Verification and Validation of OpenFOAM for High-Lift Aircraft Flows

Neil Ashton* and Vangelis Skaperdas†

*University of Oxford, Oxford OX2 0ES, United Kingdom
†BETA CAE Systems, Thessaloniki 57500, Greece

DOI: 10.2514/1.C034918

This paper presents a detailed investigation into the performance of the open-source finite volume computational fluid dynamics (CFD) code OpenFOAM for complex high-lift aircraft flows. A range of cases are investigated, including a zero-pressure gradient flat plate, a NACA0012 airfoil at varying angles of attack, a DSMA661 airfoil, a NASA High-Lift Common Research Model, and a Japan Aerospace Exploration Agency (JAXA) standard model (JSM) high-lift aircraft model. The final three cases were computed as part of the third AIAA High Lift Prediction Workshop. For all of these, the same mesh and turbulence model is used to benchmark against the commercial CFD code STAR-CCM+. The paper shows that OpenFOAM using the Spalart–Allmaras model matches the lift and drag coefficient within 3% of the commercial code for all the test cases simulated. For the JSM high-lift aircraft for which experimental data is available, both codes show good agreement at pre-stall angles of attack but fail to capture the location of separation at post-stall angles, even though the global lift appears to be well predicted. Although OpenFOAM demonstrated comparable accuracy to a range of CFD codes for these aforementioned test cases, further work is required to improve the robustness and stability of OpenFOAM for these types of flows.

Nomenclature

- \( C_D \) = drag coefficient
- \( C_f \) = skin-friction coefficient
- \( C_L \) = lift coefficient
- \( C_M \) = pitching moment coefficient
- \( C_p \) = pressure coefficient
- \( c \) = chord
- \( h \) = \( \sqrt{1/N} \)
- \( M \) = Mach number
- \( N \) = volume mesh cell count
- \( p \) = pressure
- \( Re \) = \( cU_\infty/\nu \)
- \( U_\infty \) = freestream velocity
- \( T_\infty \) = freestream static temperature
- \( \eta \) = ratio of spanwise coordinate to airplane semispan
- \( \nu \) = kinematic viscosity
- \( \rho \) = density

I. Introduction

One of the primary drivers of growth in computational fluid dynamics (CFD) has been the aerospace industry. Over the past 40 years, it has evolved from a useful analysis method to a major design tool. At companies such as Boeing, much of the early wing design work is conducted almost exclusively in CFD until integration with the other components [1], e.g., high-lift devices. A range of computational techniques, from panel methods to Reynolds-averaged Navier-Stokes (RANS) models, is used in various parts of the CFD design process, depending on the competing needs of accuracy and turnaround time. As high-performance computing (HPC) facilities become ever cheaper and more available, RANS methods [and later, high-fidelity methods such as wall-modeled large-eddy simulation and hybrid RANS/large-eddy simulation (LES) methods] will gradually take over a greater share of the entire aerodynamic toolkit as the cost savings and flexibility over experimental methods become more viable [2–4].

However, at present, even though RANS methods are commonly used, this is only true in particular parts of the flight envelope, such as the cruise phase [1]. In high-lift configurations, RANS often fails to capture the correct flow physics, typically failing to capture the separation point and the resulting extent of the separated flow region (as shown at the recent AIAA High Lift Prediction Workshop (HLPW) [5]). Higher-fidelity methods arguably offer the greatest hope of using CFD in these more challenging regions of the flight envelope. The cost of hybrid RANS/LES methods such as detached-eddy simulation (DES) [6] is unfeasibly high for a design environment at present [7]; however, the progress of HPC means that this may now be less than 5–10 years away [2].

Many aerospace codes, for reasons of national security and confidentiality, are neither open source nor available to foreign nationals. For this reason, researchers wishing to collaborate openly and globally must look for open-source codes to develop both the methods and software practices to advance the use of CFD. The aerospace sector is a large user of CFD; however, other industries such as the automotive and energy sector are also key stakeholders in the future of CFD. Thus, any such codes should ideally be numerically flexible enough to solve a range of problem types.

