

ECMM102

Group Project (Meng) (A, TRM1+2 2016/7)

004107



1027442



By submitting coursework you declare that you understand and consent to the University policies regarding plagiarism and mitigation (these can be seen online at www.exeter.ac.uk/plagiarism, and www.exeter.ac.uk/mitigation respectively), and that you have read your school's rules for submission of written coursework, for example rules on maximum and minimum number of words. Indicative/first marks are provisional only.

Coursework: Individual contribution to the group achievement

Submission Deadline: Thu 4th May 2017 12:00

Personal tutor: Dr Guangtao Fu

Marker name: G_Tabor

620007506

Word count: 13989

First marker's comments

Indicative mark

Second marker's comments

Second mark

Moderator's comments

Agreed mark



I2 Report

Automatic Mesh Generation for Shape Optimisation of a Draft Tube
Using cfMesh

Samuel Arthur Hardy

2017
4th Year MEng Group Project

I certify that all material in this thesis that is not my own work has been identified and that no material has been included for which a degree has previously been conferred on me.

Signed.....

A handwritten signature in black ink that reads "Samuel Arthur Hardy".

I2 Report

ECMM102

Title: Automatic Mesh Generation for Shape
Optimisation of a Draft Tube Using cfMesh

Word count: 13989

Number of pages: 40

Date of submission: Thursday, 04 May 2017

Student Name: Samuel Arthur Hardy

Programme: MEng Mechanical Engineering

Student number: 620007506

Candidate number: 004107

Supervisor: Professor Gavin Tabor

Abstract

This study was completed as a contribution to the group project '*An Investigation into the Optimisation of a Draft Tube*' [1], using the Turbine-99 draft tube as a case study.

The aim of this study was to optimise a draft tube by increasing its pressure recovery factor to improve the efficiency of turbines in hydropower plants. OpenFOAM 2.3.1 was used as the computational fluid dynamics (CFD) code. cfMesh, implemented within the OpenFOAM framework, was used to develop an automatic boundary layer meshing technique, which was subsequently incorporated into a Bayesian-approach Efficient Global Optimisation (EGO) optimiser, with the objective function of maximising the pressure recovery factor. This allowed a turbulence model study to be conducted, using two-equation Reynold-averaged Navier-Stokes (RANS) turbulence models $k-\varepsilon$, Re-Normalisation Group (RNG) $k-\varepsilon$, $k-\omega$ and the $k-\omega$ shear stress transport (SST) model to establish the optimum turbulence model for draft tube flow, and provide a validation of the simulation results.

The $k-\omega$ models produced the best agreement with experimental results, both on the original geometry and on an exaggerated, high-curvature model. For the exaggerated case, only a 2.6% reduction in modelling performance was found, highlighting the robustness of cfMesh in high-curvature geometries. Therefore, it can be concluded that in coordination with the optimiser, cfMesh provided meshes of adequate quality, as the occurrence of geometries of this curvature is not expected.

Experimental results were not deemed accurate enough to provide confident validation of the numerical modelling, therefore future studies are required.

Keywords: draft tube, optimisation, computational fluid dynamics (CFD), wall functions, OpenFOAM, turbulence modelling.

Acknowledgements

Many thanks to Professor Gavin Tabor for his supervision and advice regarding CFD and the use of OpenFOAM. Special thanks also to Steven Daniels and Alma Rahat for providing support throughout the project, knowledge of OpenFOAM and recommendations concerning optimisation techniques. I would also like to acknowledge the input from Michel Cervantes, who kindly provided the dimensions for the geometry of the original Turbine-99 draft tube.

Table of contents

1.	Introduction and background	1
2.	Literature review	2
2.1.	Early Draft Tube Modelling.....	2
2.1.1.	Mesh Resolution and Discretisation in Draft Tube Simulations	3
2.1.2.	RANS Turbulence Modelling for Draft Tube Simulations.....	4
2.1.	Mesh Generation for Optimisation using CFD	6
2.2.	Objective Function for Draft Tube Performance	7
3.	Theoretical background and Methodology	7
3.1.	Geometry and Strategy	7
3.2.	Objective Function.....	9
3.3.	Boundary Conditions	10
3.3.1.	RANS Two Equation Turbulence Models	10
3.3.1.1.	k- ϵ Turbulence Models	13
3.3.1.2.	k- ω Turbulence Models	13
3.4.	The Boundary Layer: Near Wall Meshing and Wall Functions	14
3.4.1.	Turbulent Kinetic Energy Wall Function	16
3.4.2.	Epsilon Wall Function	17
3.4.3.	omegaWallFunction.....	17
3.4.4.	vt Wall Function.....	18
3.4.4.1.	nutkWallFunction	18

3.4.4.2.	<i>nutUSpaldingWallFunction</i>	18
3.5.	Solution Set-up and Strategy	19
3.6.	<i>snappyHexMesh</i>	20
3.7.	<i>cfMesh</i>	21
3.7.1.	Utilising <i>cfMesh</i> for Optimisation	24
3.8.	Heel Insert	24
3.9.	Inducing Swirl	25
4.	Results and Analysis	26
4.1.	Model 2	30
4.2.	Swirl	32
4.3.	<i>cfMesh</i> for Optimisation	33
5.	Discussion and conclusions	33
5.1.	Conclusion	35
5.1.1.	Further Work	35
6.	Project management	36
7.	Contribution to group functioning	37
8.	Bibliography	37

1. Introduction and background

The modern world is enormously dependent upon electrical energy, largely generated by burning fossil fuels. Not only does this severely impact on the environment, but fossil fuels are also a finite energy resource. As a result, there is a high demand for alternative energy sources that are commercially viable and sufficient to compete with fossil fuel energy sources. One example of a potential renewable energy source to meet this demand is hydroelectric power.

Draft tubes function to improve the efficiency of a hydroelectric power plant, however up to 50% of the total head losses throughout the system can be attributed to poor draft tube performance [2]. Therefore, there is a large degree of research interest into optimising the draft tube design, in order to reduce these losses. Along with this, the high civil costs associated with installing a draft tube mean that they remain in operation for long periods of time, therefore even a minor improvement in efficiency can result in large savings both for industry and consumers over the course of its operational lifetime.

Computational fluid dynamics (CFD) provides a powerful tool for simulating and optimising flow physics, and can be used to model draft tubes with the aim of making optimisation more feasible than through simple analytic methods. On the one hand, CFD offers flexibility to simulate a number of fluid dynamic problems, and is cost-effective. On the other hand, CFD simulations can be computationally expensive, meaning that optimisation tasks take considerable time. Therefore, there is significant interest within the research community regarding machine learning that could automate the optimisation process for these fluid problems.

This study was part of a group project, *An Investigation into the Optimisation of a Draft Tube*, which aimed to optimise the pressure recovery performance of a 1:220 scale version of the Hölleforsen draft tube. The objective of this individual study was to generate automatic meshes, which was achieved using cfMesh, an extension of the OpenFOAM framework. Utilising cfMesh, a novel boundary layer meshing technique was developed, that produced a boundary layer mesh at each evaluation of the optimiser. This provided automatic meshes for CFD simulations to be input into a machine learning environment, using Efficient Global Optimisation (EGO) to maximise pressure recovery of the draft tube. The pressure recovery

factor is a single parameter that determines efficiency.

Draft tubes are challenging to model due to the development of complex flow features, such as turbulence, unsteadiness, separation and secondary flow: swirl is also a feature, however in order to simplify the modelling process, this study did not introduce swirl conditions. To validate the CFD modelling of the complex flow, experimental results were provided. These results formed the basis of a Reynolds-averaged Navier-Stokes (RANS) turbulence model study, completed in order to establish the performance of the automatic meshes across a number of turbulence models and to identify the most applicable turbulence model for this fluids dynamics problem.

This report includes a discussion of the relevant available literature, summarising the background for this study. Secondly, an introduction to the theoretical background and methodology of the project will be provided, where the cfMesh technique is introduced. Following this, the results are presented and subsequently analysed, leading to a critical discussion of the merit of the work achieved. Finally, conclusions are drawn and suggestions for future work are made.

2. Literature review

2.1. *Early Draft Tube Modelling*

To better simulate the internal flow physics of a draft tube, a number of studies have been undertaken to establish a best practice guideline for simulations of draft tube flow. A significant contribution is that of the participants of Turbine-99 draft tube workshops in Porjus 1999 [3], Älvkarleby 2001 [4] and Porjus 2005 [5]. For many applications of draft tubes, the draft tube itself has to be bent to limit excavation cost and to redirect the flow, a typical case being the Turbine-99 sharp heel draft tube [6] - this design reduced manufacturing costs and also aimed to limit cavitation. The shape of the draft tube can be seen in Figure 1 (page 8, left image) where a typical cylindrical diffuser is connected to the runner casing of the turbine and a rectangular outlet, typical of older hydropower plants. The choice of the Turbine-99 draft tube was due to its representation of approximately 20 existing industrial draft tube designs; in addition, its sharp cross sections allow for optimisations of the geometry [6]. The model

featured across all three conference proceedings was a 1:11 scale model of the Hölleforsen power station draft tube.

Early simulations of the Turbine-99 case highlighted the challenges of draft tube modelling, as separation caused problems for the experiments and for CFD simulations [7]. A study by Gublin [8] examined the flow physics and concluded that swirl, although normally attributed to reduced performance, can aid in reducing separation due to the radial forces induced in the flow, forcing the flow toward the wall, thus reducing the boundary layer thickness.

Initial studies presented in the first 1999 workshop [3] found large discrepancies between the experimental and simulated values for pressure recovery [6] [7] [9]: it was concluded that due to the complex flow within the draft tube, further analysis of simulations was required to understand the flow physics. Another key finding from the initial workshop in Porjus 1999 was that a single parameter, the pressure recovery factor, was considered as the most important factor when determining draft tube performance, taking priority over the energy loss coefficient [10]. The initial findings from the thesis [10] highlighted the great variety in flow patterns experienced at the outlet: this was attributed to small variations within the elbow which created large differences downstream of the elbow. Most importantly, the majority of pressure recovery was demonstrated to occur within the draft tube cone, before the elbow [11].

A study by Raisee and Alemi [12] discovered some important information regarding the development of turbulent flow through 90° ducts. It was found that square and rectangular cross-sections of curved ducts induce strong secondary motion: for the rectangular cross-section this secondary flow was confined to the corners. The effects of secondary flow were also experienced in the downstream flow, however slowly deteriorating with the main flow.

2.1.1. Mesh Resolution and Discretisation in Draft Tube Simulations

After the first two conference proceedings of the Turbine-99 workshops, Andersson *et al* [13] discussed the significance of grid issues on simulation results. The inlet required an adequate grid resolution to resolve the physics of the flow, and in order to capture the secondary flow physics generated at the sharp heel, a refinement of the mesh was also required. Such resolution requirements resulted in difficulty in obtaining adequate y^+ values throughout the entire domain. In addition, due to the large difference in velocities at the near wall, the y^+ criterion

for each of the various models is frequently violated; it was concluded that if wall functions are used, hexahedral elements are advised in the near wall region, and if applicable, wall functions should be replaced with near wall models [14]. Finally, the mesh generated for the workshop had imperfections in the geometry within the inlet, resulting in inaccuracies in the simulations which continued throughout the entire draft tube solution.

In order to assess the ability of OpenFOAM open-source CFD, Nilson [15] compared results obtained by OpenFOAM with other CFD codes such as CFX-5 and concluded that OpenFOAM produced results that not only compared well with CFX-5 but also with experimental data, when using a combination of the $k-\epsilon$ turbulence model and the Gamma discretisation scheme. It was mentioned that although OpenFOAM was unable to dampen the unsteadiness of the flow, a benefit of OpenFOAM is that as a result, it does not over-predict diffusion.

Due to the complex flow generated within the draft tube, Andersson *et al* [13] suggested the use of higher-order schemes so that the momentum equation accurately captures the physics of the flow. Gunnar and Hellström [16] suggested the use of second-order schemes, however they used first-order schemes to provide initial results. Their findings also showed only a small difference in the pressure recovery factor between steady and unsteady simulations when using the standard $k-\epsilon$ model, thus it was concluded that for the small change in results, it is more advantageous to use steady state simulations, as computational cost can be reduced.

During the third Turbine-99 workshops two meshes were available, one being of $y^+ = 50$ and one at $y^+ = 1$: a later study by [17] compared the two meshes with the standard, RNG and realisable $k-\epsilon$ turbulence model and found that the $y^+ = 1$ mesh displayed significant differences in the results when near wall options are used.

