Australian Government
Department of Defence
Defence Science and
Technology Group

CFD RANS Simulations on a Generic Conventional Scale Model Submarine: Comparison between Fluent and OpenFOAM

D.A. Jones

Maritime Division
Defence Science and Technology Group

DSTO-TN-1449

ABSTRACT

The ability to perform accurate Computational Fluid Dynamic (CFD) simulations of the flow around submarines is crucial for support to SEA1000. The Hydroacoustics Group in MD have been using the commercial software code Fluent for some years now to perform these simulations, but recently have been considering using the open source code OpenFOAM to replace some of the Fluent simulations. The fidelity of the Fluent code has been carefully validated, but the accuracy of parts of the OpenFOAM code have not been so extensively tested. To test the accuracy of the OpenFOAM software, CFD simulations have been performed on the DSTO generic conventional submarine model using both Fluent and OpenFOAM. A comparison of the value of the drag coefficient calculated by the two codes shows differences of up to 15%. A partial resolution of these differences has been found and the discrepancy has been reduced in some cases, but unacceptable differences are still present. This report summarizes the work performed so far to highlight and resolve these differences, and suggests further work which needs to be done to provide confidence in the use of these codes.

RELEASE LIMITATION

Approved for public release

Published by

Maritime Division
Defence Science and Technology Group
506 Lorimer St
Fishermans Bend, Victoria 3207   Australia

Telephone: 1300 333 362
Fax: (03) 9626 7999

AR-016-390
September 2015

APPROVED FOR PUBLIC RELEASE

CFD RANS Simulations on a Generic Conventional Scale Model Submarine: Comparison between Fluent and OpenFOAM

Executive Summary

To provide support to SEA1000 in the areas of submarine dynamics and acoustic signature control the Hydroacoustics Group in Maritime Division (MD) are required to perform extensive Computational Fluid Dynamics (CFD) simulations. Previously these have been performed using the commercial software package Fluent to solve the time independent Reynolds Averaged Navier-Stokes (RANS) equations. Simulation results obtained using Fluent have been well validated in the past. More recently, due to a need to perform acoustic signature simulations, the open source software suite OpenFOAM has been used to perform time dependent Large Eddy Simulations (LES). OpenFOAM also has the ability to solve the RANS equations and since the code is freely available its use in this manner could result in considerable cost savings. The accuracy of the simulation results obtained using the OpenFOAM RANS solver has not been well documented however. Both Fluent and OpenFOAM are essential tools which are used on a daily basis in our work to support SEA1000, and the accuracy of the results obtained using these tools must be continuously monitored. The aim of this work is to assess the accuracy of the RANS solver in OpenFOAM by performing simulations on geometries of interest and comparing these with the results obtained from the Fluent code.

Simulations have been performed on three different meshes; the bare hull of a generic conventional scale model submarine, the fully appended model in the cavitation tunnel test section and the fully appended model in an "open water" scenario. Since the simulation results depend significantly on the discretization method used to represent the divergence term in the momentum equation comparisons between the two codes were made for a number of different discretization schemes. All simulations were performed using the standard k-ε turbulence model.

The simulations performed have shown that while the simulated flow fields produced by the two codes are very similar when compared from an overall perspective, significant differences can occur for calculated quantities such as drag and lift coefficients and maximum and minimum velocities at specific locations in the flow. Differences of the order of 15% between results from the two codes have been found in some simulations. A partial resolution of these differences has been found and the discrepancy has been reduced in some cases, but unacceptable differences are still present. This report summarizes the work performed so far to highlight and resolve these differences, and suggests further work which needs to be done to provide confidence in the use of these codes.

This page is intentionally blank

Author

David A. Jones
Maritime Division

David obtained a B.Sc. (Hons) and Ph.D. in Theoretical Physics from Monash University and joined Materials Research Laboratories in 1983 after positions at the University of Strathclyde, Glasgow; Queen Mary College, London University; and the University of New South Wales, Sydney. His early research covered polymer dynamics, chaos theory, atomic physics and laser-plasma interactions. During 1987/88 he was attached to the Naval Research Laboratory, Washington, DC. He has used computational fluid dynamics techniques to simulate air blast events and detonation phenomena and since moving into Maritime Division has been simulating flow around maritime platforms using a variety of simulation techniques. He is currently using OpenFOAM software to perform both Reynolds Averaged Navier-Stokes and Large Eddy Simulations on submarines.

Contents

1. Introduction

To provide support to SEA1000 in the areas of submarine dynamics and acoustic signature control the Hydroacoustics Group in Maritime Division (MD) are required to perform extensive Computational Fluid Dynamics (CFD) simulations. Previously these have been performed using the commercial software package Fluent to solve the time independent Reynolds Averaged Navier-Stokes (RANS) equations. Simulation results obtained using Fluent have been well validated in the past. More recently, due to a need to perform acoustic signature simulations, the open source software suite OpenFOAM has been used to perform time dependent Large Eddy Simulations (LES). OpenFOAM also has the ability to solve the RANS equations and since the code is freely available its use in this manner could result in considerable cost savings. The accuracy of the simulation results obtained using the OpenFOAM RANS solver has not been well documented however. Both Fluent and OpenFOAM are essential tools which are used on a daily basis to support SEA1000, and the accuracy of the results obtained using these tools must be continuously monitored. The aim of this work is to assess the accuracy of the RANS solver in OpenFOAM by performing simulations on geometries of interest and comparing these with the results obtained from the Fluent code.

OpenFOAM is an open source software suite containing a large number of Computational Fluid Dynamics (CFD) solvers, including the Reynolds Averaged Navier-Stokes (RANS) solver simpleFoam. It also contains a large number of standard turbulence models, including the standard k-ε model, the RNG k-ε model, the Realizable k-ε model, the k-ω model, the k-ω SST model and the Spalart-Allmaras model. The aim of the simulations discussed here is to asses the level of agreement between the simulated results obtained from the simpleFoam solver in OpenFOAM compared with those obtained from the RANS solver in the commercial code Fluent. Simulations have been performed on three variants of the generic DSTO conventional submarine scale model; the bare hull of the generic model in a conventional wind tunnel section, the fully appended model (the bare hull plus aft control surfaces, deck and sail) in a cavitation tunnel test section and the fully appended model in an “open water” scenario. The meshes used for each of these simulations are meant to provide realistic meshes for the codes to work with, but have not been designed to reproduce the exact details of either the wind tunnel or cavitation tunnel geometries. Hence the simulation results have not been compared with experimental data, the aim of these simulations being to identify differences between the codes themselves. Since the simulation result obtained on a given mesh depends significantly on the discretization method used to represent the divergence term in the momentum equation a number of different discretization schemes were tested for each of the codes. The relative performance of some of these discretization methods was found to depend on the mesh chosen for the simulation. All simulations were performed using the standard k-ε turbulence model. This is only a first step however, and further comparative simulations should be conducted using other available turbulence models.

The simulations have shown significant differences between the results obtained from the two codes. For example, differences of the order of 15% in the value of the drag coefficient have been found in some simulations. A partial resolution of these differences has been found and the discrepancy has been reduced in some cases, but unacceptable differences are still present. This report summarizes the work performed so far to highlight and resolve these differences, and suggests further work which needs to be done to provide confidence in the use of simpleFoam.