There are a number of open-source CFD codes available, from aerospace focused (e.g., SU2 [8]) to energy (nuclear) focused: Code Saturne [9]. It is not within the scope of this paper to attempt to review all the different open-source CFD codes and, although there may be other codes that fit the bill of a general-purpose CFD code, OpenFOAM [10] is arguably one of the most widely used that meets this requirement. Over the past 15 years, it has developed from a university code to one that is used by both major industries and universities: largely because of its growing user base and the comprehensive set of solvers, turbulence models, and meshing capabilities. Its largest perceived weakness is a lack of verification and validation, which is particularly true in the aerospace sector, in which it does not have a demonstrated track record of matching the results from standard aerospace codes.
This paper attempts to address this and has the following aims:

1) Verify and validate OpenFOAM on a range of simple compressible test cases from the NASA Turbulence Modeling Resource (TMR) website.

2) As part of the AIAA HLPW, compute the NASA Common Research Model (CRM) in the high-lift configuration and compare it against major commercial code STAR-CCM+.

3) Compute a second high-lift full aircraft [the Japan Aerospace Exploration Agency (JAXA) standard model (JSM)] and compare it against experimental data and the commercial code STAR-CCM+.

Future goals are to explore improved RANS and hybrid RANS/LES methods; however, this paper focuses purely on the Spalart–Allmaras (SA) [11] model in order to allow comparison against the greatest number of other codes, given its status as one of the widely used turbulence models in the aerospace industry.

II. Verification and Validation

For all the following test cases, the computational setup is kept the same: other than the boundary conditions, which are specific for each test case. Two CFD codes are used in this study: OpenFOAM [10,12] and STAR-CCM+ by Siemens.

OpenFOAM is an open-source C++ toolbox that is most commonly used for CFD. It supports arbitrary polyhedral unstructured grids and contains a range of incompressible and compressible solvers. In these simulations, a segregated pressure-based solver (rhoPimpleFoam) is used with local time stepping to accelerate steady-state convergence; for stability purposes, the local Courant–Friedrichs–Lewy condition (CFL) was typically kept below five. The pimple approach is an OpenFOAM-specific variant of the Pressure-Implicit with Splitting of Operators (PISO) [13] approach in which the outer correction loops are used to improve convergence and stability (i.e., looping over a single time step a set number of times with underrelaxation). A second-order upwind scheme is used for the momentum and turbulent convective fluxes. This scheme was used after exhaustive attempts to achieve convergence with the density-based schemes within OpenFOAM, which coincides with the findings of Nikaido et al. [14]. More detailed information about the range of OpenFOAM solvers and numerical schemes was given by Robertson et al. [15].

STAR-CCM+ is a commercial CFD code, which uses a cell-centered finite volume discretization applied to cells of arbitrary polyhedral shapes and offers a range of available physical models and algorithms for a variety of applications. In these simulations, a fully implicit compressible density-based scheme is used with the Roe scheme for the flux. A CFL number of 5–30 was typically used based upon a balance between convergence and stability and a Green–Gauss scheme with a Minimum Modulus function (minmod) limiter was used for the gradient calculations. A second-order upwind scheme was used for the momentum and turbulent convective fluxes.

A. Zero-Pressure Gradient Flat Plate

The primary purpose of this test case is to verify the implementation of the SA turbulence model in OpenFOAM and STAR–CCM+ for a spatially developing boundary layer. As per the NASATMR website (see footnote ‡), we use the term verification in the context of establishing that the turbulence model equations are implemented correctly, and we compare against CFL3D and FUN3D that have undergone significant verification of the SA turbulence model. The mesh resolution in this study means that, by the finest grid, the only differences should be due to implementation and boundary conditions. The setup of the case is shown in Fig. 1a, and an illustration of the computational mesh and domain is shown in Fig. 1b. Four quasi-two-dimensional (quasi-2-D) meshes are used, with each successively containing double the number of cells in each spatial direction: 69 × 49, 137 × 97, 273 × 193, and 545 × 385.

It can be seen from Fig. 2 that both STAR–CCM+ and OpenFOAM agree well with the NASA structured (CFL3D and unstructured (FUN3D) codes for the skin-friction and drag coefficients, which is in agreement with Gomez et al. [16]. Both codes show less sensitivity to the mesh than CFL3D and FUN3D, which was also seen by

‡ Data available online at http://turbmodels.larc.nasa.gov [retrieved 15 January 2019].
Nikaido et al. [14]. The turbulent viscosity ratio is in good agreement between OpenFOAM, CFL3D, and FUN3D, with STAR–CCM+ having a slightly larger value. Due to the commercial nature of the code, it is not possible to see the exact implementation of the SA model, and thus there may be slight differences that cause this increase in the turbulence levels.

