8-11-2016

Testing the Accuracy of Commercial Computational Fluid Dynamics Codes: Vortex Transport by Uniform Flow and Transonic Ringleb Flow Problems Using ANSYS Fluent

Carl Leake
Embry-Riddle Aeronautical University, leakec@my.erau.edu

Follow this and additional works at: https://commons.erau.edu/pr-honors-coe
Part of the Aerodynamics and Fluid Mechanics Commons, and the Computational Engineering Commons

Scholarly Commons Citation

This Article is brought to you for free and open access by the Honors Program at Scholarly Commons. It has been accepted for inclusion in College of Engineering by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.
Testing the Accuracy of Commercial Computational Fluid Dynamics Codes: Vortex Transport by Uniform Flow and Transonic Ringleb Flow Problems Using ANSYS Fluent

Carl Leake
NASA Glenn Research Center
Major: Aerospace Engineer
2016 Summer Session
Date: 11 August 2016

This final report has been reviewed and approved by Mentor to ensure information is accurate and does not contain sensitive data.

Mentor Name & Org Code/Date
Testing the Accuracy of Commercial Computational Fluid Dynamics Codes: Vortex Transport by Uniform Flow and Transonic Ringleb Flow Problems Using ANSYS Fluent

Carl Leake\textsuperscript{1} \\
\textit{Embry-Riddle Aeronautical University, Prescott, Arizona, 86301}

This study juxtaposes the accuracy of a commercial computational fluid dynamics (CFD) software, ANSYS Fluent, against two CFD research codes, EZ4D and Flux Reconstruction. The study uses three problems to make the comparison, two vortex transport problems and one Ringleb flow problem. Fluent was used to simulate all problems, however, data sets from each CFD research code were not available for all cases. The first vortex problem showed that quadrilateral meshes perform better in Fluent than triangular meshes when the flow is aligned normal to quadrilateral cells. In addition, it revealed that Fluent has more dissipation error but less phase error than EZ4D. Furthermore, it revealed that Fluent has less velocity error than EZ4D, but this error was measured using a metric that is biased towards phase error. Fluent and EZ4D had a similar order of accuracy for this problem. The second vortex problem showed that Fluent has more velocity error as well as density error and has a lower order of accuracy compared to Flux Reconstruction. The Ringleb problem showed that EZ4D was able to establish fully developed Ringleb flow on grids coarser than Fluent; however, once Fluent established fully developed Ringleb flow, it had entropy error approximately an order of magnitude less than EZ4D. In addition, the problem showed that the entropy error in the Flux Reconstruction simulation was approximately 2.5 to 3.5 orders of magnitude less than either EZ4D or Fluent.

Nomenclature

$a$ = Speed of sound  
$C_p$ = Specific heat at constant pressure  
$h$ = Characteristic length scale  
$k$ = Streamline parameter  
$L$ = Mesh dimension  
$M$ = Mach number  
$m$ = Summation index  
$N$ = Number of cells  
$P$ = Fluid pressure  
$q$ = Velocity magnitude  
$R$ = Vortical characteristic radius  
$R_{gas}$ = Gas constant for air  
$s$ = Entropy  
$T$ = Fluid temperature  
$u$ and $v$ or $v_x$ and $v_y$ = Cartesian components of velocity  
$x$ and $y$ = Cartesian coordinates  
$\beta$ = Vortical characteristic strength  
$\gamma$ = Ratio of specific heats for air  
$\rho$ = Fluid density

\textsuperscript{1} LERCIP Intern, Inlets and Nozzles Branch, 21000 Brookpark Road, MS 5-12
\[ \infty = \text{Freestream} \]
\[ 0 = \text{Initial condition} \]

\section*{I. Introduction}

Commercial computational fluid dynamics (CFD) software are used in industry and academia to obtain numerical solutions for various fluid problems. Commercial CFD software offers an alternative to performing hand calculations, creating research codes (CFD solvers written by individuals or organizations), or creating prototypes and physically testing them; however, each of these methods has advantages and disadvantages when compared to commercial CFD software. This study explores how the accuracy of commercial CFD software compares to that of CFD research codes.

ANSYS Fluent (abbreviated as Fluent throughout the rest of the document) was the commercial CFD solver chosen for this study because it is available to interns at NASA and popular throughout industry and academia. Information from the Fluent website shows its software’s popularity through its versatility: from designing and analyzing race cars with Red Bull \cite{1} to designing and analyzing water turbines with Andritz Hydro \cite{2}.

