

# Numerical Experiments of Subsonic Jet Flow Simulations Using RANS with OpenFOAM

Negar Naghash Zadeh<sup>1</sup>, Roberto Paoli<sup>1,2</sup>

<sup>1</sup>Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, USA

<sup>2</sup>Argonne National Laboratory, Leadership Computing Facility and Computational Science Division, Lemont, USA

Email: robpaoli@uic.edu

**How to cite this paper:** Zadeh, N.N. and Paoli, R. (2022) Numerical Experiments of Subsonic Jet Flow Simulations Using RANS with OpenFOAM. *Open Journal of Fluid Dynamics*, 12, 230-247.  
<https://doi.org/10.4236/ojfd.2022.122011>

**Received:** April 5, 2022

**Accepted:** June 26, 2022

**Published:** June 29, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

This study investigates the accuracy of different numerical schemes of OpenFOAM software to simulate compressible turbulent jets. Both pressure-based schemes utilizing the implicit PIMPLE algorithm and density-based schemes relying on AUSM scheme and explicit Runge-Kutta time integration are considered. The results of the numerical tests are compared and validated against data from NASA ARN nozzle geometry. The choice of parameter setting of the schemes is discussed in depth and possible optimization strategies are proposed to increase accuracy of RANS simulations of turbulent jets.

## Keywords

CFD, Jet Flow, Axisymmetric, Turbulence, RANS, OpenFOAM

---

## 1. Introduction

The numerical simulation of compressible jet flows is relevant to many areas of aerospace engineering, such as for example, the identification of noise sources in jet aeroacoustics [1] or the prediction of exhaust dispersion in the atmosphere [2]. Modeling high speed jet flow requires a compressible flow solver of Navier-Stokes equations. If Direct Numerical Simulations (DNS) of jet flows in real airframe and operating conditions is still out of reach for current computational resources—and it will be so in a foreseeable future—large-eddy simulations (LES) of jets in single or double stream flows including nozzles appeared recently in the literature [3]. LES offers a promising approach especially when the focus is on the unsteadiness of the jet. However, their computational cost can be extremely high due to the resolution required (both in space and time) to capture the turbulent features of the flow, and the large computational domains needed

to compute the entire 3D flow-field even though the flow is statistically axisymmetric (at least for turbulent round jets). Indeed, recent reviews of experimental and numerical studies of turbulent round jets showed that DNS and LES satisfy the desired level of accuracy but the computational cost can be extremely high [3] [4].

RANS offers a valid alternative to LES when the interest is on the analysis of mean flow characteristics in complex geometries and airframes that are representative of actual engine nozzles, with or without mounted pylon and wings. In RANS models, turbulence is modeled via transport equations for Reynolds stresses in time-averaged Navier-Stokes equations. Spalart-Allmaras [5] one-equation model and  $k-\varepsilon$  [6], and  $k-\omega SST$  [7] two-equation model are among the most turbulence models and have been used extensively for incompressible and compressible jets, in subsonic or supersonic conditions [8].

Even though RANS modeling significantly reduces the simulation computational cost, jet flow prediction accuracy using RANS models is limited [9] [10]. Miltner *et al.* (2015) performed a comprehensive analysis of turbulent, free jets modeled with several RANS models and compare the results with experimental data measured by Laser-Doppler-Anemometry for three-dimensional flow [11]. Their findings suggest that, while some models can offer higher accuracy for a specific application such as  $k-\omega SST$  for free jet flow, they fail to provide the same level of accuracy for more complex problems. RANS modeling is more likely to exhibit limitations in the cases of relatively high temperature jets, when the flow cannot be treated as incompressible or when the axisymmetric conditions do not hold [12].

For applications such as aeroacoustics, accurate turbulence properties are required to assess jet noise emissions. Georgiadis and Yoder (2006), modified three different two-equation stress transport RANS models, tailored specifically for jet flow simulations. Their model modifications succeeded in predicting mean axial velocity with higher accuracy, but no improvement for the prediction of kinetic energy was detected [13]. In addition to the accuracy, the uncertainty of the turbulent models affects the simulation reliability. Mirsha and Iaccarino (2017), investigated discrepancy between the RANS simulated data and high precision data to estimate models' uncertainty [14]. They investigated the uncertainty for subsonic and supersonic, and hot and cold jet flow simulations for four different nozzle geometries.

An advantage of simulating a jet flow in axisymmetric round jet with RANS models is that the computational mesh can be reduced by exploiting the axisymmetry of the problem. Georgiadis *et al.* (2006) used a quadrant of nozzle to simulate lobed nozzle flow fields due to symmetry in both x-y and x-z planes [15]. Depending on the symmetry level, the grid can be reduced further to a two dimensional plane.

OpenFOAM (<https://www.openfoam.com>) is an open-source CFD toolbox that has been extensively applied towards simulating a wide range of engineering ap-

plications with relatively good level of success [16] [17]. It provides many features such as wide variety of solvers, boundary conditions, discretization schemes, turbulence models, etc., tailored for specific applications. Some studies report on the influence of the numerical schemes used within the OpenFOAM platform. Being open source, this toolbox has been constantly improved and expanded over the years and numerous studies have employed OpenFOAM for their simulations. Kannan(2015) used OpenFOAM to study turbulent free jets [18]. Zang *et al.* (2018) used rhoCentralFoam compressible density-based solvers to model supersonic round jets [19]. Yachao *et al.* (2017) developed a LES compressible solver, with the purpose of reducing numerical dissipation, based on rhoCentralFoam [20]. They also tried to improve the numerical stability of central scheme by adopting an alternative form of the convective terms that conserves kinetic energy. Li and Paoli [21] analyzed the scalability of rhoCentrlaFoam and developed an explicit version of the scheme called rhoCentrlaRK4Foam that is based on Runge-Kutta temporal integration.

The main objective of this work is to investigate the accuracy of two schemes of OpenFOAM for RANS simulations of turbulent compressible jets. Specifically, this study utilizes rhoPimpleFoam, a pressure based solver from the OpenFOAM baseline distribution, and rhoEnergyFoam, a density based solver which has been developed by Modesti *et al.* [22]. The benchmark case for the validation is the NASA Acoustic Reference Nozzle (ARF) geometry [23]. The paper is organized as follows: Section 2 gives an overview of the numerical schemes and the computational setup while Section 3 presents the results of the numerical experiments. Finally, Sections 4 and 5 summarize the discussions and draw some conclusion of the study.

## 2. Numerical Methods

In the finite volume approach, the Navier-Stokes equations for a compressible ideal gas are integrated over an arbitrary control volume  $V$ :

$$\frac{d}{dt} \int_V \mathbf{u} dV + \sum_{i=1}^3 \int_{\partial V} (\mathbf{f}_i - \mathbf{f}_i^v) n_i dS = 0 \quad (1)$$

where  $\mathbf{u}$ ,  $\mathbf{f}_i$  and  $\mathbf{f}_i^v$  represent the conservative variables vector and the associated Eulerian and viscous fluxes, respectively.

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u_i \\ \rho E \end{bmatrix}, \quad \mathbf{f}_i = \begin{bmatrix} \rho u_i \\ \rho u_i u_j + p \delta_{ij} \\ \rho u_i H \end{bmatrix}, \quad \mathbf{f}_i^v = \begin{bmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ik} u_k - q_i \end{bmatrix} \quad (2)$$

In OpenFOAM, the control volume is the volume of the grid cell, and the solvers differ in the way the numerical fluxes are discretized and the grid-average equations are evolved in time.

### 2.1. Compressible Solvers: rhoPimpleFoam

RhoPimpleFoam is a transient solver for turbulent compressible flow. It uses the

PIMPLE solution algorithm which combines PISO and SIMPLE methods. The PISO algorithm, or Pressure-Implicit with Splitting of Operators, is a non-iterative solver for the computation of unsteady flows. SIMPLE algorithm, or Semi-Implicit Method for Pressure Linked Equations, is a corrector-predictor solver for steady flow. A PIMPLE algorithm solutions requires an outer-corrector and inner-corrector to be set.