2.1.2. RANS Turbulence Modelling for Draft Tube Simulations

During the second Turbine-99 workshop [4], Grotjians [18] concluded that the $k-\epsilon$ model correctly computed the flow field, where convergence of both first- and second-order discretisation schemes was fast. Lai *et al*'s [19] simulations showed significant differences in the streamline patterns of the $k-\epsilon$ and $k-\omega$ simulations, which developed differences in pressure distributions and secondary flow physics. The turbulence investigation by Gunnar and Hellström [16] revealed not only convergence issues with the $k-\omega$ SST model but also that the

results obtained by the k- ε model showed greater agreement with the experimental data than previous simulation results of by Marjavaara [20], on a matching geometry.

During the third Turbine-99 workshop [5] participants were provided with improved boundary conditions and meshes. Cervantes and Engström [11] compared the use of the k- ε model and the k- ω SST and also experienced issues with second-order scheme convergence of the k- ω SST model. Little was concluded as to the cause of the lack of stability and it was attributed to the unsteady flow within the draft tube; however, results showed little quantitative difference in the pressure recovery for each model. Marjavaara et al [21] and Marjavaara and Lundstrom [22] investigated more optimal draft tube designs where a small radius was added to the heel, with the use of the standard k- ε and k- ω SST models. The two models showed significant differences in the pressure recovery and energy loss factors, yet provided similar flow physics. Similar results were found in Tokyay and Constantinescu [23] study, whereby the standard k- ε and k- ω SST models showed significant differences in the main flow vortices, which in this study produced significantly different values of pressure recovery: this study also investigated the use of Large Eddy Simulations (LES) and found that, due to computational expense, the Reynolds number had to be reduced, thus the use of LES was unable to be quantitatively compared with RANS modelling. LES simulations were however achieved by Su *et al* [24] who found that, when compared to the standard k- ε model, LES shows better agreement with experimental results than the RANS model. During Marjavaara *et al*'s [25] turbulence study, it was found that the use of the k- ω SST model provided the closest results to experimental data during first-order steady state calculations, and found that as they experimented with second-order discretisation schemes, there was no correlation between the pressure recovery values generated by the two CFD codes (CFX-5 and Loci-STREAM), when using RANS turbulence models, highlighting the unpredictable flow physics generated within draft tubes.

Foroutan and Yavuzkurt [26] investigated steady state simulations using both the standard k- ε model and the k- ω SST model on a smooth-heeled draft tube. Both of the steady simulations using RANS models under-predicted the velocity along the centreline of the geometry, yet the k- ω SST model yielded more accurate results compared to experimental data. During Galvan *et al*'s [17] assessment of the standard, RNG and realisable k- ε turbulence models, the RNG turbulence model's performance was affected by the discretization scheme and the use of near wall functions, however, when a lower resolution mesh was used, the quantitative difference

in static wall pressure was small. A previous study assessing the k- ϵ turbulence models by Galvan *et al* [27] found that although sensitivity to discretisation schemes was higher for the RNG k- ϵ model, more precise details about the flow can be assessed using this model.

2.1. Mesh Generation for Optimisation using CFD

A number of techniques have been used within CFD to regenerate and deform geometries to strive for optimisation. The most common geometric alterations to grids are achieved through parameters and algorithms that modify the original grid structure, such as Reuther *et al* [28], who used a CFD solver in conjunction with a mesh perturbation algorithm to explore the optimal design space. In order for the original mesh to be modified and to generate a mesh with sufficient quality, blending functions are being introduced into existing grid generation software [29]. An example is Soni's [30] mesh adaption processes, the most crucial element of his research being the redistribution of existing mesh points and using algebraic functions to refine and improve orthogonality within the mesh.

Successful attempts have been made to integrate optimisation techniques into computer-aided design (CAD) software [31], where shape perturbations could be obtained from the CAD software and used to generate modified grids, however this technique requires the use of multiple software platforms. Leblond *et al* [32] also used CAD to alter the geometry and subsequently the mesh; however, as the geometry was cut to create perturbations, the engineer had to specify a valid cutting surface for each optimisation loop in order to create an appropriate boundary mesh, which was not cost effective.

A number of studies have used OpenFOAM to undertake CFD optimisation, using a variety of mesh deformation methods, without the requirement of a CAD model within the optimisation loop. A typical example is a study by Petropoulou [33] who utilised the built-in adjoint solver within OpenFOAM and combined this with a mesh-morphing process. The original geometry is used to create a mesh, which is then deformed at each iteration of the optimisation loop, whereby the deformation is based upon stretching or contracting node locations. The degree of freedom is defined by a finite number of control points that move in the x,y and z directions. Othmer *et al* [34] used morphing information extracted from ANSA to utilise OpenFOAM's meshing and adjoint solver for optimisation. However, there are a number of pre-existing mesh generation and deformation facilities within OpenFOAM. Daniels *et al* [35] highlighted the

use of techniques such as the immersed boundary and moving boundary method and highlighted that these techniques can compromise skewness and orthogonality of cells. For Daniels *et al*'s research [36], researchers used an OpenFOAM automated meshing generator called snappyHexMesh, a three stage process of castellation, snapping and boundary layer refinement, where the user inputs a Stereolithography (STL) geometry file to form the base mesh. Cells on the edge of the STL surface are identified: if they lie outside of the domain these cells are subsequently removed. Although the mesh was generated automatically, quality controls were adjusted using a variety of settings. It must be noted however that the downfall of Daniels *et al*'s use of snappyHexMesh was that the geometry setup was such that it was only able to remove sections from the base geometry, ultimately limiting the optimiser's exploration space.

2.2. *Objective Function for Draft Tube Performance*

A study by Galván *et al* [37] analysed the most appropriate objective function and revealed that the energy loss coefficient factor is highly sensitive to inlet conditions; this finding concurs with the findings from Marjavaara and Lundström [38] who found that the energy loss coefficient was difficult to predict and thus for their application of shape optimisation, such difficulty led to the conclusion that pressure recovery factor is a better measure of draft tube efficiency [5]. It was found during a study by Karlsson *et al* [14] that a source of uncertainty for the energy loss coefficient is that the evaluation of the energy loss factor was ill-conditioned and therefore a geometric optimisation based upon the energy loss factor would not be advised.

3. Theoretical background and Methodology

3.1. *Geometry and Strategy*

A significant number of papers and information have been provided on the Turbine-99 base case, a 1:11 scaled model of the Höllerforsen power station draft tube. As this is a base case for draft tube simulations, this geometry was also chosen for this study. The first stage of the project was to generate a surface geometry Stereolithography (STL) file, as the chosen mesh generation techniques used in this study require STL file formats; this was achieved using SolidWorks (a Dassault Systèmes 3D computer-aided design software). The dimensions of the

geometry were provided from Cervantes at Luleå University of Technology, a key member and organiser of the Turbine-99 workshops, as fully dimensioned drawings were not available. It was decided to experiment using 3D-printed geometries to aid manufacturing for the complex optimised designs; however, due to printing facilities available within the University of Exeter Engineering department, a 1:20 scale of the previously simulated 1:11 scale model was used (the final model was a 1:220 scale). The flow rate used within the third Turbine-99 workshop was $0.522 \text{ m}^3\text{s}^{-1}$: due to the flow bench and electronic equipment tolerances available, a flow rate of 0.5 Ls^{-1} was used. The flow rate of this study was significantly lower, therefore results are not comparable with the results obtained from the Turbine-99 workshops.

The primary objective of the optimisation task was to create a functional work product, with the aim of subsequently increasing accuracy, therefore a simplification of the optimisation was required. It was decided that the optimisation task would be limited to the optimisation of the diffuser section of the draft tube. This is contrary to the flow theory outlined in the literature that suggests that much of the pressure recovery is experienced within the cone of the draft tube [11]. The search space was also limited by pressure tappings at the outlet of the experimental models, and the inlet and outlet cross-sections were fixed to mimic industrial constraints. These factors not only affect the potential increase in pressure recovery but also require the accuracy of simulations to be high, as the percentage change in performance could lie within the simulation error, misleading the optimiser.

In order to validate the numerical modelling, turbulence and mesh sensitivity studies were used to assess the performance of the novel boundary layer automatic mesh generation on two geometries. Model 1 is the original, 1:220 scaled model of the Hölleforsen draft tube, and Model 2 shows an exaggerated geometry aiming to test the performance of the automatic meshes generated on a high-curvature geometry, a ‘worst case’ scenario for the meshing techniques.



Figure 1: Experimental draft tube models. Original geometry (left) and Model 2 (right).

For each of the simulations completed within this study, as well as the optimisation run, the computer specifications can be found in Table 1.

Table 1: Computer specifications used for simulations

2 Processors	12 Cores	Intel® Xenon® CPU 3.47GHz	99GB RAM	OS Linux	
--------------	----------	---------------------------	----------	----------	--

3.2. Objective Function

As previously stated, up to 50% of a low head operation hydropower plant's total losses can be attributed to the draft tube [2]: engineering quantities such as the pressure recovery factor and energy loss factor can be analysed to reduce total losses of the operation. The objective function of this project's optimiser was to maximise the pressure recovery within the draft tube, as this can provide a better measure of efficiency [5]. Internal pressure measurements can be taken directly during CFD simulations, however it is difficult to generate accurate internal pressure measurements in experiments, therefore wall pressures are used to determine the experimental pressure recovery coefficient.

The pressure recovery factor C_p given by Equation 1 is a dimensionless number based upon the averaged static pressure across the inlet section, P_{in} , and the averaged static pressure across the outlet section, P_{out} . The pressure recovery factor denotes the level of conversion of kinetic energy into static pressure. The wall pressure recovery factor is given by:

$$C_p = \frac{P_{out} - P_{in}}{\frac{1}{2}\rho \left(\frac{Q}{A_{in}}\right)^2} \quad (1)$$

where A_{in} is the area of the inlet cross-section, Q is the flow rate and ρ is the density of the fluid.

The optimiser within the project is formulated such that the optimisation function is set to maximise the value of C_p . In this study, the outlet boundary condition was set as a zero pressure condition; this allowed recirculation flow at the outlet. The guage pressure at the inlet cross-section of the draft tube will always be negative, therefore the maximisation of C_p results in the decrease in pressure at the turbine outlet, increasing the work done across the turbine.

3.3. Boundary Conditions

Many of the papers listed within Section 2 induce swirl at the inlet to simulate turbine motion, however, as this is an optimisation task, swirl was not induced, with the aim of reducing the complexity of the flow and reducing computational expense. Previous studies have also used both smooth and rough walls: for this study the walls were set as smooth, to again limit computational expense.

The draft tube simulations within this study were compared to experimental data taken by Angus [39] and Dye [40]. The experimental flow rate was limited to approximately 0.5 Ls^{-1} as the pressure sensors used to measure such data had an accuracy tolerance up to 0.52 Ls^{-1} . The initial conditions were therefore set to match the given flow rate; this flow rate generated an approximate Reynolds number (Re) of 20,600 based upon the inlet length.

$$Re = \frac{U_{ref}L}{\nu} \quad (2)$$

where U_{ref} is the fluid velocity, L is the characteristic length and ν is the kinematic viscosity. For this case the boundary conditions and flow properties velocity was set as follows: $U = 1.1386 \text{ ms}^{-1}$ $L = 2.36 \times 10^{-2} \text{ m}$ and $\nu = 1.307 \times 10^{-6} \text{ m}^2 \text{s}^{-1}$. The characteristic length was set as the inlet diameter; this is because this length scale was deemed to provide the turbulent characteristics of the flow. The outlet boundary condition was set as a zero pressure condition, this allowed recirculation flow at the outlet.

3.3.1. RANS Two Equation Turbulence Models

As turbulence models are required to simulate the flow, RANS models are used, which are two equation models based upon the Boussinesq approximation, that modify the Navier-Stokes equations.

Within the literature, RANS models have provided results in concordance with the experimental results [25]. This study used the two-equation RANS turbulence models k- ε , Re-Normalisation Group (RNG) k- ε , k- ω and the k- ω shear stress transport (SST) model to assess the turbulence model performance of the automatic meshes generated using cfMesh, (see Section 3.7).

