10^{-6} and 2.48×10^{-5}, respectively. As expected, the mean Courant number doubled in size, and was still well below the maximum allowable value of 1. This meant that the time step size doubled as well, going from $2e^{-7}$ for one iteration up to $4e^{-7}$. The same applied for lowering the speed of sound by a factor of four; the time step size increased four times. Again, the limiting factor for each case is controlled by the speed of sound curves shown in Figure 4.3. Although this process works, it is restricted to velocities generated in the flow field, so caution is required when altering the speed of sound in the fluid and vapor in order to maintain realistic results.

5.2.1.4.3 interPhaseChangeFoam This section details the headform results using OpenFOAM's incompressible cavitation solver interPhaseChangeFoam. It should also be noted that unlike cavitatingFoam, interPhaseChangeFoam has been designed with the implementation of turbulence models. An evaluation of the phase change models has been performed and the results are compared to that of the compressible solver.

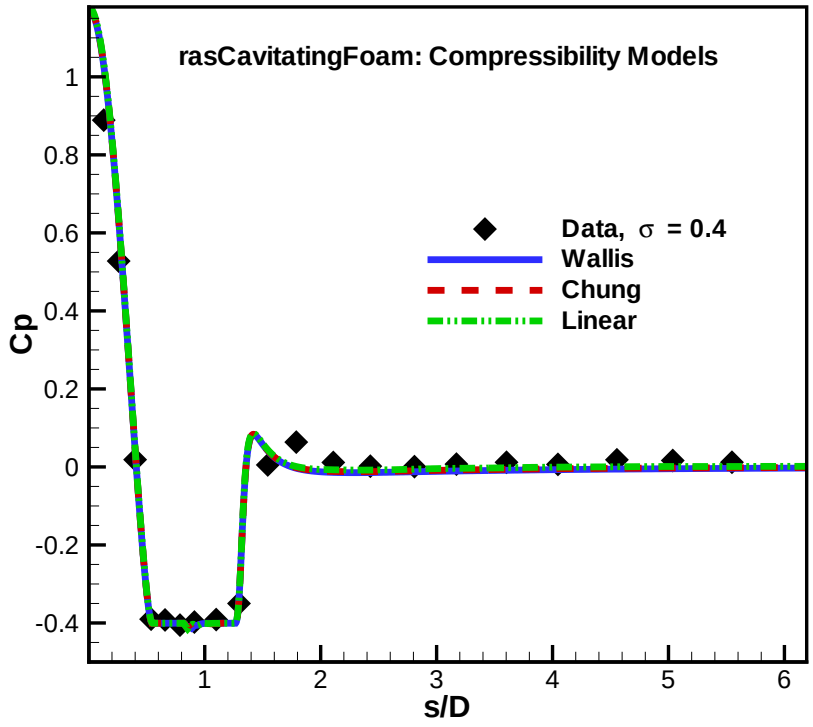

Figure 5.47. Comparison of C_p for each compressibility model at $\sigma = 0.4$.

5.2.1.4.3.1 Phase Change Models Before a full array of computations was performed using a one of the three phase change models, a comparison of performance was required to determine which model to use. The hemispherical headform with $\sigma = 0.4$ was chosen to be the verification case for the Merkle, Kunz, and Schnerr-Sauer phase change models. Although the boundary conditions were the same as the compressible case, these models require different constants to be specified. For example, the Kunz and Merkle mass transfer models are very similar in design (refer to Chapter 4) and use the same constants. Since the Schnerr-Sauer model is based on modeling the bubble dynamics rather than mass transfer, it has its own sets of constants that need to be specified. A quick summary and overview of the incompressible cavitation solver constants used for the hemispherical headform case can be seen in Table 5.8.

Table 5.8. Incompressible Phase Change Model Constants

Model	C_c	C_v	T_∞	U_∞	n	dNuc
Kunz	1000	1000	1.75×10^{-4}	5.35	—	—
Merkle	1000	1000	0.1	5.35	—	—
Schnerr-Sauer	1000	1000	—	—	1×10^8	6×10^5

The variables in Table 5.8 are the condensation empirical constant (C_c), the vaporisation

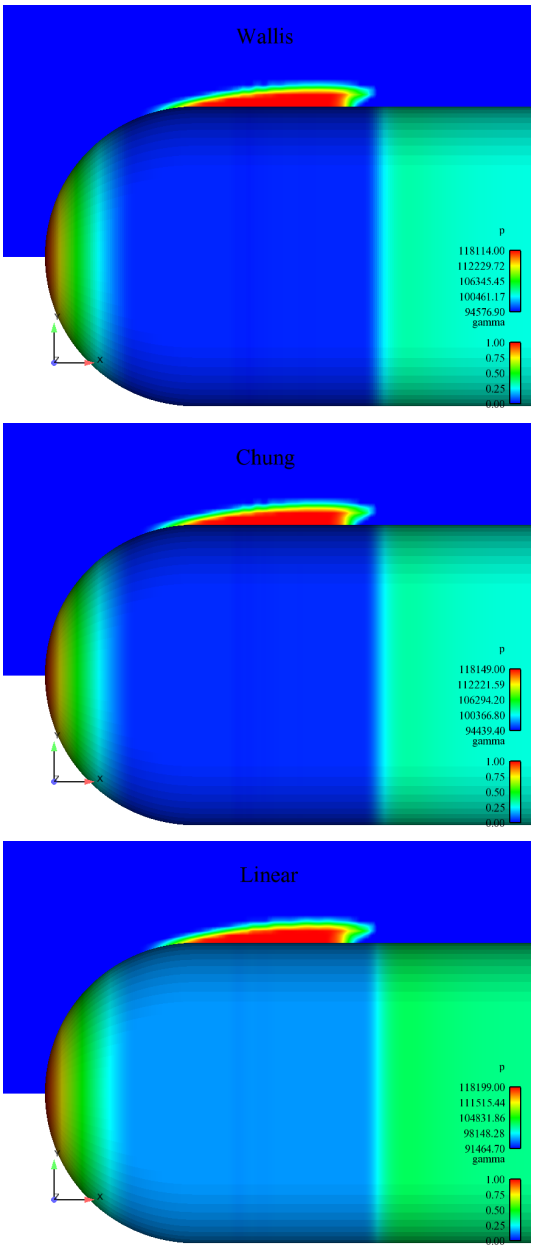

Figure 5.48. Volume fraction and surface pressure results from the three compressibility models.

empirical constant (C_v), the mean flow time scale (T_∞), the velocity scale U_∞, the bubble number density (n), and the nucleation site diameter (dNuc). All models use the empirical constants for condensation and vapor, but the Schnerr-Sauer model differs from the other two in requiring bubble properties rather than the free stream velocity and time scales.