1) The inner-corrector sets the number of times the pressure is corrected in each iteration.

2) The outer-corrector, specifies the number of iterations the system of equations are solved in each time step. Hence, the solver looks for a steady state solution iteratively in each time step within the SIMPLE phase. Then the time step is increased through the PISO phase.

This makes PIMPLE able to advance in time using relatively large time steps, allowing for high Courant number  $\nu > 1$  without compromising the numerical stability. This gives PIMPLE algorithm a substantial advantage over other transient explicit solvers that need to satisfy the CFL condition,  $\nu \leq 1$ , to ensure numerical stability.

## 2.2. Compressible Solvers: rhoEnergyFoam

RhoEnergyFoam is a density-based solver of transient compressible Navier-Stokes equations. It is designed obtain low diffusive solutions in complex geometries [22]. With local use of the AUSM flux splitting method, rhoEnergyFoam preserves total kinetic energy and maintains properties conservation on unstructured mesh. In addition, the AUSM diffusive flux provides the solver with shock-capturing capabilities [24]. The Eulerian fluxes are split into convective and pressure terms.

$$\mathbf{f}_{iON} = \mathbf{f}_{iON} + \mathbf{p}_{iON} = \begin{bmatrix} \rho u_i \\ \rho u_i u_j \\ \rho u_i H \end{bmatrix} + \begin{bmatrix} 0 \\ p \delta_{ij} \\ 0 \end{bmatrix} \quad (3)$$

$\mathbf{f}_{ON}$  is the numerical flux calculated at the interface of two adjacent cells, denoted as Owner and Neighbor, respectively (see **Figure 1**). Each term of the Eulerian flux consists of a central part and a diffusive part.

$$\begin{aligned} \mathbf{f}_{ON} &= \hat{\mathbf{f}}_{ON}^C + \hat{\mathbf{f}}_{ON}^D \\ \mathbf{p}_{ON} &= \hat{\mathbf{p}}_{ON}^C + \hat{\mathbf{p}}_{ON}^D \end{aligned} \quad (4)$$

The central part of the convective flux is interpolated with the midpoint scheme, whereas for the pressure flux, the central part is interpolated with the standard central scheme. The midpoint scheme allows the solver to preserve the flow's kinetic energy. The convective and pressure central fluxes are calculated as follows:

$$\begin{aligned} \hat{\mathbf{f}}_{ON}^C &= 1/8(\rho_O + \rho_N)(u_{nO} + u_{nN})(\varphi_O + \varphi_N) \\ \hat{\mathbf{p}}_{ON}^C &= 1/2(\mathbf{p}_O + \mathbf{p}_N) \end{aligned} \quad (5)$$

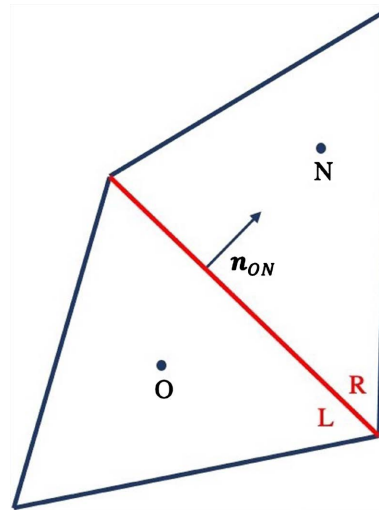
To ensure numerical stability when using an unstructured mesh or where shock waves are present, numerical diffusion is locally added. A shock sensor,  $\theta$ , is introduced to determine if artificial diffusion is needed. The diffusive flux is calculated as follows:

$$\begin{aligned} \hat{f}_{ON}^D &= ICH(\theta_{ON} - \theta^*) \hat{f}_{ON}^{AUSM} \\ \hat{p}_{ON}^D &= IP\theta_{ON} \hat{p}_{ON}^{AUSM} \end{aligned} \tag{6}$$

$IP$  and  $IC$  are flags that control the activation of diffusive fluxes while  $H(\theta_{ON} - \theta^*)$  is a Heaviside function that takes the value 1 wherever  $\theta_{ON}$  exceeds the threshold. By specifying appropriate thresholds for both the pressure and convective terms and activating  $IP$  and  $IC$  flags in line with the simulation need, the artificial diffusion terms provided by the AUSM scheme,  $\hat{f}^{AUSM}$  and  $\hat{p}^{AUSM}$  are added to the solution field.  $IP$  and  $IC$  activation create different configuration modes that affect the final converged solution (see **Table 1**). Utilizing mode B can lead to numerical oscillations in the solution. Mode C is appropriate when using an unstructured grid or where shock waves are present to offer the highest smoothness. In all cases, an explicit third-order Runge-Kutta scheme is used for time integration.

### 2.3. Computational Grid

The axisymmetric nature of the RANS solution for round jets allows the geometry to be reduced to a two-dimensional grid. In order to replicate a two dimensional



**Figure 1.** Owner (O) and Neighbor (N) computational adjacent cells with ON interface.

**Table 1.** Different numerical setups in rhoEnergyFoam.

Mode	$IP$	$IC$
A	0	0
B	1	0
C	1	1

domain using volumetric cells, the planar grid is rotated  $1^\circ$  around the axial direction to create a single-cell thickness wedge.

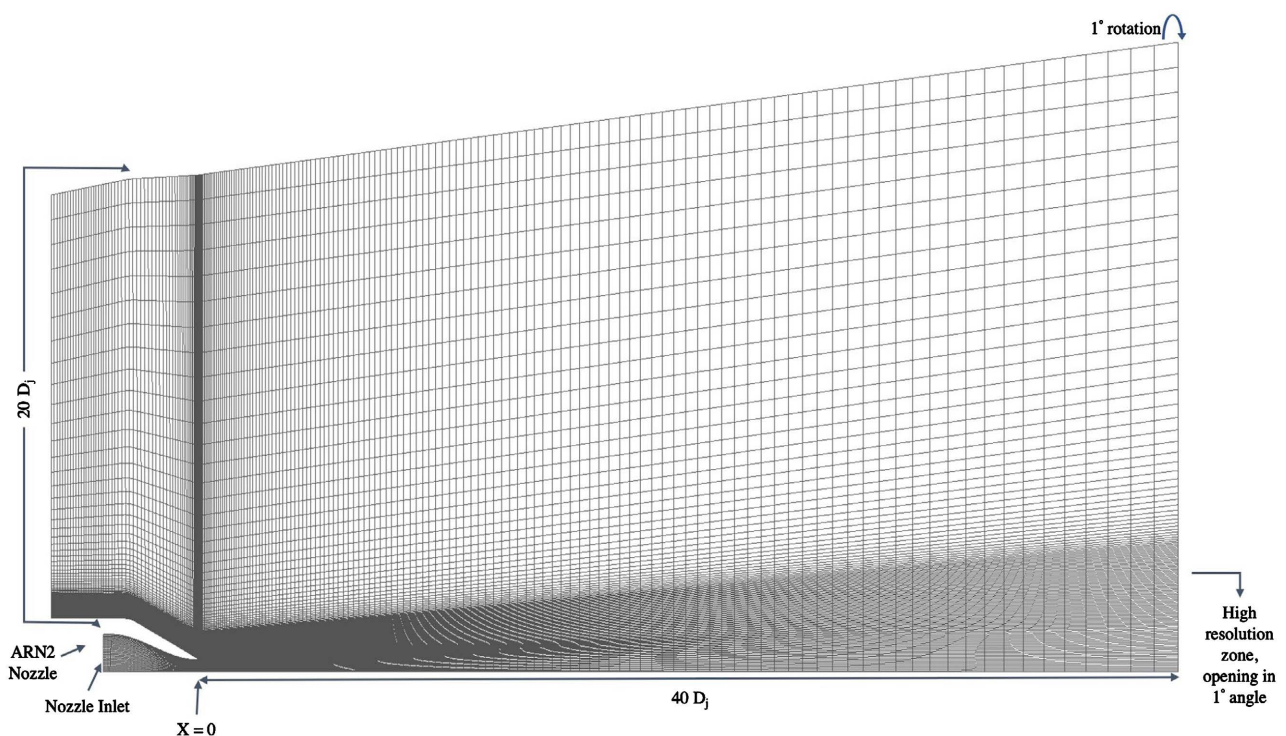
The NASA Acoustic Research Nozzle (ARN) is used for the geometry's outline. The nozzle diameter and length are  $D_j = 0.05$  m and  $L_j = 0.196$  m, respectively, while the nozzle lip thickness is 1 mm. The computational domain shown in **Figure 2** is constructed using structured cells and extends up  $20D_j$  in the radial direction and  $40D_j$  in the axial direction. The mesh is created using Pointwise grid generator.