2. Previous Comparisons between Fluent and OpenFOAM

A number of different computational fluid dynamics groups have recently been investigating the possibility of using OpenFOAM software to replace commercial CFD codes such as Fluent for performing standard RANS simulations. Nebenführ [1] used the simpleFoam solver and the k-ε Realizable turbulence model to calculate the lift and drag forces on two model car geometries (designated as the VRAK model and the S80 model). For the VRAK model the OpenFOAM drag coefficient was 2.4% higher than the Fluent value and agreed better with experiment, while the lift coefficient differed from the Fluent value by 53% and also gave slightly better agreement with the experimental value. For the S80 model the OpenFOAM drag coefficient was 9% higher than the Fluent value and the lift coefficient was 16.5% higher than the Fluent value. Both model car geometries were meshed using Harpoon, which is a commercial software package which provides fully automatic meshing of complex geometries and produces meshes which are hex-dominant. It is similar to the snappyHexMesh software provided in OpenFOAM. Unfortunately no information was provided on the range of y^+ values on the meshes.

Clarke et al. [2] have also compared the simpleFoam solver in OpenFOAM with the standard Fluent solver by simulating the flow around the fuselage of the ARH Tiger helicopter. They used the ICEM software package to produce an unstructured mesh containing prism layers on the surface of the fuselage and tetrahedral cells elsewhere. The average y^+ value was 68, but varied between 0.0986 and 223.76. Simulations were performed using the k-ω SST turbulence model. They found that the value of the drag coefficient calculated using OpenFOAM was 1.2% higher than the Fluent value, but the OpenFOAM lift coefficient was 9.1% lower than the Fluent value.

Tapia [3] compared OpenFOAM with Fluent for the modelling of wind flow over complex terrain. Velocity profiles were compared for two test cases; flow over a flat plate and flow over an axisymmetric hill. The mesh for the flat plate simulation was constructed from quadrilateral cells and had y^+ values varying between 40 and 75, while the mesh for the axisymmetric hill was a structured hex mesh with y^+ values varying between 30 and 60. The simpleFoam solver was used for OpenFOAM and both the Fluent and OpenFOAM simulations used the standard k-ε turbulence model. The velocity profiles at various locations for the plate simulation were identical for the two codes. For the axisymmetric hill the experimental data showed a shallow separation occurring on the lee side of the hill. This was well reproduced by the Fluent simulation but was significantly over predicted by the simpleFoam simulation, even though both codes used the same turbulence model and second order solution algorithms. Outside the separation region however the velocity profiles calculated by the two codes were identical.

Andersen and Nielsen [4] compared the performance of OpenFOAM with Fluent by simulating flow in a Burner Flow Reactor (BFR). Two cases were considered, a cold-flow simulation using the simpleFoam solver in OpenFOAM and a gas combustion simulation using the reactingFoam solver in OpenFOAM. A structured two-dimensional quadrilateral mesh and the standard k-ε turbulence model were used. For the cold-flow simulations the results from both codes were very similar, but for a fuel-rich combustion simulation there were significant differences. The gas combustion model used in OpenFOAM however is different to the one used in Fluent and so these differences were not unexpected. Even though the velocity plots in the cold-flow simulations showed very similar profile shapes and agreed very closely at most locations, there were some locations where the peak values differed by up to 30%.

Murcia [5] compared OpenFOAM and Fluent by simulating steady, viscous flow along a section of the Mississippi river. A structured hex mesh with the standard k-ε turbulence model and standard wall functions were used in both codes. The OpenFOAM solver used was simpleFoam. Contour plots of the turbulent kinetic energy, turbulent energy dissipation, and pressure and velocity magnitude were virtually identical for the two codes. More accurate comparisons were made by comparing velocity profiles at specified locations and whilst the overall shapes of the plots were similar differences of around 50% were found in the velocity magnitude at some locations.

3. Bare Hull Mesh

These simulations model the flow around the bare hull of a generic conventional scale model submarine of length 1.35 m in the Aerospace Division Low Speed Wind Tunnel (LSWT). The geometry is based on the work of Joubert [6-7] and was designed to achieve low resistance and flow noise. The hull form is made of an axisymmetric body of revolution with a length to diameter ratio of 7.3. The bow shape profile is based on a NACA 0014.2-N00.20 curve, the midbody is a uniform cylindrical section while the after body is based on a number of successive parabolic sections. The support stand used in the experimental arrangement has not been included in the mesh as the aim of these simulations is to compare simulated results between codes, rather than to obtain exact agreement with experimental results. A one quarter geometry mesh was used for the simulation and the bare hull was oriented at zero angle of incidence. The y^+ value was approximately 100 over the majority of the cylindrical mid-section of the hull surface. The mesh contained approx 0.9M hex cells. The standard k-ε turbulence model was used with standard wall functions. Figures 1 and 2 show the bare hull and mesh configuration. The mesh has been reflected through one of the symmetry planes for greater visibility.

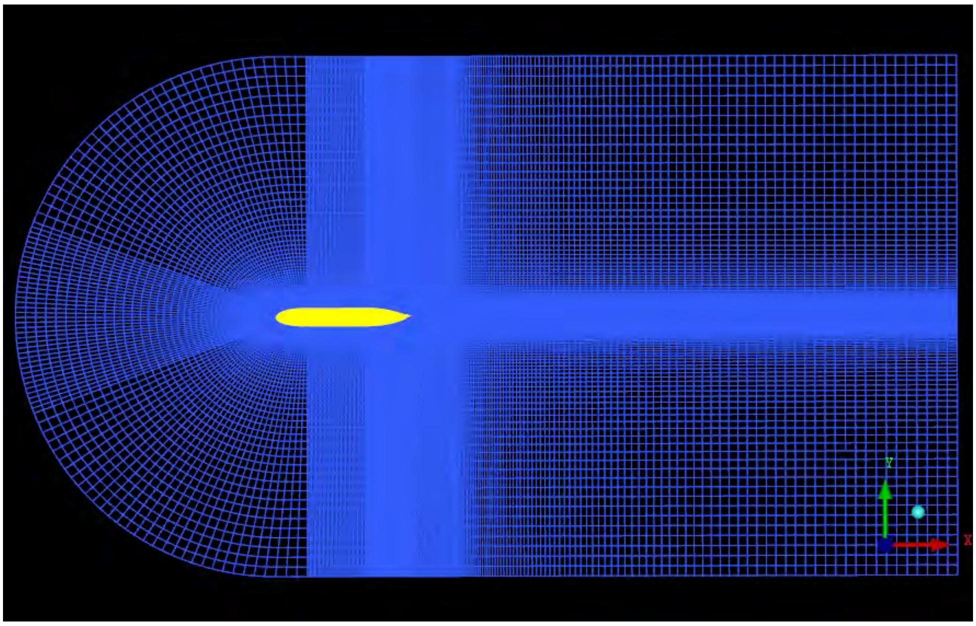

Figure 1: Domain used for the simulation on the bare hull of the DSTO generic conventional scale model submarine.

