

Spring 2024

## Development of Eagle3D Solver For Wall Modeled LES of Transonic Flows

Spencer Moore

*Embry-Riddle Aeronautical University*, moores40@my.erau.edu

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Aerodynamics and Fluid Mechanics Commons](#)

---

### Scholarly Commons Citation

Moore, Spencer, "Development of Eagle3D Solver For Wall Modeled LES of Transonic Flows" (2024).  
*Doctoral Dissertations and Master's Theses*. 797.  
<https://commons.erau.edu/edt/797>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Doctoral Dissertations and Master's Theses by an authorized administrator of Scholarly Commons. For more information, please contact [commons@erau.edu](mailto:commons@erau.edu).

By

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aerospace Engineering

Embry-Riddle Aeronautical University

Daytona Beach, Florida

By

THESIS COMMITTEE

---

---

---

Graduate Program Coordinator,  
Dr. Hever Moncayo

---

Date

---

Dean of the College of Engineering,  
Dr. James W. Gregory

---

Date

---

Associate Provost of Academic Support,  
Dr. Kelly Austin

---

Date

## ACKNOWLEDGMENTS

I would first like to acknowledge my thesis supervisor and advisor, Dr. William Engblom. His support in my work is what has allowed me to grow a career in CFD and foster a love for research and engineering. I always came away from our discussions feeling as if I was doing something exciting and couldn't wait to work on the next step, even when the challenges felt insurmountable. I feel so lucky to have had his mentorship, dedication, and support over the past few years as I have learned so much from him. Thanks for all the hard work and involvement you have put into my time here at Riddle, I will keep what I've learned from you for the rest of my career and life.

I would like to acknowledge my parents for their love and support throughout my life. It is because of you that I have been able to achieve everything I have. You have afforded me many great opportunities to pursue what I love and given me the tools and guidance needed to seize them. I feel forever grateful that I have had both of you with me and I can't wait to move forward into this next stage of my life and continue to learn and grow because of everything you have done for me.

I would also like to acknowledge Abhinav Balamurugan, Arjun Vedam, and Ben Malczewski for their insight and friendship. They helped me bounce ideas, develop my research and made my work environment fun and interesting.

Lastly I would like to acknowledge all of the faculty and staff at ERAU who helped me along the way.

## ABSTRACT

Wall modeled Large Eddy Simulation (LES) is an area of interest due to its ability to lower computational costs of LES simulation. Even with the application of wall models, LES still proves to have practicality issues when it comes to use in industry, due to the expertise, time, and computational resources required to get results. A case described by an axisymmetric transonic bump is explored utilizing the Embry-Riddle Aeronautical Universities in house unstructured finite volume multi-element CFD code, Eagle3D.

Eagle3D, has been brought to the state of the art and validated against current research using this transonic bump case as a benchmark. Added to Eagle3D includes features such as an HLLC and skew symmetric scheme switched via Ducros sensor, integrated Synthetic Eddy Method (SEM) such that realistic turbulence can be produced at domain inlets, and the implementation of a wall model. Additionally, a novel technique for generating a lean LES grid is explored. The technique utilizes a RANS solution to extract turbulence information and infer a lean grid optimized for wall modeled LES. The solution found by the generated grid is then compared against the validation cases ran in Eagle3D.

## TABLE OF CONTENTS

<b>ACKNOWLEDGMENTS</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>LIST OF FIGURES</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>vii</b>
<b>NOMENCLATURE</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Methodology</b>	<b>3</b>
2.1 Eagle3D Flow Solver	3
2.2 Wall Model	4
2.2.1 Wall Model Description	4
2.2.2 Wall Model Implementation	5
<b>3 Validation Of Eagle3D for Wall Modeled LES</b>	<b>8</b>
3.1 Taylor-Green Vortex	8
3.2 Transonic Bump Problem Description	9
3.3 Computational Grid	10
<b>4 Grid Generation for Wall Modeled LES</b>	<b>12</b>
4.1 Problem Introduction	12
4.2 RANS Case	12
4.3 Shock Sensing	13
4.4 Near-Field/Background Acoustic Sizing	14
4.5 Grid Sizing from Turbulence Parameters	15
4.6 Wall Spacing	17

4.7 Smoothing of Grid Around Shocks	18
<b>5 Results</b>	<b>22</b>
5.1 Validation Results	22
5.2 Validation of WMLES Results for Generated Grid	25
<b>6 Conclusion and Future Work</b>	<b>30</b>
<b>REFERENCES</b>	<b>31</b>
<b>A Appendix</b>	<b>34</b>