RANS models provide the most computationally-economic method for simulating turbulent flows in industry. The models provide time-averaged Navier-Stokes equations where a flow variable, such as velocity u , is decomposed into a mean time-averaged component U and a fluctuating component u' . For incompressible steady flow, the equation [41] can be given as:

$$\rho \bar{u}' \frac{\partial \bar{u}_i}{\partial x_j} = \rho \bar{f}_i + \frac{\partial}{\partial x_j} \left[-\bar{p} \delta_{ij} + \mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \rho \bar{u}' u'_j \right] \quad (3)$$

$\rho \bar{u}' \frac{\partial \bar{u}_i}{\partial x_j}$ signifies the change in mean momentum of the fluid due to the convection and unsteadiness within the mean flow. This momentum change is equal to the change in mean body force $\rho \bar{f}_i$, the isotropic stresses due to the mean pressure $-\bar{p} \delta_{ij}$, the viscous stresses $\mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right)$ and the stress due to the fluctuating velocity field $\rho \bar{u}' u'_j$. The fluctuating velocity field, called the Reynold stress, requires modelling to evaluate this term; this can be achieved using turbulence models [41].

All two equation models by definition contain two extra transport equations to simulate the turbulent physics of the flow and to model the Reynolds stresses using the eddy viscosity approach. The transported variables are turbulence kinetic energy k , turbulence dissipation ε , and the specific turbulence dissipation rate ω .

Within this study, k was calculated using:

$$k = \frac{3}{2} (U_{ref} T_i)^2 \quad (4)$$

where U_{ref} is the inlet flow velocity and T_i is the turbulence intensity. For high turbulence cases with complex flow, a typical turbulence intensity of between 5% and 20% is expected [42], therefore a value of 0.1 was selected for the value of T_i in this study, resulting in $k = 0.0195 \text{ m}^2 \text{s}^{-2}$.

This can then be used to calculate the estimated turbulence dissipation (ε):

$$\varepsilon = C_\mu^{3/4} \frac{k^{3/2}}{l} \quad (5)$$

Where $l = 0.07L$, $C_\mu = 0.09$ is a coefficient from the k- ε turbulence model and L is the characteristic inlet scale. From the values given above, $\varepsilon = 0.2692 \text{ m}^2\text{s}^{-3}$.

In order to use the k- ω and k- ω SST model, a value of ω was calculated. Omega denotes the rate at which thermal internal energy is produced by conversion of turbulence kinetic energy, per unit volume and time. Omega can be defined implicitly in Equation 6 as the specific turbulence dissipation rate, using both the turbulence dissipation and the turbulence kinetic energy, providing a value of 13.84s^{-1} as follows:

$$\omega = \frac{\varepsilon}{k} \quad (6)$$

The Boussinesq eddy viscosity assumption is the basis for two equation models, which state that the Reynolds stress is proportional to the viscous stress:

$$-\rho \overline{u'_i u'_j} = \mu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (7)$$

Where μ_t is the eddy viscosity that is computed by the two transported variables in the chosen turbulence model. For incompressible flow, the last term is included to make sure that the definition of turbulence kinetic energy is maintained [43].

Both the strength and the weakness of two equation models is the Boussinesq assumption. Flow theory for simple flows suggests that the Reynolds stress tensor is proportional to the mean strain rate tensor, however in complex flow with separation, strong curvature, rotation and stagnation regions, the Boussinesq assumption is not valid. The benefit of the assumption is that the effects of molecular viscosity in laminar flow can be interpreted similarly to the effect of turbulence in the main flow, removing the requirement to solve for each of the six turbulent stresses independently, thus significantly reducing computational expense, which is vital for this optimisation study. Investigation of turbulence models within this study are aimed at minimising the resulting effects of this approximation and provide validation of the CFD work function.

3.3.1.1. $k-\varepsilon$ Turbulence Models

The standard $k-\varepsilon$ model was first utilised within this project to provide understanding of the flow physics, as it has been shown to provide adequate accuracy for draft tube modelling and reasonable computational speed [11]. This two equation model solves for both the turbulence kinetic energy and the turbulence dissipation. For internal, wall-bounded flows the standard model has shown good results, however this is only for cases where the mean pressure gradients are small [44]. The $k-\varepsilon$ models typically under-predict separation and also delay the onset of separation. The RNG $k-\varepsilon$ model was introduced to provide extra sophistication to the standard $k-\varepsilon$ model by introducing anisotropic forces in the boundary layer. The eddy viscosity in the standard model is calculated from a single turbulence length scale however in fluid flow, there are numerous scales of motion and the RNG $k-\varepsilon$ modifies the epsilon equation in an attempt to account for different turbulence length scales [45].

3.3.1.2. $k-\omega$ Turbulence Models

In the standard $k-\omega$ model, the two transport variables that are solved are k (turbulence kinetic energy) and ω (specific turbulence dissipation rate). The standard $k-\omega$ model is able to model turbulent flow in the viscous sub-layer without any additional terms; this makes the model robust as it is less sensitive to y^+ values (see Section 3.4). The result of the formulation is that the $k-\omega$ models are better at predicting boundary layer flow with adverse pressure gradients and separation. The models do however show significant sensitivity to the free stream values of both k and ω .

Another formulation of the $k-\omega$ model is the SST model of the $k-\omega$ turbulence model. This turbulence model is a hybrid of the $k-\omega$ model used within the near wall region and the $k-\varepsilon$ in the free stream. As the $k-\omega$ model's formulation can model the inner regions of the boundary layer, all the way into the viscous sub-layer, it can better simulate flows with adverse pressure gradients and separating flow than the $k-\varepsilon$ model, yet avoid the $k-\omega$ sensitivity to turbulent conditions in the free stream. The $k-\omega$ model still produces heightened turbulence levels in regions of strong acceleration and stagnation, however this is an improvement from the $k-\varepsilon$ model.

3.4. The Boundary Layer: Near Wall Meshing and Wall Functions

Wall functions are present within CFD to model the strong shear gradients that appear at no-slip walls; these gradients appear as viscous effects and affect the flow structure. Numerically, these viscous effects can be modelled through the use of wall functions where empirical formulas are used to prescribe conditions at wall boundaries, without the requirement to resolve the boundary layer with fine meshes, and eliminating the requirement for turbulence models to calculate the viscous effects, thus reducing CPU time. This reduction of CPU time was critical to the success of the project as the optimiser would require significant computational expense.

Through both mathematical and experimental analysis of turbulent flat-plate boundary layers, it has been found that turbulent boundary layers are split into a number of sub layers, these being the viscous sub-layer, buffer layer and the log-law region.

Wall distance is a dimensionless number identified as y^+ ; this value of y^+ denotes the distance to the first cell of a mesh structure. In order to achieve consistent y^+ values throughout the domain, a boundary mesh is used. A boundary layer mesh consists of a number of layers of ideally hexahedral cells expanding in the direction normal to the wall boundary. In order to analyse the flow close to the wall, a number of viscous scales are used and defined in Table 2, where ρ is density, U the freestream velocity and Re is the Reynolds number.

Table 2: Viscous scales

Skin Friction	Wall Shear Stress	Friction Velocity	Viscous Velocity	Wall Units
$C_f = 0.079Re^{-0.25}$ [11]	$\tau_w = \frac{1}{2}C_f\rho U^2$	$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$	$u^+ = \frac{U}{u_\tau}$	$y^+ = \frac{u_\tau y}{v}$

The layer nearest the wall is a viscous sub-layer, where laminar flow is experienced. In this region, the Reynolds stress is zero and the molecular viscosity dominates the momentum of the fluid [16]. Within the viscous sub-layer, where $y^+ < 5$, the velocity follows a linear relationship with the increase in wall distance:

$$u^+ = y^+ \quad (8)$$

As the perpendicular distance y from the wall increases, the next region of a typical wall boundary layer is the buffer layer, approximately $5 < y^+ < 30$; this buffer layer provides an

intermediate stage between the viscous sub-layer and the main turbulent flow. The largest area of variation between the viscous sub-layer and the buffer region occurs where their respective equations overlap, at approximately $y^+ = 11$. It can be said that before a wall distance of $y^+ = 11$ the linear approximation is more valid, and beyond $y^+ = 11$, the logarithmic law should be applied [46] (see Figure 2). This is also described in Sections 3.4.1 - 3.4.4 when discussing OpenFOAM implementations of wall functions.

Finally, the logarithmic layer, where turbulence dominates, as the inertial forces of the flow dominate the viscous forces generated at the boundary. Within this region, standard wall functions can numerically compute the shear and viscosity effects as a function of the logarithmic profile of velocity:

$$u^+ = \frac{1}{\kappa} \ln y^+ + B \quad (9)$$

Where κ is the von Kármán's constant of 0.41 and B , a well-known constant generally taken as 5.2 for high Reynolds number flow. This logarithmic region is apparent for $30 < y^+ < 300$.

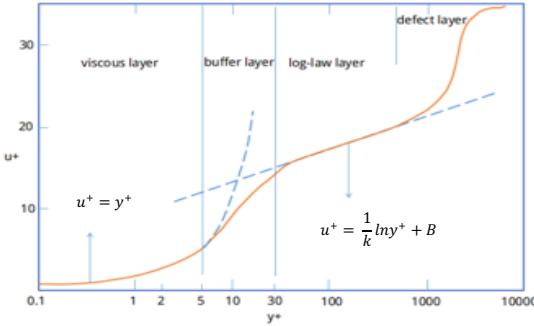


Figure 2: Layer Wall Functions [47]

The findings of Anderson *et al* [13] and [14] highlighted how draft tube modelling often results in void turbulence models due to the range in values of wall velocities, resulting in invalid y^+ values for the given turbulence model used. This study employed a number of wall functions and turbulence models: this was carried out to manage a number of difficulties experienced with mesh generation due to the size of the model used (1:220 scale), see Section 3.7. The boundary layer thickness of the internal flow within the 1:220 scale draft tube was a large percentage of the internal flow area and therefore when aiming to produce large enough y^+ values to provide valid turbulence models, the overall mesh resolution is reduced, resulting in a mesh that is potentially unable to capture the flow accurately.

Simulations were run with existing OpenFOAM wall functions to find the most accurate function to simulate this complex flow, and the OpenFOAM implementations are found in Sections 3.4.1 - 3.4.4. The nutkWallFunction and nutUSpaldingWallFunction were compared with each mesh used within the study and finally, a low Reynolds number simulation of each of the turbulence models. Due to the complexity of the flow described in the literature, it was suggested to use mesh densities that resolve the boundary layer [14]; however, in order to achieve the objectives of the study, wall functions were a priority as this would significantly reduce the computational time of the optimiser.

3.4.1. Turbulent Kinetic Energy Wall Function

Within the project, two available k wall funtions were used, the kQRWallFunction and kLowReWallFunction. The kQRWallFunction is implemented for use in high Reynolds number situations, where it returns a zero gradient condition; however, the kLowReWallFunction is implemented such that it can be used for both high and low Reynolds number applications, so only the kLowReWallFunction will be explained [47]. The low Reynolds number implementation can operate in two modes, by computing the y^+ value at which the laminar flow becomes turbulent yPlusLam: this calculation of the laminar-turbulent y^+ condition allows the first cell centre to be located within the buffer layer.

The OpenFOAM source code calculates the value of yPlusLam through a ten iteration loop, where the value of kappa (κ) is the von Kármán constant and E is the roughness coefficient of 9.8, the resulting y^+ is approximately 11.5. Upon calculating the y^+ value, the calculation of k as a function of wall distance is given by:

For $y^+ > y\text{PlusLam}$

$$k^+ = \frac{C_k}{\kappa} \times \log(y^+) + B_k \quad (10)$$

where $C_k = -0.416$, $B_k = 8,366$ and $C_{eps2} = 1.9$ are default in OpenFOAM.

When $y^+ < y\text{PlusLam}$

$$k^+ = \frac{2400}{C_{eps2}^2} \times C_f \quad (11)$$

where C_f is now calculated as

$$C_f = \frac{1}{(y^+ + C)^2} + \frac{2y^+}{C^3} - \frac{1}{C^2} \quad (12)$$

Following this, k can be computed from the non-dimensional y^+ on each cell face

$$k = k^+ \times u_\tau$$

3.4.2. Epsilon Wall Function

The epsilon wall function fundamental is the weighting function used. The weight is described by the number of faces of one cell that will use the boundary condition; for example, if n faces of a cell use the wall function boundary condition, the weight of the cell is n . The epsilon value is calculated at the n faces and summed to provide the cell centre epsilon value.

$$\varepsilon_{centre} = \frac{1}{n} \varepsilon_1 + \frac{1}{n} \varepsilon_2 \dots \frac{1}{n} \varepsilon_n \quad (13)$$

$$\varepsilon = \frac{1}{n} \sum_{f=i}^n \left(\frac{c_\mu^{3/4} k^{3/2}}{\kappa y_i} \right) \quad (14)$$

Similar to the turbulence kinetic energy low Reynolds number wall function implementation, the `epsilonLowReWallFunction` uses a y^+ calculator to determine the cell centre epsilon value.