Three problems were chosen for this study: the vortex problem \cite{3} from the 1st, 2nd, and 3rd International Workshops on High-Order CFD Methods (abbreviated the Workshop throughout the rest of the document) \cite{4}, the Shu vortex from a NASA technical report \cite{5} \cite{6}, and the transonic Ringleb flow problem \cite{7} from the Workshop. All three problems were chosen because they assessed the accuracy of the solution on a variety of grid sizes, discretization orders, and, in the case of the two vortex problems, grid types. In addition, data sets for these problems from research codes were easily accessible for comparison with Fluent results.

\section*{II. The Workshop Vortex}

\subsection*{A. Overview and Relevant Equations}

The Workshop vortex problem tests “a high-order method’s capability to preserve vorticity in an unsteady inviscid flow,” which is valuable in the field of CFD because the “accurate transport of vortices … is very important for Large-Eddy and Detached-Eddy simulations” \cite{3}. The vortex problem starts from the analytical solution and runs for a specified amount of time. Afterwards, the error in the simulation is calculated by comparing the velocity in each cell to its initial solution counterpart. The simulations are run on different meshes that vary in cell shape and number of cells; this allows one to compare how the shape and number of cells affect the accuracy of the simulation. (The word mesh will be used interchangeably with the word grid throughout the rest of this document).

The meshes used for this problem were two dimensional squares that were divided into multiple cells; examples are shown in Fig. 1.
Figure 1. Example Workshop vortex meshes.

Figure 1 shows all four mesh shapes on the coarsest mesh level: quadrilateral, triangular, quadrilateral perturbed, and triangular perturbed. There are 16 total meshes used in the Workshop vortex problem. The fineness of the meshes ranges from 32x32 grids to 256x256 grids and all grid fineness levels have four meshes that correspond to the four mesh shapes. Each mesh uses periodic boundaries on all sides.

In order to create the vortex within the mesh, the individual cells were all initialized with a pressure, temperature, density, and velocity vector that were calculated using Eqs. (1-12) [3]. The vortex was then simulated for one or 50 time-periods (τ), calculated in Eq. (13) [3].

\[
\begin{align*}
\rho_\infty &= P_\infty / (\text{R}_{\text{gas}} \ast T_\infty) \\
U_\infty &= M_\infty \sqrt{\gamma R_{\text{gas}} T_\infty} \\
r &= \sqrt{(x - X_c)^2 + (y - Y_c)^2} / R \\
C_p &= \frac{\gamma}{\gamma - 1} \text{R}_{\text{gas}} \\
\delta u &= -(U_\infty \ast \beta) \ast \frac{y - Y_c}{R} \ast \exp(-r^2/2) \\
\delta v &= (U_\infty \ast \beta) \ast \frac{x - X_c}{R} \ast \exp(-r^2/2) \\
\delta T &= 0.5 \ast (U_\infty \ast \beta)^2 \ast \exp(-r^2) / C_p \\
U_0 &= U_\infty + \delta u
\end{align*}
\]
\[ v_0 = \delta v \]
\[ T_0 = T_\infty - \delta T \]
\[ \rho_0 = \rho_\infty \cdot (T_0/T_\infty)^{\gamma/\gamma-1} \]
\[ P_0 = \rho_0 \cdot R_{gas} \cdot T_0 \]
\[ \tau = L_x / U_\infty \]

where \( P_\infty = 1e5 \ Pa, T_\infty = 300 \ K, M_\infty = 0.5, \gamma = 1.4, R_{gas} = 287.15 \frac{J}{kg-K}, L_x = 0.1 \ m, X_c = 0.05 \ m, Y_c = 0.05 \ m, R = 0.005 \ m, \) and \( \beta = 0.2 \) are constants of the problem [3].

**B. Error Calculation**

After each simulation ran for the specified number of time-periods, the velocity components of each cell were used to calculate the error in the simulation. The Guidelines page from the Workshop [8] was used to create the equations for the L2-Norm, abbreviated \( L_2(variable) \), and L2-Error, abbreviated \( Error_{L2(variable)} \), of the \( u \) velocity, the \( v \) velocity, and the velocity vector, \( \vec{V} = u \hat{i} + v \hat{j} \). The equations used to calculate the L2-Norm and L2-Error for these quantities are shown in Eqs. (14-19).