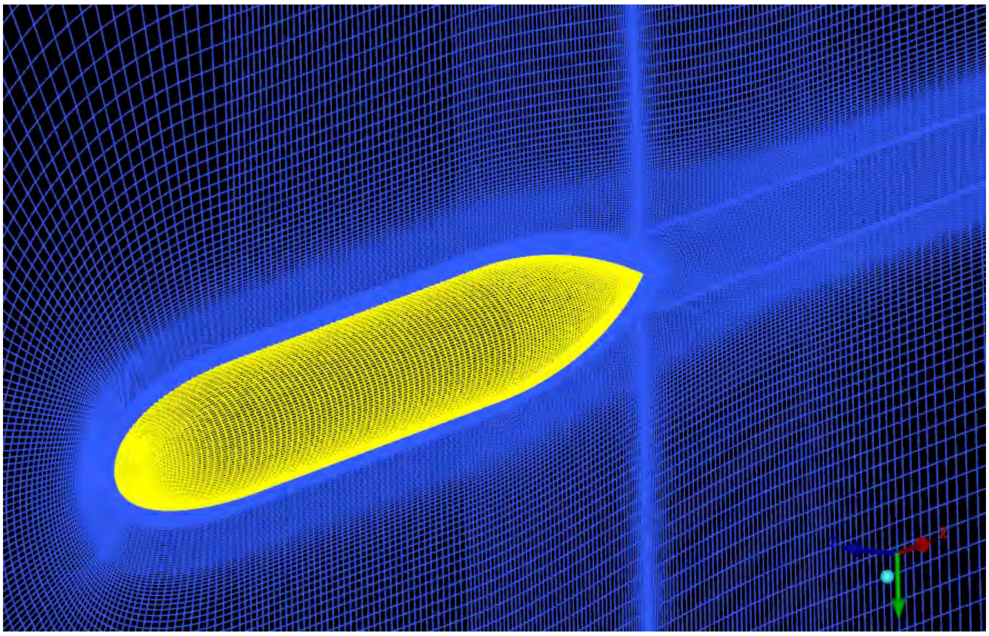

Figure 2: Detailed view of the mesh on the hull surface and the symmetry plane for the DSTO generic conventional scale model submarine.

Simulations were conducted for an inflow velocity of 60 m/s corresponding to a Reynolds number of 5.5 x 10^6, which is in agreement with the value used in the LSWT. The initial turbulent intensity was set to 3% and the turbulent viscosity ratio to 5. Therefore k = 4.86 m^2/s^2 and ε = 2.9 x 10^4 m^2/s^3. These are representative values used to initiate the simulation. Previous experience with both codes has shown that the final simulation results are insensitive to these initial values. The density ρ = 1.225 kg/m^3 and viscosity μ = 1.789 x 10^{-5} kg/(m-s). A velocity inlet boundary condition was applied at the inflow and a pressure outlet boundary condition was applied at the outflow. A no slip boundary condition was applied to the hull surface and the remaining surfaces, consisting of the side wall, symmetry wall and far field, were set to symmetry.

3.1 Fluent simulations

The simulated value of the drag coefficient is shown in Table 1 for each of the available discretization schemes available in Fluent. The gradient discretization used the least squares, cell based scheme. The divergence discretization in the momentum equation was varied as in Table 1 below. The equations for *k* and ε were solved using the upwind scheme in all cases. The pressure discretization was second order.

Table 1. Effect of discretization scheme used for the convective divergence term in the momentum equation on the simulated drag coefficient. Fluent simulation on bare hull.

Discretization scheme - Fluent	C_d	$F_{pressure}$ (N)	$F_{viscous}$ (N)
2nd order upwind	0.001385	0.1369	1.2549
QUICK	0.001376	0.1311	1.2517
MUSCL	0.001376	0.1321	1.2502
Power Law	0.001701	0.4588	1.2499
upwind	0.001687	0.4459	1.2490

3.2 OpenFOAM simulations

All runs were performed using OpenFOAM-1.5 and the simpleFoam code. In the fvSchemes file the gradSchemes were set to Gauss linear, the laplacianSchemes to Gauss linear corrected and the div(phi,k) and div(phi, epsilon) terms set to Gauss upwind.
In the fvSolution file the pressure solver used was PCG with the DIC preconditioner. The tolerance was set to 1.0e-08 and relTolerance to 0.0. For U, k and epsilon the PBiCG solver was used with the DILU preconditioner. The tolerance was typically set to 1.0e-06 and the relTolerance to 0.0. The number of nonorthogonal correctors was set to zero. All runs showed excellent convergence with the residuals decreasing by 5 or 6 orders of magnitude. The simulated value of the drag coefficient is shown in Table 2 for a small subsection of the many discretization schemes available in OpenFOAM (approximately 40 to 50 different schemes).

Table 2. Effect of discretization scheme used for the convective divergence term in the momentum equation on the simulated drag coefficient. OpenFOAM simulation on bare hull.

Discretization scheme - OpenFOAM	C_d	$F_{pressure}$ (N)	F_{visc} (N)
upwind	0.001652	0.4339	1.2259
linearUpwind Gauss linear	0.001316	0.1020	1.2206
linearUpwind cellLimited Gauss linear 1.0	0.001322	0.1105	1.2172
blended 0.9	0.001343	0.1313	1.2183
Gamma 0.5	0.001317	0.1005	1.2229
Gamma 0.9	0.001323	0.1010	1.2285

3.3 Discussion

Both Fluent and OpenFOAM store discrete values of the flow variables at the cell centres, but during the solution procedure values for these variables must also be known at the cell faces. The simplest way to achieve this is to assume that the cell centre values of any variable represent an average value and that this average value is valid throughout the entire cell. When using Upwind interpolation the face value is set equal to the cell centre value in the upstream cell. This is equivalent to using a backward or forward difference approximation for the first derivative and hence has first order accuracy. Fluent also has a Power Law discretization scheme which interpolates the face value of a variable using the exact solution to a one-dimensional convection-diffusion equation. When the flow is dominated by convection however, as is the case in these simulations, the Power Law scheme is effectively equivalent to the Upwind scheme. This is evident in the simulated values shown in Table 1 and 2. The drag coefficient calculated using the Fluent formulation of the Upwind and Power Law schemes is 0.001687 and 0.001701 respectively, a difference of 0.8%, while the OpenFOAM Upwind value is 0.001652, which is only 2% different from the average first order Fluent result.

Higher order accuracy can be obtained in a number of different ways. Fluent has a second order Upwind scheme based on using a Taylor series expansion about the cell centroid. Both the variable and its gradient are evaluated at the centroid and the value of the gradient is limited so that no new maxima or minima are introduced into the solution. Fluent also has the QUICK (Quadratic Upstream Interpolation for Convective Kinetics) scheme of Leonard [8] which uses a three-point upstream-weighted quadratic interpolation for cell face values. Since the scheme is based on a quadratic function it has third order accuracy on a uniform mesh, although Ferziger and Peric [9] state that the overall approximation is still only second order accurate. The scheme can result in minor overshoots or undershoots for the flow variables in some cases however and this can be problematic. A negative value for the turbulent kinetic energy in the k-ε model needs to be avoided, for example. A large number of second order schemes have been developed which avoid the problems of overshoot and undershoot which occur in the more standard second order approaches, such as the Central Differencing Scheme. Fluent contains the MUSCL (Monotonic Upstream-Centred Scheme for Conservation Laws) scheme introduced by van Leer [10]. This scheme uses linear piecewise approximations in each cell which are slope limited (similar to flux limiting) to ensure a fully Total Variation Diminishing (TVD) scheme, thereby avoiding the spurious oscillations which could otherwise occur around discontinuities. Such schemes are basically a nonlinear combination of first and second order approximations to the variables fluxes which tend to first order at local extrema but tend to second order over smooth parts of the domain. The MUSCL scheme implemented in Fluent however is claimed to be a third order scheme obtained by blending a central differencing scheme and a second order upwind scheme [11]. It is interesting to note however that the value of the drag coefficient calculated from the MUSCL scheme in Fluent is identical to the value obtained from the QUICK scheme, 0.001376, which is only 0.6% lower than the value obtained from the second order Upwind scheme in Fluent, which is 0.001385.