## LIST OF FIGURES

Figure	Page
2.1 Flow chart of the wall models solve process.	5
2.2 Example of how the wall model searches for a list of cells to sample from.	6
2.3 Example tet grid with path the wall search algorithm takes.	7
3.1 Time evolution of mean kinetic energy of Eagle3D as compared to Morris [1].	8
3.2 Comparison of iso-contours for normalized vorticity magnitude between Morris [1] (left) and Eagle3D (right). Contour levels for the Morris image are 1, 5, 10, 20 and 30.	9
3.3 Q criterion visualization for $t^* = 8$ .	9
3.4 Grid used for the in the simulation of the axisymmetric bump.	10
4.1 RANS grid used for LES grid generation	13
4.2 Point cloud Generated from RANS grid.	13
4.3 Cloud points flagged by the ducros sensor in blue.	14
4.4 Cloud points heat map based on assigned spacing.	16
4.5 Grid generated by near field/background spacing and Kolmogorov scale.	16
4.6 Spacing generated by only considering wall $y^+$ .	17
4.7 Grid generated only considering wall $y^+$ .	18
4.8 A diagram showing the distance from a Ducros flagged cell and the $i$ th cell	19
4.9 The shape functions of the grid smoothing dependent on $n$ .	19
4.10 Cloud points of final spacing.	20
4.11 Final grid generation.	20
5.1 Skin friction and pressure coefficient plots for validation cases.	22
5.2 Q criterion visualization of the $Y^+$ target of 60 run	24
5.3 Nondimensional velocity plot of the $Y^+$ target of 60 run	24



5.4	Mach number of the $Y^+$ target of 60 run	25
5.5	Gradient of density of the $Y^+$ target of 60 run	25
5.6	Q criterion visualization of the $Y^+$ target of 60 run with generated grid	26
5.7	Q criterion visualization comparison with grid between validation grid (left) and generated grid (right).	27
5.8	Skin friction and pressure coefficient plots for grid generation case.	27
5.9	Gradient of density comparison of validation case (top) vs grid generation case (bottom).	28
5.10	Nondimensional velocity plot of the $Y^+$ target of 60 run with generated grid.	29
5.11	Mach number of the $Y^+$ target of 60 run with generated grid.	29

## LIST OF TABLES

Table		Page
3.1	Grid spacings at various locations used in the simulation of the axisymmetric bump. Boundary layer thickness $\delta$ is assumed to be 1 cm, and $c = 20.32$ cm.	11

## NOMENCLATURE

$C_f$	Coefficient of skin friction
$C_p$	Coefficient of pressure
$F$	Flux
$\eta$	Kolmogorov scale.
$\gamma$	Ratio of specific heats.
$\mu$	Dynamic Viscosity
$\omega$	Rate of turbulent dissipation from $k - \omega$ model.
$\varepsilon$	Turbulent dissipation.
$C_{kolm}$	User specified multiple of Kolmogorov scale.
$D$	Ducros sensor
$f$	Frequency.
$k$	Turbulent kinetic energy.
$k$	Turbulent kinetic energy.
$N$	Number of Points.
$n$	User defined exponent used in the shock smoothing function.
$R$	User defined radius used in the shock smoothing function.
$S$	Spacing used by point cloud.
$T$	Temperature.

$u$  Velocity

$y$  Distance from the wall

## **SUBSCRIPTS**

$+$  Viscous Nondimensional

$BK$  Background

$i$  Relating to the  $i$ th cell.

$kolm$  Relating to the Kolmogorov scale

$NF$  Near Field

$sh$  Relating to the shock.