The grid radial resolution is  $\Delta r/D_j = 0.005$ . To ensure a smooth change in the high gradient zone, the radial grid growth factor is 1% within two jet diameters from the centerline, and 15% rate for the rest of the radial domain. The axial grid spacing is in the range  $\Delta x/D_j = [0.01, 1]$ , with the most resolved cells located at the jet exit, and grows as it gets close to the domain outlet.

**Figure 3** shows a magnified view of the high resolution zone within two diameter distance from the centerline. To ensure grid independency, two additional grids were introduced with the highest resolution twice as big and half as big as the initial grid. **Table 2** shows the grid details for each mesh.

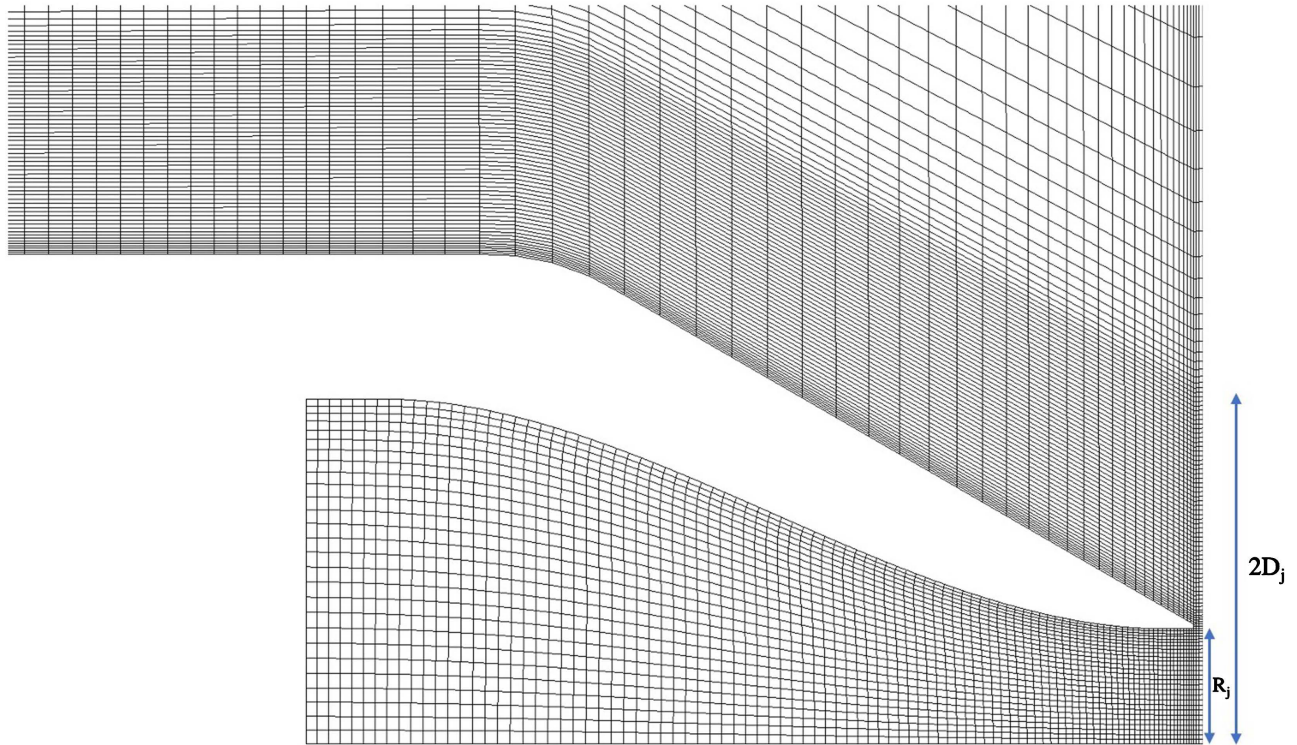
## 2.4. Simulation Setup

The subsonic axisymmetric jet flow exits into quiescent ambient air with  $M_j = 0.5$ . To avoid the numerical complications of modeling the ambient air to be absolutely quiescent, the air flow moves in the direction of the jet flow with  $M_\infty \approx 0.003$ . The reference pressure and temperature are  $p_\infty = 101300$  Pa and



**Figure 2.** Two-dimensional wedge grid.





**Figure 3.** Magnified view of the mesh in the high-resolution zone, 1% radial grid growth within two jet diameters from the centerline.

**Table 2.** Grids radial and axial resolution and domain length and width.

Mesh	Min $\Delta r/D_j$	Min $\Delta x/D_j$	Domain length $L$	Domain width $D$
Coarse	0.0100	0.01	$40D_j$	$20D_j$
Medium	0.0500	0.01	$40D_j$	$20D_j$
Fine	0.0025	0.01	$40D_j$	$20D_j$

$T_\infty = 293$  K, respectively. The diameter ratio of the nozzle is  $D_{inlet}/D_{exit} = 3$  which dictates the nozzle pressure ratio of  $NPR = 1.197$  to reach the desired Mach number at exit. The simulations are run for subsonic cold jet with the temperature ratio of  $T_t/T_\infty = 1$ . The boundary conditions at the nozzle inlet are fixed total pressure,  $p_t = 120141$  Pa and total temperature,  $T_t = 293$  K.

### 3. Results

Section 3.1 discusses a series of tests conducted using rhoPimpleFoam with different turbulence models. Then the grid dependency test is conducted using this pressure-based solver. In Section 3.2, the medium size mesh and the  $k-\omega SST$  turbulence model are chosen for comparison of data obtained with rhoPimpleFoam and rhoEnergyFoam. The NASA cold jet subsonic Particle Image Velocimetry(PIV) dataset is used for validation [23].