OpenFOAM contains a large number of discretization schemes for the convective divergence term in the momentum equation. The one which corresponds most closely to the second order Upwind scheme in Fluent is the linearUpwind scheme, which uses the gradient term to extend the flux between neighbouring cells to second order accuracy. When using this scheme the user also has the option of specifying the discretization scheme used for the gradient term. Gauss linear is the simplest approximation to use, but it is also possible to limit the gradient in a number of ways. In the OpenFOAM simulations listed in Table 2 the result of using both Gauss linear and Gauss cellLimited are shown. The difference between the two schemes in this case is minimal, the value of the drag coefficient for the linearUpwind Gauss linear simulation being 0.001316 while that using the linearUpwind cellLimited scheme is 0.001322, a difference of 0.45%.

A number of non-linear flux limited schemes are also available in OpenFOAM. The Gamma scheme is a smooth and bounded blend between a second order central differencing scheme and a first order upwind scheme. Central differencing is used wherever it satisfies the boundedness requirements and upwind is used wherever central differencing is unbounded [12]. The degree of blending is controlled by the numerical parameter. Table 2 shows that the calculated values of the drag coefficient using the Gamma 0.5 and Gamma 0.9 schemes are 0.001317 and 0.001323 respectively. The difference between these two schemes is again minimal, being only 0.45%.

The blended 0.9 scheme is a blend of 90% central differencing and 10% upwind. The value of the drag coefficient using this scheme is 0.001343, which is 1.7% higher than the average result from the Gamma scheme and 1.8% higher than the average result from the linearUpwind scheme.

Our experience with Fluent has shown that the second order upwind discretization scheme gives the best agreement with experiment. The drag coefficient calculated using this scheme has the value 0.001385. In OpenFOAM the linearUpwind scheme corresponds most closely to the second order upwind scheme in Fluent. For the two slightly different implementations of this scheme tested here the drag coefficient varied between 0.001316 and 0.001322, giving an average value for the linearUpwind scheme of 0.001319, which is 4.8% lower than the value calculated using Fluent and surprisingly high given that the two discretization schemes are very similar.

4. Fully Appended Hull – Cavitation Tunnel Mesh

The geometry considered in this section is the fully appended DSTO generic conventional submarine model based on the work of Joubert [6-7]. This consists of the axisymmetric hull described in the previous section with the addition of a casing, sail and aft control surfaces. The edge of the casing is tangent to the hull midbody, the fin is a rectangular planform with constant cross-section based on a NACA-0015 airfoil section and has minimum rounding at the tip. The aft control surfaces are four rudders in an X configuration. A fully structured 3D hex mesh containing approximately 16M cells was used to simulate flow around this model geometry in the cavitation tunnel at the Australian Maritime College in Launceston. The mesh includes both the support swords used in the experimental arrangement as well as the presence of the walls of the cavitation tunnel. Figure 3 shows the mesh on the symmetry plane and on the surface of the model while Figure 4 shows a close up view of the mesh on the symmetry plane and on the sail/casing junction. The y^+ value over the hull, sail and control surfaces was in the range 20 to 40.

Simulations were performed at zero degrees angle of incidence at a flow speed of 7 m/s, corresponding to a Reynolds number of 9.45 x 10^6, in agreement with typical values used in the experimental runs. The goal of these simulations however is not to attempt to obtain precise agreement with a set of experimental results, but rather to compare the simulation results from the two codes.

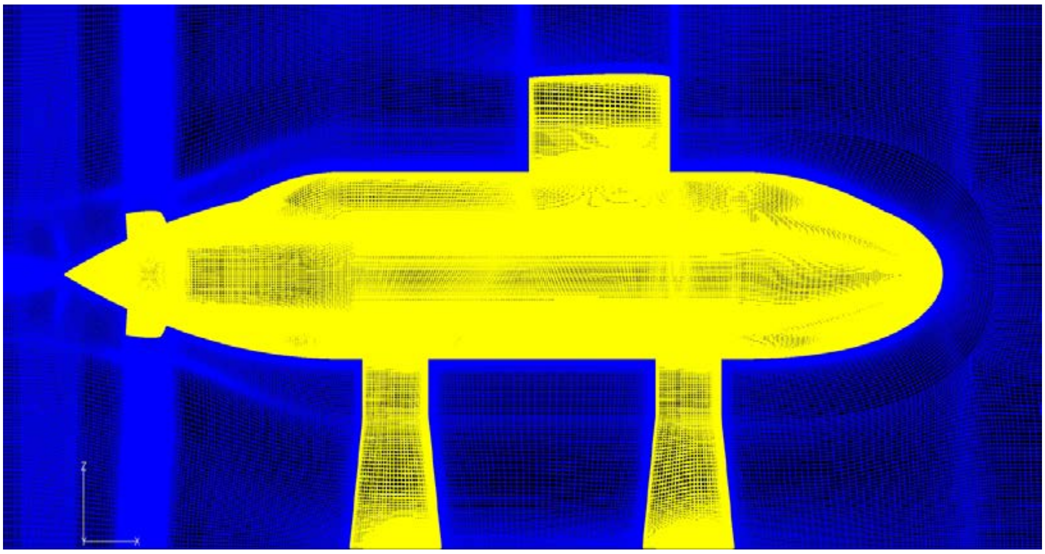

Figure 3: Mesh on the symmetry plane and on the surface of the fully appended hull of the DSTO generic conventional scale model submarine in the cavitation tunnel.

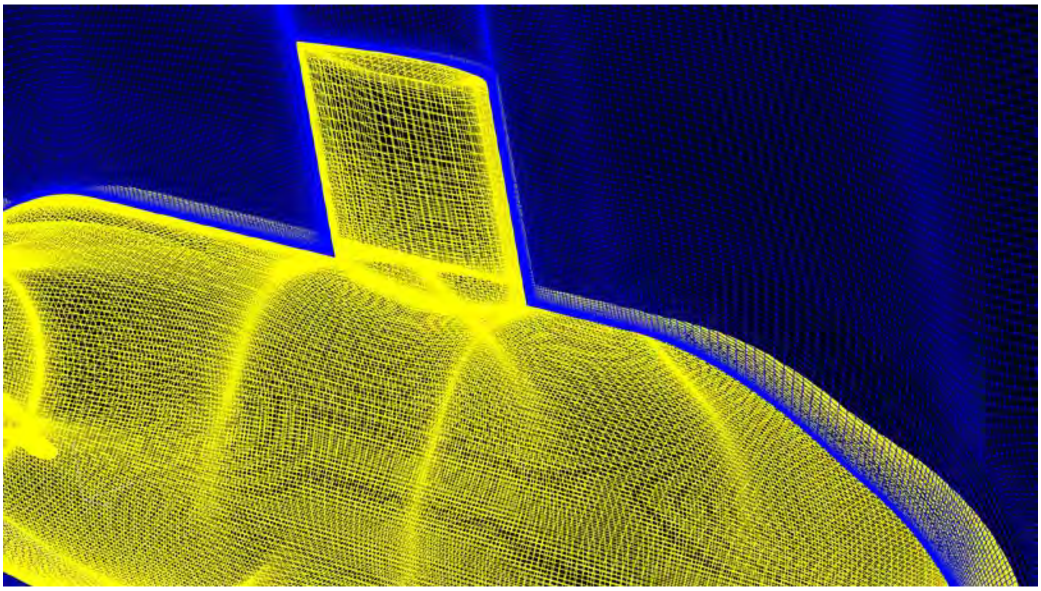

Figure 4: Close up view of the mesh on the symmetry plane and on the sail/casing junction for the fully appended hull of the DSTO generic conventional scale model submarine in the cavitation tunnel.