$t$  Turbulent

$wm$  Wall model

## **ACRONYMS**

CFD Computational Fluid Dynamics

DES Detached Eddy Simulation

DNS Direct Numerical Simulation

ERAU Embry-Riddle Aeronautical University

LES Large Eddy Simulation

RANS Reynolds Averaged Navier-Stokes

SBLI Shock-Boundary Layer Interaction

SEM Synthetic Eddy Method

SGS Sub Grid Scale

TKE Turbulent Kinetic Energy

WMLES Wall Modeled Large Eddy Simulation

## 1 Introduction

Computational Fluid Dynamics (CFD) has been an area of interest in the aerospace industry for many years due to its ability to simulate and accurately predict flow across a wide number of applications. In the case of aerodynamics, Reynolds Averaged Navier-Stokes (RANS) equations have been an industry standard approach due to its accuracy in a large number of situations and its relatively fast computation time. However, for situations with complex flow structures such as in propulsive flows, high speed flows, shock-boundary layer interactions (SBLI), and other current areas of industry focus, modeling turbulence through RANS yields inaccurate results. One approach to more accurately predict these complex flows is through Large Eddy Simulation (LES). LES has mainly been a tool reserved for academia due to its accuracy and high computational cost. As industry moves more towards more complex aerodynamic challenges, the difference between RANS and LES becomes greater and greater.

One common approach to reducing the computational cost of LES is the implementation of a Wall Model. Wall modeled LES (WMLES) aims to reduce the cost of LES simulations by modeling the near wall turbulence instead of resolving the turbulence fully within the grid. This saves on overall grid count as well as loosens the restriction on time step due to grid volume. Reviews on the state of the art of wall modeled LES can be found including papers by Piromelli and Balaras [2], Piromelli [3], Larsson et al [4] for general applications. There also exist examples of WMLES with specific application such as the paper relating to jet acoustics by Lyrintzis et al [5].

Some examples of application of WMLES include in the analysis of converging-diverging beveled nozzle jets as presented by Aikens et al. [6], the application of WMLES in high speed flows such as by Mettu et al. [7], and lastly the case used as a benchmark for the remainder of the paper WMLES on a transonic axisymmetric bump as presented by Iyer et al. [8].

Wall models can be put into several categories. One approaches is with a Hybrid

RANS/LES method. An example of this is detached Eddy Simulation (DES). Refer to a review by Spalart [9] for further exploration of the area. More specific approaches involve RANS based calculations blended with the spatially filtered Navier-Stokes equations by an addition of eddy viscosity such as presented by Choi et al. [10] and Giesekeing et al. [11]. Another approach is to blend RANS and LES based on local turbulent length scales as suggested by Shur et al. [12]. The last common method for WMLES is through wall stress modeling. This is the method focused on for the rest of the paper. Wall stress models work by either algebraically or through ODEs solving for a velocity and temperature profile from which to extract a wall shear stress and heat flux. These models are solved through sampling the primitive vector some normal distance from the wall boundary condition in the LES grid, solving the model, and applying these stresses and fluxes as a boundary condition onto the wall. Two examples of an ODE approach to the wall stress model are a model with equilibrium assumptions suggested by Kawaii and Larsson [13], [4], or a non equilibrium assumption suggested by Park and Moin [14]. The former being the model chosen for the rest of the paper.

The ultimate objective of this paper is to broaden the use case of LES by lowering the computational cost. First a wall model was implemented into Eagle3D. The wall model developed by Kawaii and Larsson [13] was chosen, as previously discussed, and validated against experiment and results of a similar simulation performed by Iyer et al. [8]. Additionally, in order to further reduce computational cost, a novel method of lean grid generation is discussed. The objective being to use a RANS solution in order to generate a good grid geared towards WMLES. The combination of lean grid generation with computational savings of WMLES is proposed as a path to lower the threshold for LES in industry.

## 2 Methodology

### 2.1 Eagle3D Flow Solver

Eagle3D is an in house CFD code developed for Embry-Riddle Aeronautical University (ERAU). It is a finite volume multi-element adaptive unstructured grid solver. It is Density based and MPI driven solver developed for general purpose and high speed flows. The solver uses, for this current application, a four stage Runge Kutta time scheme. In order to get Eagle3D to the state of the art, CharLES was used as a benchmark, and based on the schemes and setup described by Iyer et al. [8] updates were made to Eagle3D. For spatial schemes a combination skew symmetric flux scheme layed out in Morris et al. [1], and a shock capturing Harten-Lax-van Leer-Contact (HLLC) low dissipation scheme detailed in Toro et al. [15] were implemented into Eagle3D. The shock capturing is performed via a Ducros sensor [16] blending of the schemes. The spatial accuracy of the schemes is second order and the sub grid scale is modeled implicitly.