As Figure 5.49 shows, all three cases produce similar results in terms of volume fraction and surface pressure distribution. The waviness at the intersection of the liquid and vapor fields can be attributed to the discretization scheme used on γ, the volume fraction field. The Van Leer

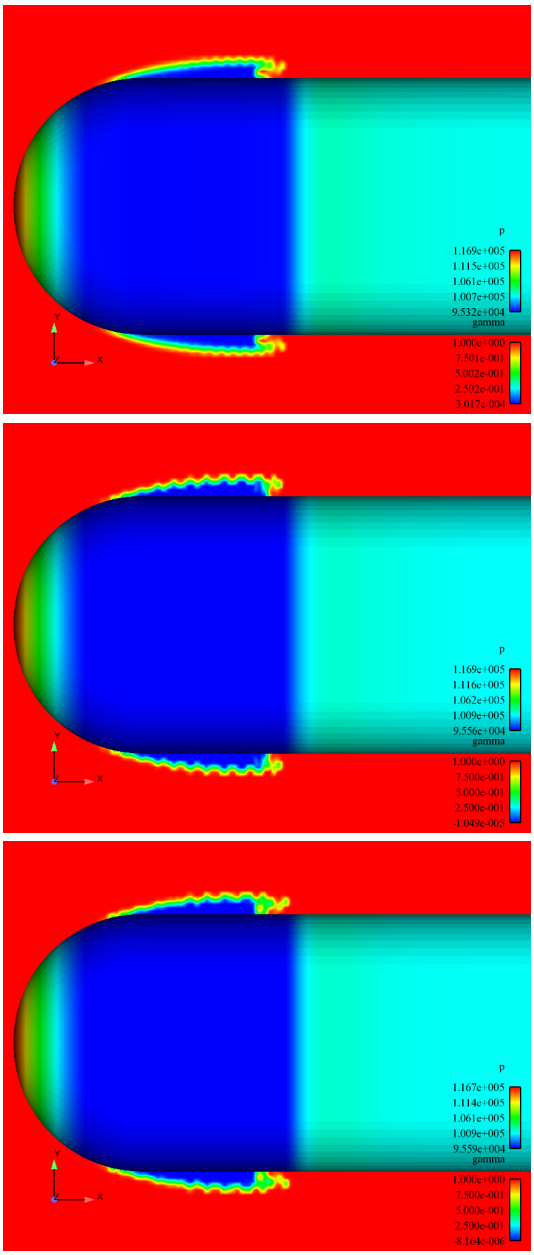

Figure 5.49. Volume fraction and surface pressure results from the three phase change models: Kunz on the top, Merkle in the middle, and Schnerr-Sauer on the bottom.

scheme was used for boundness but also tends to lead to such numerical errors. One solution could be to switch to an upwind scheme that would blend out the numerical errors, but also decrease the accuracy to first-order.

All three cases shown in Figure 5.49 form stable cavities without shedding, and solution times for each were comparable. The differences lie in the sensitivity of each model. The Merkle model was very sensitive to the time constant. For example, the relation of the length scale divided by

the velocity scale to calculate a time scale did not produce a converged solution using the Merkle model. A range of values for the time constant was tested in order to produce a converged solution. On the other hand, the Kunz model was flexible with the time constant supplied, performing equally as well across a wide range of values, including the standard length scale divided by the velocity scale. The Schnerr-Sauer model reached almost an identical solution to the other two by specifying a realistic nucleation site radius and bubble density.

In order to compare all the models, the cavitation results for all solvers at a $\sigma = 0.4$ have been plotted in Figure 5.50. Although the solution differences between the phase change models are small, the Kunz model has been chosen for future cavitation cases. It has shown more flexibility than the Merkle model, and is more comparable to UNCLE-M results that utilize a similar approach rather than the bubble dynamics solution procedure of the Scherr-Sauer model.

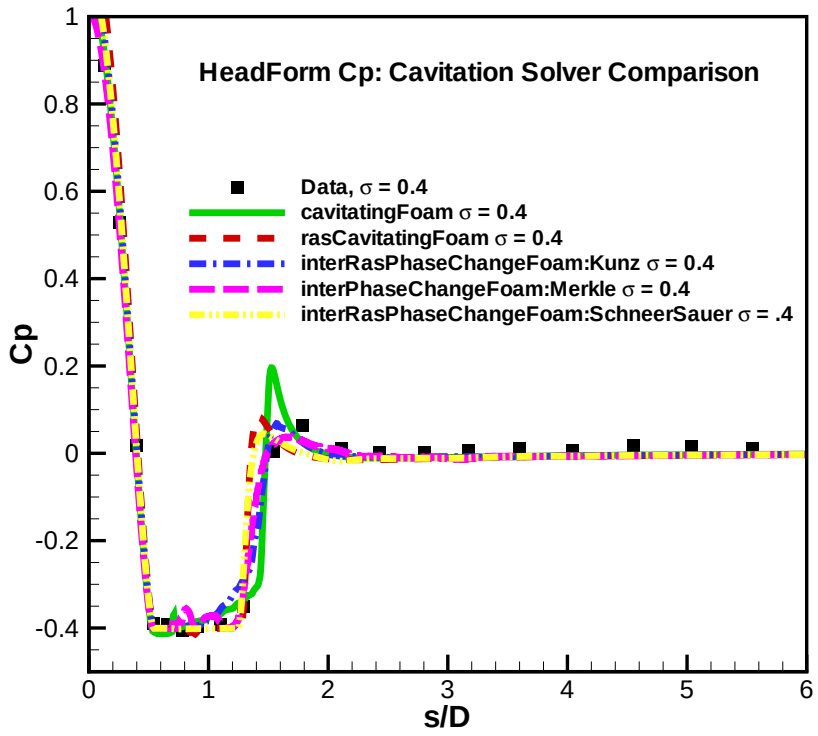

Figure 5.50. Comparison of C_p for each cavitation solver at $\sigma = 0.4$.

Figure 5.51 was generated to study how the Kunz model performs across the same range of cavitation numbers as the compressible cavitation solver. What is first evident is that the cavity length is fairly accurate in all cases and it does a good job at predicting the peak pressure after the cavity. The only downfall, which was also seen with the turbulence modeled compressible solver, is over-prediction of the pressure recovery region. It appears that the pressure recovery after the

cavity occurs at a faster rate than found in the experimental work of Rouse and McNown. The overall performance of the solver is in line with the expected experimental results

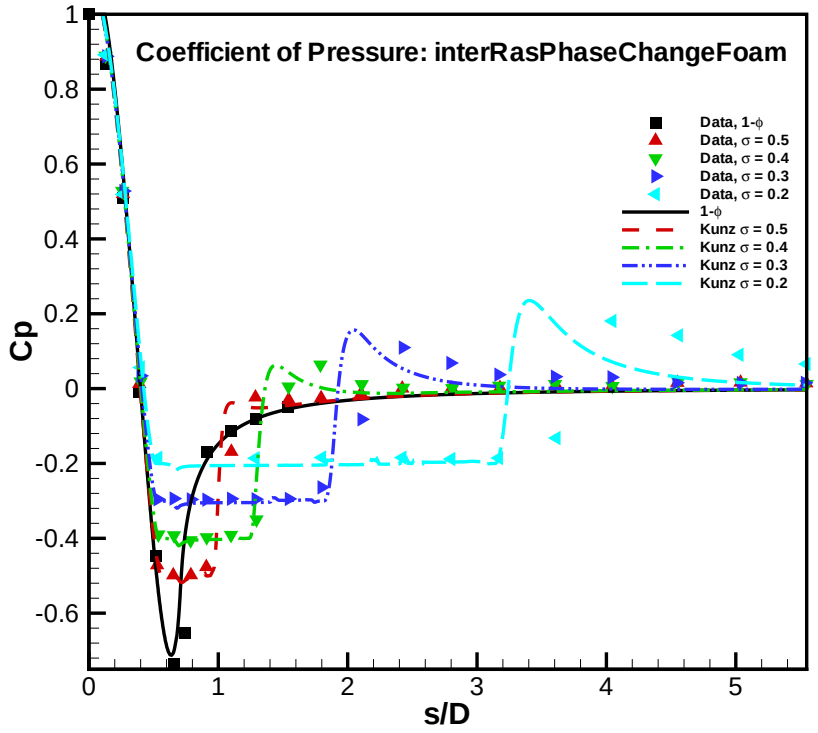

Figure 5.51. The full range of cavitation results from the incompressible solver with the Kunz phase change model.