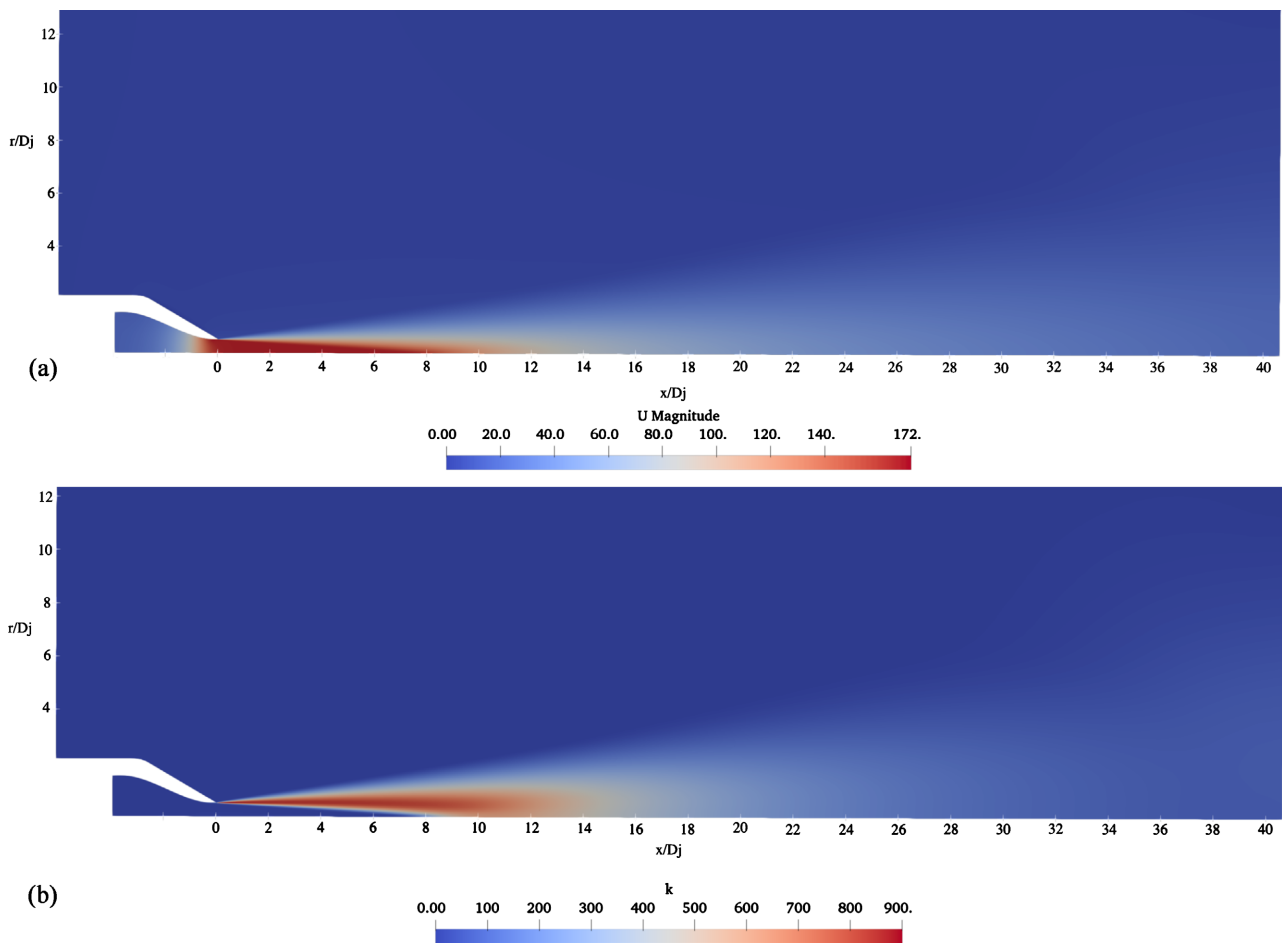
#### 3.1. rhoPimpleFoam Solver

The Spalart-Allmaras,  $k-\omega SST$  and  $k-\epsilon$  turbulence models are used for the 2D

RANS modeling tests. To visualize the spatial distribution of the flow in the domain, **Figure 4(a)** and **Figure 4(b)** show the streamwise contours of velocity and kinetic energy for the medium grid resolution using the  $k-\varepsilon$  turbulence model.

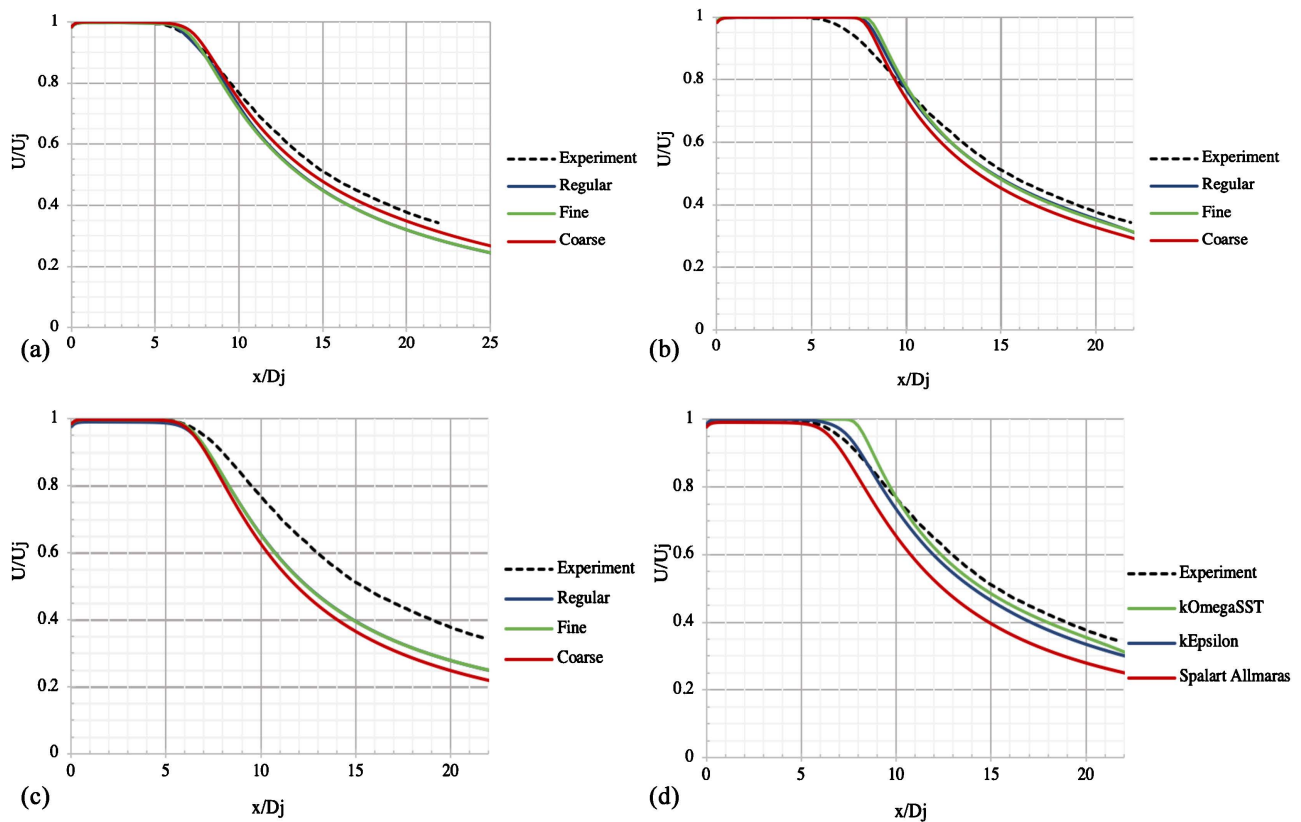
**Figures 5(a)-(c)** report the axial velocity along the centerline normalized by jet exit velocity for different grid resolutions. The Spalart-Allmaras,  $k-\omega SST$ , and  $k-\varepsilon$  turbulence models are used for this test. It can be seen that for all three turbulence models, the medium and fine grids show no significant difference, which indicates that the medium grid is sufficiently resolved for the RANS simulations.

**Figure 5(d)** shows the centerline mean axial velocity obtained with same grid and three different turbulence models. The velocity remains constant in the potential core region and spreads gradually into the domain as the shear layer expands downstream for all three turbulence models. The simulated data are compared to the cold jet experimental data. The Spalart-Allmaras model shows the most accurate prediction of the potential core length (0.1% error), the  $k-\varepsilon$  model slightly over-predicts it (11% error), and the  $k-\omega SST$  model shows the highest error (33% error). On the other hand, the Spalart-Allmaras model shows the largest axial velocity decay compared to the other turbulence models. This is