\[
L_2(u) = \sqrt{\frac{1}{N} \sum_{m=1}^{N} u_m^2} 
\]

\[
L_2(v) = \sqrt{\frac{1}{N} \sum_{m=1}^{N} v_m^2} 
\]

\[
L_2(\vec{V}) = \sqrt{\frac{1}{N} \sum_{m=1}^{N} (u_m^2 + v_m^2)}
\]

\[
Error_{L2(u)} = \left[ \frac{\sum_{m=1}^{N} (u_m - u_0)^2 |V_m|}{\sum_{m=1}^{N} |V_m|} \right]^{1/2} 
\]

\[
Error_{L2(v)} = \left[ \frac{\sum_{m=1}^{N} (v_m - v_0)^2 |V_m|}{\sum_{m=1}^{N} |V_m|} \right]^{1/2} 
\]

\[
Error_{L2(\vec{V})} = \left[ \frac{\sum_{m=1}^{N} ((u_m - u_0)^2 + (v_m - v_0)^2) |V_m|}{\sum_{m=1}^{N} |V_m|} \right]^{1/2} 
\]

where \( u_0 \) is the \( u \) velocity in cell \( m \) at the start of the simulation, \( v_0 \) is the \( v \) velocity in cell \( m \) at the start of the simulation, \( u_m \) is the \( u \) velocity in cell \( m \) at the current time-step, \( v_m \) is the \( v \) velocity in cell \( m \) at the current time-step, and \( V_m \) is the volume of cell \( m \). It is important to note that the variables \( u_m, v_m, u_0, \) and \( v_0 \) were all nondimensionalized using the freestream velocity, \( U_\infty \), before any L2-Norm or L2-Error calculations were performed.
C. Results

The results section is broken down into two subsections: Fluent results only and Fluent results compared to EZ4D results. For Fluent, the accuracy dependence on the number of cells and the type of cells was explored, whereas only the number of cells was explored in EZ4D because EZ4D can only run simulations on triangular meshes.

1. Fluent Results

The results in this section show the error in the vortex simulation after one time-period qualitatively as well as quantitatively. The u velocity contour plots in Fig. 2 show that a grid with more cells does a better job of preserving vorticity and maintaining the major flow features of the simulation than one with less cells. The top row of the contour plots shows a 32x32 grid, and the bottom row shows a 256x256 grid: both grids used quadrilateral meshes.

(a) 32x32 mesh at initialization.  
(b) 32x32 mesh after one time-period.  
(c) 256x256 mesh at initialization.  
(d) 256x256 mesh after one time-period.

Figure 2. U velocity contours on quadrilateral meshes.

Figure 2 shows that the vortex strength in the coarser grid has diminished more after one time-period than in the finer grid. This phenomena was observed for all four levels of grid fineness, and was observed across all four mesh shapes; however, the fineness of the grid is not the sole determining factor in the simulation error.
The shape of the mesh plays a factor in the amount of simulation error. Figure 3 shows the $u$ velocity contour plots after one time-period on the second finest mesh for each of the four cell shapes.

![Figure 3. U velocity contours on the 128x128 meshes.](image)

Figure 3 shows the accuracy of the simulation as a function of cell shape. One can determine the accuracy of a particular cell shape in comparison to the others by looking at the size of the darkest blue and darkest red portions of the vortex. The closer the size of the darkest portions of the vortex are to their size at initialization, the more accurate the solution is. According to the results shown in Fig. 3, the order of the cell shapes from most accurate to least accurate is quadrilateral, quadrilateral perturbed, triangular, and triangular perturbed.
The remaining figures in this section use what is known as the characteristic length scale, defined in Eq. (20), rather than reporting the number of grid cells directly.

\[ h = \frac{1}{\sqrt{N}} \]  

(20)

Algebraically, as the number of cells increases, the characteristic length scale decreases.

As the number of cells increases and the length scale decreases, it is expected that the error in the simulation will decrease. The results from Fluent support this expectation when referring to grids of the same cell shape; however, this was not necessarily true when referring to grids with different cell shapes. In addition, it was expected that the triangular meshes would perform better than the quadrilateral meshes because they double the number of cells for a given grid fineness level. It was also expected that the perturbed meshes would perform slightly worse than their unperturbed counterparts.

Figure 4 shows a graph of the L2 velocity vector error versus length scale for all four types of meshes on all four fineness levels after one time-period of simulation. Additionally, Fig. 4 shows the second-order accurate line.

![Fluent L2 velocity error graph](image)

**Figure 4. Fluent L2 velocity error after one time-period.**

Figure 4 shows that the accuracy of the mesh shapes listed from most accurate to least accurate is quadrilateral, quadrilateral perturbed, triangular, and triangular perturbed. The accuracy of cell shapes matches the order found when looking at the \( u \) velocity contour plots in Fig. 3. Furthermore, the last two data points of the quadrilateral and quadrilateral perturbed meshes are second-order accurate. The order of accuracy of data points is defined by the slope of two consecutive data points, and increases as steepness of the slope increases. If the data points have a slope steeper than the second order accurate line, then the data points are considered more than second order accurate, otherwise, the data points are not considered second order accurate.