5.2.2 NACA66(mod) Airfoil

For the cavitation study, both leading edge and midchord cavitation were under investigation at one, four, and six degrees angle of attack. The NACA66 airfoil was chosen to study cavitation effects because the shape and camber of the foil are well suited for marine applications.

5.2.2.1 Validation Data and Flow Parameters

In order to validate the CFD results, comparison was made to the experimental pressure data from Shen and Dimotakis [51] which has been used extensively in previous cavitation studies [11], [55], [56]. The parameters used for each of the cases can be seen in Table 5.9

The simulations were all started from a single phase steady-state solution to aid in convergence of the two phase computation.

Table 5.9. NACA66(MOD) Case Summary

Angle of Attack	Reynolds Number (Re)	Cavitation Number (σ)	Cavitation
1°	3×10^6	0.99 0.43 0.38 0.34	Mid-Chord
4°	2×10^6	1.76 1 0.91 0.84	Leading-Edge
6°	2×10^6	3.08 1.25	Leading-Edge

5.2.2.2 Grid Generation

Both of the grids from the single phase solutions were used in these cavitation studies. In order to study the effects of varying grid structures and surface cell counts, several hybrid structured/unstructured grids that ranged from a lower number of surface cells to a higher concentration of surface cells over the hydrofoil. The original O-shaped grids contained 417 cells around the surface of the foil. The slightly coarser, hybrid grid contained 364 faces on the foil surface, whereas the fine hybrid grid had 728 faces around the surface.

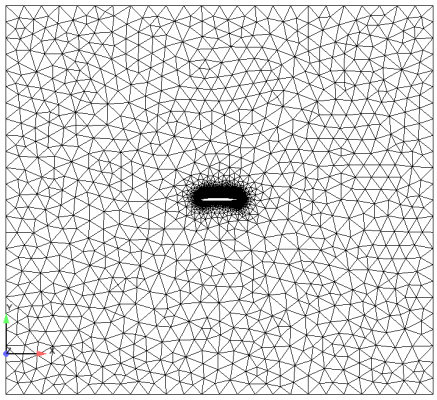

Figure 5.52. Full view of the hybrid $k-\epsilon$ wall function NACA66(MOD) grid. The inlet is on the left, outlet to the right, foil in the middle, and top and bottom are set to symmetry conditions.

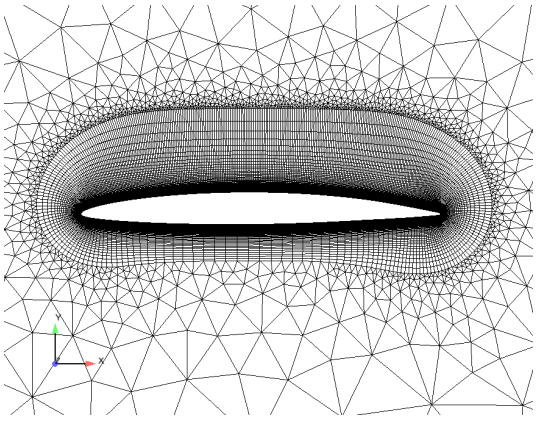

Figure 5.53. Close up of the coarse hybrid $k-\epsilon$ wall function NACA66(MOD) grid showing the structured near wall cells.

5.2.2.3 Numerical Parameters

For these NACA66 cavitation cases, the discretization schemes were set to second-order accuracy. Most schemes were set to linear (center differencing), while the γ field was discretized using a Van Leer scheme to ensure boundness. The Courant number ranged between 0.2 and 0.5 depending on the degree of cavitation in the particular case. Cases with lower cavitation numbers generally

were given lower Courant numbers to adhere to, in order to ensure stability. These cavitation cases were run out until either a stable cavity had been reached or several pulsations over the same area occurred and proper time averaging of the pressure and volume fraction fields could occur. The tolerance for all variables was set to 1×10^{-8} with a relative tolerance of 0.0. The incompressible cavitation solver interPhaseChangeFoam was used with the $k-\epsilon$ turbulence model.

5.2.2.4 Simulation Results

The midchord cavitation at one degree angle of attack was the first case studied. As previously mentioned in the single phase case, the area of low pressure located uniformly across the suction side of the foil leads to midchord cavitation once that region of low pressure drops below the vapor pressure. OpenFOAM was able to produce the best results for this angle of attack (Figure 5.54) due to the lack of any sharp pressure gradients over the suction side of the hydrofoil.

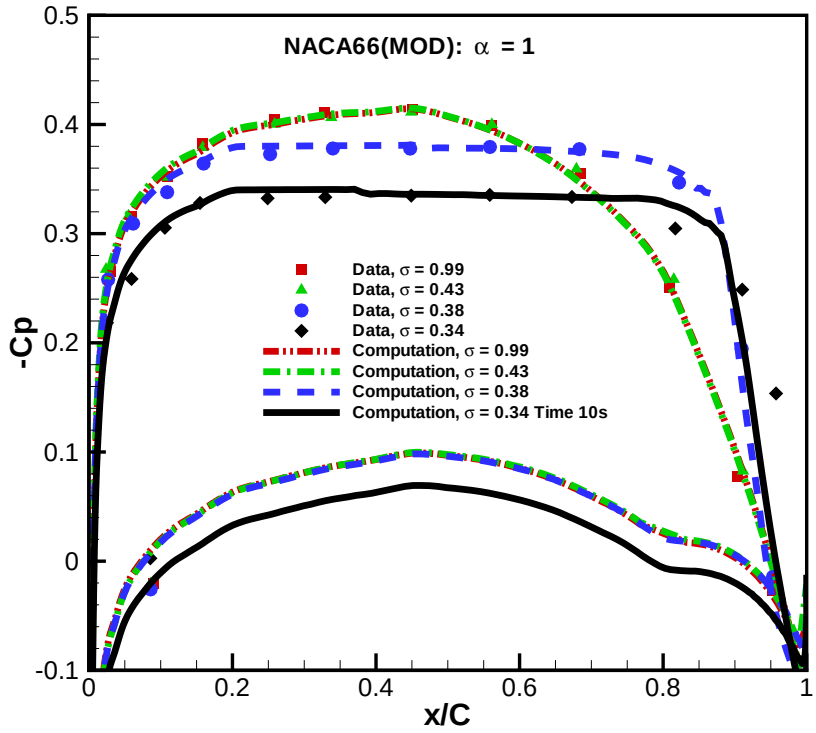

Figure 5.54. Surface pressure distribution at different values of σ over the NACA66(MOD).