**Figure 4.** Streamwise contours of (a) mean axial velocity, and (b) kinetic energy.



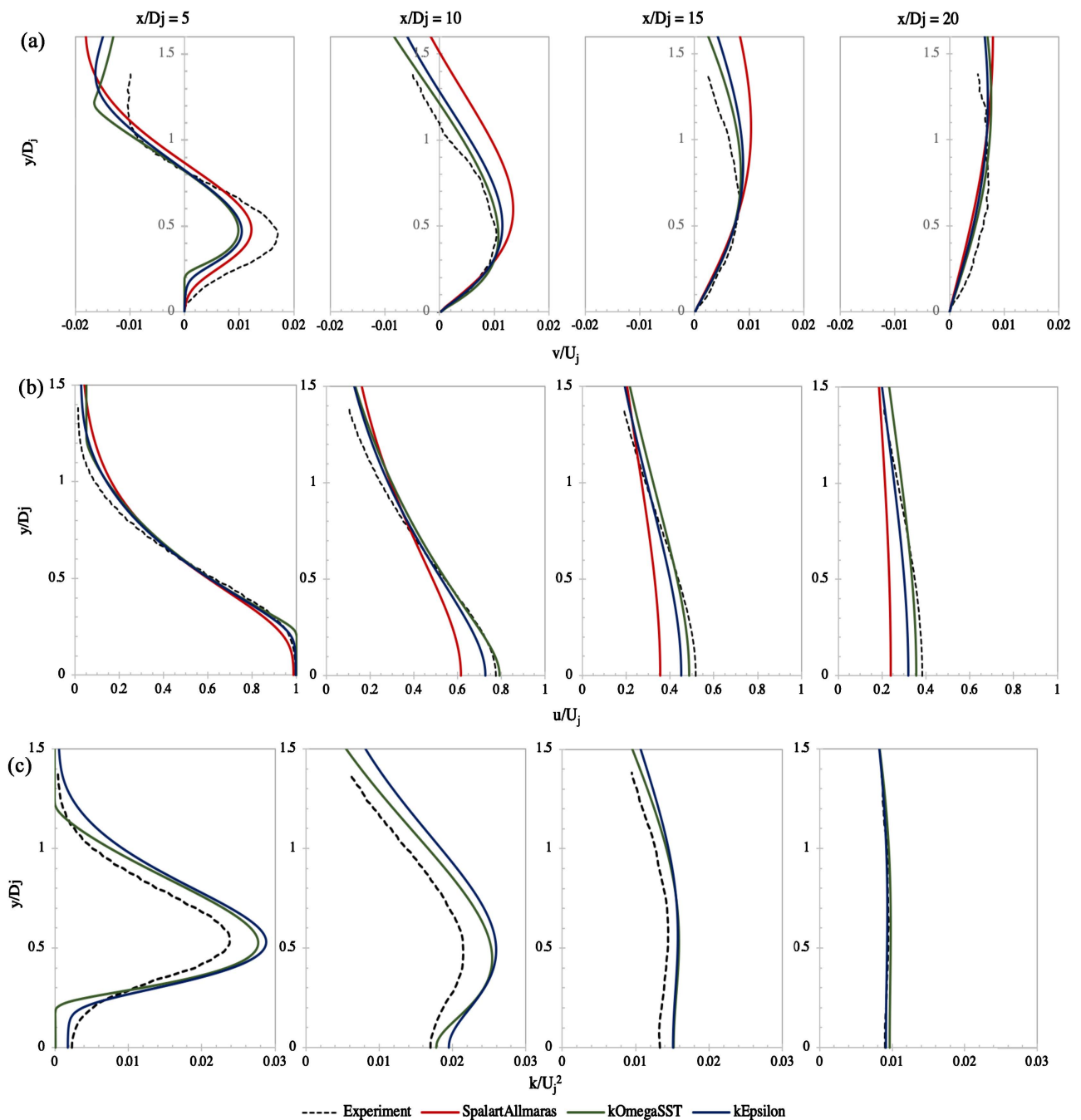


**Figure 5.** Grid dependency test for 3 grid resolution, comparing normalized mean axial velocity in the streamwise direction at the centerline using (a)  $k-\epsilon$ , (b)  $k-\omega SST$ , and (c) Spalart-Allmaras turbulence models. (d) Normalized velocity comparison of models.

a consequence of the high dissipation behavior of the Spalart-Allmaras model. The decay error for each turbulence model is calculated using the root mean square error, RMSE, for  $k-\omega SST$ ,  $k-\epsilon$ , and Spalart-Allmaras are 3%, 3.6%, and 9.1% respectively.

**Figure 6** shows the normalized radial velocity, axial velocity, and kinetic energy profiles in the radial direction at  $5D_j$ ,  $10D_j$ ,  $15D_j$ , and  $20D_j$  axial distances from the jet exit. **Figure 6(a)** shows that the peak radial velocity occurs farther from the centerline compared to the experimental data, and the radial difference between peaks becomes wider when moving away from the nozzle exit. In addition, **Figure 6(a)** shows that, as the axial distance increases, the simulations fail to accurately predict the location where the radial velocity changes sign (from outward to inward).

**Figure 6(b)** compares the normalized axial velocity along radial direction with experimental data. Along the streamwise direction, the  $k-\epsilon$  and  $k-\omega SST$  results tend to underpredict the velocity at the centerline more, whereas they reach a better agreement with the experimental data between 1 to 1.5 diameter distances from the centerline. Moreover, all three models represent the experimental data accurately at a distance of  $5D_j$  from the jet exit. The accuracy deteriorates when moving further away from the exit for  $k-\omega SST$  and Spalart-Allmaras models, which is in line with the trend observed in **Figure 5**.



**Figure 6.** Radial profiles for (a) normalized radial velocity, (b) normalized axial velocity, and (c) normalized kinetic energy at  $5D_j$ ,  $10D_j$ ,  $15D_j$ , and  $20D_j$  distance from jet exit.

**Figure 6(c)** shows that the largest overprediction of kinetic energy occurs close to the lip line,  $y/D_j = 0.5$ , while the accuracy increases further away in the radial direction.

### 3.2. rhoEnergyFoam Solver

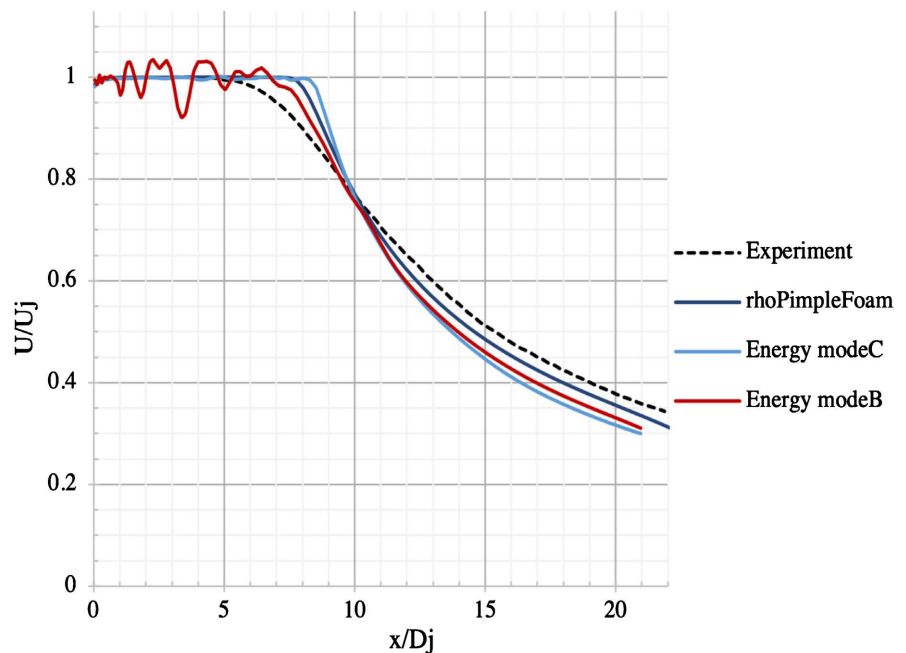
Based on the grid dependency test results in Section 3.1, the medium resolution grid is sufficient to insure grid convergence and is then used for the simulations

with rhoEnergyFoam. The  $k-\omega SST$  model is used as turbulence model. **Table 3** presents the potential core length and velocity decay errors with respect to the experimental data for all tests conducted in this section.