The standard k-ε turbulence model was used with standard wall functions. The initial turbulent intensity and turbulent viscosity ratio were left at the Fluent default values of 10% and 10 respectively. Hence the initial turbulent kinetic energy *k* and turbulent dissipation rate ε have the following values; $k = 0.735$ m^2/s^2 and $\varepsilon = 4.84 \times 10^3$ m^2/s^3. The working fluid was water having a density of 998.201 kg/m^3 and viscosity of 1.003×10^{-3} kg/(m-s). A velocity inlet boundary condition was applied at the inflow and a pressure outlet boundary condition was applied at the outflow. A no slip boundary condition was applied to the hull and appendage surfaces and the tunnel walls.

4.1 Fluent simulations

The gradient discretization used the least squares, cell based scheme. The divergence discretization in the momentum equation was varied as in Table 3 below. The equations for *k* and ε were solved using the upwind scheme in all cases. The pressure discretization was second order. In most cases the residuals decreased by at least four orders of magnitude.

Table 3. Effect of discretization scheme used for the convective divergence term in the momentum equation on the simulated drag coefficient. Fluent simulation on fully appended hull in cavitation tunnel.

Discretization scheme - Fluent	C_d	$F_{pressure}$ (N)	$F_{viscous}$ (N)
2nd order upwind	0.002525	43.300	69.227
MUSCL	0.002429	39.724	68.523
QUICK	0.002960	60.519	71.392
Power Law	0.002938	62.907	67.985
upwind	0.002941	62.919	68.157

4.2 OpenFOAM simulations

In the fvSchemes file the gradSchemes were set to Gauss linear, the laplacianSchemes to Gauss linear corrected and the div(phi,k) and div(phi, epsilon) terms set to Gauss upwind. In the fvSolution file the pressure solver used was PCG with the DIC preconditioner. The tolerance was set to 1.0e-08 and the relTolerance to 0.0. For the momentum, k and ε equations the PBiCG solver was used with the DILU preconditioner. The tolerance was typically set to 1.0e-06 with and the relTolerance to 0.0. These settings are the same as those used for the simulations of the bare hull in the LSWT.

Table 4. Effect of discretization scheme used for the convective divergence term in the momentum equation on the simulated drag coefficient. OpenFOAM simulation on fully appended hull in cavitation tunnel.

Discretization scheme OpenFOAM	C_d	$F_{pressure}$ (N)	$F_{viscous}$ (N)
upwind	0.003398	77.838	73.609
blended 0.9	0.003406	78.165	73.642
linearUpwind Gauss linear	0.002908	55.955	73.646
Gamma 0.9	0.002886	55.029	73.622

4.3 Discussion

As was found in the previous section for the bare hull simulations, the values for the drag coefficient calculated using the Fluent upwind and Power Law schemes are very similar, 0.002941 and 0.002938 respectively, differing by 0.1%. For the bare hull all the second order Fluent schemes gave very similar values for the drag coefficient, while for the fully appended hull there are differences. The MUSCL value of 0.002429 is 3.8% lower than the second order upwind value of 0.002525, while the QUICK value of 0.002960 is 17.2% higher than the second order upwind value and much closer (within 0.7%) to the first order upwind value.

For the OpenFOAM results we again find good agreement between the drag coefficient calculated using the linearUpwind scheme, 0.002908, and the value calculated using the Gamma scheme, 0.002886, with the two values differing by 0.76%. For the bare hull simulations the blended 0.9 scheme predicted a value for the drag coefficient only 1.8% different from the value calculated using the linearUpwind scheme, but for the fully appended hull the difference is 17%, with the calculated value of 0.003406 agreeing much more closely with the upwind result of 0.003398, a difference of only 0.2%.

With regards a comparison between the two codes, the linearUpwind OpenFOAM value for the drag coefficient of 0.002908 differs from the Fluent second order upwind value of 0.002525 by 15.2%, which is significantly different given the similarity between the two schemes. It is also much larger than the 4.8% difference found for the bare hull simulations.

5. Fully Appended Hull – Open Water Mesh

The geometry considered in this section is again the fully appended DSTO generic conventional submarine model based on the work of Joubert [6-7]. In this case however no support structure was included in the mesh. Figure 5 shows the mesh on the symmetry plane and on the surface of the hull while Figures 6 and 7 show close up views of the mesh on the nose and stern regions respectively. The mesh is a fully structured 3D hex mesh with 8.5M cells. Although referred to as "open water" the simulations were performed in air with a velocity of 35 m/s, corresponding to a Reynolds number of 3.2×10^6, similar to the value obtained in experimental runs performed on the fully appended geometry in the LSWT. As mentioned in the previous sections however, the goal of these simulations is not to attempt to obtain precise agreement with a set of experimental results, but rather to compare simulation results from the two codes on a number of different meshes. For example, although the open water mesh in this section is smaller than the one used for the simulations in the cavitation tunnel, the y^+ values are lower, in a range between 10 and 12 over most of the hull, sail and control surfaces.

The standard k-ε turbulence model was used with standard wall functions. The initial turbulent intensity was set to 3% and the turbulent viscosity ratio to 5, hence $k = 1.65$ m^2/s^2 and $\varepsilon = 3.37 \times 10^3$ m^2/s^3. These are representative values used to initiate the simulation. Previous experience with both codes has shown that the final simulation results are insensitive to these initial values. The density $\rho = 1.225$ kg/m^3 and viscosity $\mu = 1.789 \times 10^{-5}$ kg/(m-s). A velocity inlet boundary condition was applied at the inflow and a pressure outlet boundary condition was applied at the outflow. A no slip boundary condition was applied to the hull, sail and aft control surfaces and the far field was set to symmetry.

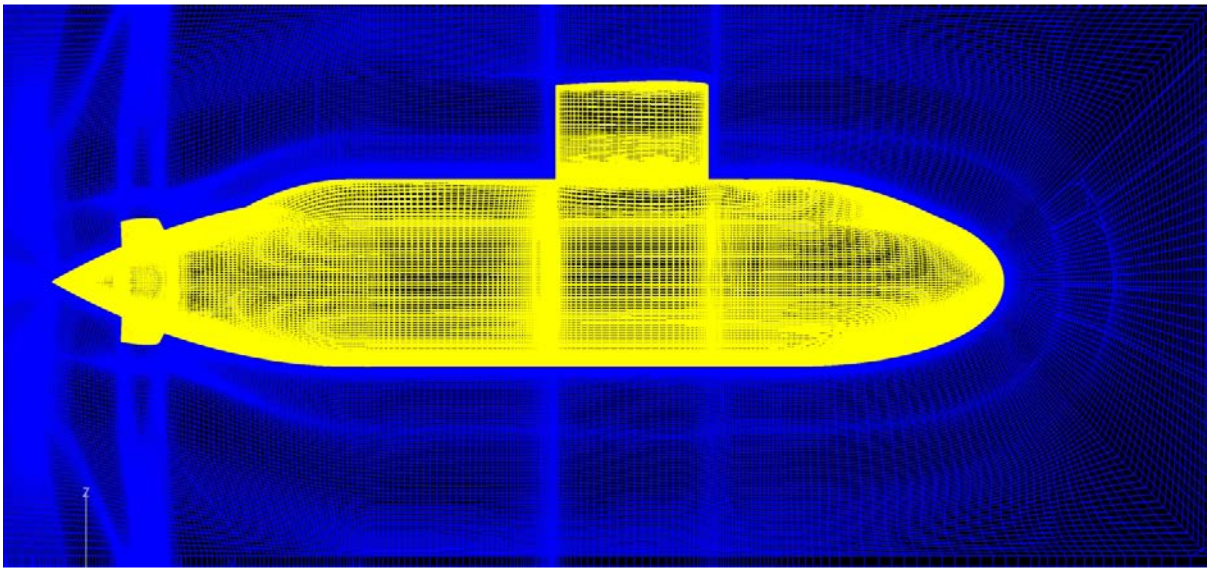

Figure 5: Mesh on the symmetry plane and on the surface of the fully appended hull of the DSTO generic conventional scale model submarine for the open water simulation.