Looking individually at each of the four mesh shapes shown in Fig. 4 reveals that as the number of cells increases within a grid, the error in the simulation decreases; however when looking at each of the four mesh shapes comprehensively, one can conclude that the number of cells in the grid is not the sole determining factor in simulation error. For example, the left-most blue dot has twice the number of cells as the left-most red dot, but the left-most blue dot has more error than the left-most red dot.

The hypothesis that the triangular meshes would perform better than the quadrilateral meshes was proved false. Although the triangular meshes do have more cells, they have “inherently larger truncation error than quad [meshes] which are aligned with the flow direction” [9]. The hypothesis that the perturbed meshes would perform worse than their unperturbed counterparts was true overall as the average error of the perturbed meshes was greater than the average error of the unperturbed meshes, but this was not true for every mesh. The meshes where the hypothesis was false were coarser than the meshes where it was true. Reviewing the \( u \) velocity contour plots reveals that the coarser meshes do not have nearly the same vorticity strength as the finer meshes and may be subject to biases of the error metric (discussed in detail in the next section); thus, the data collected from these meshes are less meaningful than the data collected from the finer meshes.
2. Comparison Results

The results in this section compare the error in the vortex simulation between Fluent and EZ4D after 50 time-periods qualitatively and quantitatively.

Figure 5 shows $u$ velocity contour plots from Fluent and EZ4D [10] on a 256x256 mesh. It is important to note that the Fluent contour plots are on quadrilateral meshes and the EZ4D contour plots are on triangular meshes. Fluent contour plots on triangular meshes were not included in Fig. 5 because all simulations on the Fluent triangular meshes, although not a complete uniform freestream, appeared washed out and unclear; it was not possible to complete a Fluent simulation on a triangular grid finer than 128x128 due to time and computational constraints. EZ4D is only capable of running simulations on triangular meshes, so displaying a quadrilateral mesh from an EZ4D simulation was not possible. For clarity, a quadrilateral mesh was chosen from the Fluent simulations and the most similar triangular mesh chosen from the EZ4D simulations.

The left column of Fig. 5 shows the Fluent simulation at initialization and the EZ4D simulation after one time-period because a contour plot of the EZ4D simulation at initialization was not obtainable. Although this leads to an indirect comparison between the solution initializations, this figure is shown to highlight features of the vortex which do not change significantly between initialization and one time-period.

The contours after 50 time-periods shown in Fig. 5 reveal that Fluent has more dissipation error, error due to diminishing vortex strength, than EZ4D; this is shown by EZ4D’s final vortex having darker colors closer to its original vortex than Fluent’s final vortex. In addition, it shows that Fluent has less phase error, error due to the
movement of the vortex away from its original position, after 50 time-periods than EZ4D because Fluent’s vortex has moved less from its original position than EZ4D’s: the vortices translate a similar amount to the right and down, but the Fluent vortex is rotated less than the EZ4D vortex.

It is expected that the error in the vortex simulation will decrease as the number of cells in the mesh increases. This expectation was true for both Fluent and EZ4D once the length scale of the meshes was below approximately 0.95. On meshes with a length scale greater than 0.95, there was significant dissipation error and the flow field became a uniform freestream. These cases are not included in the graphs below because they lack the major flow features, and therefore are not good representations, of vortex transport by uniform flow. Furthermore, the uniform flow cases would convolute the error data if included because the method used to calculate the solution error is biased towards phase error. Consequently, until the meshes that retained the vortex reached a fineness level such that dissipation error was the dominating source of error, the finer mesh’s velocity error was greater than the freestream cases.

Figure 6 shows the $u$ velocity error versus length scale on the left and the $v$ velocity error versus length scale on the right.

![Graph showing L_2 error in velocity versus length scale for Fluent and EZ4D simulations.](image)

Figure 6. Fluent and EZ4D, simulation error versus length scale.

The graph on the left in Fig. 6 shows that the error in the $u$ velocity is the smallest on the Fluent triangular mesh simulations, largest on EZ4D simulations [10], and Fluent quadrilateral mesh simulations lie in the middle; however, the EZ4D data points have the steepest slope, indicating the highest order of accuracy, the Fluent triangular mesh simulations have the shallowest slope, indicating the lowest order of accuracy, and the Fluent quadrilateral meshes lie in the middle. The graph on the right in Fig. 6 shows that the error in the $v$ velocity follows approximately the same trend line for all simulations.