When $y^+ > y_{PlusLam}$ the `epsilonWallFunction` is employed, however when $y^+ < y_{PlusLam}$

$$\varepsilon = \frac{1}{n} \sum_{f=i}^n \left(\frac{2k v_i}{y_i^2} \right) \quad (15)$$

3.4.3. omegaWallFunction

The omega function is used to determine the turbulence specific dissipation rate. The benefit of OpenFOAM's implementation of the `omegaWallFunction` is that it can blend between the logarithmic region and the viscous sub-layer, again based upon a calculated y^+ value. Menter's wall function [48] calculates the omega value at the face of each cell (like epsilon), as its value is known in both the viscous (ω_{vis}) and logarithmic region (ω_{log}), and these face values are blended to calculate the omega value for the cell centre.

$$\omega_{vis} = \left(\frac{6.0v}{\beta_1 y^2} \right) \omega_{vis} \quad (16)$$

$$\omega_{log} = \left(\frac{k^{1/2}}{c_\mu^{1/4} \kappa y} \right) \quad (17)$$

where OpenFOAM's default value of $\beta_1 = 0.075$ and blending to form Equation 18.

$$\omega = \sqrt{\omega_{vis}^2 + \omega_{log}^2} \quad (18)$$

This blending function is more powerful than an abrupt change in the solution, based on y^+ , however it must be noted that this cannot correctly represent the buffer layer; this is currently unavailable in any OpenFOAM wall function.

3.4.4. ν_t Wall Function

3.4.4.1. nutkWallFunction

OpenFOAM's nutkWallFunction provides a wall boundary condition for the turbulence viscosity. The nutkWallFunction used within this study uses the turbulence kinetic energy to provide a turbulence viscosity. This standard wall function changes its turbulence viscosity calculation based upon y^+ values.

When $y^+ > y_{PlusLam}$

$$\nu_t = \nu \left(\frac{\kappa y^+}{\ln(Ey^+)} - 1 \right) \quad (19)$$

and when $y^+ < y_{PlusLam}$ the turbulence viscosity is set to zero. As there is no blending between the two functions, it is important to maintain y^+ values in the logarithmic region when using this wall function.

3.4.4.2. nutUSpaldingWallFunction

The nutUSpaldingWallFunction provides a more advanced relationship, where the turbulence viscosity is based upon the wall velocity of the fluid. The wall function's viscosity calculations fit that of the boundary layer theory, whereby in the viscous sub-layer region $u^+ = y^+$ and in the logarithmic region $u^+ = \frac{E y^+}{\kappa}$, this fitted curve is given by Equation 20 [47]:

$$y^+ = u^+ + \frac{1}{E} \left[e^{\kappa u^+} - 1 - \kappa u^+ - \frac{1}{2} (\kappa u^+)^2 - \frac{1}{6} (\kappa u^+)^3 \right] \quad (20)$$

The value of y^+ is then input into Equation 21, where the turbulence viscosity is calculated. The returned value is then the maximum of zero or the value given from Equation 21.

$$\nu_t = \frac{(u_\tau)^2}{\frac{\partial U}{\partial n}} - \nu \quad (21)$$

As this calculates the turbulence viscosity down to $y^+=1$, this wall function was used to avoid the potentially void nutkWallFunction, due to the challenges of generating y^+ values (see Section 3.7); therefore, this wall function was used for all turbulence models.

3.5. Solution Set-up and Strategy

Within the literature issues with second-order convergence of the $k-\omega$ SST model were noted [16], this became an import factor when determining a valid solution for the Bayesian optimiser. Convergence issues were also present within this study, where second-order simulations could only achieve residual values of approximately 1×10^{-2} , therefore first-order simulations were used. First-order simulations are more dissipative and therefore less sensitive to mesh quality, whereas in second-order schemes, a single cell value propagates further into the solution. Therefore, if a mesh has a region of lower quality, it will more greatly affect the solution. As complex geometries may be created by the optimiser, resulting in the potential for lower quality meshes, the use of first-order schemes within this study was confirmed. The solution strategy for the Pointwise and blockMesh meshing of Gowans [49] and Burns [50] in the group project was different as their studies used second-order schemes and as a result, experienced higher accuracy unstable simulations with convergence residual values of 1×10^{-2} .

Different solvers are available within OpenFOAM and are selected in the fvSolution dictionary. Two solvers are used within this study, however the GAMG (Geometric Agglomerated Algebraic Multigrid) solver was used as the primary solver, over the PCG (Preconditioned Conjugate Gradient) solver. The PCG was selected from a default OpenFOAM tutorial, as the solver is commonly known to be stable and sufficiently accurate for many applications, and on the contrary, the GAMG provides a more accurate solution that compromises stability. However, when using the $k-\omega$ SST model in this study, the simulations would frequently fail after up to 900 iterations of the simpleFoam simulation when using the PCG solver. It is common for simulations to fail in the initial stages of the simulation, as initial conditions and significant developments within the flow are experienced. However, as simulations were failing at a later stage, it was postulated that the PCG, due to its reduced accuracy, would divert the solution, resulting in instability. Changing to the GAMG solver resolved these instability

issues as the accuracy of the solution potentially provided improved convergence to the flow solution.

Within the fvSolution dictionary, the convergence residuals can be set. For this study, the residual for pressure, velocity, k and epsilon were set to 1×10^{-5} . In order to aid convergence, potentialFoam was used to initialise the flow field, then OpenFOAM's SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) simpleFoam to conduct steady state solutions for incompressible turbulent flow.

3.6. *snappyHexMesh*

During Daniels *et al*'s study, an OpenFOAM utility called snappyHexMesh was used as an automatic meshing technique to optimise the PitzDaily case. The snappyHexMesh utility takes Stereolithography (STL) files, which are triangulated surface meshes, and generates a hexahedra based (hex and split hex) three-dimensional mesh based on an arbitrary geometry [51]. STL files of the draft tube geometry were generated using SolidWorks and were provided in ASCII format. In order to limit the faceting of the geometry surface, the tolerances of the 'deviation' and the 'angle' fields were reduced to the minimum, reducing the distance and the angle deviation between the flat triangulated STL surfaces and the true modelled surface. In order to use the snappyHexMesh utility, a larger mesh surrounding the geometry is required, which is then castellated and finally smoothed, referred to as 'snapped', to the surface during mesh generation. Either an external or internal flow can be simulated; a location within the mesh to be retained is specified and the cells outside this region are removed. In this case, a location within the draft tube was specified, creating an internal mesh of the draft tube. Addition of boundary layers are achieved through refinement, in order to maintain consistent y^+ values throughout the domain. Boundary layer refinement generates layers of hexahedral cells that inflate normal to the surface. The snappyHexMesh utility is a robust and fast meshing technique that can also be decomposed to run simulations using MPI (Message Passing Interface) distributed memory parallelisation. Meshes can be generated within quality constraints set in the meshQualityControls dictionary: a number of these quality controls were investigated within the study, such as the non-orthogonality, aspect ratio and skewness. The mesh quality can be checked using the checkMesh dictionary.

During the initial stages of this project snappyHexMesh was used due to the contribution of Daniels *et al*'s study [36], outlined within the literature, that used snappyHexMesh to optimise the PitzDaily case. The snappyHexMesh utility provided sufficient mesh quality for their investigation, however for this draft tube application the boundary layer generation was not of sufficient quality for this study, nor was the creation of inverted cells.

As seen in Figure 3, boundary layers were unable to be created for the complete domain and a number of inverted cells were present within the mesh; these issues were present after improving the number of smoothing iterations and tightening the minimum allowable mesh quality. For studies undertaken in this report, cfMesh was used.

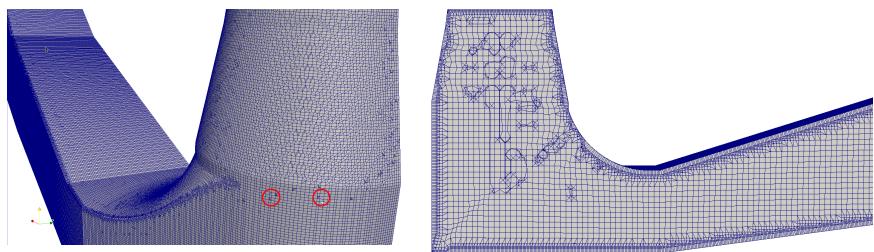


Figure 3: Poor quality boundary layer mesh generation using snappyHexMesh.

3.7. ***cfMesh***

The purpose of cfMesh and the reason for its use within this study is that cfMesh is a robust and accurate ‘inside-out’ meshing technique. cfMesh distributors claim a speed advantage over existing OpenFOAM automatic mesh utilities, due to its updated features; this reduction in CPU time is beneficial to the optimisation task [52]. cfMesh was also developed to be learned and used with little user input required, and is implemented within the OpenFOAM framework.

The ‘inside out’ workflow is such that the software starts by generating a mesh template from the input geometry and adjusts the template to fit the input geometry more closely, using smoothing and refinement iterations [53]. Three meshing workflows are available within cfMesh, these being Cartesian, Tetrahedral and Polyhedral (`cartesianMesh`, `tetMesh` and `pMesh` respectively). The `cartesianMesh` utility primarily generates hexahedra-based cells with polyhedral cells in the transitions between different cell sizes; equivalent meshes using the `tetMesh` and `pMesh` (generating tetrahedral and polyhedral cells respectively) utilities took significantly longer, therefore only the `cartesianMesh` utility was used in this study.

The cfMesh tool requires triangulated surface files as geometry input files, where the suggested file types are FMS, STL and FTR, the preferred file type being FMS. These files are required to store all necessary information for the meshing process, such as feature edges, patches and subsets, unlike snappyHexMesh where patches can be stored within different triSurface files. In order to create high quality FMS files with the correct patch naming and edge information, Salome was installed to the Linux machine. A STEP file of the geometry was created within SolidWorks and imported into Salome. Salome provides a meshing triangulation procedure whereby the user defines each patch by generation of a triangulated mesh on each patch surface (more information is available in [54]). A python script located within the source code of cfMesh is then used to generate FMS files from the Salome mesh and place them into the file in which the code is located. The process of FMS file generation was deemed too complex for the optimisation application, as it requires processes across numerous software and only provided a semi-automated procedure, as the Salome process could not be automated. Therefore, alternative investigations into the use of STL files was required, as STL files could be generated automatically within the Linux command framework or through python scripting.

Merging of STL files provides an effective measure to provide a stand-alone file that contains all the meshing requirements, as per the cfMesh requirements. In the standard Turbine-99 geometry, the simulation was set up such that the main body of the draft tube was given as one STL file and the inlet and outlet were given as two separate STL files: these three files were then merged with the ‘cat’ Linux command, thus creating a single closed manifold of different patches. Patches are specified within each STL, each triangulated surface of the STL is assigned that patch name and each patch is identified in the solution by its name and type.

The only requirements for mesh generation using cfMesh are the single geometry file, in this case an STL file, and the specification of a maximum cell size. These are specified in a separate meshDict dictionary, along with other mesh quality and boundary layer conditions. However, unlike snappyHexMesh, there is no option for the choice of either an internal or external flow simulation, as cfMesh requires single closed manifold STL file to function [55], where the mesh is created from within the geometry file. Therefore, external flows cannot be prescribed as simply as in snappyHexMesh.

Mesh parameters within the meshDict allowed for boundary layers to be generated on the wall boundary STL, where the number, thickness and expansion ratio of the boundary layer was

specified. Even if no boundary layers are specified, cfMesh by default generates one boundary layer. As previously stated, cfMesh only requires the geometry file and the maximum cell size to be specified; this maximum cell size is a function of the geometry size, therefore when generating grids with increasing y^+ values, the maximum cell size becomes a larger percentage of the domain length, missing details of the geometry. This became evident when generating the heel of the draft tube, where the curved surface created a tangent with the cone inlet to the draft tube (this also became an issue within Gowans study [49] where this tangent was removed from the Pointwise mesh) and also resulted in facet walls of the draft tube heel. As this study aimed to analyse the effectiveness of cfMesh as a completely automatic meshing tool for optimisation, it was deemed important to not remove the tangent feature.