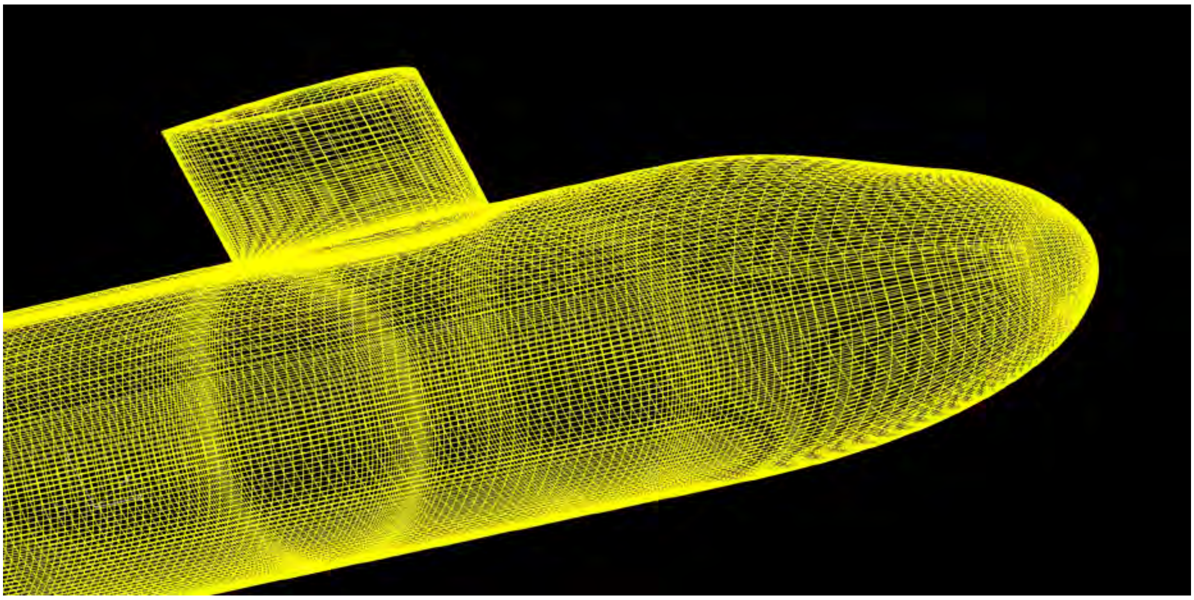

Figure 6: Close up view of the mesh on the surface of the fully appended hull of the DSTO generic conventional scale model submarine for the open water simulation. Bow view.

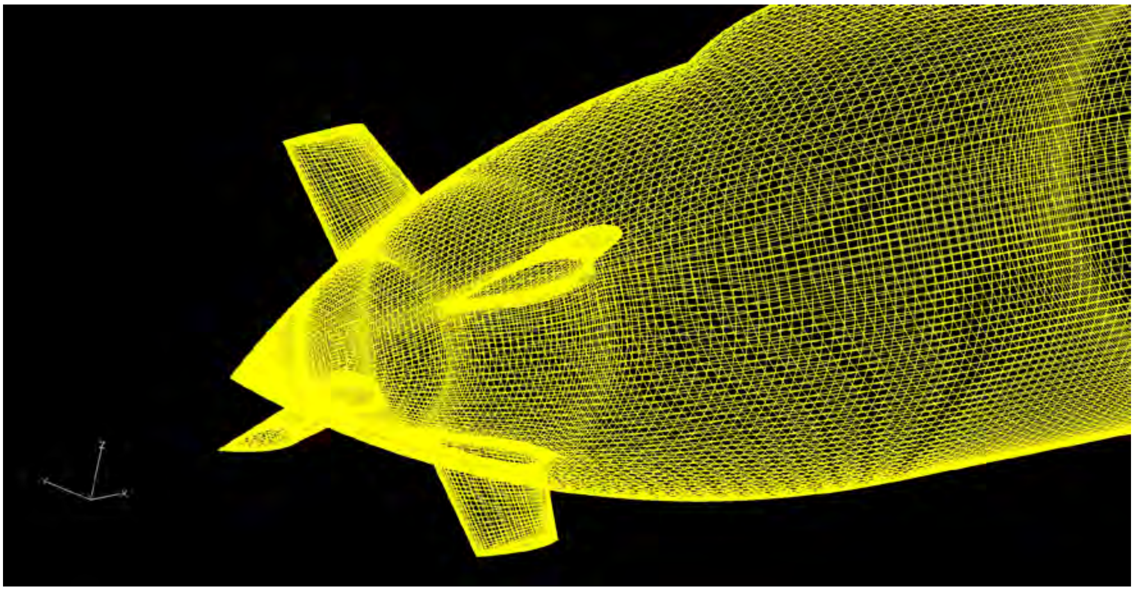

Figure 7: Close up view of the mesh on the surface of the fully appended hull of the DSTO generic conventional scale model submarine for the open water simulation. Stern view.

5.1 Fluent simulation

Only one run was performed using the second order upwind scheme for the divergence term in the momentum, k and ε equations. The gradient discretization used the least squares, cell based scheme. The result is shown in the table below.

Table 5. Simulated drag coefficient. Fluent simulation on fully appended hull, open water mesh.

Discretization scheme – Fluent	C_d	$F_{pressure}$ (N)	$F_{viscous}$ (N)
2nd order upwind	0.002070	0.4385	2.3921

5.2 OpenFOAM simulation

In the fvSchemes file the gradSchemes were set to Gauss linear, the laplacianSchemes to Gauss linear corrected and the div(phi,k) and div(phi, epsilon) terms set to Gauss upwind. The GAMC solver was used for the pressure equation with the tolerance set to 1.0e-10 and the relTol to 0.0. For the momentum, k and ε equations the PBiCG solver was used with the tolerance set to 1.0e-6 and the relTol to 0.0. The number of nonorthogonal correctors was set to 4. The pressure relaxation factor was set to 0.1 and the other relaxation factors to 0.3. Two runs were performed, one using the linearUpwind scheme for the convective term in the momentum equation and the other using a blended scheme. The results are shown in Table 6 below.

Table 6. Effect of discretization scheme used for the convective divergence term in the momentum equation on the simulated drag coefficient. OpenFOAM simulation on fully appended hull, open water mesh.