Although the Fluent simulations outperformed EZ4D in the Workshop vortex problem for the data points shown in Fig. 6, the graphs do not reveal the entire picture. The contour plots in Fig. 5 show that EZ4D actually does a better job preserving vortex strength in uniform flow, but has more phase error than Fluent. As was aforementioned in the paragraphs above, the error metric used to judge simulation error is biased towards phase error. In summation, the Fluent simulations outperformed EZ4D in reducing phase error, but underperformed EZ4D in dissipation error. The Fluent triangular mesh follows the same line of reasoning: they performed the best in terms of the metric used to determine simulation accuracy, but did the worst in preserving vortex strength.
III. Shu Vortex

A. Overview and Relevant Equations

The Shu vortex tests CFD software in the same way as the Workshop vortex problem. The Shu vortex was run on all fineness levels of the unperturbed meshes shown in Workshop vortex problem, but was only run for one time-period.

The initialization process for the Shu vortex is similar to the vortex problem from the Workshop; however, it uses a different set of nondimensionalized equations. Equations (21-30) were taken from a NASA technical report [5] and dimensionalized using a freestream temperature and pressure.

\[
f(x, y) = -\frac{1}{2R^2} [(x - X_c)^2 + (y - Y_c)^2]
\]

\[
\Omega = \beta e^{f(xy)}
\]

\[
\delta v_x = -\frac{y}{R} \Omega
\]

\[
\delta v_y = \frac{x}{R} \Omega
\]

\[
\delta T = -\frac{y - 1}{2} \Omega^2
\]

\[
v_{x,0} = M_\infty \cos \alpha + \delta v_x
\]

\[
v_{y,0} = M_\infty \sin \alpha + \delta v_y
\]

\[
p_0 = \frac{1}{\gamma} (1 + \delta T)^{\gamma-1}
\]

\[
\rho_0 = (1 + \delta T)^{1\gamma-1}
\]

\[
T_0 = p_0/(R^*_{gas} \rho_0)
\]

where \(R^*_{gas}\) is equal to one in nondimensional space. After calculating the nondimensionalized values for each cell, \(p_0, \rho_0, T_0, v_{x,0}\), and \(v_{y,0}\), the values were dimensionalized using a freestream temperature and pressure. Freestream conditions were chosen such that (1) \(\gamma\) and \(R^*_{gas}\) maintained their standard values throughout the flow field, (2) they satisfied the isentropic flow equations, and (3) they scaled the mesh size so the unperturbed meshes from the Workshop vortex problem could be reused.

B. Error Calculation

Equations (14) and (17) from the Workshop vortex problem were used to calculate the L2 norm and L2 error of the u velocity, respectively. Equations (31-32), shown below, were used to calculate the L2 norm and L2 error of the density.

\[
L_2(\rho) = \sqrt{\frac{1}{N} \sum_{m=1}^{N} \rho_m^2}
\]

\[
Error_{L_2(\rho)} = \left[ \frac{\sum_{m=1}^{N} (\rho_m - \rho_0)^2 |V_m|}{\sum_{m=1}^{N} |V_m|} \right]^{1/2}
\]
C. Results

The results in this section compare Fluent to Flux Reconstruction. The data is compared quantitatively only. Contour plots are not used in this section to make a qualitative comparison because contour plots from Flux Reconstruction were not available for one time-period simulations.

Figure 7 shows a graph of \( u \) velocity error versus length scale on the left and a graph of density error versus length scale on the right.

The graph on the left in Fig. 7 shows that Fluent has more error and a lower order of accuracy in terms of \( u \) velocity than Flux Reconstruction. In addition, the graph shows that Fluent triangular meshes have less error than quadrilateral meshes. The triangular meshes may have done better in this simulation than the quadrilateral meshes because the flow was not aligned normal to one of the sides of the quadrilateral cells, so it did not reduce the truncation error in these cells like in the Workshop vortex problem [9]. However, the triangular meshes have a shallower slope than the quadrilateral meshes do, and thus, a lower order of accuracy. None of the Fluent data has a slope steep enough to be considered second order accurate.

The graph on the right in Fig. 7 shows that Fluent has more error and a lower order of accuracy in terms of density than Flux Reconstruction. In addition, the graph shows that for density, the Fluent quadrilateral meshes performed about the same as the Fluent triangular meshes. Furthermore, both Fluent data sets have approximately the same slopes between consecutive points, and therefore, have approximately the same order of accuracy.