**B. NACA0012 Airfoil**

The NACA0012 airfoil provides an opportunity to assess both codes for a flow exhibiting flow separation. The setup of the case is shown in Fig. 7a, and an illustration of the computational mesh and domain is shown in Fig. 7b. Four grids of 281 × 49, 561 × 97, 1121 × 193, and 2241 × 385 cells were initially used from the NASA TMR site. It can be seen in Figs. 8a and 8b that STAR–CCM+, FUN3D, and CFL3D show similar mesh convergence trends for the lift and drag coefficient. For these codes, the streamwise velocities (Fig. 8c) are also in good agreement with each other and the experimental data by the finest grid.

For OpenFOAM, the result on the original mesh from the NASA TMR website shows clearly higher drag and lower lift as compared to the other codes. This trend is consistent between grids, and thus suggests that numerical dissipation is not causing this inaccuracy. Considering the close agreement between both codes for the flat-plate test case and the differing results between OpenFOAM and the other codes for NACA0012, it was concluded that the major difference between the codes was the wall-distance calculation.

For the SA model, the wall distance d is a key component of the turbulent dissipation term, acting as the turbulent length scale [11]. There are a range of methods to compute the distance to the closest wall, which typically trade off computational cost and accuracy. These range from approximate Partial Differential Equation (PDE)-based approaches [18,19] to exact k-dimensional tree (KD-tree) search algorithms. A detailed discussion of the wall-distance calculation method within OpenFOAM can be found in the work of Kareem et al. [20], but Fig. 9 summarizes the different ways the wall

![Fig. 3 NACA0012 airfoil a) boundary conditions and b) computational grid and domain (courtesy of NASA TMR website).](image)

**Table 1** Force coefficients for a variety of CFD codes using the 897 × 256 grid from the NASA TMR website

<table>
<thead>
<tr>
<th>Code</th>
<th>α = 0 deg</th>
<th>α = 10 deg</th>
<th>α = 15 deg</th>
<th>α = 0 deg</th>
<th>α = 10 deg</th>
<th>α = 15 deg</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFL3D</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
<tr>
<td>FUN3D</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
<tr>
<td>NTS</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
<tr>
<td>JOE</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
<tr>
<td>SUMB</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
<tr>
<td>TURNS</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
<tr>
<td>GGNS</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
<tr>
<td>OVERFLOW</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
<tr>
<td>OpenFOAM</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
<td>0.001349</td>
<td>0.001235</td>
<td>0.001212</td>
</tr>
</tbody>
</table>
distance could be computed for a simple example grid. In Fig. 9, each image shows the same grid, where \( C \) is the cell center and the shaded bottom line is the wall. The shaded cell is the cell where we wish to compute the wall distance, and the two images show how the wall distance could be computed for a different cell location. The wall distance used in the SA model is defined as the distance from the cell center to the nearest wall, and errors in the calculation of this variable cause errors in \( \nu \) to grow as \( d^2 \). The default wall-calculation method in OpenFOAM is to compute the nearest wall-face center, i.e., distance from \( C \) to the middle of \( A \) and \( B \). In the first image (Fig. 9a), this method is close to the exact wall distance (shown in the blue arrow) with incorrect wall distances being the distance to the nearest wall node (CA, CB). For the second image (Fig. 9b), the approach of computing the distance to the middle of the nearest face center is also incorrect and deviates from the correct distance (C-B shown by the blue arrow). Within OpenFOAM, this wall-distance calculation method therefore has the potential to compute the wrong wall distance, particularly for highly nonorthogonal or skewed grids.

To test the accuracy of the wall-distance computational, two additional grids (shown in Fig. 10) were generated in the mesh generation tool ANSA using the same surface mesh as the NASA TMR grids. This was done for four refinement levels from \( 281 \times 49 \).
to 2241 × 385 cells. One grid focused on ensuring orthogonality, as shown in Fig. 10c, and the other created deliberately skewed cells (Fig. 10b); the reference grid is shown in Fig. 10a. These were initially run in STAR–CCM+, to which the result was the same for all three grid families within convergence errors, e.g., not visible on a graph. For OpenFOAM, the results from the three grid families are shown in Fig. 8. It can be seen that, for the grid that ensures orthogonality, the results are in very good agreement with those from the other CFD codes. This is because, when the grid is orthogonal, the wall-distance computational is accurate (as shown for the flat-plate test case). When the grid is skewed, the wall distance is now incorrect, and this results in an overprediction of the drag and an underprediction of the lift. The NASA TMR site mesh is somewhat between both, which suggests that the reason OpenFOAM differs from the other code is that the wall-distance calculation is not accurate when the near-wall grid is nonorthogonal or skewed.