A Ducros sensor is applied by comparing the dilatation of velocity and the local vorticity and produces a value between zero and one through the following equation:

$$D = \frac{(\nabla \cdot \vec{u})^2}{(\nabla \cdot \vec{u})^2 + (|\nabla \times \vec{u}|)^2 + \epsilon} \quad (2.1)$$

Where  $\epsilon$  is a small value to prevent divide by zero. The value of D is then used to weight the fluxes between the skew symmetric flux scheme, and the low dissipation HLLC such that HLLC is only active around regions containing shocks as described in equation 2.2.

$$F_{inv} = F_{skew}(1 - D) + F_{HLLC} * D \quad (2.2)$$

Additionally, a Synthetic Eddy Method (SEM), developed by Kopper [17] based off of the model proposed by Jarrin et al. [18], was integrated into Eagle3D. SEM allows for realistic



turbulence to be produced at the inlet instead of needing sufficient run up, trip, filter, or other method to generate the desired turbulence. The SEM code was its own stand alone code that produced input files that were read by Eagle3D every iteration to produce the desired turbulence. In order to streamline simulations and to accommodate the large size of this case, the two codes were integrated such that the SEM is generated on the fly as iterations are run.

## 2.2 Wall Model

### 2.2.1 Wall Model Description

The wall model implemented into Eagle3D was first developed by Kawai and Larsson [13]. The code for the wall model is listed in the appendix. The model consists of two equations, a momentum equation 2.3, and an energy equation with a damping function 2.4.

$$0 = \frac{d}{dy} \left[ (\mu_{wm} + \mu_{t,wm}) \frac{dU_{wm}}{dy} \right] \quad (2.3)$$

$$\frac{d}{dy} \left[ c_p \left( \frac{\mu_{wm}}{Pr} + \frac{\mu_{t,wm}}{Pr_{t,wm}} \right) \frac{dT_{wm}}{dy} \right] = - \frac{d}{dy} \left[ (\mu_{wm} + \mu_{t,wm}) U_{wm} \frac{dU_{wm}}{dy} \right] \quad (2.4)$$

Where:

$$\mu_{t,wm} = \kappa \rho_{wm} \sqrt{\frac{\tau_{w,wm}}{\rho_{wm}}} y \left[ 1 - \exp - \frac{y^+}{A^+} \right]^2 \quad (2.5)$$

$$y^+ = \frac{y \sqrt{\tau_{w,wm} / \rho_{wm}}}{\mu_{wm}} \quad (2.6)$$

$$A^+ = 17, \quad \kappa = 0.41, \quad Pr_{t,wm} = 0.9 \quad (2.7)$$

The model is assumes equilibrium boundary layer ie. parallel flow without acceleration or pressure gradient and a constant shear stress. It also assumes the profile spans from the log layer to the wall (a  $y^+ > 50$  to a height of 10% of the boundary layer thickness). The boundary layer being modeled in this region implies that it's profile holds true irrespective of the complexity of the flow field. More details on the wall model can be found by Kawai

and Larsson [13].

### 2.2.2 Wall Model Implementation

The wall model implementation follows the steps listed in figure 2.1.

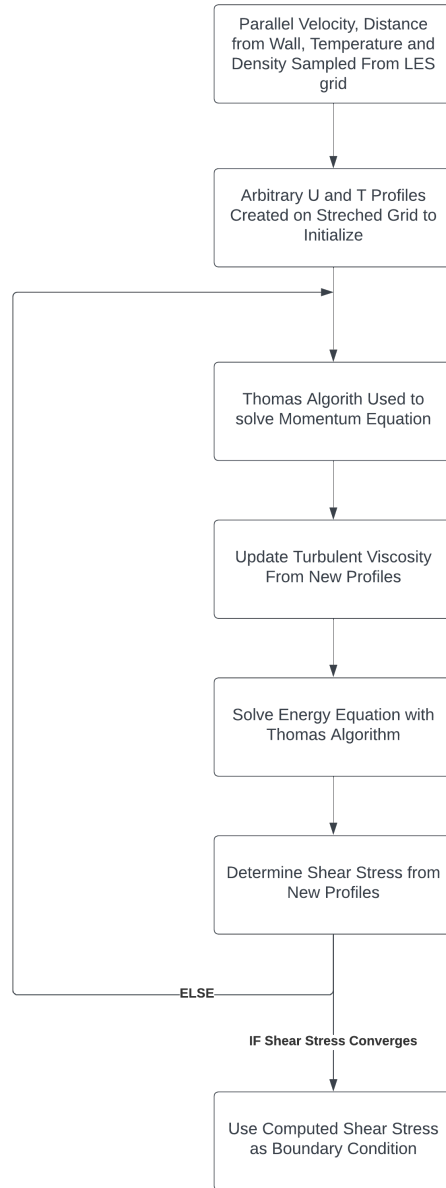


Figure 2.1 Flow chart of the wall models solve process.