The Fluent data shows asymptotic behavior as the length scale decreases, which differs from what was observed in the Workshop vortex problem. One possible explanation is written in a NASA Technical Report [5]. The characteristic vortex radius for the Shu vortex is larger in proportion to the size of the mesh than the Workshop vortex problem which creates two artificial shear layers. “The errors associated with these shear layers are orders of magnitude larger than the error from the numerical method,” [5] and may be responsible for the asymptotic behavior observed in Fig. 7.
IV. Transonic Ringleb Flow

A. Overview and Relevant Equations

The transonic Ringleb flow problem tests “the spatial accuracy of high-order methods” [7] which is important for characterizing the accuracy of a solver. The Ringleb problem was initialized using the analytical solution, and was run in Fluent until a steady-state solution was reached. The convergence of the problem was judged by the L2 norm of the density residual, according to instructions in the problem statement [7]. The error in the simulation was calculated by comparing the entropy in each cell to its initialization counterpart.

The mesh used for the Ringleb problem is a two-dimensional curved geometry that is divided up into multiple cells; examples are shown in Fig. 8.

There were five total Ringleb meshes that ranged in coarseness from grid level zero to grid level four. All of the Ringleb meshes used quadrilateral shaped cells. Air flows in through a pressure inlet at the top and flows out through a pressure outlet at the bottom. The black lines on the left and the right of the meshes represent wall boundaries.

The nondimensional initialization quantities for the Ringleb problem were calculated using Eqs. (33-40). First, a numerical solver solved for \( q \) and \( k \) given a grid point \((x,y)\) and Eqs. (33-37). Then Eqs. (33-34) and (38-40) were calculated using \( q \).

\[
a = \sqrt{1 - \frac{\gamma - 1}{2} q^2}
\]

(33)

\[
\rho_0 = a \left( \frac{2}{\gamma - 1} \right)
\]

(34)

\[
J = \frac{1}{a} + \frac{1}{3a^2} + \frac{1}{5a^5} - \frac{1}{2} \ln \left( \frac{1 + a}{1 - a} \right)
\]

(35)

\[
x(q,k) = \frac{1}{2\rho_0} \left( \frac{2}{k^2 - q^2} \right) - \frac{J}{2}
\]

(36)

\[
y(q,k) = \pm \frac{1}{k\rho_0 q} \sqrt{1 - \left( \frac{q}{k} \right)^2}
\]

(37)
\[ M = \frac{q}{a} \]  
\[ P_0 = \frac{1}{\gamma a^{(2\gamma - 1)/\gamma}} \]  
\[ T_0 = \frac{P_0}{(R_{\text{gas}} \rho_o)} \]

It can be shown from Eq. (33) and the isentropic flow equations that total temperature and total pressure are constant throughout the nondimensional flow field. Thus after calculating the nondimensional values, \( P_0, \rho_0, T_0, \) and \( a \), for each cell, they were dimensionalized using a total temperature and total pressure; total conditions were chosen such that \( \gamma \) and \( R_{\text{gas}} \) maintained their standard values throughout the flow field and the isentropic flow equations were satisfied.

Once \( a \) was dimensionalized, the velocity vector for each cell was computed using Eqs. (41-44).

\[ \vec{V} = M \, a \, \vec{\hat{V}} \]  
\[ \vec{\hat{V}} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \]  
\[ \theta = \arctan\left(\frac{dy}{dx}\right) \]  
\[ \frac{dy}{dx} = \frac{\partial y}{\partial x} / \frac{\partial q}{\partial x} \]

B. Error Calculation

Equation (45), shown below, was used to calculate the L2 entropy error.

\[ \text{Error}_{L_2(s)} = \left[ \frac{\sum_{m=1}^{N}(s_m - s_0)^2|V_m|}{\sum_{m=1}^{N}|V_m|} \right]^{1/2} \]

C. Results

The results in this section compare three CFD software: Fluent, EZ4D, and Flux Reconstruction. The results between Fluent and EZ4D are compared qualitatively and the results of all three CFD software are compared quantitatively.
Figure 9 shows contour plots of Mach number for Fluent and EZ4D [10]: plots include the Ringleb flow at initialization (a), Fluent solutions (b) and (c), and EZ4D solutions (d) and (e). Contour plot (b) is labeled pseudo-steady because a steady-state solution was unobtainable in Fluent for that particular mesh.

(a) Ringleb initialization.

(b) Fluent, grid two, pseudo-steady state.

(c) Fluent, grid four, steady state.

(d) EZ4D, grid two, steady state.

(e) EZ4D, grid four, steady state.