Discretization scheme – OpenFOAM	C_d	$F_{pressure}$ (N)	$F_{viscous}$ (N)
linearUpwind cellLimited Gauss linear 1	0.002396	0.6936	2.5829
blended 0.9	0.002385	0.6818	2.5795

5.3 Discussion

Only one simulation was run for the Fluent code using the second order upwind scheme as previous experience has shown this to be the most accurate scheme. For OpenFOAM two simulations were run; a cell limited linearUpwind scheme for comparison with the Fluent second order upwind scheme, and a blended 0.9 scheme. The drag coefficient calculated from the two OpenFOAM simulations differ by less than 0.5%. This is quite different to the situation for the fully appended model simulation in the cavitation tunnel, where the linerUpwind and blended 0.9 results differed by approximately 15%. It is similar to the bare hull case however, where the blended 0.9 scheme predicted a value for the drag coefficient only 1.8% different from the value calculated using the linearUpwind scheme.

The predicted value for the drag coefficient using the OpenFOAM linearUpwind scheme is 0.002396, which is 15.7% higher than the value calculated using the Fluent second order upwind scheme.

6. Partial Resolution of Differences

For the three different simulation cases, the bare hull, the fully appended hull in the cavitation tunnel and the fully appended hull open water mesh, the OpenFOAM results differ from the Fluent results by 4.8%, 15.2% and 15.7% respectively. Some insight into the cause of these differences can be obtained by noting that the turbulent viscosity is zero over a large portion of the hull during the OpenFOAM simulation on the fully appended model in the open water mesh. This indicates that OpenFOAM has turned off the production of turbulent kinetic energy in the vicinity of the body. In OpenFOAM the turbulent viscosity and epsilon wall functions check the local y^+ value and if it is less than the laminar value ($y^+ \sim 11$) then the turbulent viscosity is set to zero and the production of kinetic energy is also set to zero. To prevent this from happening, the epsilon wall function and turbulent viscosity wall function were modified by commenting out the check on the y^+ value (shown in Appendix 1 and Appendix 2). When running the simulation using these modified wall functions the value for the drag coefficient is 0.0020691, which reduces the difference from the Fluent result to less than 0.05%, a significant improvement.

The same modified wall functions were then used to re-run the simulations for the bare hull mesh and the fully appended model in the cavitation tunnel. In these cases the results were mixed. For the cavitation tunnel mesh the simulation was re-run using the Gamma 0.9 scheme for the divergence term in the momentum equation. Previously this gave a drag coefficient of 0.002886, which is 14.3% different from the Fluent result. Using the modified wall functions the simulated value was 0.002792, which is 10.6% different from the Fluent result. The average y^+ value was close to 30, although it varied between 2 and 108.

For the bare hull mesh the OpenFOAM simulation was re-run using the linearUpwind cellLimited Gauss linear 1 discretization of the divergence term in the momentum equation. Previously this gave a drag coefficient of 0.001322, which is 4.5% different from the Fluent result. Using the modified wall functions gave a simulated value of 0.001306, which is a difference of 5.7% from the Fluent result.

Given that the y^+ value for the bare hull mesh is close to 100 over most of the hull it isn't surprising that the modified wall functions have such a small effect on the simulated value for the drag coefficient as the switch to the laminar behaviour at $y^+ \sim 11$ would be triggered in very few cells. For the fully appended model in the cavitation tunnel however the average y^+ value is close to 30 but it does dip to 2 over parts of the hull, so we would expect the turbulent energy production to be turned off in these cells. Preventing this from happening by using the modified wall functions would therefore be expected to have a greater effect on the simulated drag coefficient, which is evident in the reduction in the difference in the simulated value between the two codes from 14.3% to 10.6%. For the open water mesh, where the y^+ value lies in a range between 10 and 12, the modified wall functions would be expected to have maximum impact, as has been demonstrated.

7. Conclusion

The discussion in the previous sections has shown that while the simulated flow fields produced by the two codes are very similar when compared from an overall perspective, significant differences can occur for calculated quantities such as drag and lift coefficients and maximum and minimum velocities at specific locations in the flow. While these differences have been documented in several publications over the last few years, no attempts appear to have been made to explain the source of these differences. The modification to the wall functions in OpenFOAM described in Section 6 partially explains the difference between simulated values calculated by OpenFOAM and Fluent when using the standard k-ε turbulence model, but differences between the results from the two codes remain and imply subtle differences between the codes. These differences could be due to slightly different implementations of the turbulence models, the treatment of near wall boundary conditions, or the numerical solution procedures employed. Stephens [13] has noted that the implementation of the k-ω and k-ω SST turbulence models in OpenFOAM is different to that in Fluent in order to provide more stable solutions.

Other differences between the codes could occur due to implementation of slightly different versions of each of the turbulence models. OpenFOAM-1.6 (the version we are currently running) contains the *k*-ω turbulence model as described by Wilcox in 1998 [14]. Since then different versions of this model have appeared. The NASA Langley Research Center [15] lists four of these. The version of the *k*-ω model currently implemented in Fluent incorporates modifications for low Reynolds number effects, compressibility and shear flow spreading [11]. The recent paper by Kim et al. [16] shows conclusively that the low Reynolds number version of the *k*-ω turbulence model is significantly better at predicting turbulent flow around a submarine hull in a turning manoeuvre. Comparisons between the *k*-ω model in OpenFOAM-1.6 and the *k*-ω model in Fluent 14.0 could not be expected to provide exact agreement unless the improvements contained in the Fluent implementation were turned off. Alternatively, since we have access to the source code, the *k*-ω model in OpenFOAM-1.6 could be modified to include the low Reynolds number effects. This would appear to be the more appropriate strategy.

8. Acknowledgements

The author would like to thank Dr. Darrin Stephens from Applied CCM Pty. Ltd for providing the modified wall function coding, Dr. Ronny Widjaja for the bare hull mesh, and Dr. Greg Seil for provision of the fully appended cavitation tunnel and open water meshes.

9. References

1. Nebenführ, B., “OpenFOAM: A tool for predicting automotive relevant flow fields”, Masters Thesis in Automotive Engineering, Division of Fluid Dynamics, Department of Applied Mechanics, Chalmers University of Technology, Göteborg, Sweden 2010.

2. Clarke, G., Vun, S., Giacobello, M. and Reddy, R., “Estimation of ARH Tiger Fuselage Aerodynamic Characteristics Using Computational Fluid Dynamics Tools”, DSTO-TN-0965, October, 2010.

3. Tapia, X.P., “Modelling of wind flow over complex terrain using OpenFOAM”, Masters Thesis, University of Gävle, Sweden, June 2009.

4. Andersen, C. and Nielsen, N.E.L., “Numerical Investigation of a BFR using OpenFOAM”, AAU- Institute of Energy Technology, Aalborg University, Denmark, February 2008.

5. Murcia, O.E.H., “Comparison of OpenFOAM and Fluent for steady, viscous flow at Pool 8, Mississippi River”, Iowa Institute of Hydraulics Research, The University of Iowa. Graduate student presentation, available on the internet at http://www.personal.psu.edu/dab143/OFW6/Presentations/oscar_eduardo_hernandez_murcia_slides.pdf

6. Joubert, P.N., “Some Aspects of Submarine Design Part 1 – Hydrodynamics”, Defence Science and Technology Organisation Technical Report DSTO-TR-1622, 2004.

7. Joubert, P.N., “Some Aspects of Submarine Design Part 2– Shape of a Submarine 2026”, Defence Science and Technology Organisation Technical Report DSTO-TR-1920, 2004.