Changing the angle of attack to 4 degrees and decreasing the Reynolds number had a significant impact on the flow field, as previously seen in the single phase results section. This change in the surface pressure had a direct effect on the cavitation results as one would expect. Figure 5.55 shows how the sharp pressure gradient that the angle of attack creates causes problems for

the cavitation solver. The surface pressure, and therefore the vapor cavity, fall significantly short of the experimental values. The spike in peak pressure after the cavity closure matches what the experiments show; however, the cavities themselves are just too short. At the lowest cavitation number, $\sigma = 0.84$, the cavity should reach roughly half the chord length, whereas the solver is showing the cavity approach four tenths of the chord length. The main cause for this error is the sharp pressure gradient seen over the suction side of the foil, something not seen in the previous lower angle of attack case.

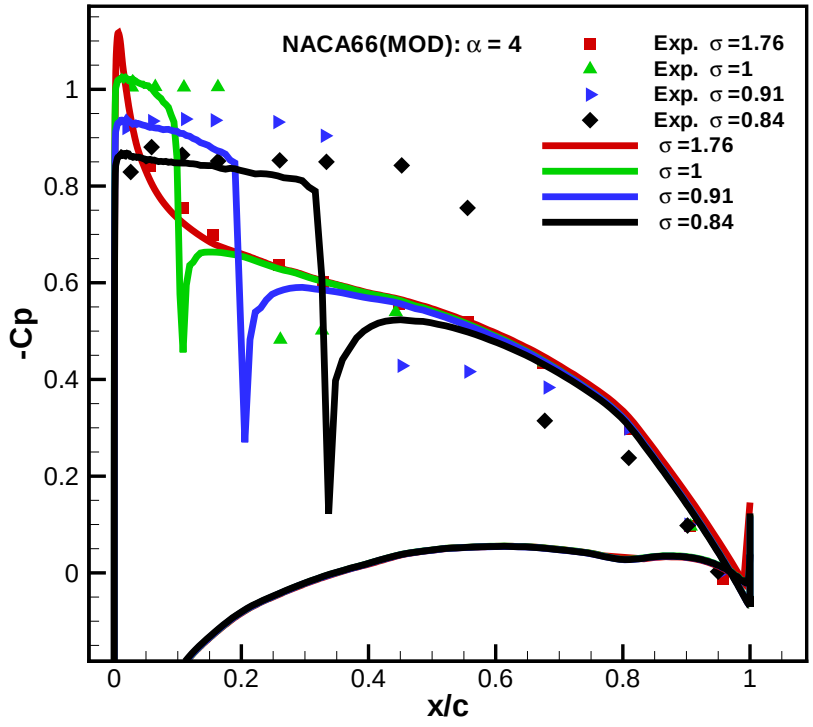

Figure 5.55. Although the peaks in pressure recovery correspond with experimental results, the cavity length does not.

For the 6° cavitation case, the $\sigma = 1.25$ value showed similar results to the previous case: under-prediction of the cavity length. This case had an even greater pressure gradient and was unable to accurately predict the pressure recovery region (Figure 5.56).

The flow field for each of the NACA66(MOD) cases is shown in Figure 5.57. The midchord cavitation volume fraction field is an averaged value due to the shedding of the cavity. The other two cases showed fairly stable cavities without any shedding. Again, for the gamma field, 1 represents liquid and 0 represents vapor for this incompressible solver.

For all the NACA66 cases, the pressure in the cavity matched the experimental data, but the cavity closure occurred prematurely. Several actions have been implemented to try to correct

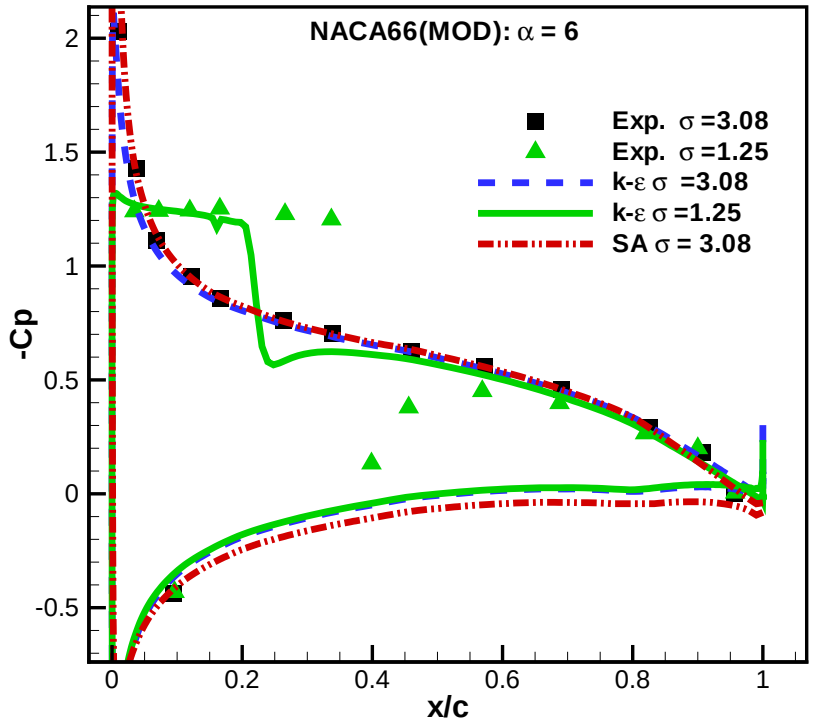

Figure 5.56. Pressure distribution at six degrees angle of attack for the single phase and cavitating cases.

this issue. First, a grid study was performed in order to determine if lack of grid resolution was responsible for the errors. Second, the constants, C_c, C_v, and t_∞ were altered and tested for more agreement with the data. The hybrid grids shown in the grid generation section were used to compare to the baseline results. Again, in order to compensate for the increased number of cells on the surface of the hydrofoil, the far field was constructed using unstructured cells. In addition, this provides insight into how OF handles hybrid grids.

The results from the grid study show that changing the number of cells around the hydrofoil had no impact on the solution. In fact, when attempting to run the same case using the Spalart-Allmaras turbulence model, the case could not converge. This can be attributed to the hybrid nature of the grid. For this $k-\epsilon$ study, the solutions converged, but suffered from the same under-prediction of cavity length seen in the baseline case. These time averaged pressure values do not feature the same distinct spikes in pressure as the instantaneous coefficient of pressure graph, similar to the Figure 5.55.