These problems were also present in [55] and [52] that stated issues with generating sharp features. In order to improve this the surfaceFeatureEdges was used to specify edges of the mesh, which drastically improved the generated mesh, see Figure 4. Furthermore, the improveMeshQuality function adds additional smoothing and mesh solving algorithms to improve mesh quality parameters. The results of the utility can be seen in Table 3.



Figure 2: Image displaying the surfaceFeatureEdges where sharp features were able to be captured.

It must be noted that when using improveMeshQuality, issues can arise when improving the mesh quality further as this can lead to over-solving the mesh, resulting in a poor quality mesh, see Table 3.

Table 3: improveMeshQuality iterations.

	Iteration 0	Iteration 1	Iteration 2
Max cell openness	8.54E-16	3.42E-16	3.37E-16
Max aspect ratio	14.97	7.57	8.88
Mesh non-orthogonality	Max: 70.80 Avg: 4.75	Max: 61.43 Avg: 6.18	Max: 60.42 Avg: 6.86
*Severely non-orthogonal faces	3	0	0
Max skewness	2.577	1.437	1.759
Mesh OK.	Yes	Yes	Yes

With these findings in mind, four meshes were proposed, meshes A, B, C and D, with y^+ values of 15, 11, 8, 2 and cell counts of 175162, 332112, 419756 and 547466, respectively. Mesh D was used in simulations using low Reynolds number turbulence models.

3.7.1. Utilising cfMesh for Optimisation

The primary objective of this study was to generate an effective meshing system that can be integrated with Ng's [56] Bayesian optimiser. As discussed in the analytics of Daniels *et al*'s study [36], snappyHexMesh, due to its 'outside-in' geometry snapping technique, was able to only remove regions of the geometry, limiting the search space and potentially the optimised design. In order to combat the limitations of Daniels *et al*'s study, an extended draft tube geometry was created. The python code written by Ng [56] generates an STL file with each evaluation of the optimiser. Using a python code generated by the author and Gowans [49], the STL was merged with the extended draft tube geometry, see Figure 5.

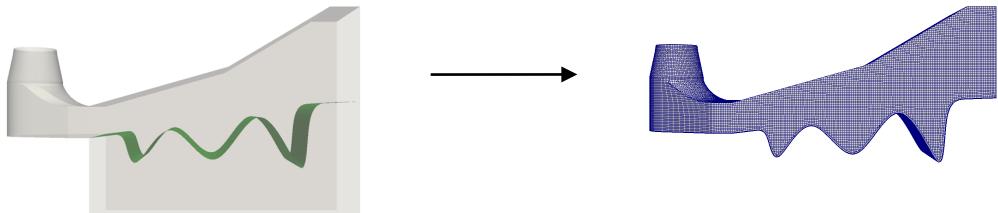


Figure 3: Demonstration of cfMesh generating new geometries based on the optimiser STL output.

As the cfMesh is an 'inside-out' meshing technique, as soon as the geometry is closed by the STL surface, the mesh is generated, creating a closed manifold body. This proved to be successful as not only could the geometry be changed, but also because a hexahedra-dominated mesh with boundary layers was created for each evaluation, regardless of the curvature of the STL provided. Finally, within the code, the mesh was improved for each evaluation using the a single improveMeshQuality iteration. cfMesh was also combined with pointwise in Gowans' study [49] where if the mesh quality fell below a threshold, cfMesh was used to generate the mesh; this became evident in the validation of Model 2, see Section 4.1.

3.8. Heel Insert

cfMesh provided a benefit to the study due to its ability to mesh geometry changes quickly; this was evident when a radius was introduced into the heel. Prior to the manufacture and optimisation of the draft tube, the heel was redesigned, with the intention of removing the

stagnation region and secondary flow in the draft tube, that was evident in initial simulations and was predicted by Raisee and Alemi [12], where secondary flow was found to generate in the corners of square and rectangular cross-sections of ducts. The redesign aimed to improve the accuracy of CFD results, as complex regions of the flow would be minimised, reducing the requirement of fine meshes to resolve the physics of such flow. The redesigned heel reduced the stagnation at the bottom of the draft tube heel, however it was unable to remove the secondary flow, see Figure 6.

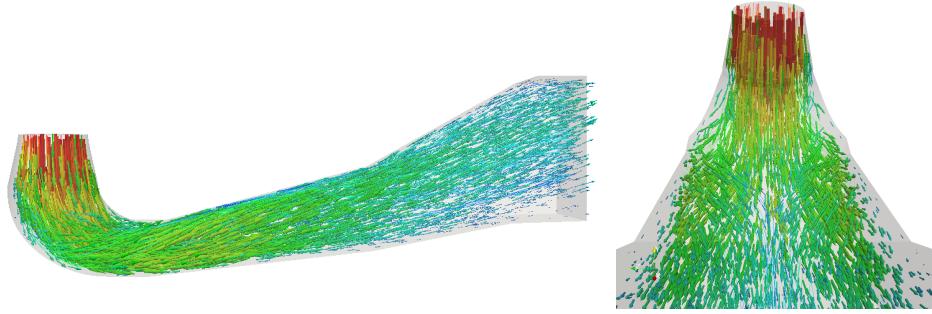


Figure 4: Images displaying the draft tube heel insert design with a reduction in stagnation but reoccurrence of secondary flow generation.

As the redesign was unable to remove the secondary flow within the draft tube, it was decided to maintain the existing geometry, so that the methods outlined for this project could prove their significance on an existing base, the Turbine-99 draft tube.

3.9. Inducing Swirl

At the Turbine-99 workshop [5], attendees were given fixed values of velocity components based upon LDV measurements, where both the axial and tangential velocity components were measured. Within the workshop, the formula for computing the radial velocity $V(r)$ at the inlet assumed that the flow was attached to both the walls of the draft tube and/or the runner cone, dependent upon radial position as follows:

$$V(r) = U(r) \tan(\theta) \quad (22)$$

$$\theta = \theta_{cone} + \frac{\theta_{wall} - \theta_{cone}}{R_{wall} - R_{cone}} (r - R_{cone}) \quad (23)$$

where $R_{cone} \leq r \leq R_{wall}$ and $\theta_{wall} = +2.8^\circ$ and $\theta_{wall} = -12.8^\circ$.

Initially, compiling an existing OpenFOAM velocity boundary condition and implementing a new condition were deemed most suitable for specifying swirl at the inlet, however due to the experimental limitations of the project, velocity components were not available to be measured and thus a new boundary condition could not be made within the scope of the study.

In order to simulate the motion of the runner, a moving wall boundary condition was set at the workshop value $N=595$ rpm, then using Equation 23, three locations along the radius r were used to specify the radial velocity, each with a tangential velocity calculated using an angular velocity of 62.3rads^{-1} and a constant axial velocity of 1.1386ms^{-1} . The three locations were: at the runner cone, the midpoint between the runner cone and the draft tube walls and at the draft tube wall ($r = R_{cone}$, $r = \frac{R_{wall}-R_{cone}}{2}$, $r = R_{wall}$ respectively). In order to specify the said conditions within OpenFOAM, a cylindricalInletVelocity patch field was chosen.

This investigation was not aiming to provide conclusive statements as to the effect of swirl on the solution, rather it was primarily aimed at briefly touching upon the potential effects and trends of swirl on the performance of the draft tube. This study was conducted on Mesh D, the finest mesh available, in order to capture the swirling flow physics in as much detail as possible.

4. Results and Analysis

This section will comprise the results obtained on Model 1, using meshes A, B, C and D with the various turbulence models and wall functions listed in Section 3.3.1, and secondly an assessment of simulations of the random geometry (Model 2) used to validate potential outputs of the optimiser. Finally, the results of the investigation of swirling flow on draft tube performance will be detailed.

The experimental data provided by Angus [39] was subject to a large degree of error due to an observed periodic fluctuation in the flow, however the fluctuation remained stable across the experiment. Another error present was the accuracy of the flow rate measurement equipment, which had an uncertainty of 3%, therefore further sources of error may be as a result of inaccuracies in the flow rate delivered by the hydraulics bench. During the CFD simulations, a fluid temperature of 10°C was used, as experimental data was not available; however, during experiments the pump within the flow bench generated a large amount of heat energy, raising

the temperature to 23°C. The change in temperature resulted in a change in density of less than 0.2% and therefore the effect on the flow simulation is expected to be negligible (for further information on experimental errors, please see Angus' study [39]). To validate the CFD results, the results should agree with the mean experimental pressure recovery value.

Figures 7-10 display a mesh and turbulence model sensitivity study, where for meshes, A, B, C and D the results of the different turbulence models and wall functions are presented.

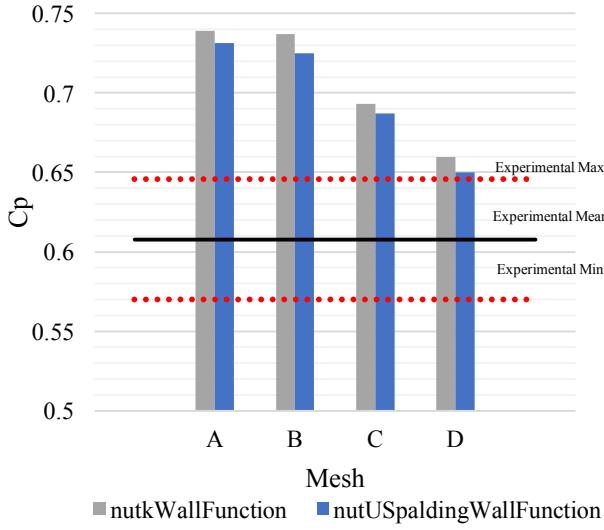


Figure 7: Graph displaying the C_p values generated by k -Epsilon simulations

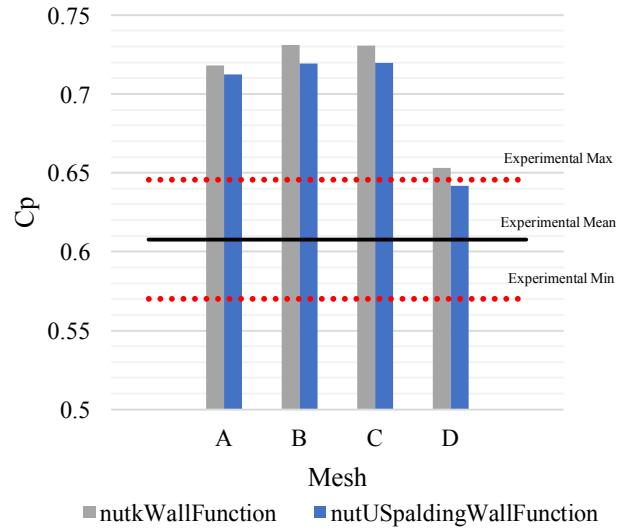


Figure 8: Graph displaying the C_p values generated by RNG k -Epsilon simulations