Unlike typical wall model implementations that use the first cell height as a sample location, Eagle3D has three options: to use the height of the first hex cell as is standard; to use the height of the first cell center; to search for a cell on the fly to sample from based

off a calculated  $Y^+$  target. The solver will create a list of candidate cells for each wall face during the pre-processing of the grid and, from this list of cells, will interpolate a primitive vector to feed the wall model based on the desired  $Y^+$  target.

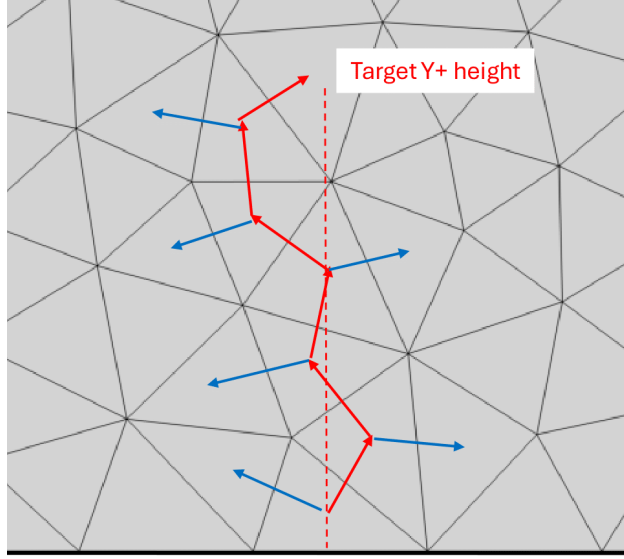


Figure 2.2 Example of how the wall model searches for a list of cells to sample from.

The algorithm used to create the list of candidate cells works regardless of element type. Starting at the first cell of the wall, each connecting cell is evaluated, and the best fit cell is chosen and added to the list. Then the connecting cells of the previously chosen cell are checked and a new cell is chosen and added to the list. This process repeats until a predetermined number of cells is reached. As shown in figure 2.2 best fit cells are represented by the red arrows, and rejected cells in blue. The cells are evaluated by two dot products,

$$\text{Test 1 : } \max(\hat{n}_{wall} \cdot \|\langle (x_i - x_{wall}), (y_i - y_{wall}), (z_i - z_{wall}) \rangle\|) \quad (2.8)$$

$$\text{Test 2 : } 0 < \hat{n}_{wall} \cdot \langle (x_i - x_{prev}), (y_i - y_{prev}), (z_i - z_{prev}) \rangle \quad (2.9)$$

Where  $\hat{n}_{wall}$  is the unit normal vector of the wall face, the  $i$ th cell is the one being tested, and the  $prev$  cell is the previously chosen cell in the list. Test 1 looks for the connecting cell that that is most directly above the wall face, while Test 2 allows that the only possible next

candidate cells are in the wall normal direction as compared to the last chosen cell. The cell with the highest value of Test 1, that also has Test 2 equal to a number greater than zero becomes the chosen cell.

The path taken by the wall search algorithm is shown on two test grids in figure 2.3. Both grids are 2D for simplicity of the visual. One for a tet dominated grid and the other for a hybrid hex and tet grid.