Finally, to study the effect of changing the empirical constants of condensation and vaporization (C_c and C_v), a simple test was performed on the NACA66(mod) at 6° angle of attack. The cavitation number was set to 1.25 and the constants were changed from $C_c = C_v = 1000$ to

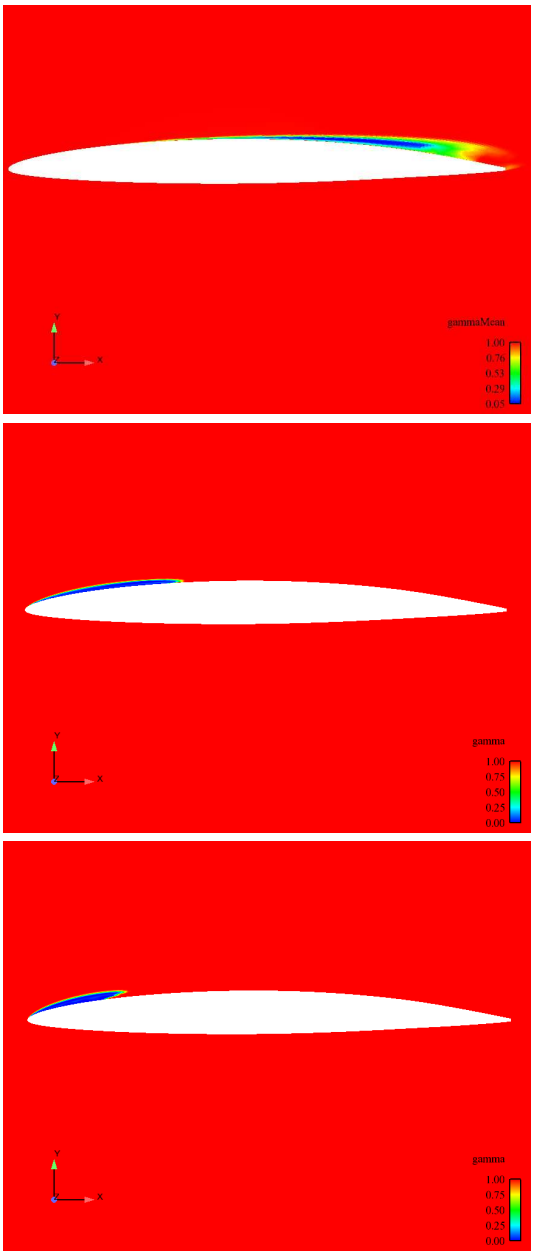

Figure 5.57. NACA66 volume fraction field at the lowest cavitation number for each angle of attack (Top = 1°, Middle = 4°, Bottom = 6°).

ranges from 1 to $1e^6$. The incompressible Kunz cavitation solver was used. Changing the values of the empirical condensation and vaporization constants had a direct result on the shape and progression of the vapor cavity. The volume fraction field, as expected, showed increased levels of vapor when the vaporization constant was increased and marked decrease in levels of vapor when the condensation constant was increased. Figure 5.59 highlights some of the results from the test cases. It should be noted that at some extremes, the solver failed to converge. This

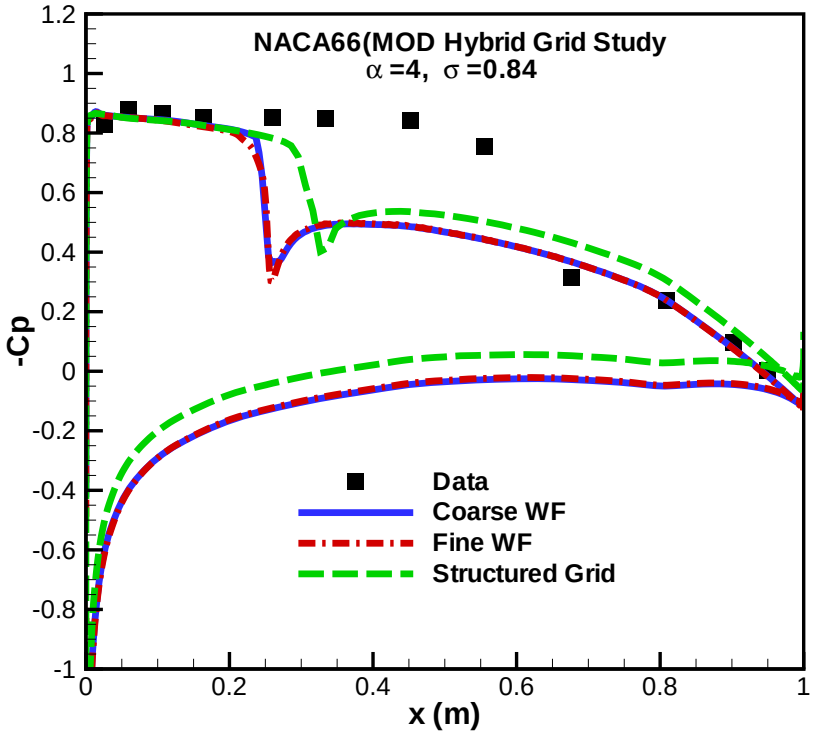

Figure 5.58. The pressure distribution of the hybrid grids compared to the original, structured grid.

indicates that although the solver is flexible with the value of the production and destruction terms, there are still limits.

The increased terms for the production constant not only resulted in increased vapor, but also reduced the destruction which led to shedding of the cavity. When the production constant was reduced by an order of magnitude from $1e^5$ to $1e^4$, the effect of cavity shedding slightly diminished, but remained strong due to the low destruction term. When the destruction term was increased to $1e^6$, the cavity formed at the leading edge as expected, but failed to develop. The large magnitude of the destruction term is too much for the production term to overcome. The results show that a proper balance of production and destruction is needed to properly model cavitation.

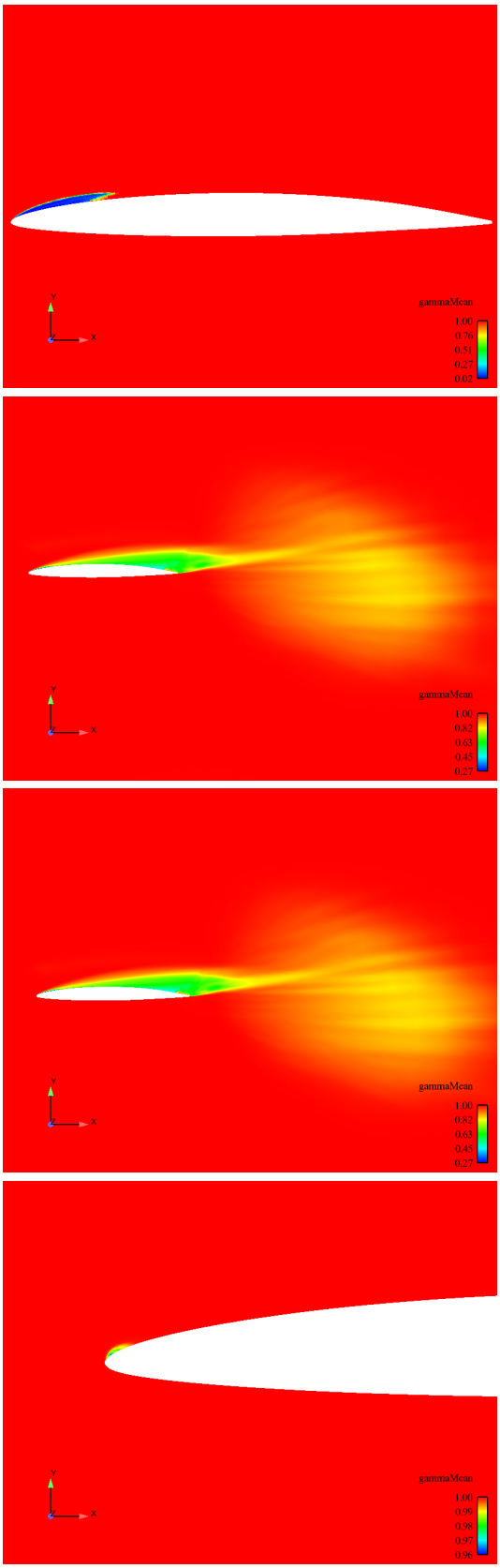

Figure 5.59. Results for volume fraction field with varying empirical constants for condensation and vaporization. From top to bottom: $C_c = C_v 1000$, $C_c = 1 C_v = 1e^4$, $C_c = 1 C_v 1e^5$, $C_c = 1e^6 C_v = 1$

Chapter 6

Conclusions and Future Work

6.1 Single Phase Computations

This thesis has shown, with the use of RANS CFD modeling in both single phase and cavitating flow, that an open-source tool, OpenFOAM, has been successful in terms of verifying and validating several benchmark hydrodynamic cases. The variety of cases, coupled with the testing of various numerical parameters, boundary conditions, constants, and turbulence models have shown a wide range of functionality and applications. These findings will provide the groundwork for more complicated marine applications such as waterjets and open water propellers.