This idea is reinforced, given that the shear-stress transport (SST) model, shown in Fig. 11, run on the NASA TMR grids is in close
agreement with the reference CFL3D, FUN3D results. For the SST model, the wall distance is only used in the F1 and F2 blending functions (which control the switch between the $k−\varepsilon$ and $k−\omega$ models [21] within the SST model); thus, any errors in the wall distance have less impact on the overall solution.

Based upon the findings of these validation cases, all the grids used in the study of the high-lift aircraft were designed to have as little nonorthogonality and skewness as possible to minimize the inaccuracy of the wall-distance computation in OpenFOAM. Future work will be to implement and validate the prototype wall-distance method of Kareem et al. [20] into the latest version of OpenFOAM.

### III. NASA High-Lift CRM

In the context of the Third AIAA HL PW [22], the NASA CRM in a new high-lift configuration (HL-CRM) [23] is simulated. This is a wing–body high-lift system that is studied in a nominal landing configuration (slat and flaps deployed at 30 and 37 deg, respectively) without the nacelle, pylon, tail, or support brackets. At the time of this publication, no experimental data are available; however, tests are planned for 2019/2020. The purpose of this test case in this paper is to assess the accuracy of OpenFOAM for a complex full aircraft geometry. To the best knowledge of the authors, no such geometry has been comprehensively investigated and published using...
OpenFOAM. To provide a benchmark to which OpenFOAM can be judged, simulations are undertaken using the same computational grid in the commercial CFD code STAR–CCM+ v11.6. This is done in a spirit of openness rather than a desire to promote one code over another. As stated in the Introduction (Sec. 1), the inability to implement custom turbulence models (among other things) in the majority of commercial codes means that codes like OpenFOAM, SU2, or others are a necessity. STAR–CCM+ was chosen as an example of a mature, widely used code that has been extensively used for compressible high-lift geometries.

### A. Computational Setup

The HL-CRM is computed in both OpenFOAM and STAR–CCM+. The flow conditions are shown in Table 2; however, as the geometry is full scale, the flow parameters are adjusted to achieve the required Reynolds number. The viscosity is computed using Sutherland’s law, and the density is based upon the ideal gas law. Simulations are conducted at 8 and 16 deg angles of attack, and all simulations use the SA turbulence model (SA-no1t2 according to the NASA TMR website). Second-order upwind schemes are used for the momentum and turbulent quantities in both OpenFOAM and STAR–CCM+.

### B. Computational Grid

OpenFOAM has its own mesh generation utility, called SnappyHexMesh, which is a Cartesian-prismatic unstructured generation tool; however the experience of the authors has shown that it is not suitable for low y⁺ grids, and the region between the prismatic and Cartesian is often subject to severe nonorthogonality and large cell size jumps [24]. For this reason, an alternative mesh generator is used, which is capable of generating high-quality grids that represent the kinds of unstructured grids that are typically used by the aerospace industry. ANSA 17.1, which is a preprocessor from BETA CAE Systems, was used to generate all the following grids. The meshing process used for this study was described in detail in the work of Skaperdas and Ashton [25]; thus, a brief description is only given here.

The geometry was cleaned up, and a hemispherical domain of 78,740 in.² [285 mean aerodynamic chord (MAC)] was created for the half-symmetric model. An unstructured triangular surface mesh (Fig. 12) with extruded prisms (and some Hexahedral cells (hexas)), pyramids, and tetras in the volume mesh approach was then produced. A factor of 1.5 increase in total cell count was completed twice to form a family of coarse, medium, and fine grids.

### C. Results

The first objective of this study was to assess the mesh convergence using a series of three successively finer grids: coarse, medium, and fine. Figure 14 shows the mesh convergence of OpenFOAM and STAR–CCM+ for \( C_1 \) and \( C_D \) at 8 and 16 deg angles of attack. Several conclusions can be made: first, the overall correlation between STAR–CCM+ and OpenFOAM is within 1% for the lift coefficient and 3% for the drag coefficient. Second, OpenFOAM and STAR–CCM+ tend toward the same mesh-converged solution, with OpenFOAM showing a greater sensitivity to the mesh. Finally, although there are no experimental data for this geometry, the results agree well with other participants running the SA model from the Third AIAA HLWP [26]. Even though there are only two flow angles simulated, the flow does not show any inboard separation at the fuselage–wing junction or tip stall with the main separation occurring on the outboard flap. Interestingly, the separation actually reduces at the higher angle of attack, suggesting that the design of the wing is not well optimized at the lower angle of attack. Further simulations at pre- and poststall conditions are needed to investigate the stall mechanisms further.