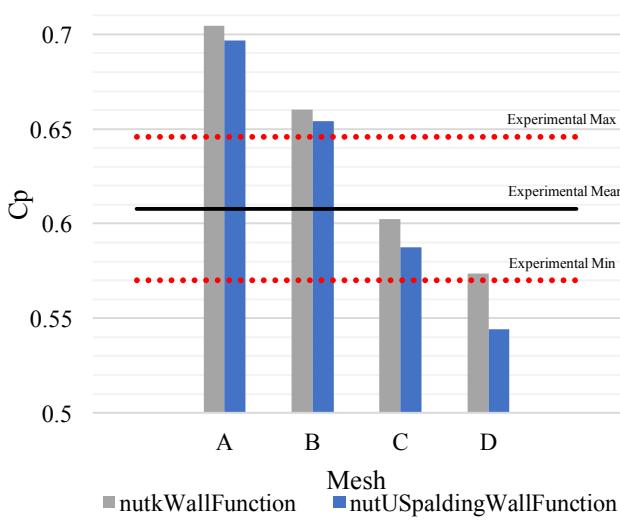


Figure 9: Graph displaying the C_p values generated by k - ω SST simulations

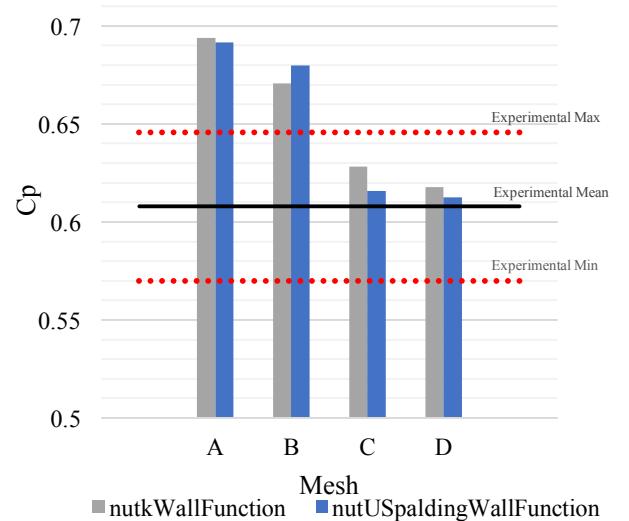


Figure 10: Graph displaying the C_p values generated by k - ω simulations

The turbulence model study indicates that the $k-\omega$ model turbulence models show most agreement with the experimental results; this is followed by the $k-\omega$ SST, $k-\varepsilon$ and RNG $k-\varepsilon$ in that order. For each of the models and subsequent turbulence viscosity wall function, the performance of meshes A and B do not fall within the bounds of the experiment data. For all turbulence models, as the mesh was refined, the pressure recovery factor was shown to reduce, therefore one can deduce that the poor performance of meshes A and B is due to the inability to resolve secondary flow within the draft tube. Post processing of the results confirmed this, where Figure 12, a $k-\omega$ simulation using the nutkWallFunction with mesh D, captured in more detail the secondary flow, whereas in Figure 11 the simulation using the same model on mesh A shows the secondary flow to be more diffusive; this is expected as the velocity is interpreted across larger cells in mesh A. The inability of meshes A and B to adequately model the secondary flow experienced within the draft tube resulted in a lower prediction of minor head losses, thus predicting a higher pressure recovery factor across each of the turbulence models used.

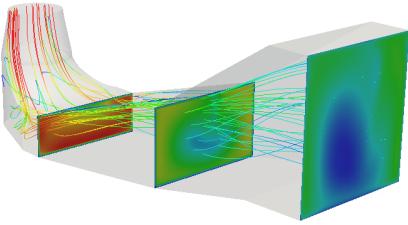


Figure 11: Image showing the velocity profile a cross sections of the geometry using a $k-\omega$ simulation on mesh A

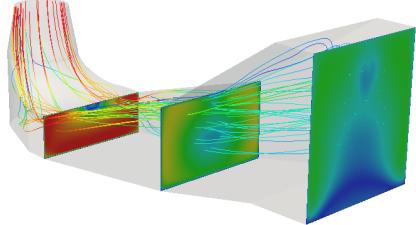


Figure 12: Image showing the velocity profile a cross sections of the geometry using a $k-\omega$ simulation on mesh D

The $k-\omega$ model with mesh D provided an error from the experimental mean of 0.75%, and for meshes C and D using the $k-\omega$ model, the maximum percentage difference was 3.24%. The worst performing model, $k-\varepsilon$ model with mesh A, simulated a pressure recovery value of 17.7% from the mean experimental value. Simulations run on meshes generated by Pointwise and a hybrid of Pointwise and blockMesh, generated in Gowans' [49] and Burns' studies [50] used the $k-\varepsilon$ model. The Pointwise and blockMesh meshes had cell counts of around 300,000 cells and were therefore similar to mesh B used in this study. The Pointwise and the hybrid mesh simulations for $k-\varepsilon$ generated values much closer to the experimental mean (1.74% and 8.79% respectively), yet still over-predicted the pressure recovery, due to the mesh density. The use of a second-order $k-\varepsilon$ simulations was evident here, as the results produced were in better agreement with the experimental data than the equivalent $k-\varepsilon$ simulation generated in this

study, however the residuals were less stable at convergence residuals of 1×10^{-2} , as the second order simulations are more dependent on the mesh quality. It was suspected that the reduced performance of blockMesh can be attributed to non-orthogonality within the mesh.

Within this study the use of $k-\varepsilon$ and RNG $k-\varepsilon$ returned results in less agreement with the experimental data; this is due to the inferior performance of $k-\varepsilon$ models in the near wall region, when compared to $k-\omega$ models. These $k-\varepsilon$ models are also more sensitive to y^+ values, whereas the ω functions provide a blending factor from the viscous sub-layer to the logarithmic region, showing that the $k-\omega$ formula is better formulated to cope with lower than desirable y^+ values. However, when using mesh D, the low Reynolds number wall models used with the RNG $k-\varepsilon$ model displayed a significant change in results; this could be attributed to the improved modelling of the low Reynolds number wall functions in the near wall regions. The RNG $k-\varepsilon$ model's introduction of anisotropic forces in the boundary layer showed an improved result over the $k-\varepsilon$ model, as more accurate modelling of the flow in the near wall region was achieved.

As shown by Figures 7-10 the nutUSpaldingWallFunction for the turbulence viscosity simulates a lower pressure recovery across the majority of turbulence models and meshes. The nutkWallFunction wall function relies on the turbulence kinetic energy modelled value: this is only valid for the logarithmic region of the boundary layer, thus it predicted less turbulence viscous effects within the flow, producing a higher value of pressure recovery. On the contrary, the nutUSpaldingWallFunction is able to model the viscous forces down to the wall ($y^+ = 0$), potentially predicting a higher turbulence viscosity, thus reducing the pressure recovery.

In order to understand how the pressure recovery develops along the draft tube, the pressure recovery factor was found along the upper and lower centrelines of the draft tube, using the best performing simulation from each of the turbulence models investigated, see Figure 13 and 14. The pressure recovery values were found by finding the difference in pressure at each position against the lowest value of pressure on the centrelines. Where the inlet patch meets the wall patch of the draft tube, the point pressure value is more negative than the average inlet pressure, therefore the C_p values along the centreline are higher than in Figures 7-10. The single pressure value for the $k-\omega$ model was greater than the point values for the $k-\omega$ SST, RNG $k-\varepsilon$ and $k-\varepsilon$ models, therefore this value of pressure recovery was unfortunately shifted by this value. For the lower centreline, the values at the inlet do not equal zero, as the largest

negative pressure is experienced on the upper centreline. The largest negative pressure is always experienced in this location as the flow is attached and accelerated along the curved surface of the draft tube heel.

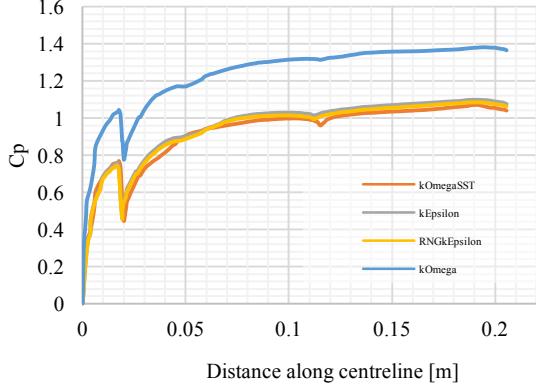


Figure 13: Pressure recovery factor along the upper centreline of the draft tube.

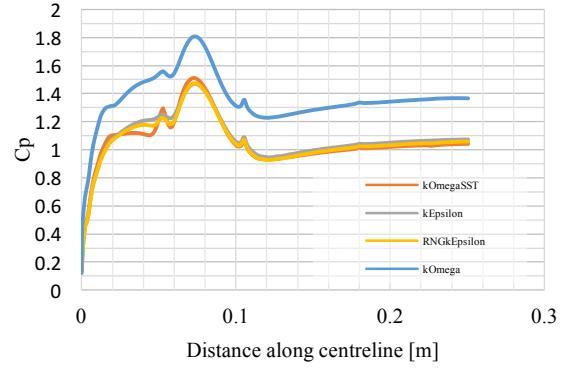


Figure 14: Pressure recovery factor along the upper centreline of the draft tube.

The results from Figure 13 and 14 show that the majority of the pressure recovery occurs in the heel of the draft tube. For each of the models presented in Figure 13 and 14, the pressure recovery distribution appears almost identical, therefore one may deduce that as long as sufficient resolution is provided within the mesh, each model can be used within the optimiser, as the flow development is very similar.

For the upper centreline, the pressure recovery increases until approximately 0.02m where the cone section meets the vertical: the flow quickly attaches to the curved section, accelerating the flow and reducing C_p , before rising quickly as the curved section ends. Figure 14 shows how the pressure recovery increases until the cone section meets the vertical, then a small rise in pressure is experienced in the sharp heel, where the first region of stagnation begins. Finally, at 0.07m the flow stagnates, resulting in a large increase in pressure recovery, due to the increase in pressure. This peak then reduces as the flow begins to reattach, remaining at a near constant pressure recovery.

4.1. Model 2

Using the results presented in Figures 7-10, the best results of each of the turbulence models were used to validate the CFD results of the second geometry. The pressure recovery factors within the experimental error bounds for Model 1 are as follows: mesh C for $k-\omega$ SST, C and D for $k-\omega$ and only mesh D for the two $k-\epsilon$ models. The narrowing down of the best performing

models reduced CPU time. The results used for validation of Model 2 are presented in Figures 15-18.

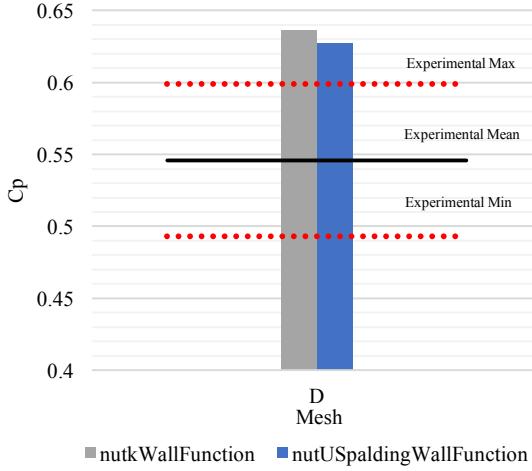


Figure 15: Graph displaying the use of the $k-\epsilon$ turbulence model for Model 2.

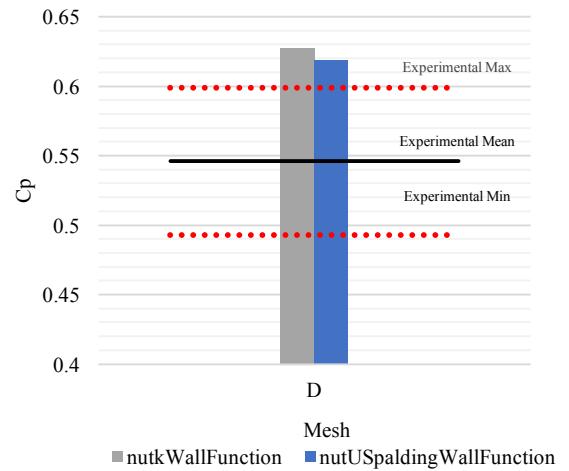


Figure 16: Graph displaying the use of the RNG $k-\epsilon$ turbulence model for Model 2.

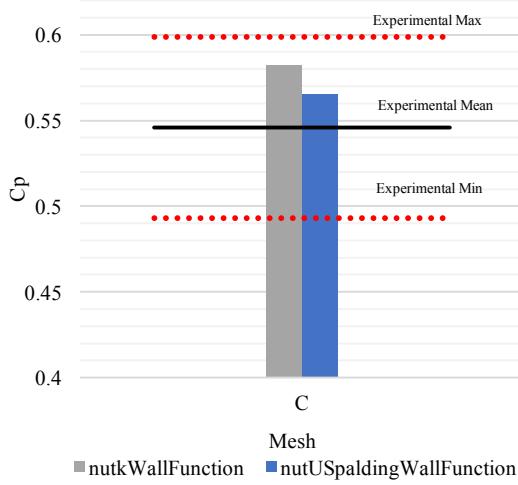


Figure 17: Graph displaying the use of the $k-\omega$ SST turbulence model for Model 2.