Verification and validation of single phase computations have provided insight into the functionality of OpenFOAM. First, the flat plate case was investigated and demonstrated the capabilities of both the wall function turbulence models and the sublayer resolved turbulence models. The range of dimensionless near wall spacing values was determined for each case, and the finite volume schemes were tested for accuracy. Total drag and local skin friction coefficients, along with boundary layer profiles, were all used as verification in determining what parameters and turbulence models were best suited for this baseline turbulent flat plate case. The Spalart-Allmaras and $k-\epsilon$ models were the best performing turbulence models, producing results within reasonable range of the expected published values. The sublayer resolved SA model showed excellent results in terms of skin friction and total drag for $y^+ = 0.1$ and $y^+ = 1$. In addition, the $k-\epsilon$ turbulence model produced similar results for grids with much less near wall resolution, with $100 \leq y^+ \leq 150$ producing the best drag and skin friction results.

Next, the hemispherical headform provided a more complex geometry to investigate, as well as a reliable benchmark to validate the results with experimental data. A full 3-D headform case and an axisymmetric wedge headform both showed agreement with experimental surface pressure data from Rouse and McNown. In addition to testing the difference in numerical results with the $k-\epsilon$ and Spalart-Allmaras turbulence models, unsteady solvers utilizing the PISO method were compared directly to the steady-state results which used the SIMPLE algorithm. All cases

resulted in matching the single phase coefficients of pressure on the surface of the headform. The differences in the cases arose in solution times. Most notably was the 3D case, with 10 times as many cells as the wedge boundary condition case, which took much longer per iteration to reach a converged solution (19,000 seconds on 2 processors for the wedge case, compared to 108,000 seconds on 4 processors for the full 3-D case). Both cases produce almost identical results which shows that the periodic boundary conditions work sufficiently and can be used to drastically reduced grid size and ultimately computational resources.

Although generally the increased near wall resolution required by low-Re turbulence models has a direct result of increasing computational time, since the overall cell count of the headform case was relatively small (under 100,000 cells) and the difference between the wall function grid and the sublayer resolved grid was roughly 15,000 cells, the computational time between the two turbulence models was almost negligible. However, the unsteady, Courant number driven solver was significantly slower in reaching a converged solution than the steady-state solver. For example, the unsteady simulation required a little over 70,000 seconds on 4 processors to advance the solution 0.30 seconds, which was a sufficiently converged solution for this simple test case. The steady-state solver, without the restrictions of monitoring a Courant number, was able to solve for 5,000 time steps, converging completely in only 19,000 seconds with just 2 processors. This study has shown while both the steady and unsteady solvers are reliable in OpenFOAM, the steady-state solver is much more efficient in producing a converged solution.

The NACA0012 airfoil provided another benchmark for validation with proven experimental data. For these cases, the lift, drag, and coefficient of pressure were used as metrics for validation of single phase flow. In addition, total skin friction coefficient values were compared to theoretical values. A grid study was also performed using a coarse, medium, and fine resolution grid. The results showed that as grid resolution improved, the surface pressure and skin friction coefficient also showed improved accuracy. In addition, the lift and drag values corresponded to experimental data after first reaching a converged solution using an upwind scheme then switching to a linear, second order linearization scheme for the divergence of the velocity field. The technique provided better results for the sublayer resolved grid than the wall function grid, which had difficulty identifying the stall angle of the foil. The separation of flow at higher angles of attack was properly modeled by the Spalart-Allmaras model, which historically has been used for just such airfoil simulations.

With more marine applications than its NACA0012 counterpart, the NACA66(MOD) hydrofoil was computed at several angles of attack and the results were validated against experimental data [51]. The coefficient of pressure results over the surface of the foil laid the foundation for the cavitation cases. An initial study of the effect of wall functions versus sublayer resolution of this non-symetric foil showed that the $k - \epsilon$ turbulence model provided accurate results for a fraction of the computational time. The pressure of the foil for each angle of attack corresponded with the experimental results and highlighted the correct areas of low pressure that eventually equate to cavitation.

6.2 Cavitating Flow

Two primary benchmarks were investigated for the cavitation section: the hemispherical headform and the NACA66(MOD) foils. These two cases, already verified and validated for the single phase flows, were used to study the differences between compressible and incompressible solvers, along with their respective compressibility models and phase change models. In addition to having cavitation data readily available for direct comparison, the singles phase results provided a reliable starting point to test the cavitation models.

First the hemispherical headform, starting from the wedge boundary condition case, was solved at four different cavitation numbers (0.2, 0.3, 0.4, and 0.5) using the compressible cavitation solver. The first comparison involved solving cavitating flows with and without the turbulence model. Next, three different compressibility models were compared against one another. Then the incompressible solver was used to solve the cavitating flow fields, comparing each of its three phase change models. All of these results were validated using the experimental data of Rouse and McNown. A summary of the compressible cavitation solver results is as follows:

- The compressible solver without the turbulence model produced excellent cavity lengths and the downstream surface pressure agreed with the data.
- The compressible solver without the turbulence modle slightly over predicts the recovery pressure after the cavity.
- The compressible solver with turbulence corrected the over-prediction of the pressure recovery immediately after the cavity.
- The compressible solver with turbulence produced shorter cavity lengths, and hence shorter regions of constant vapor pressure compared to the solver without turbulence modeling.
- The solver equipped with the turbulence model is more physical, but some modification to the solver may be needed to better correct the effects of turbulence on the vapor cavity.
- The three compressibility models, did not produce any noticeable change in the solution. As long as the speed of sound was kept below the limits of the model, the physics were unchanged.
- The speed of sound for the compressibility model can be altered from the actual physical value to decrease computational time, but caution must be used as to not allow the Mach number of the flow to reach too high (above 0.3).
- Although the compressible solvers produced accurate results, lengthy solution times are inevitable to maintain a significantly low acoustic Courant number. Furthermore, even for these simple benchmark cases, stability was an issue for the non-turbulence modeled cases, since for such high Reynolds numbers, a lack of turbulence is not practical.

- The compressible solver appears more suited for smaller grids with less stringent near wall spacing requirements.

Using the same wedge shaped headform grid as in the compressible cavitation cases, the incompressible solver was run through the same range of cavitation numbers. The three different phase change models were also validated against the experimental data. A summary of the results is as follows:

- In a similar fashion to the compressible case, the overall surface pressure and resulting volume fraction field varied very little model to model, even though the Schnerr-Sauer mode used bubble properties rather than velocity and time scales.

- Although results are similar to the turbulence modeled compressible solver, these incompressible cases are not limited by the acoustic Courant number. The solution times for these cases are 27,000 seconds on 4 processors for the incompressible case to reach 0.1 seconds, compared to 687,000 seconds (over a week) on 4 processors for the compressible case.