Although the lift and drag coefficients are important quantities, such a global quantity can hide error cancellation. In Figs. 15 and 16, we see a number of planar cuts along the wingspan for the pressure and skin-friction coefficients (the locations of these cuts are shown in Fig. 17a). For the pressure coefficient, the agreement between both codes is very close, with only small differences in the suction peaks. The mesh refinement only made a small change to the pressure coefficient: mainly over the flap, as shown in Fig. 17. For the skin-friction coefficient, the agreement is close in the inboard sections but shows some small differences in the inboard section, which may explain the ≈3% difference in the drag coefficient.

In Figs. 18 and 19, the surface streamlines at \( \alpha = 8 \) deg and \( \alpha = 16 \) deg are shown for OpenFOAM and STAR–CCM+. It can be seen that the flow over the entire wing is very similar between both

### Table 2 Flow properties for HL-CRM aircraft

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOA, deg</td>
<td>8 and 16</td>
</tr>
<tr>
<td>( M )</td>
<td>0.2</td>
</tr>
<tr>
<td>MAC, in.</td>
<td>275.8</td>
</tr>
<tr>
<td>( \rho_{ref}, Pa )</td>
<td>101,353</td>
</tr>
<tr>
<td>( T_{ref}, K )</td>
<td>288</td>
</tr>
<tr>
<td>( Re )</td>
<td>3,260,000</td>
</tr>
</tbody>
</table>

### Table 3 HL-CRM mesh properties

<table>
<thead>
<tr>
<th></th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface mesh face count</td>
<td>1.8M</td>
<td>2.4M</td>
<td>2.8M</td>
</tr>
<tr>
<td>Volume mesh cell count</td>
<td>89M</td>
<td>147M</td>
<td>236M</td>
</tr>
<tr>
<td>Trailing-edge rows of elements</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Number of layers around wing/nacelle/pylon</td>
<td>40</td>
<td>45</td>
<td>65</td>
</tr>
<tr>
<td>Number of layers around body</td>
<td>45</td>
<td>57</td>
<td>84</td>
</tr>
<tr>
<td>Layers growth</td>
<td>1.25</td>
<td>1.16</td>
<td>1.1</td>
</tr>
<tr>
<td>Layers first height, in.</td>
<td>0.000787</td>
<td>0.000787</td>
<td>0.000787</td>
</tr>
</tbody>
</table>

---

Fig. 12 HL-CRM medium surface mesh.
codes at both angles of attack. For engineering purposes, this demonstrates that OpenFOAM can be used for the analysis of this particular configuration of aircraft, given that tools such as STAR–CCM+ are used by many engineering companies for major design projects. This also agrees with recent work by Ashton et al. on helicopter fuselage geometries [27].

So far, however, all quantities have been global or surface quantities; in Fig. 20, velocity profiles are shown according to the positions shown in Fig. 21. In a similar vein to the previous quantities, the agreement between STAR–CCM+ and OpenFOAM is close in the offbody region, differing by an account likely within the experimental error range.
Fig. 16 Skin-friction coefficient at a) \( \eta = 0.151 \), b) 0.329, and c) 0.552 at \( \alpha = 8 \) deg for OpenFOAM and STAR-CCM+.

Fig. 17 Schematic of a) planes analyzed in this section, and b,c) mesh convergence of pressure coefficient at \( \eta = 0.552 \) for \( \alpha = 8 \) deg using OpenFOAM.

Fig. 18 Surface streamlines at \( \alpha = 8 \) deg for the SA using a) OpenFOAM and b) STAR-CCM+.

Fig. 19 Surface streamlines at \( \alpha = 16 \) deg for the SA using a) OpenFOAM and b) STAR-CCM+. 
IV. Japan Aerospace Exploration Agency’s Standard Model High-Lift Aircraft

The results from the HL-CRM model have shown excellent agreement between OpenFOAM and STAR–CCM+ for the same turbulence model and mesh. From an engineering perspective, a maximum of a 3% error between the codes is acceptable and suggests that OpenFOAM can be used to analyze these types of flows. At the time of this paper, there are no experimental data available for the HL-CRM, limiting a full discussion of the accuracy of STAR–CCM+ or OpenFOAM. To address this, a further high-lift aircraft model (the JSM) is now investigated that includes experimental data from 0 to 21 deg AOAs [28]. The JSM is a more realistic aircraft model with slat brackets.