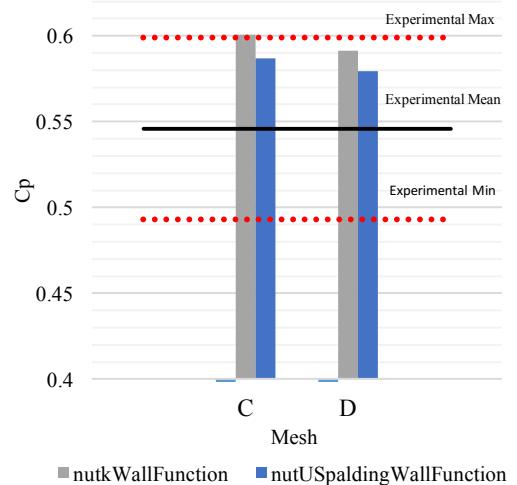


Figure 18: Graph displaying the use of the $k-\omega$ turbulence model for Model 2.

The $k-\epsilon$ models again over-predict the pressure recovery, however they are able to predict a lower value than the original geometry, which is fitting with the lower experimental mean value. The best performance for this geometry came from the $k-\omega$ SST model with a percentage difference from the mean of 3.35%. The blockMesh [50] and pointwise [49] meshes provided pressure recovery values of 0.641 and 0.447 or 14.88% and 20.65% from the experimental mean value, respectively. This translates to an 11% reduction in performance for blockMesh and 12% reduction in performance for Pointwise, when compared to the structured meshes of

the original geometry, compared to cfMesh's 2.6% reduction in performance. This reduced accuracy in the simulation of Model 2 is attributed to the mesh quality. The high curvature provided challenges in the structured meshing of Pointwise and blockMesh, and with the use of second-order schemes, results generated were greatly influenced by poor mesh quality. However, if geometries of this curvature are generated within the optimiser, the Pointwise mesh quality would fall below quality thresholds and would switch to cfMesh, highlighting its flexibility and robustness.

4.2. Swirl

The pressure recovery results shown in Figure 19 are found using the $k-\omega$ model on mesh D, the best performing model for the original geometry. The results from Figure 19 indicate that the swirling motion generated within the flow reduces the pressure recovery, when compared to simulation of the no swirl inlet condition; this result is expected, as the vortex rope generated from the runner cone creates complex flow and produces additional minor head losses within the flow, reducing the pressure recovery of the draft tube.

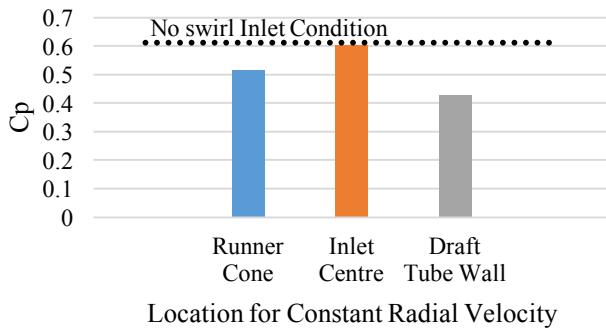


Figure 19: Simulation results generated for swirling motion, where the radial velocity is fixed by location on the inlet. This is compared to the simulated of pressure recovery of the no swirl inlet condition.

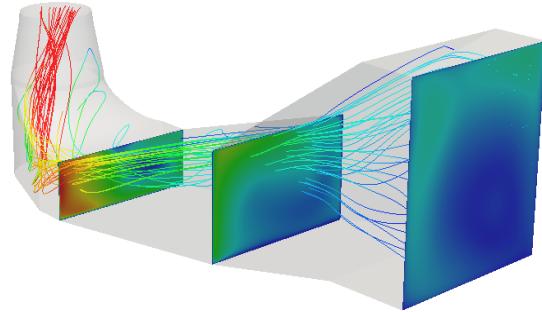


Figure 20: Image displaying the early generation of a vortex rope within the flow

Figure 20 demonstrates the generation of a vortex rope, which was experienced in the literature [5], due to the radial velocity of the swirling motion attaching the flow to the runner cone. As this section of the study aimed to provide insight into the use of cfMesh and the turbulence models for the potential introduction of swirl in future work, a conclusive answer as to the validation of such results cannot be formulated. However, as a basis for future work, the simulations show realistic results; this will be discussed in Section 5.1.1.

4.3. *cfMesh for Optimisation*

The methods outlined within this study provided an automatic meshing technique that was combined with Ng's optimisation code [56] to generate an optimised solution. As the optimisation forms part of a group project, the analysis of the flow, the increase in performance and validation will be discussed in the G2 report [1] (experimental results were also not available at the time of this study). Figure 21 shows the optimised geometry after 100 iterations. In short, the optimiser has aimed to smooth the sharp heel of geometry by expanding immediately after the heel, where in a complete draft tube optimisation, a radius would be added to the heel to reduce separation. Along with this, an expansion of the outlet has aimed to reduce the flow velocity, increasing the pressure at the outlet.

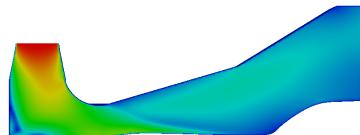


Figure 21: Image displaying the flow development of the Bayesian optimised solution after 100 iterations.

5. Discussion and conclusions

Within this study both snappyHexMesh and cfMesh have been used with the aim of providing an automatic mesh generation technique for the optimiser. However, due to problems with boundary layer formation and mesh quality, the use of cfMesh was preferred. The use of cfMesh resulted in the generation of boundary layer meshes throughout the entire geometry. The limitations of this meshing technique are that for small geometries, where the mesh size becomes larger than the feature size, edges are poorly generated and facet faces are created, resulting in a maximum y^+ average of 15 for this study. However, this may not have impacted the solution significantly, as the regions of stagnation produced within the flow will have reduced the average y^+ values, therefore in regions where the turbulent flow is present, the y^+ values may still be valid for the turbulence models used within this study. As cfMesh provides a fast meshing process, a mesh sensitivity study was achieved. The study revealed that as the mesh was refined for each of the turbulence models used, the value of pressure recovery reduced; this was due to the improved resolution of the secondary flow generated within the draft tube.

The turbulence model study revealed that the RNG k- ε and standard k- ε models over-predicted the pressure recovery, due to the k- ε model's poor performance within the near wall region, when compared to the k- ω model. The turbulence model with the best agreement with the experimental results was the k- ω SST model, being 0.75% from the experimental mean. In order to achieve more accurate results using first-order schemes, an enhanced grid resolution was required, whereas the Pointwise and blockMesh mesh generation studies used second-order standard k- ε simulations which provided better agreement with the experimental results, with a lower resolution meshed. However, the improved accuracy compromised the solution stability, resulting in convergence residuals of 1×10^{-2} .

It was also found during the simulations of Model 2 that structured meshing from Pointwise and blockMesh showed poor agreement with the experimental results, due to poor mesh quality as a result of the high curvature geometry. The enhanced sensitivity of second-order schemes to mesh quality was evident, as the poor quality meshes reduced their performance by over 11%, when compared to their use in the standard geometry. The results achieved by cfMesh showed a reduction in performance of 2.6%, for a significant increase in curvature and complexity, highlighting its robustness. It must be noted however that the improved performance for the cfMesh simulations may be due to the use of first-order simulations that are less sensitive to mesh quality, potentially showing an advantage of using first-order schemes for the optimiser. The increase in robustness shown by cfMesh provided valuable information for Gowans' study [49], that subsequently combined the Pointwise and cfMesh technique within the optimiser, if the Pointwise mesh quality fell below given thresholds. One cannot be confident as to the accuracy of the experimental results produced for this study, as potential fluctuations in flow rate may have impacted results, therefore more work is required to confirm the validation of the numerical modelling.

Results of the pressure recovery along the upper and lower centrelines of the geometry confirmed that much of the pressure recovery occurs within the heel of the draft tube. This is important when validating the optimiser as the expected improvement of optimising the diffuser section alone is low, and therefore the instabilities of the second-order Pointwise and blockMesh simulations may have mislead the optimiser. However, the optimiser solution agrees with intuition, where the optimiser has aimed to smooth the heel and increase the diffuser section, which suggests that the inaccuracies in the solutions are small enough for the

optimiser to establish an optimised solution. It may be expected that without the geometry limitations of the optimisation search space, a more profound geometry change could be experienced.

Swirling motion was introduced into the simulation, however as there was no experimental data regarding values of tangential and axial velocity, used to generate radial velocity components, no validation could be achieved. However, as a process to inform future work, it was valuable, as a vortex rope was generated from the turbine runner, reducing the pressure recovery factor.

5.1. Conclusion

Presented within this report is investigation of automatic mesh generation techniques. This study developed an automated boundary layer meshing technique using cfMesh, for the application of draft tube shape optimisation. The objective function of the optimiser was to increase the pressure recovery factor, a single parameter that determines the draft tube's efficiency. A large degree of secondary flow was generated within the draft tube and in order to capture such flow, meshes with high resolution are required. During a turbulence model study, the $k-\omega$ and $k-\omega$ SST models provided the best agreement with experimental data, however, due to the geometry constraints of the experimental models and the accuracy of the data provided, a conclusive answer as to the validity of the models cannot be provided with a high degree of certainty. cfMesh was better able to handle high curvature geometries, where structured meshing techniques performed poorly. Most of the pressure recovery was found to lie in the heel of the draft tube, therefore any inaccuracies in the modelling are a concern, as the percentage change in performance may be attributed to modelling errors. Although the accuracy of the optimiser presents some uncertainty, the process of automatic mesh generation using cfMesh has proved successful.

5.1.1. Further Work

As the results in this study show, a higher mesh resolution resulted in improved modelling of secondary flow, therefore with more time and computational resources, meshing with grid resolutions of less than $y^+ 1$ and without the use of wall functions could improve the accuracy of simulations. Many of the challenges present within the study were due to the limitations of

the printing size and experimental equipment, therefore if larger scale models could be attained, more convincing validation could be achieved, as the generation of appropriate y^+ values would not be such a challenge. To develop the automatic meshing within the optimiser further, a study should be accomplished to see the sensitivity of the solution to the meshing technique; this will determine the validity of combining the structured meshing for low curvature geometries and cfMesh for more complex geometries. With the initial understandings of the effect of swirl generated within this study, it is recommended to compile a new inlet boundary condition, using the existing cylindricalInletVelocity condition; this could be achieved by using LDV data of experimental studies, to provide true velocity components, that can be prescribed at the inlet.

6. Project management

As this study formed part of a group project, it was important that work packages were clearly defined. The project was split into three sub-teams under the categories of experimental, optimisation and CFD. Much of the work was closely linked and in order to maintain direction within the project, weekly meetings were set up with each of the teams, research members and the project supervisor in attendance. For each meeting a chair and secretary were elected, who were responsible for setting an agenda for the meeting and recording meeting minutes: the roles were rotated throughout the project.

As this was a research project, the timeline or predictions of work could not be predicted, however a Gantt chart generated for the project was used as a guideline as to when deliverables were required. A handwritten logbook was used to record any important deadlines for deliverables and requirements from the group. A progress meeting was called at the midpoint of the project, providing an opportunity to outline the work achieved and allowed further links to be drawn between the group. A presentation in the final stage of the project gave an opportunity to receive feedback from the project supervisor and his research team, and to take suggestions for improvement of the work delivered.

The main area requiring project management within this study was the time required to run CFD simulations. In order to ensure the best use of the time available for the project, simulations were set up during the day and run on multiple processors overnight, allowing results to be generated without needing a researcher present to monitor solutions. As the

optimiser took over 4 days to run, careful planning was needed in order to coordinate work to fit in around these lengthy simulations.

All data was backed up using cloud storage and on a backup drive to mitigate the risk of data loss. All meeting minutes were also available on the cloud to coordinate any collaborative work.

7. Contribution to group functioning

The group objective was to optimise the shape of the draft tube to increase pressure recovery. This study achieved the generation of automatic meshes which produced solutions for Ng's optimiser [56]. In order to facilitate the easy implementation of the meshing technique into the optimiser, an efficient work flow was developed.

This study was achieved alongside the study conducted by Gowans [49], who used a structured meshing technique. When mesh quality fell below prescribed thresholds, cfMesh replaced Pointwise. This was a difficult task that required the input of much knowledge gained from the use of cfMesh in this report.

This study used experimental results generated by Angus [39] to validate the CFD simulations. This was achieved through a turbulence model study, aiming to better understand the flow characteristics. Validation was provided for the studies conducted by Gowans [49] and Burns [50], and for the findings of the group project.

Advice was sought from Gilbert regarding CFD due to his industrial experience and high level knowledge of fluid dynamics. This advice impacted on the turbulence model study and the solution set up.