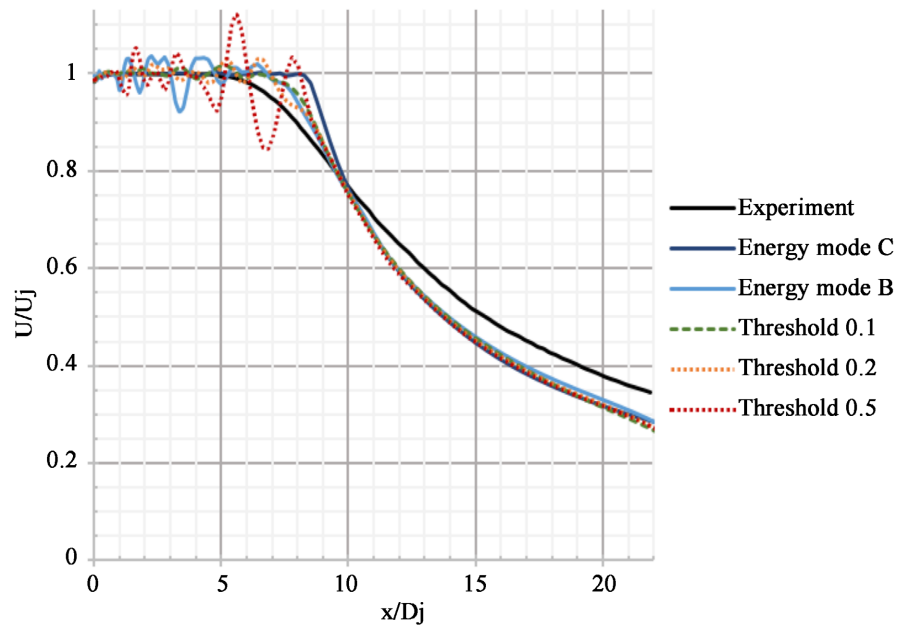
**Figure 7** compares the results obtained using rhoPimpleFoam and rhoEnergyFoam with the NASA subsonic cold jet experimental data. The figure shows that the mode-C setting of rhoEnergyFoam is the most dissipative scheme and produces the highest decay rate in the streamwise direction compared to the experimental data. The first step to reduce the numerical dissipation is by switching to mode-B configuration. However, as anticipated in Sec. 2.2, this leads to higher oscillations in the core region as shown in **Figure 7**. The overprediction of the potential core length halves by simply disabling the convective diffusive flux (see **Table 3**).

The next approach to reduce the generated numerical diffusion of rhoEnergyFoam solution is to alter the threshold value  $\theta^*$ . As discussed in Sec. 2.2, Equation (6) describes how the convective threshold operate as an activation switch in the Heaviside function for all shock sensor values captured above the predefined threshold. The change of the threshold affects the number of cells that are treated with the shock sensor.

**Figure 8** shows the axial velocity in streamwise direction using different settings and thresholds: mode C and mode B with shock sensors  $\theta^* = 0.1$ ,  $\theta^* = 0.2$ , and  $\theta^* = 0.5$ . As  $\theta^*$  increases from the recommended threshold,  $\theta^* = 0.05$ , the numerical oscillations in the potential core region increase. The threshold alteration test does not provide any significant improvement on the decay rate and the potential core length prediction only improves for the  $\theta^* = 0.1$  test.



**Figure 7.** Normalized mean axial velocity in the streamwise direction at the centerline simulate with pressure-based and density-based solvers.



**Figure 8.** Normalized mean axial velocity in the streamwise direction at the centerline of mode C, mode B and the threshold tests for  $\theta^* = 0.1, 0.2, \text{ and } 0.5$ .

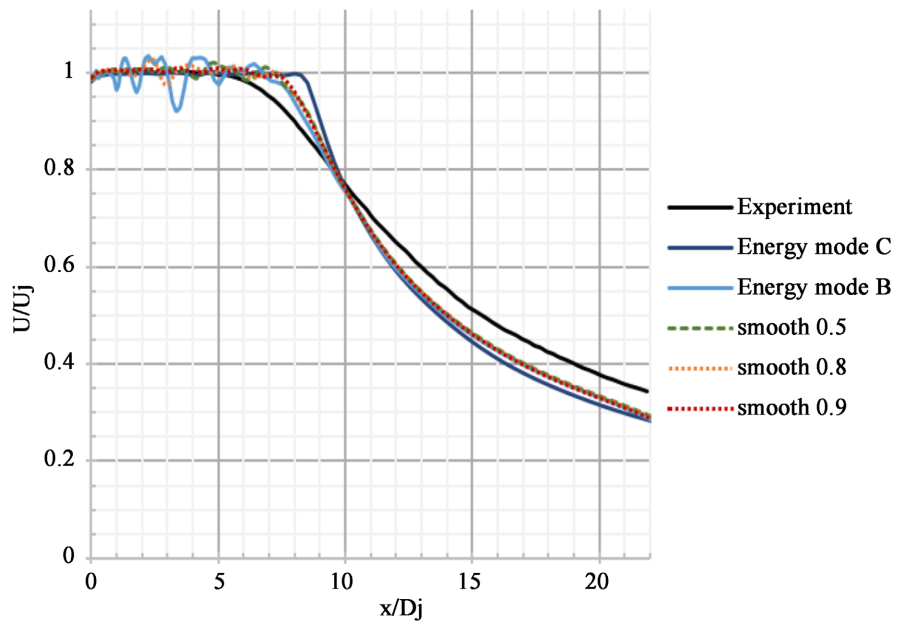
Another approach tested to alleviate the excessive dissipation is to introduce a smoothness factor. The idea is to modify the numerical flux by adding a localized correction to the scheme with  $smooth = 1$  corresponding to the mode C and  $smooth = 0$  corresponding to mode B configuration of AUSM scheme. These tests were carried out in mode-C configuration with the recommended threshold  $\theta^* = 0.05$ :

$$\hat{f}^D = smooth * ICH(\theta_{ON} - \theta^*) \hat{f}^{AUSM} \quad (7)$$

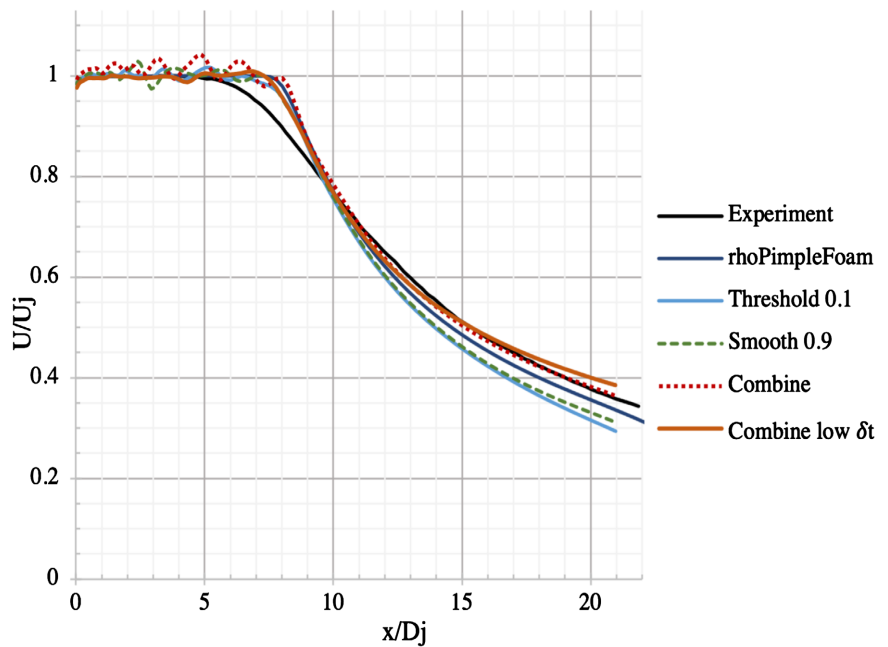
**Figure 9** compares the centerline velocity in the streamwise direction using different models and smoothness factors. Increasing the smoothness tends to reduce the overprediction of the potential core length. The decay rate is also improved compared to the baseline mode-C (see **Table 3**).

With the aim of minimizing both numerical dissipation and oscillations, we finally considered different combination of the threshold and smoothness factor. The results of these tests showed that the values  $\theta^* = 0.1$  and  $smooth = 0.9$  represent an optimal combination in that they improve the decay rate substantially. **Table 3** shows that with this combination of parameters, the error in the decay rate was reduced to 2.9%, the lowest among all tests performed in the study. However, this combination cannot completely eliminate the oscillations in the potential core region. The same test was then repeated by reducing the time-step by one order of magnitude, which leads to a substantial damping of numerical oscillations as shown in **Figure 10**.