A. Computational Setup

The flow conditions are shown in Table 4, in which the viscosity is computed using Sutherland’s law and the density is based upon the ideal gas law. Simulations are conducted at 4.36, 10.47, 14.54, 18.58, 20.59, and 21.57 deg angles of attack; and all simulations use the SA turbulence model (SA-noft2 according to the NASA TMR website). Second-order upwind schemes are used for the momentum and turbulent quantities in both OpenFOAM and STAR–CCM+.

**Table 4** Flow properties for JSM aircraft

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOA, deg</td>
<td>4.36, 10.47, 14.54, 18.58, 20.59, 21.57</td>
</tr>
<tr>
<td>M</td>
<td>0.172</td>
</tr>
<tr>
<td>MAC, m</td>
<td>0.5292</td>
</tr>
<tr>
<td>$p_{ref}$, Pa</td>
<td>99,684</td>
</tr>
<tr>
<td>$T_{ref}$, K</td>
<td>306.55</td>
</tr>
<tr>
<td>Re</td>
<td>1,930,000</td>
</tr>
</tbody>
</table>
orthogonality and skewness.

Fig. 23 Representations of a) prism layer coverage around the JSM and b) details of the imposed orthogonality near the wall for better orthogonality and skewness.

Table 5 JSM mesh properties

<table>
<thead>
<tr>
<th>Surface mesh face count</th>
<th>1.7M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume mesh cell count</td>
<td>109M</td>
</tr>
<tr>
<td>Trailing-edge rows of elements</td>
<td>8</td>
</tr>
<tr>
<td>Number of layers around wing/nacelle/pylon</td>
<td>49 (57 fuselage)</td>
</tr>
<tr>
<td>Layers growth</td>
<td>1.16</td>
</tr>
<tr>
<td>Layers first height, mm</td>
<td>0.0015</td>
</tr>
</tbody>
</table>

C. Results

B. Computational Grid

The meshing tool ANSA v17.1 was again used to generate the computational grid for the JSM geometry. An unstructured triangular surface mesh (Fig. 22) with extruded prisms (and some hexas) as well as pyramids and tetrads in the volume mesh approach was again used. Only a medium mesh (detailed in Table 5) was used in this study; however, a fine mesh with a factor of 1.5 increase in total cell count was created and made available to the HLPW participants.

The differing numbers of layers around the wing and fuselage were followed in order to ensure a smooth volume ratio transition between the last extruded layer and the pyramid/tetra mesh. Again, orthogonality was ensured in the layers region as shown in Fig. 23.

C. Results

Figure 24 shows the lift and drag curves for OpenFOAM and STAR–CCM+ using the SA model. It can be seen that, first, the agreement between OpenFOAM and STAR–CCM+ is within 3%, which is in line with the results from the HL-CRM geometry. The lift coefficient is in good agreement with the experimental data during the linear range but overpredicts the lift toward the stall region.

The previous work has shown that the standard SA model struggles to capture the anisotropic nature of the turbulence in the wing–root junction [5,30]; thus, it is not surprising that the SA fails to capture this region and has been discussed widely for flows subject to corner separation [31]. The use of a Reynolds stress model [17] or the SA-QCR [29] model was predicted to help this; however, none of the Third HLPW participants using the SA-QCR or Reynolds stress models captured the inboard separation [31]. The use of a Reynolds stress model [17] or the SA-QCR [29] model was predicted to help this; however, none of the Third HLPW participants using the SA-QCR or Reynolds stress models captured the inboard separation [31]. The use of a Reynolds stress model [17] or the SA-QCR [29] model was predicted to help this; however, none of the Third HLPW participants using the SA-QCR or Reynolds stress models captured the inboard separation [31]. The use of a Reynolds stress model [17] or the SA-QCR [29] model was predicted to help this; however, none of the Third HLPW participants using the SA-QCR or Reynolds stress models captured the inboard separation [31]. The use of a Reynolds stress model [17] or the SA-QCR [29] model was predicted to help this; however, none of the Third HLPW participants using the SA-QCR or Reynolds stress models captured the inboard separation [31].

D. Computational Expense

The previous results have shown that OpenFOAM does largely compare well with the accuracy of a major commercial code when using the same mesh, turbulence model, and numerical scheme order. This agrees with the recent work of Ashton et al. for a helicopter fuselage [32]. This is encouraging for users of OpenFOAM, but this is the experimental data. At this low angle of attack, the flow is attached throughout the wing, providing less of a challenge to the SA turbulence model.