Figure 9. Ringleb contour plots.
Looking at the bottom half of the inner wall of solutions (c) and (e) in Fig. 9 shows that Fluent has less simulation error than EZ4D on grid level four; however, it is unclear from Fig. 9 which performs better on grid level two. Figure 9 also reveals that both Fluent and EZ4D have difficulty developing the major features of the flow field on grids as coarse as grid level two, but are able to develop these features on grid level four. Throughout the rest of this section, the simulations that contain all of the major flow features will be referred to as fully developed Ringleb flow.

Figure 10 compares the entropy error versus length scale between Fluent, EZ4D, and Flux Reconstruction. Fluent only has four data points in this graph. The data point corresponding to grid level three could not be obtained, because neither a steady nor pseudo-steady converged solution could be found. The data points shown for EZ4D correspond to grid levels three through six; grids five and six were created for EZ4D to obtain more data points that contained fully developed Ringleb flow. The length scales for EZ4D and Fluent do not align in Fig. 10 for two reasons: (1) the grids used in Fluent took the high-order nodes from the original grids and converted them into low order cells because Fluent did not recognize high-order nodes and (2) EZ4D had to diagonalizable all of the quadrilateral cells because EZ4D only used triangular grids.

![Figure 10. Fluent, EZ4D, and Flux Reconstruction, entropy error versus length scale.](image)

The three right-most data points in the Fluent data set from Fig. 10 correspond to Fluent simulations that did not contain fully developed Ringleb flow. The left-most data point in the Fluent data does have fully developed Ringleb flow; this explains why the Fluent data set has a shallow slope for the three right-most data points, and then suddenly drops more than an order of magnitude in error. EZ4D and Flux Reconstruction have fully developed Ringleb flow for all of the data points shown.

EZ4D was able to establish fully developed Ringleb flow on coarser grids than Fluent and had less simulation error than Fluent when Fluent did not have fully developed Ringleb flow; however, Fluent had less error than EZ4D once Fluent was able to establish fully developed Ringleb flow. It is impossible to make an order of accuracy comparison between EZ4D and Fluent because Fluent only has one data point with fully developed Ringleb flow. In order to make an order of accuracy comparison, one would need to complete more Fluent simulations that had grids fine enough to have fully developed Ringleb flow; however, due to time constraints, it was not possible to run these simulations.

Flux Reconstruction only had one data point available for the Ringleb flow, so it is impossible to establish an order of accuracy for Flux Reconstruction; however, this data point shows that Flux Reconstruction has a simulation error approximately 2.5 to 3.5 orders of magnitude lower than either Fluent or EZ4D.

V. Conclusion

This study provided insight as to how the accuracy of Fluent compares to two CFD research codes, EZ4D and Flux Reconstruction. In the Workshop vortex problem, Fluent did better quantitatively than EZ4D due to a biased error metric, but did worse in terms of dissipation error than EZ4D when making a qualitative comparison using the \( u \) velocity contour plots. In the Shu vortex problem, Fluent did worse than Flux Reconstruction in terms of error value and order of magnitude. In the Ringleb problem, Fluent did better than EZ4D once it was able to establish fully developed Ringleb flow, but EZ4D was able to establish Ringleb flow on grids coarser than Fluent. In
addition, Fluent had more entropy error than Flux Reconstruction. Challenges and observations made during this study generated recommendations for future studies.

The error metric used in the Workshop vortex problem is biased towards phase error, so much so that a freestream solution may be better in some cases than solutions where the vortex is still present. A study that looked at the Workshop vortex problem with an updated error metric that weights dissipation error and phase error more equally may produce better results that more accurately depict which software is best at preserving vorticity. The Shu vortex problem showed that the proportion of the characteristic vortex radius to the mesh size may be too high, because an artificial shear layer develops that convolutes the error calculation. A NASA Technical Paper recommends halving this proportion to help eliminate the effects of the artificial shear layer. Performing a study using this recommendation may more clearly show the order of accuracy of Fluent for the Shu vortex. Data sets from more CFD research codes would help further answer the question on how commercial CFD software compare to other codes. Furthermore, the data shown, specifically the Ringleb problem, would be more impactful and better supported if more Fluent trials could be completed on finer meshes and appended to the current data.

Appendix

The tables included in this section show the Fluent solver settings used for each of the three problems. The name of the solver parameter in Fluent is listed on the left, and the setting of that solver parameter is listed on the right. Choosing the correct solver parameters for each problem was done using a combination of the Fluent User’s Guide [11], Fluent Theory Guide [12], and a Solver Settings Fluent training [13].