8. Leonard, B.P., “A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation”, *Comput. Methods Appl. Mech. Eng*., **19**, pp. 59 – 98 (1979)

9. Ferziger, J.H. and Peric, M. “Computational Methods for Fluid Dynamics”, Springer-Verlag, 1996.

10. van Leer, B., “Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov’s Method”, *J. Com. Phys*., **32**, pp. 101 – 136 (1979).

11. Ansys Fluent 14.0 User Guide.

12. Jasak, H., “Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows”, Ph.D. thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, University of London, June 1996.

13. Stephens, D., Applied CCM Pty. Ltd., private communication, July 24th, 2013.

14. Wilcox, D.C., “Turbulence Modeling for CFD”, 2nd Edition, DCW Industries, Inc., 1998.

15. Turbulence Modeling Resource, NASA Langley Research Center. Available on the internet at http://www.turbmodels.larc.nas.gov

16. Kim, S.E., Rhee, B.J. and Miller, R.W., “Anatomy of Turbulent Flow around DARPA SUBOFF Body in a Turning Maneuver Using High-Fidelity RANS Computation”, 29th Symposium on Naval Hydrodynamics, Gothenburg, Sweden, 26-31 August, 2012.

Appendix 1 - Portion of modified nutWallFunction

```
tmp<scalarField> nutWallFunctionFvPatchScalarFieldMod::calcNut() const
{
    const label patchI = patch().index();

    const RASModel& rasModel = db().lookupObject<RASModel>("RASProperties");
    const scalarField& y = rasModel.y()[patchI];
    const tmp<volScalarField> tk = rasModel.k();
    const volScalarField& k = tk();
    const scalarField& nuw = rasModel.nu().boundaryField()[patchI];

    const scalar Cmu25 = pow(Cmu_, 0.25);

    tmp<scalarField> tnutw(new scalarField(patch().size(), 0.0));
    scalarField& nutw = tnutw();

    forAll(nutw, faceI)
    {
        label faceCellI = patch().faceCells()[faceI];

        scalar yPlus = Cmu25*y[faceI]*sqrt(k[faceCellI])/nuw[faceI];

//      if (yPlus > yPlusLam_)
//      {
            nutw[faceI] = nuw[faceI]*(yPlus*kappa_/log(E_*yPlus) - 1.0);
//      }
    }

    return tnutw;
}
……….
```

Apendix 2- Portion of modified epsilonWallFuntion

```
// Set epsilon and G
   forAll(nutw, faceI)
   {
      label faceCellI = patch().faceCells()[faceI];

      scalar yPlus = Cmu25*y[faceI]*sqrt(k[faceCellI])/nuw[faceI];

      epsilon[faceCellI] = Cmu75*pow(k[faceCellI], 1.5)/(kappa_*y[faceI]);

//       if (yPlus > yPlusLam)
//       {
           G[faceCellI] =
              (nutw[faceI] + nuw[faceI])
             *magGradUw[faceI]
             *Cmu25*sqrt(k[faceCellI])
             /(kappa_*y[faceI]);
//       }
//       else
//       {
//           G[faceCellI] = 0.0;
//       }
   }
```

DISTRIBUTION LIST

CFD RANS Simulations on a Generic Conventional Scale Model Submarine: Comparison between Fluent and OpenFOAM

D.A. Jones

AUSTRALIA

Task Sponsor

Director General Future Submarine Program, CDRE Michael Houghton	1

S&T Program

Navy Scientific Adviser, Andrew Bailey	1
Chief of Maritime Division, Ms Janis Cocking	1
SEA 1000 S&T Advisor, Mr Kevin Gaylor	1
Research Leader Underwater Platform Systems, Mr Kevin Gaylor	1
Research Leader ASW Systems, Dr David Liebing	1
Research Leader Littoral Warfare Systems, Dr Bryan Jessup	1
Research Leader Signature Management, Mr Leo de Yong	1
Research Leader Platform Survivability, Dr Chris Norwood	1
Research Leader Surface Platform Systems, Dr Stuart Cannon	1
Research Leader Aircraft Performance and Survivability, Dr Greg Bain	1
Staff Officer Science SEA 1000, Mr Ross Susic	1
Head Hydrodynamics: Mr Brendon Anderson	1
Dr David A Jones	1
Dr Daniel Norrison	1
Dr David Clarke	1
Dr. Matteo Giacobello	1
DST Group SEA 1000 Project Coordinator, Ms Carol Batras	1
Head Naval Architecture and Platform Systems, Mr Andrew Tynan	1
SEA 1000 Adelaide Technology and Operational Analysis Liaison, Dr Robert O'Dowd	1

Capability Development Group

Director Future Submarine Program Capability Development, CAPT Stephen Dalton	1

Navy

Director General Submarine Capability, CDRE Peter Scott	
SO Science Fleet Headquarters, Sean Franco	1

SEA 1000 Integrated Project Team

Deputy IPT Lead and Whole Boat Designer, Dr Chris Edmonds (UK, contractor)	1
Simon Binns (IPT, UK, contractor)	1

Future Submarine Program – Adelaide Office

Chief Engineer, Mr Pat Donovan	1

Capability Acquisition and Sustainment

Head Future Submarine Program, RADM Greg Sammut	1
Director of Engineering Future Submarine Program, Mr David Simcoe	1
Chief Engineer Collins Submarines Program, CAPT Adam Lindsay	1
Submarine Program Chief of Staff, Mr Bob Clarke	1
SEA 1000 Senior Naval Architect, Mr Tim Gates	1

FOI

Dr. Christer Fureby	1
Dr. Mattius Liefvendahl	1
Dr. Mattius Johansson	1

Total number of copies: 33

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY GROUP DOCUMENT CONTROL DATA	1. DLM/CAVEAT (OF DOCUMENT)

2. TITLE CFD RANS Simulations on a Generic Conventional Scale Model Submarine: Comparison between Fluent and OpenFOAM	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)
4. AUTHOR(S) D. A. Jones	5. CORPORATE AUTHOR Defence Science and Technology Group 506 Lorimer St Fishermans Bend Victoria 3207 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DST-Group-TN-1449	AR-016-390	Technical Note	September 2015

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
eg: 2009/1034056	07/386r	DGFSP	16	12

13. DSTO Publications Repository	14. RELEASE AUTHORITY
http://dspace.dsto.defence.gov.au/dspace/	Chief, Maritime Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DST RESEARCH LIBRARY THESAURUS

Computational fluid dynamics, CFD, Simulation, Reynolds Averaged Navier-Stokes (RANS) equations

19. ABSTRACT

The ability to perform accurate Computational Fluid Dynamic (CFD) simulations of the flow around submarines is crucial for support to SEA1000. The Hydroacoustics Group in MD have been using the commercial software code Fluent for some years now to perform these simulations, but recently have been considering using the open source code OpenFOAM to replace some of the Fluent simulations. The fidelity of the Fluent code has been carefully validated, but the accuracy of parts of the OpenFOAM code have not been so extensively tested. To test the accuracy of the OpenFOAM software, CFD simulations have been performed on the DSTO generic conventional submarine model using both Fluent and OpenFOAM. A comparison of the value of the drag coefficient calculated by the two codes shows differences of up to 15%. A partial resolution of these differences has been found and the discrepancy has been reduced in some cases, but unacceptable differences are still present. This report summarizes the work performed so far to highlight and resolve these differences, and suggests further work which needs to be done to provide confidence in the use of these codes.

Page classification: UNCLASSIFIED