- The Kunz Model showed flexibility in the range of time scales that result in a converged solution. The Merkle model was much more sensitive and in some cases failed to converge with the same time scales as the Kunz model case. This result originates from the slight differences in computation of the source terms in the pressure and volume fraction equations.

- Overall, accuracy of the solvers being comparable, the incompressible cavitation solver with turbulence modeling and the Kunz phase change model demonstrated the most versatility with variables, good solution stability, acceptable correlation with data in terms of surface pressure and cavity lengths, and rapid convergence speeds over the range of cavitation numbers for the hemispherical headform test case.

The NACA66(MOD) provided more insight into the application of the cavitation solver. Based upon the performance of the hemispherical headform cavitation cases, the incompressible Kunz model was chosen to be the cavitation solver for these test cases. The three angles of attack that were studied, 1°, 4°, and 6° showed both leading edge cavitation and mid-chord cavitation. The midchord cavitation cases showed excellent agreement with experimental data in terms of surface pressure and cavity lengths. When the angle of attack was increase to create leading edge cavitation, the results showed under-prediction of the cavities. This under-prediction can be attributed to the increased pressure gradient over the suction side of the foil. Cases with such steep pressure gradients will need to be studied further with the $k-\epsilon$ turbulence model to obtain more accurate cavitation results.

To determine if the under-prediction of cavity length was caused by insufficient surface grid points, a grid study was performed using hybrid structured/unstructred grids. The results showed that changing the number of surface cells had no direct influence in obtaining a more accurate

cavitating flow field. In fact, the hybrid grid results showed even greater under-prediction of the cavities. Additionally, the sublayer resolved hybrid grids were unable to reach converged solutions. More work is needed on the hybrid grid studies in the future.

Finally, a few simple cases were run with varying extremes of empirical constants (C_c and C_v). The results showed that the incompressible cavitation solver was not sensitive to small changes in constants, such as one order of magnitude, but when larger variations were set for initial constants, the cavitation results varied greatly. When the production constant was increased several orders of magnitude, while simultaneously decreasing the destruction term, the vapor was observed to shed off of the hydrofoil and travel some distance downstream. The opposite effect occurred when the condensation term was increased several orders of magnitude, while also lowering the production term; the pocket of vapor was limited to the stagnation point and failed to develop. These extreme cases illustrated the bounds of the cavitation solver and indicate that the incompressible cavitation solver is fairly stable and accurate when provided with reasonable values for the empirical constants.

6.3 Future Work

With the successful verification and validation of the hydrodynamic benchmarks presented in this thesis, the next step is to apply the lessons learned to more complicated geometries and real life problems. The open-source and accessibility of OpenFOAM have allowed the cavitation models to be investigated and have shown the ability to predict cavity lengths for simple test cases. More optimization will need to occur before the long term goal of predicting the exact location of the cavity closure and hence points of erosion on complicated marine propulsor geometries can be realized.

Bibliography

[1] Franc, J. and J. Michel (2004) *Fundamentals of cavitation*, Kluwer Academic Publishers Boston.

[2] McNown, H. R. J. (1945) *Cavitation and Presssure Distribution: Head Forms at Zero Angle of Yaw*, *Tech. rep.*, Iowa Institute of Hydraulix Research.

[3] Abbott, I. and A. Von Doenhoff (1959) *Theory of Wing Sections*, Dover Publications.

[4] Boswell, R. (1971) "Design, Cavitation Performance, and Open-Water Performance of a Series of Research Skewed Propellers." .

[5] Allison, J. (1993) "Marine Waterjet Propulsion," *SNAME Transactions*, **101**, pp. 275–335.

[6] Lindau, J., D. Boger, R. Medvitz, and R. Kunz (2005) "Propeller Cavitation Breakdown Analysis," *Journal of Fluids Engineering*, **127**, p. 995.

[7] Athavale, M., H. Li, Y. Jiang, and A. Singhal (2002) "Application of the Full Cavitation Model to Pumps and Inducers," *International Journal of Rotating Machinery*, **8**(1), pp. 45–56.

[8] CHEN, Y. and S. HEISTER (1995) "Two-phase modeling of cavitated flows," *Computers & fluids*, **24**(7), pp. 799–809.

[9] Kunz, R., D. Boger, T. Chyczewski, D. Stinebring, and H. Gibeling "Multi-phase CFD analysis of natural and ventilated cavitation about submerged bodies," *Proceedings of FEDSM*, **99**, pp. 1999–07.

[10] Robert F. Kunz, D. R. S. T. S. C. J. W. L. H. J. G. S. V., David A. Boger and T. R. Govindan (2000) "A preconditioned Navier-Stokes method for two-phase flows with application to cavitation prediction," *Computers & Fluids*, **29**, pp. 849–875.

[11] Merkle, C., J. Feng, and P. Buelow (1998) "Computational Modeling of the Dynamics of Sheet Cavitation," *3rd International Symposium on Cavitation, Grenoble, France*, **2**, pp. 47–54.

[12] Singhal, A., M. Athavale, H. Li, and Y. Jiang (2002) "Mathematical Basis and Validation of the Full Cavitation Model," *Journal of Fluids Engineering*, **124**, p. 617.

[13] Senocak, I. and W. Shyy (2002) "A Pressure-Based Method for Turbulent Cavitating Flow Computations," *Journal of Computational Physics*, **176**, pp. 363–383.

[14] Wikstrm, N. (2005) "Modeling of Cavitating Flow Around a Stationary/Moving Wing Profile," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, **1287**, p. 13.

[15] Venkateswaran, S., J. W. Lindau, R. F. Kunz, and C. L. Merkle (2002) "Computation of multiphase mixture flows with compressibility effects," *J. Comput. Phys.*, **180**(1), pp. 54–77.

[16] Schmidt, C. J. C. M. L., D. P. Rutland (1999) "A Fully Compressible, Two-Dimensional Model of Small, High-Speed, Cavitating Nozzles," *ATOMIZATION AND SPRAYS*, **9**, pp. 255–276.

[17] Schnerr, G. and J. Sauer (2001) "Physical and Numerical Modeling of Unsteady Cavitation Dynamics," *Proc. 4th Int. Conf. on Multiphase Flow (ICMF 2001)*.

[18] Brajdic-Mitidieri, P. (2005) *Advanced Modelling of Elastohydrodynamic Lubrication*, Ph.D. thesis, Imperial College London.

[19] Kubota, A., H. Kato, and H. Yamaguchi (1992) "A new modelling of cavitating flows: a numerical study of unsteady cavitation on a hydrofoil section," *Journal of Fluid Mechanics*, **240**, pp. 59–96.

[20] Jasak, H. (1996) *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows.*, University of London.

[21] Ubbink, O. (1997) *Numerical Prediction of Two Fluid Systems with Sharp Interfaces*, Ph.D. thesis, University of London.

[22] Hill, D. (1998) *The computer simulation of dispersed two-phase flows*, Ph.D. thesis, University of London.

[23] Brennan, D. and U. of London (2001) *The Numerical Simulation of Two Phase Flows in Settling Tanks*, Ph.D. thesis, University of London.