8. Bibliography

- [1] J. Angus, P. Burns, T. Dye, R. Gilbert, J. Gowans, S. Hardy, S. Hutchings and C. Ng, "An Investigation into the Optimisation of a Draft Tube," Exeter, 2017.
- [2] M. J. Cervantes and M. Lövgren, "Radial Velocity at the Inlet of the Turbine-99 Draft Tube," in *2nd IAHR International Meeting of the Workgroup on Cavitation and Dynamic Problems in Hydraulic Machinery and Systems*, Timisoara, 2007.
- [3] B. Gebart, L. Gustavsson and R. Karlsson, "Proceedings of the Turbine-99 – Workshop on draft tube flow," Porjus, 1999.

- [4] T. Engström, L. Gustavsson and R. Karlsson, “Turbine-99 Workshop 2 on Draft Tube Flow,” in *Proceedings of 21st IAHR Symposium on Hydraulic Machinery and Systems*, Lausanne, 2001.
- [5] M. Cervantes, T. Engström and L. Gustavsson, “Proceedings of the third IAHR/ERCOFTAC Workshop on draft tube flows - Turbine-99 iii,” Luleå University of Technology Department of Applied Physics and Mechanical Engineering Division of Fluid Mechanic, Porjus, 2005.
- [6] U. Andersson, “An Exerimental Study of the Flow in a Sharp-Heel Kaplan Draft Tube,” Luleå University of Technology, Älvkarleby, 2009.
- [7] U. Andersson and N. Dahlbäck, “Experimental evaluation of draft tube flow. – A test case for CFD-simulations,” in *Proceedings of the XIX IAHR Symposium on Hydroaulic Machinery and Cavitation*, 1998.
- [8] M. Gublin, “Draft tubes of Hydro- Electric Stations,” Amerind Publishing Co. Pvt. Ltd., New Delhi, 1973.
- [9] U. Andersson and R. Karlsson, “Quality aspects of the Turbine 99 draft tube experiments,” in *Proceedings from Turbine 99 - workshop on draft tube flow, Technical report*, Sweden.
- [10] U. Andersson, “Licentiate Thesis - An exerimental study of the flow in a sharp-heel draft tube,” Luleå University of Technology, Älvkarleby, 2000.
- [11] M. Cervantes and F. Engström, “Eddy Viscosity Turbulence Models and Steady Draft Tube Simulations,” in *Proceedingds of the third IAHR/ERCOFTAC workshop on draft tube flow*, Porjus, 2005.
- [12] M. Raisee and H. Alemi, “Performance of Linear and Non-Linear Low-Re $k - \epsilon$ Models In Prediction of Developing Turbulent Flow Through 90° Curved Ducts,” in *Turbine-99 III Proceedings of the third IAHR/ERCOFTAC workshop on draft tube flow*, Porjus, 2005.
- [13] U. Andersson, F. Engström, H. Gustavsson and R. Karlsson, “The Turbine-99 workshops - conclusions and recommendations,” in *Proceedings of the 22th IAHR Symposium on Hydraulic Machinery and Systems*, Stockholm, 2004.
- [14] R. Karlsson and U. Andersson, “The Turbine-99 Workshops - Conclusions and Recommendations - Best Practice Advice for AC6-07 Draft Tube, Thematic Network for Quality and Trust in the Industrial Application of CFD (QNET-CFD),” in *22nd IAHR Symposium on Hydraulic Machinery and Systems* , Stockholm, 2004.
- [15] H. Nilsson, “Evaluation of OpenFOAM for CFD of turbulent flow in water turbines,” Chalmers University of Technology, Göteborg, 2006.
- [16] I. Gunnar and J. Hellström, “Redesign of an Existing Hydropower Draft Tube,” Luleå University of Technology , Sweden, 2005.
- [17] S. Galván, M. Reggio and F. Guibault, “Assessment Study of K- ϵ Turbulence Models and Near-Wall Modeling for Steady State Swirling Flow Analysis in Draft Tube Using Fluent,” *Engineering Applications of Computational Fluid Mechanics*, vol. 5, no. 4, p. 459–478, 2011.
- [18] H. Grotjans, “Simulations on draft tube with CFX,” in *Proceedings of Turbine-99, Second ERCOFTAC Workshop on Draft Tube Flow*, Älvkarleby, 2001.
- [19] Y. Lai, L. Weber and V. Patel, “Non hydrostatic three-dimensional model for hydraulic flow simulation. I: Formulation and verification,” *Journal of Fluids Engineering*, vol. 129, no. 3, pp. 196-205, 2003.
- [20] B. D. Marjavaara, “Parameterisation and Flow Design Optimisation of Hydraulic Turbine Draft Tubes,” Sweden, 2004.
- [21] B. Marjavaara, T. Lundstrom, T. Goel, Y. Mack and W. Shyy, “Hydraulic turbine diffuser shape optimization by multiple surrogate model approximations of pareto fronts,” *Journal of Fluids Engineering*, vol. 129, no. 9, pp. 1228-1240.

- [22] B. Marjavaara and T. Lundstrom, "Redesign of a sharp heel draft tube by a validated cfd-optimization," *Journal for Numerical Methods in Fluids*, vol. 50, no. 8, pp. 911-924, 2006.
- [23] T. Tokyay and G. Constantinescu, "CFD SIMULATIONS OF FLOW IN A HYDRAULIC TURBINE DRAFT TUBE USING NEAR WALL RANS AND LES MODELS," in *Turbine-99 III Proceedings of the third IAHR/ERCOFTAC workshop on draft tube flow*, Porjus, 2005.
- [24] W.-T. Su, F.-C. Li, X.-B. Li, X.-Z. Wei and Y. Zhao, "Assessment of Les Performance in Simulating Complex 3D Flows in Turbo-Machines," *Engineering Applications of Computational Fluid Mechanics*, vol. 6, pp. 356-365, 2012.
- [25] B. Marjavaara, R. Kamakoti, T. Lundström, W. Shyy, S. Thakur and J. Wright, "Steady and Unsteady CFD Simulations of The Turbine-99 Draft Tube Using CFX-5 and STREAM," in *Turbine-99 III Proceedings of the third IAHR/ERCOFTAC workshop on draft tube flow*, Porjus, 2005.
- [26] H. Foroutan and S. Yavuzkurt, "Simulation of flow in a simplified draft tube: turbulence closure considerations," in *26th IAHR Symposium on Hydraulic Machinery and Systems - Conf. Series: Earth and Environmental Science*, Pennsylvania, 2012.
- [27] S. Galvan, M. Page, F. Guibault and M. Reggio, "Numerical Validation of Different CFD K-Epsilon Turbulence Models Using Fluent Code," in *Turbine-99 III Proceedings of the third IAHR/ERCOFTAC workshop on draft tube flow*, Porjus, 2005.
- [28] J. Reuther, A. Jameson, J. Farmer, L. Martinelli and D. Saunders, "Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation," American Institute of Aeronautics and Astronautics, Nevada, 1996.
- [29] J. A. Samareh, "Grid Generation for Multidisciplinary Design and Optimization of an Aerospace Vehicle: Issues and Challenges," Hampton, 2000.
- [30] B. K. Soni, "GRID GENERATION FOR INTERNAL FLOW CONFIGURATIONS," *Computers and Mathematics with Applications*, pp. 191-201, 1992.
- [31] M. Nemec and M. J. Aftosmis, "Adjoint Algorithm for CAD-Based Shape Optimization Using a Cartesian Method," 17th Computational Fluid Dynamics Conference, Toronto, 2005.
- [32] T. Leblond, P. Froment, P. de-Nazelle, R. Sellakh, P. Serré and G. Chevallier, "Gradient-based Optimization of Parameterized CAD Geometries," in *11th World Congress on Structural and Multidisciplinary Optimization*, Sydney, 2015.
- [33] S. Petropoulou, "Industrial Optimisation Solutions Based on OpenFOAM® Technology," in *5th European Conference on Computational Fluid Dynamics*, Lisbon, 2010.
- [34] C. Othmer, E. M. Papoutsis-Kiachagias and K. Haliskos, "CFD Optimization via Sensitivity-Based Shape Morphing," in *4th ANAA & μETA International Conference*, Thessaloniki, 2011.
- [35] S. Daniels, A. Rahat, G. Tabor, J. Fieldsend and R. Everson, "A review of shape distortion methods available in the OpenFOAM framework for automated design optimisation".
- [36] S. Daniels, A. Rahat, J. Fieldsend and R. Everson, "Shape Optimisation Using Computational Fluid Dynamics and Evolutionary Algorithms," in *OpenFOAM Workshop Portugal*, Portugal, 2016.
- [37] S. Galván, C. Rubio, J. Pacheco, C. Mendoza and M. Toledo, "Optimization Methodology Assessment for the Inlet Velocity Profile of a Hydraulic Turbine Draft Tube. part I: Computer Optimization Techniques," *Journal of Global Optimization*, vol. 55, no. 1, pp. 53-72, 2013.
- [38] B. Marjavaara and T. Lundström, "Automatic Shape Optimisation of a Hydropower Draft Tube," in *ASME/JSM 2003 4th Joint Fluids Summer Engineering Conference*, Honolulu, 2003.

- [39] J. Angus, “Experimental Investigation of Flow Inside Draft Tubes with Varied Diffuser Geometry,” Exeter, 2017.
- [40] T. Dye, “Design and Prototyping of a Custom Data Collection System for Testing Scale Model Draft Tubes,” Exeter, 2017.
- [41] Stanford University, “Evaluation of RANS turbulence models for the simulation of channel flow,” [Online]. Available: <https://web.stanford.edu/class/me469b/handouts/turbulence.pdf>. [Accessed 17 April 2017].
- [42] CFD Online, “Turbulence intensity,” 15 March 2017. [Online]. Available: https://www.cfd-online.com/Wiki/Turbulence_intensity. [Accessed 19 April 2017].
- [43] CFD Online, “Two equation turbulence models,” 25 October 2016. [Online]. Available: https://www.cfd-online.com/Wiki/Two_equation_models. [Accessed 28 April 2017].
- [44] J. Bardina, P. Huang and T. Coakley, “Turbulence Modeling Validation, Testing, and Development,” NASA Technical Memorandum 110446, 1997.
- [45] CFD Online, “RNG k-epsilon model,” 5 June 2010. [Online]. Available: https://www.cfd-online.com/Wiki/RNG_k-epsilon_model. [Accessed 20 April 2017].
- [46] M. Ong, T. Utne, L. Holmedal, D. Myrhaug and B. Pettersen, “Numerical simulation of flow around a smooth circular cylinder at very high Reynolds numbers,” *Marine Structures*, vol. 22, no. 2, pp. 142-153, 2009.
- [47] S. Liu, “Implementation of a Complete Wall Function for the Standard $k - \epsilon$ Turbulence Model in OpenFOAM 4.0,” in *In Proceedings of CFD with OpenSource Software*, Gothenburg, 2016.
- [48] F. Menter and T. Esch, “Elements of industrial heat transfer predictions,” in *16th Brazilian Congress of Mechanical Engineering (COBEM)*, Uberlandia, 2001.
- [49] J. Gowans, “Automatic Structured and Unstructured Mesh Generation Using Pointwise,” Exeter, 2017.
- [50] P. Burns, “The Development of an Automatic Numerical Draft Tube Model,” Exeter, 2017.
- [51] C. Greenshields, “OpenFOAM User Guide: 5.4 Mesh generation with snappyHexMesh,” 2 March 2015. [Online]. Available: <https://cfd.direct/openfoam/user-guide/snappyhexmesh/>. [Accessed 23 April 2017].
- [52] J. Nagawkar, “Evaluate the use of cfMesh for the Francis-99 turbine,” Chalmers University of Technology, Göteborg, 2016.
- [53] D. F. Juretić, “cfMesh v1.1 User Guide,” Creative Fields, Zagreb, 2015.
- [54] A. Cukrov, “Gap Handling with cfMesh,” 2017. [Online]. Available: <http://cfmesh.com/portfolio/case-study-gap-handling-with-cfmesh/>. [Accessed 26 April 2017].
- [55] T. Välikangas, “Simulation method development for Fin-and-Tube Heat Exchanger with Open-source software,” Chalmers University of Technology, Gothenburg, 2015.
- [56] C. Ng, “Implementation of Bayesian Optimiser on CFD Modelled Draft Tube,” Exeter, 2017.