Moving to $\alpha = 18.59$ deg, the agreement between CFD and the experimental data worsens, as shown in Figs. 29–31. The agreement between STAR–CCM+ and OpenFOAM is, however, close, with little observable differences, which is reflected in the lift and drag coefficients. Using the SA model, both codes overpredict the outboard flow separation and underestimate the beginning of the root stall. This is reflected in the inboard plane A-A in Fig. 30, in which the pressure coefficient is overpredicted. At the outboard plane H-H (Fig. 31), an opposite trend is observed in which the pressure coefficient is underpredicted because of the flow separation. The under- and overprediction of flow separation at the inboard and outboard sections cancel out to provide a good prediction of the global lift: a clear example of error cancellation. This was seen by nearly all participants of the Third HLPW: even those using more advanced RANS models such as the SA Quadratic Constitutive Relation (QCR) model [29] or full Reynolds stress models [26].

At the stall angle of $\alpha = 21.57$ deg, the trends observed at $\alpha = 18.59$ deg continue, as shown in Figs. 32–34. The discrepancy between the experimental data and CFD is broader at this angle, with the CFD predicting much larger outboard separation while also showing little inboard stall. This is clearer in the pressure coefficient plots, in which the outboard underprediction of the pressure coefficient due to greater stall is present for both CFD codes. At this angle, the disagreement between STAR–CCM+ and OpenFOAM grows, although the large-scale unsteadiness at this angle likely makes the differences between the numerical schemes greater. The previous HL-CRM case also showed that the differences between both CFD codes reduced as the grid was refined. The JSM grid was only 100 Million as compared to the 236 Million fine HL-CRM grid, and thus it is predicted that a finer grid for this JSM geometry would bring the results for both CFD codes even closer.

Previous work has shown that the standard SA model struggles to capture the anisotropic nature of the turbulence in the wing–root junction [5,30]; thus, it is not surprising that the SA fails to capture this region and has been discussed widely for flows subject to corner separation [31]. The use of a Reynolds stress model [17] or the SA–QCR [29] model was predicted to help this; however, none of the Third HLPW participants using the SA–QCR or Reynolds stress models captured the inboard separation [26]. In addition, participants who used some form of a transition model also did not capture the inboard separation. Thus, further research and investigation are needed to understand why CFD is not able to capture the correct stall mechanisms. It may be related to subtle differences in the CFD geometry model and the model used in the experiment, or it may be related to some transient effect that needs globally time-accurate transient methods to capture.
typically only one aspect of engineering CFD analysis. A major consideration is the robustness and computational cost.

1. Robustness and Convergence

In terms of robustness, STAR–CCM+ was found to need less user input to ensure a stable, converged simulation. Given the correct initial ramping of the CFL number (typically, a CFL of 10–30 ramped from 0.1 over 500 iterations), the simulation needed no further user input and typically reached convergence of the residuals and forces by 15,000–40,000 iterations (dependent on angle of attack and grid refinement). Examples of the force and residual convergence are shown in Figs. 35 and 36a for the coarse HL-CRM mesh at a 16 deg AOA. The convergence of the residuals results in at least a four- to five-order drop in their magnitude, which is typical for complex geometries and meshes. For the forces, it means the lift, drag, and moment coefficients are reaching a point where the standard deviation over a given number of iterations is less than $5 \times 10^{-4}$. For the higher angles of attack for the JSM geometry, this was not always possible due to the large-scale unsteadiness.

For OpenFOAM, trial and error were needed to initially find the most stable setup; this resulted in a limit in the largest local CFL, which is typically two to five. This typically results in convergence from 50,000 iterations to sometimes more than 100,000 for the stalled JSM cases. An example of the force and residual convergence is shown in Figs. 35 and 36b for the coarse HL-CRM mesh at a 16 deg AOA. Second-order upwind schemes were used for both the momentum and turbulent qualities; thus, accuracy was prioritized over stability, although, typically, a solution was required to run initially with first order and later switched to second order, which also increased the total number of iterations to achieve convergence. It is not within the scope of this paper to provide an exhaustive analysis of the stability/robustness of OpenFOAM; however, considerable time was spent to find the optimum configuration.

2. Computational Expense

Simulations were conducted on the University of Oxford Advanced Research Computing (ARC) service as well as the U.K. National Supercomputer ARCHER. Figure 37 indicates the strong
Fig. 26  Surface streamlines at $\alpha = 4.36$ deg from a) the experiment and, for the SA, using b) OpenFOAM and c) STAR-CCM+.