[24] Rusche, H. and U. of London (2003) *Computational Fluid Dynamics of Dispersed Two-phase Flows at High Phase Fractions*, Ph.D. thesis, University of London.

[25] Karac, A. (2003) *Drop Impact of Fluid-Filled Polyethylene Containers*, Ph.D. thesis, Imperial College, University of London.

[26] Nilsson, H. (2007) "Some Experiences on the Accuracy and Parallel Performance of OpenFOAM for CFD in Water Turbines," *LECTURE NOTES IN COMPUTER SCIENCE*, **4699**, p. 168.

[27] Huuva, T. (2008) *Large Eddy Simulation of Cavitating and Non-Cavitating Flow*, Ph.D. thesis, Chalmers University of Technology.

[28] R.E. Bensow, G. B., T. Huuva (2008) "Large Eddy Simulation of Cavitating Propeller Flows," in *27th Symposium on Naval Hydrodynamics*, Seoul, Korea.

[29] Issa, R. (1985) "Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting," *Journal of Computational Physics*, **62**, pp. 40–65.

[30] Patankar, S. (1981) *Numerical heat transfer and Fluid Flow*, Hemisphere Publishing Corporation.

[31] Edge, B. (2007) *An analytical and numerical study of cavitation scale effects in high-Reynolds number circular jet flows*, Ph.D. thesis, The Pennsylvania State University, University Park, PA.

[32] Edge, B. and E. Paterson (2004) "CFD Simulation of the Human Thermal Wake," *Computers and Fluids*, in preparation.

[33] Craven, B. A., T. Neuberger, E. G. Paterson, A. G. Webb, E. M. Josephson, E. E. Morrison, and G. S. Settles (2007) "Reconstruction and morphometric analysis of the nasal airway of the dog (Canis familiaris) and implications regarding olfactory airflow," *Anat Rec (Hoboken)*, **290**(11), pp. 1325–1340.
URL `http://www.hubmed.org/display.cgi?uids=17929289`

[34] Craven, B., E. Paterson, and G. Settles (2008) "Reverse Engineering Nature's Best Sniffer: The Aerodynamics of Canine Olfaction," *Nature*.

[35] Kunz, R., B. Siebert, W. Cope, N. Foster, S. Antal., and S. Ettorre (1998) "A coupled phasic exchange algorithm for three-dimensional multi-field analysis of heated flows with mass transfer," *Computers and Fluids*, **27**(7), pp. 741–768.

[36] Kunz, R., D. Boger, D. S. T. Chyczewski, J. W. Lindau, H. Gibeling, S. Venkateswaran, and T. R. Govindan (2000) "A preconditioned Navier-Stokes method for two-phase flows with application to cavitation prediction," *Computers and Fluids*, **29**(8), pp. 849–875.

[37] Venkateswaran, S., J. W. Lindau, R. F. Kunz, and C. Merkle (2002) "Computation of Multiphase Mixture Flows with Compressibility Effects," *Journal of Computational Physics*, **180**(1), pp. 54–77.

[38] Medvitz, R. B., R. F. Kunz, D. A. Boger, J. W. Lindau, A. M. Yocum, and L. L. Pauley (2002) "Performance Analysis of Cavitating Flow in Centrifugal Pumps Using Multiphase CFD," *Journal of Fluids Engineering*, **124**(2), pp. 377–383.
URL `http://link.aip.org/link/?JFG/124/377/1`

[39] Lindau, J. W., R. F. Kunz, D. A. Boger, D. R. Stinebring, and H. J. Gibeling (2002) "High Reynolds Number, Unsteady, Multiphase CFD Modeling of Cavitating Flows," *Journal of Fluids Engineering*, **124**(3), pp. 607–616.
URL `http://link.aip.org/link/?JFG/124/607/1`

[40] Lindau, J. W., D. A. Boger, R. B. Medvitz, and R. F. Kunz (2005) "Propeller Cavitation Breakdown Analysis," *Journal of Fluids Engineering*, **127**(5), pp. 995–1002.
URL `http://link.aip.org/link/?JFG/127/995/1`

[41] Kinzel, M., W. Willits, J. Lindau, D. Boger, R. Kunz, and R. Medvitz (2006) "CFD Simulation of Oscillating Hydrofoils with Cavitation," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2006-1279, AIAA, Reno, NV.

[42] Kinzel, M., J. W. Lindau, R. Kunz, L. Peltier, E. Paterson, and R. Noack (2007) "Computational Investigations of Air Entrainment, Hysteresis, and Loading for Large-Scale, Buoyant Cavities," in *9th International Conference on Numerical Ship Hydrodynamics*, Ann Arbor, MI.

[43] Trujillo, M., C. Hsiao, J. Choi, E. Paterson, G. Chahine, and L. Peltier (2007) "Numerical and Experimental Study of a Horizontal Jet Below a Free Surface," in *9th International Conference on Numerical Ship Hydrodynamics*, Ann Arbor, MI.

[44] Senocak, I. and W. Shyy (2002) "A Pressure-Based Method for Turbulent Cavitating Flow Computations," *Journal of Computational Physics*, **176**, pp. 363–383.

[45] Wikström, N. (2005) "Modeling of Cavitating Flow Around a Stationary/Moving Wing Profile," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005-1287, AIAA, AIAA, Reno, NV.

[46] Wallis, G. (1969) *One-Dimensional Two-Phase Flow*, McGraw-Hill.

[47] White, F. (1974) *Viscous fluid flow*, McGraw-Hill New York.

[48] Lee-Rausch, P. M. J. P. M. R. S. R. C. M. D., EM Buning (2003) "CFD Sensitivity Analysis of a Drag Prediction Workshop Wing/Body Transport Configuration," *AIAA Paper*, **3400**.

[49] Jacobs, E., K. Ward, and R. Pinkerton (1933) "The Characteristics of 78 Related Airfoil Sections from Tests in the Variable-Density Wind Tunnel," *NACA Rep*, **460**.

[50] Amick, J. (1950), "Comparison of the Experimental Pressure Distribution on an NACA 0012 Profile at High Speeds With That Calculated by the Relaxation Method," .

[51] Shen, Y. and P. Dimotakis (1989) "The Influence of Surface Cavitation on Hydrodynamic Forces," *Proc. 22nd ATTC, St. Johns*.

[52] Brockett, T. (1966), "MINIMUM PRESSURE ENVELOPES FOR MODIFIED NACA-66 SECTIONS WITH NACA A= 0.8 CAMBER AND BUSHIPS TYPE 1 AND TYPE 2 SECTIONS," .

[53] Lindau, J., S. Venkateswaran, R. Kunz, and C. Merkle (2003) "Computation of Compressible Multiphase Flows," *AIAA Paper*, **1285**, p. 2003.

[54] Liu, L., J. Li, and Z. Feng (2006) "A numerical method for simulation of attached cavitation ows," *Int. J. Numer. Meth. Fluids*, **52**, pp. 639–658.

[55] Avva, R., A. Singhal, and D. Gibson (1995) "AN ENTHALPY BASED MODEL OF CAVITATION," *Cavitation and Gas-liquid Flow in Fluid Machinery and Devices*.

[56] Vaidyanathan, R., I. Senocak, J. Wu, and W. Shyy (2003) "Sensitivity Evaluation of a Transport-Based Turbulent Cavitation Model," *Journal of Fluids Engineering*, **125**, p. 447.