**Figure 11** shows the normalized radial velocity, axial velocity, and kinetic energy in the radial direction at  $x = 5D_j$ ,  $x = 10D_j$ ,  $x = 15D_j$ , and  $x = 20D_j$ . It can be observed that the radial velocity in all cases is close to zero within  $0.1D_j$



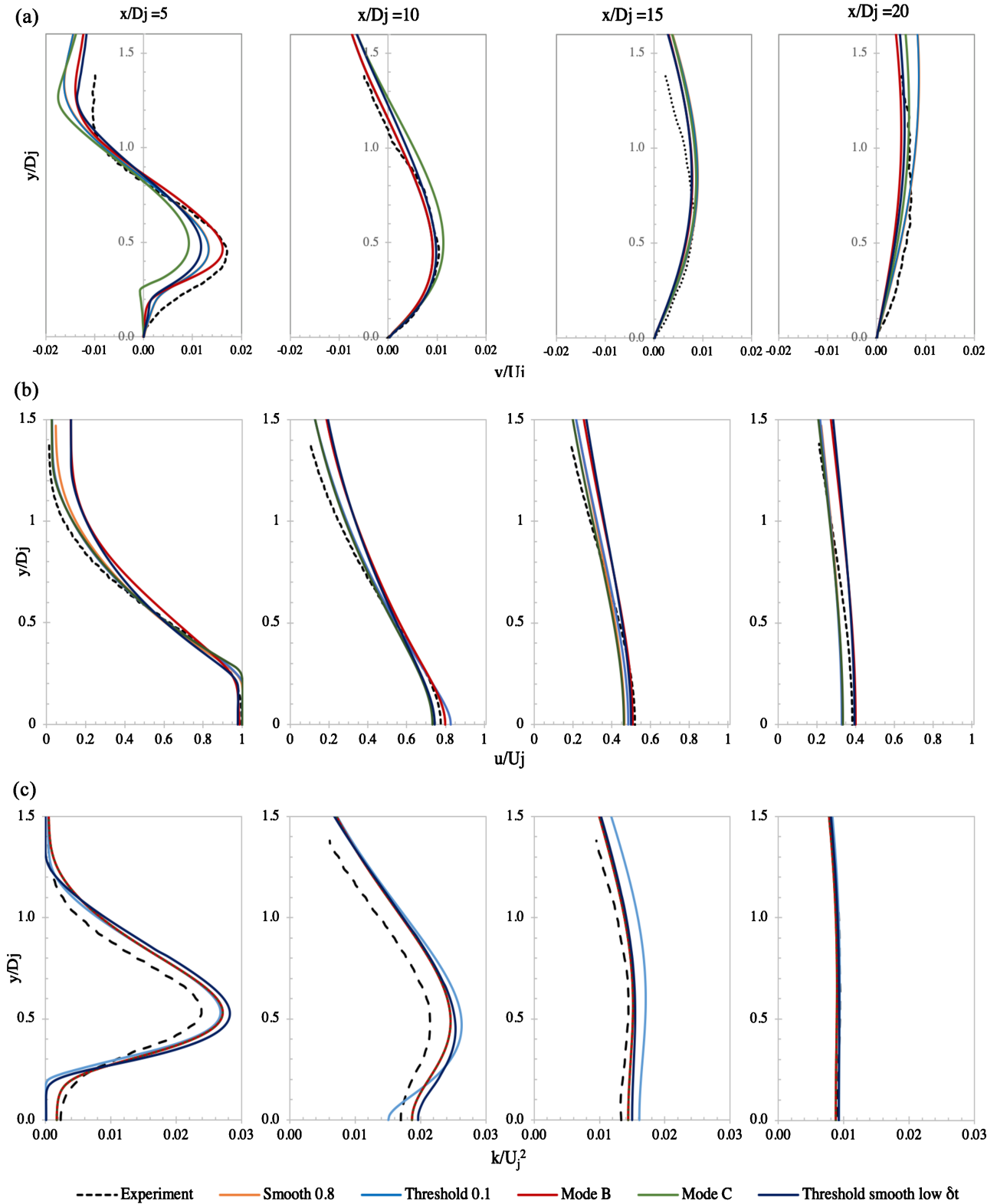
**Figure 9.** Normalized mean axial velocity in the streamwise direction at the centerline of mode C, B, and C with the smoothness factors of 0.5, 0.8 and 0.9.



**Figure 10.** Normalized mean axial velocity in the streamwise direction at the centerline simulated with rhoPimpleFoam and rhoEnergyFoam in mode B, mode C, mode C  $\theta = 0.1$ , mode C smoothness 0.9, and mode C combined test.

**Table 3.** The potential core length and velocity decay prediction error compare to experimental data.

error	rhoPimpleFaom	Mode C	Mode B	$\theta^* = 0.1$	<i>smooth</i> = 0.9	combine	reduced $\Delta t$
potential core length	33%	36.2%	16.5%	16.1%	23.1%	21.8%	17.6%
velocity decay	3%	6.1%	4.7%	5.1%	3.9%	2.9%	2.5%



**Figure 11.** Radial profiles for normalized a) radial velocity, b) axial velocity, and c) kinetic energy at  $5D_j$ ,  $10D_j$ ,  $15D_j$ , and  $20D_j$  distance from jet exit simulated with rhoEnergyFoam solver, in mode B, C, smooth, and threshold test.



distance from the centerline while the experimental data shows a much sharper increase from the centerline. This behavior is expected to appear when using the  $K-\omega SST$  turbulence model which was also observed in **Figure 6**. The location where radial velocity peaks was well captured at all four axial locations. It is also worth noting that while all simulated cases predicted the change of velocity vector direction at  $x = 5D_j$  accurately, they gradually overpredict it further away from the jet exit.

**Figure 11(b)** shows the normalized axial velocity in the radial direction. The shear layers development is responsible for the gradual spread of the jet, therefore, the overprediction of potential core length emphasizes the delay in shear layer expansion that affects the axial velocity decay trend at the first data point. Moving further away from the jet exit, the simulations show a better agreement with experimental data.

**Figure 11(c)** shows the normalized kinetic energy in the radial direction. It can be seen that the kinetic energy peaks at the lip line,  $y/D_j = 0.5$ , where the highest overprediction occurs as well. The simulation curves reach a better agreement with the experimental data when moving away from the jet exit in the axial direction.

## 4. Discussion

In Section 3.1, the solution's dependency on the grid resolution was investigated. The grid dependency test showed that the medium size grid has the necessary and sufficient resolution for the solution to be independent of the grid size. Based on the centerline velocity comparison in **Figure 5(d)**, the choice of turbulence model depends on whether the study prioritizes a more accurate prediction of the potential core length or velocity decay rate.

Section 3.2 investigates the solution of a density-based solver run with several different configurations. The analysis shows that deactivating the convective shock sensor reduces the overprediction of the potential core length but leads to the appearance of numerical oscillations in the potential core region.

An intermediate configuration that applies a smoothness factor to the shock sensor and alters its threshold can minimize both numerical oscillations and numerical dissipation. The smoothness factor controls the strength of the AUSM flux on the flow field, whereas the threshold controls the value of AUSM flux. Each parameter setting defines an intermediate condition between mode-B and mode-C solutions, however, neither of them offers a considerable enhancement to the results individually. Indeed, the main difference between the two parameters is the way they affect the solution: the smoothness factor reduces the added dissipation by letting fewer cells to meet the shock sensor's criteria (**Figure 8**), whereas the threshold increases the number of cells for the artificial dissipation, reducing at the same time the strength of the numerical dissipation (**Figure 9**).

The last test consisted in applying both threshold and smoothness factors to the simulation configuration. While the two factors did not show any substantial

change in the results on their own, combining them resulted in the best agreement of the velocity decay with the experimental data among the tests conducted with the rhoEnergyFoam solver.