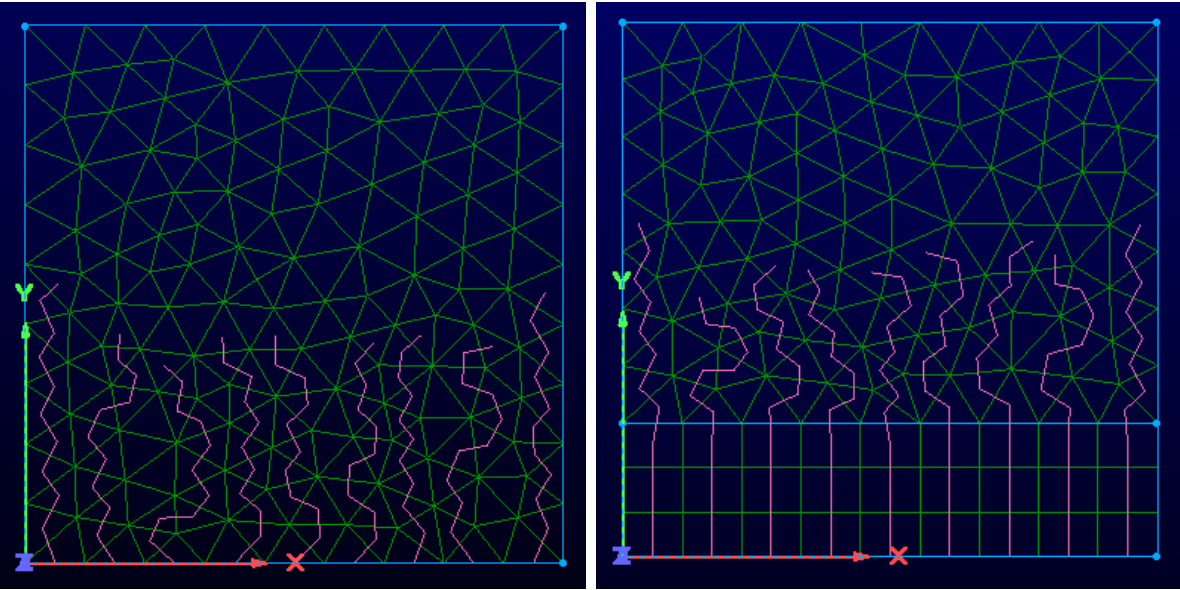


Figure 2.3 Example tet grid with path the wall search algorithm takes.

The pink lines represent the path chosen by the algorithm. The cells along the line are the ones chosen to make up the list of candidate cells. The wall normal distance of each cell is recorded and used to determine which cell will feed the wall model based on the desired  $Y^+$ .

### 3 Validation Of Eagle3D for Wall Modeled LES

#### 3.1 Taylor-Green Vortex

The classic Taylor-Green vortex problem was computed to test the newly modified numerical scheme (i.e., Skew Symmetric + HLLC inviscid flux with Ducros sensor). Eagle3D results are compared with those presented for a low-dissipation 2nd order scheme results by Morris. In this case the solution is initialized with a single large vortex at sea level pressure and temperature and a 3-D wave distribution for x and y velocity components (no z velocity). Figure 3.1 shows that the time history of the normalized Mean Kinetic Energy from eagle3D agrees very well with the Skew Symmetric "CD-2" scheme from Morris. Figure 3.2 shows that the iso-contours of normalized vorticity magnitude on a slice along the  $x=0$  plane at  $t^* = 8$  from Eagle3D and CD-2 also agree quite well. Figure 3.3 depicts Q-criterion shaded by velocity magnitude at  $t^*=8$  from Eagle3D to illustrate the expected rich vortical behavior within the solution.

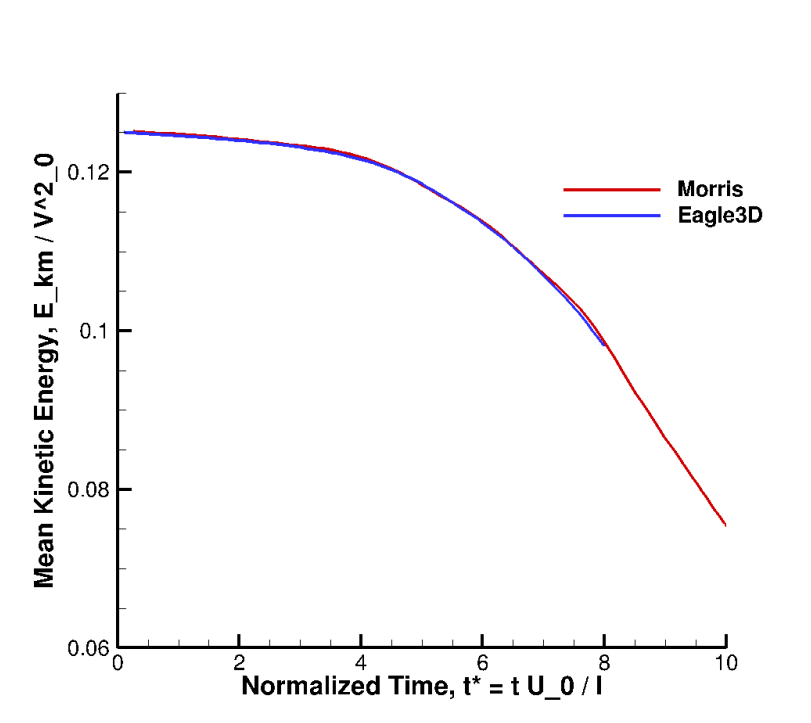


Figure 3.1 Time evolution of mean kinetic energy of Eagle3D as compared to Morris [1].