<table>
<thead>
<tr>
<th>Fluent Solver Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Solver Parameter</strong></td>
</tr>
<tr>
<td>All boundaries</td>
</tr>
<tr>
<td>Solver type</td>
</tr>
<tr>
<td>Air density</td>
</tr>
<tr>
<td>Solver formulation</td>
</tr>
<tr>
<td>Solver flux type</td>
</tr>
<tr>
<td>Solver spatial discretization gradient</td>
</tr>
<tr>
<td>Solver spatial discretization flow</td>
</tr>
<tr>
<td>Solver transient formulation</td>
</tr>
<tr>
<td>CFL number</td>
</tr>
<tr>
<td>Residual convergence criterion</td>
</tr>
<tr>
<td>Solution initialization</td>
</tr>
<tr>
<td>Time step size</td>
</tr>
<tr>
<td>Number of time steps</td>
</tr>
<tr>
<td>Maximum iterations per time step</td>
</tr>
</tbody>
</table>
Table 2. Ringleb steady-state solver settings.

<table>
<thead>
<tr>
<th>Fluent Solver Settings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Solver Parameter</strong></td>
<td><strong>Setting</strong></td>
</tr>
<tr>
<td>Inflow boundary</td>
<td>Pressure inlet</td>
</tr>
<tr>
<td>Outflow boundary</td>
<td>Pressure outlet</td>
</tr>
<tr>
<td>Inner and outer wall boundaries</td>
<td>Wall boundary</td>
</tr>
<tr>
<td>Solver type</td>
<td>Density based steady solver</td>
</tr>
<tr>
<td>Air density</td>
<td>Ideal gas law</td>
</tr>
<tr>
<td>Solver formulation</td>
<td>Explicit</td>
</tr>
<tr>
<td>Solver flux type</td>
<td>Roe-FDS</td>
</tr>
<tr>
<td>Solver spatial discretization gradient</td>
<td>Green-Gauss node based</td>
</tr>
<tr>
<td>Solver spatial discretization flow</td>
<td>Second order upwind</td>
</tr>
<tr>
<td>CFL number</td>
<td>Initial guess 1.8 but was changed if necessary based on initial solution convergence</td>
</tr>
<tr>
<td>Residual convergence criterion</td>
<td>1.00E-08</td>
</tr>
<tr>
<td>Solution initialization</td>
<td>UDF function based on problem statement</td>
</tr>
<tr>
<td>Iterations</td>
<td>2.00E06</td>
</tr>
</tbody>
</table>
### Table 3. Ringleb transient solver settings.

<table>
<thead>
<tr>
<th>Solver Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflow boundary</td>
<td>Pressure inlet</td>
</tr>
<tr>
<td>Outflow boundary</td>
<td>Pressure outlet</td>
</tr>
<tr>
<td>Inner and outer wall boundaries</td>
<td>Wall boundary</td>
</tr>
<tr>
<td>Solver type</td>
<td>Density based transient solver</td>
</tr>
<tr>
<td>Air density</td>
<td>Ideal gas law</td>
</tr>
<tr>
<td>Solver formulation</td>
<td>Explicit</td>
</tr>
<tr>
<td>Solver flux type</td>
<td>Roe-FDS</td>
</tr>
<tr>
<td>Solver spatial discretization gradient</td>
<td>Green-Gauss node based</td>
</tr>
<tr>
<td>Solver spatial discretization flow</td>
<td>Second order upwind</td>
</tr>
<tr>
<td>Solver transient formulation</td>
<td>Second order implicit</td>
</tr>
<tr>
<td>CFL number</td>
<td>Initial guess 1.8 but was changed if necessary based on initial solution convergence</td>
</tr>
<tr>
<td>Residual convergence criterion</td>
<td>1.00E-08</td>
</tr>
<tr>
<td>Solution initialization</td>
<td>UDF function based on problem statement</td>
</tr>
<tr>
<td>Time step size</td>
<td>Calculated based off of convergence or divergence of solution</td>
</tr>
<tr>
<td>Number of time steps</td>
<td>Run until solution reaches pseudo-steady state</td>
</tr>
<tr>
<td>Maximum iterations per time step</td>
<td>5000</td>
</tr>
</tbody>
</table>
Acknowledgments

The author would like to thank David Friedlander, H.T. Huynh, and Xiao-Yen Wang for their mentorship and support throughout this project. Furthermore, the author would like to thank David Friedlander, H.T. Huynh, and Seth Spiegel for providing data sets from the EZ4D and Flux Reconstruction CFD research codes for comparison with Fluent.

References