A closer look at the normalized centerline velocity in **Figure 10** shows that the density-based solver tends to generate oscillation even when running in mode C. It appears that the slightest change in the runtime configuration can magnify any existing oscillation. To overcome this stability issue, the time-step was lowered by an order of magnitude for the last simulation. Although the combined reduced  $\Delta t$  simulation shows a satisfactory result, decreasing the time step increases the computational cost. As a final remark, the time step for rhoPimpleFoam and rhoEnergyFoam solutions are of the order of  $10^{-6}$  and  $10^{-8}$  s, respectively. Using the same number of processors, simulating one flow through time takes approximately 200 s and 10,000 s computing time using rhoPimpleFoam and rhoEnergyFoam solvers, respectively.

## 5. Conclusion

This study tested different numerical schemes within the OpenFoam open source software to simulate compressible turbulent jets. Overall, the comparison between pressure-based and density-based solvers showed the latter has potential for increasing the accuracy of the prediction of the jet decay rate and potential core length. However, it requires careful choice of the parameter setting to optimize the accuracy and numerical stability of the solution. The PIMPLE algorithm in rhoPimpleFoam requires significantly less computational effort compared to the rhoEnergyFoam solver which uses an explicit Runge-Kutta algorithm for temporal integration. The mismatch in the computational cost would be reduced for: 1) high Mach numbers, eventually supersonics flows, where the CFL constraint is less restrictive, and 2) LES and DES of turbulent flows where small time steps are needed to resolve turbulent fluctuations and calculate statistics, irrespective of numerical stability. As a final remark, a semi-implicit version of rhoEnergyFoam where Eulerian fluxes are treated explicitly and viscous fluxes implicitly could be an interesting and useful upgrade of the solver in the future.

## Acknowledgments

This work was supported by the National Science Foundation, through grant #1854815, titled “HighPerformance Computing and Data-Driven Modeling of Aircraft Contrails”, awarded to R. Paoli.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Moreau, S. (2022) The Third Golden Age of Aeroacoustics. *Physics of Fluids*, **34**,

Article ID: 031301. <https://doi.org/10.1063/5.0084060>

- [2] Paoli, R. and Shariff, K. (2015) Contrail Modeling and Simulation. *Annual Review of Fluid Mechanics*, **48**, 393-427. <https://doi.org/10.1146/annurev-fluid-010814-013619>
- [3] Brés, G., P. Jordan, V. Januet, M. Le Rallic, A. Cavalieri, A. Towne, Lele, S., Colonius, T. and Schmidt, O. (2018) Importance of the Nozzle-Exit Boundary-Layer State in Subsonic Turbulent Jets. *Journal of Fluid Mechanics*, **581**, 83-124. <https://doi.org/10.1017/jfm.2018.476>
- [4] Ball, C., Fellouah, H. and Pollard, A. (2012) The Flow Field in Turbulent Round Free Jets. *Progress in Aerospace Sciences*, **50**, 1-26. <https://doi.org/10.1016/j.paerosci.2011.10.002>
- [5] Boguslawski, A., Tyliczszak, A., Wawrzak, A. and Wawrzak, K. (2017) Numerical Simulation of Free Jets. *International Journal of Numerical Methods for Heat & Fluid Flow*, **27**, 1056-1063. <https://doi.org/10.1108/HFF-03-2016-0103>
- [6] Spalart, P. and Allmaras, S. (1992) A One-Equation Turbulence Model for Aero-Dynamic Flows. *30th Aerospace Sciences Meeting and Exhibit*, Reno, 6-9 January 1992, 439. <https://doi.org/10.2514/6.1992-439>
- [7] Jones, W. and Launder, B.E. (1972) The Prediction of Laminarization with a Two-Equation Model of Turbulence. *International Journal of Heat and Mass Transfer*, **15**, 301-314. [https://doi.org/10.1016/0017-9310\(72\)90076-2](https://doi.org/10.1016/0017-9310(72)90076-2)
- [8] Menter, F.R. (1994) Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. *AIAA Journal*, **32**, 1598-1605. <https://doi.org/10.2514/3.12149>
- [9] Cebeci, T. (2013) *Analysis of Turbulent Flows with Computer Programs*. Butterworth-Heinemann, Oxford.
- [10] Kenzakowski, D. (2004) Turbulence Modeling Improvements for Jet Noise Prediction Using PIV Datasets. *10th AIAA/CEAS Aeroacoustics Conference*, Manchester, 10-12 May 2004, 2978. <https://doi.org/10.2514/6.2004-2978>
- [11] Koch, L., Bridges, J. and Khavaran, A. (2002) Flow Field Comparisons from Three Navier-Stokes Solvers for an Axisymmetric Separate Flow Jet. *40th AIAA Aerospace Sciences Meeting & Exhibit*, Reno, 14-17 January 2002, 672. <https://doi.org/10.2514/6.2002-672>
- [12] Miltner, M., Jordan, C. and Harasek, M. (2015) CFD Simulation of Straight and Slightly Swirling Turbulent Free Jets Using Different RANS-Turbulence Models. *Applied Thermal Engineering*, **89**, 1117-1126. <https://doi.org/10.1016/j.applthermaleng.2015.05.048>
- [13] Kaushik, M., Kumar, R. and Humrutha, G. (2015) Review of Computational Fluid Dynamics Studies on Jets. *American Journal of Fluid Dynamics*, **5**, 1-11.
- [14] Georgiadis, N.J., Yoder, D.A. and Engblom, W.A. (2006) Evaluation of Modified Two-Equation Turbulence Models for Jet Flow Predictions. *AIAA Journal*, **44**, 3107-3114. <https://doi.org/10.2514/1.22650>
- [15] Mishra, A.A. and Iaccarino, G. (2017) Uncertainty Estimation for Reynolds-Averaged Navier-Stokes Predictions of High-Speed Aircraft Nozzle Jets. *AIAA Journal*, **55**, 3999-4004. <https://doi.org/10.2514/1.J056059>
- [16] Georgiadis, N.J., Rumsey, C.L., Yoder, D.A. and Zaman, K.B. (2006) Turbulence Modeling Effects on Calculation of Lobed Nozzle Flow Fields. *Journal of Propulsion and Power*, **22**, 567-575. <https://www.openfoam.com>  
<https://doi.org/10.2514/1.17160>
- [17] Weller, H.G. and Tabor, G. (1998) A Tensorial Approach to Computational Conti-

- num Mechanics Using Object-Oriented Techniques. *Computers in Physics*, **12**, 620-631. <https://doi.org/10.1063/1.168744>
- [18] Kannan, B. (2015) Computation of an Axisymmetric Jet Using OpenFOAM. *Procedia Engineering*, **127**, 1292-1299. <https://doi.org/10.1016/j.proeng.2015.11.486>
- [19] Zang, B., Vevek, U., Lim, H., Wei, X. and New, T. (2018) An Assessment of OpenFOAM Solver on RANS Simulations of Round Supersonic Free Jets. *Journal of Computational Science*, **28**, 18-31. <https://doi.org/10.1016/j.jocs.2018.07.002>
- [20] Lee, Y.C., Yao, W. and Fn, X. (2017) A Low-Dissipation Scheme Based on OpenFOAM Designed for Large Eddy Simulation in Compressible Flow. *21st AIAA International Space Planes and Hypersonics Technologies Conference*, Xiamen, 6-9 March 2017, 2017-2444. <https://doi.org/10.2514/6.2017-2444>
- [21] Li, S. and Paoli, R. (2020) Scalability of OpenFOAM Density-Based Solver with Runge-Kutta Temporal Discretization Scheme. *Scientific Programming*, **2020**, Article ID: 9083620. <https://doi.org/10.1155/2020/9083620>
- [22] Modesti, D. and Pirozzoli, S. (2017) A Low-Dissipative Solver for Turbulent Compressible Flows on Unstructured Meshes, with OpenFOAM Implementation. *Computers & Fluids*, **152**, 14-23. <https://doi.org/10.1016/j.compfluid.2017.04.012>
- [23] Bridges, J. and Wernet, M.P. (2011) The NASA Subsonic Jet Particle Image Velocimetry (PIV) Dataset.
- [24] Liou, M.-S. and Steffn Jr., C.J. (1993) A New Flux Splitting Scheme. *Journal of Computational Physics*, **107**, 23-39. <https://doi.org/10.1006/jcph.1993.1122>