Fig. 27  Pressure coefficient at cut A-A at $\alpha = 4.36$ deg on the a) slat, b) wing, and c) flap.

Fig. 28  Pressure coefficient at cut H-H at $\alpha = 4.36$ deg on a) slat and b) wing.

Fig. 29  Surface streamlines at $\alpha = 18.59$ deg from a) experiment and, for the SA, using b) OpenFOAM and c) STAR-CCM+.
Fig. 31 Pressure coefficient at cut H-H at $\alpha/\theta = 18.58$ deg on the a) slat and b) wing.

Fig. 32 Surface streamlines at $\alpha/\theta = 21.57$ deg from a) experiment and, for the SA, using b) OpenFOAM and c) STAR-CCM+.

Fig. 33 Pressure coefficient at cut A-A at $\alpha/\theta = 21.57$ deg on the a) slat, b) wing, and c) flap.
scaling of OpenFOAM for production runs on the ARCHER system. It can be seen that close to linear scaling is achieved down to 50,000 cells per core (5000 cores for fine HL-CRM mesh). Unfortunately, at the time of access to ARCHER, STAR-CCM+ was not available on the system, and thus scaling tests were not possible.

To compare the computational efficiency of STAR-CCM+ and OpenFOAM, both codes were compiled in double precision on the Oxford ARC system, which contains dual Intel E5-2640v3 Haswell CPU nodes that have 16 cores per node and 64 GB of memory. The coarse HL-CRM mesh was used for this comparison at a 16 deg AOA and run for 300 iterations three times to obtain an average time per iteration. These numbers do not reflect the optimum speed of each code, which would require both specific compiler optimization work and a detailed assessment of individual solver convergence tolerance settings (which impact the number of inner iterations for each solver). These numbers are, however, added to help to provide an overall view of the differences in accuracy and computational efficiency between OpenFOAM and STAR-CCM+ on a typical HPC system. At higher angles of attack, more iterations were typically required with a lower CFL number; conversely, for lower angles of attack, higher CFL numbers could be used, resulting in fewer iterations, which was true of both OpenFOAM and STAR-CCM+.

It is clear from Table 6 that, although OpenFOAM is approximately 30% faster per iteration, the biggest difference is the number of iterations required to reach convergence in the forces and residuals. It is no surprise that the coupled implicit density-based scheme within STAR-CCM+ is able to achieve convergence in fewer iterations as compared to the implicit segregated pressure-based...
approach within OpenFOAM. This provides important evidence needed to encourage the OpenFOAM community to develop a robust coupled implicit density-based scheme for the official version of OpenFOAM so that these large-scale complex aerodynamics applications can be undertaken with greater computational efficiency.

V. Conclusions

This paper presents an assessment of a popular open-source code, OpenFOAM, for compressible high-lift aircraft flows. A range of cases is investigated including a zero-pressure gradient flat plate, a NACA0012 airfoil at varying angles of attack, the DSMA661 airfoil, the NASA high-lift common-research model, and a JSM high-lift aircraft model. For all cases, the same mesh and turbulence models are also simulated in the commercial code STAR–CCM+ to provide a benchmark solution from a widely used CFD code. For the HL–CRM and JSM, the lift and drag coefficients from OpenFOAM are within 3% of the STAR–CCM+ solution. For the JSM case with experimental data, the CFD simulations agree well with the experimental at pre-stall AOA; but, for post-stall, although the global loads are well predicted, the local separation is not well predicted and the global force agreement is due to error cancellation. This result is, however, in line with expectations for the SA model with no corrections for curvature or corner flows. It is noted that, even with great care to generate high-quality unstructured hybrid prismatic-tetrahedral meshes using the commercial tool ANSA, the robustness of the standard version of OpenFOAM 4.1 was inferior to STAR–CCM+ and typically required twice as long to reach a satisfactory force convergence, largely due to the inability to run at higher Courant numbers as STAR–CCM+. It is hoped that this work can be inspire further development of compressible methods within OpenFOAM to improve the computational efficiency and robustness.

Acknowledgments

The authors gratefully acknowledge computational support from the Engineering and Physical Sciences Research Council (EPSRC) for the United Kingdom’s National High-Performance Computing Facility ARCHER. The authors would also like to acknowledge the use of the University of Oxford Advanced Research Computing facility in carrying out this work.

References