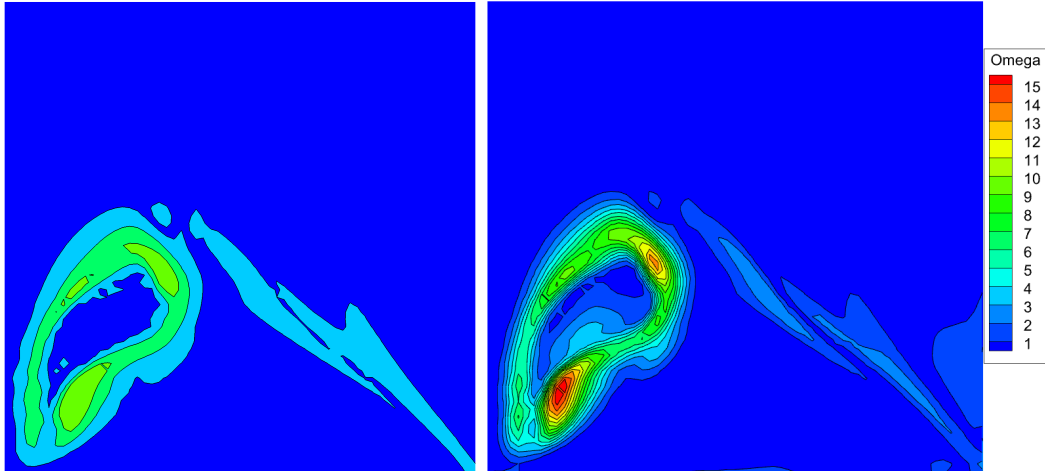


Figure 3.2 Comparison of iso-contours for normalized vorticity magnitude between Morris [1] (left) and Eagle3D (right). Contour levels for the Morris image are 1, 5, 10, 20 and 30.

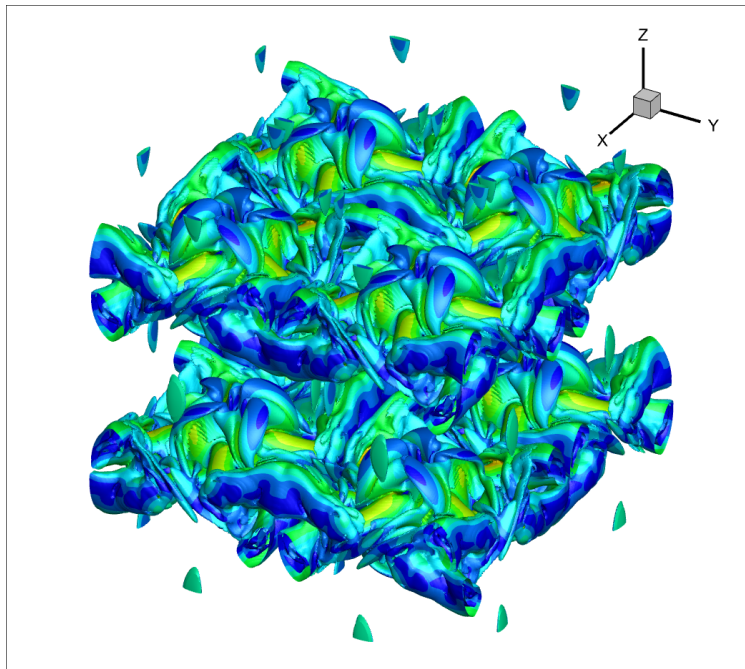


Figure 3.3 Q criterion visualization for  $t^* = 8$ .

### 3.2 Transonic Bump Problem Description

The main validation case chosen for the wall model is an axi-symmetric transonic bump based off the experiment originally run by Bachalo et al. [19]. The experiment took place in a NASA Ames 2ft x 2ft wind tunnel at a Mach number of 0.875. The wind tunnel is a continuously run, closed return, variable density design with 21% open porous slotted upper

and lower walls. The Reynolds number based off the length of the bump is 2.763 million, at a temperature of 300 K, a pressure of 70 kPa. The bump is characterized by a circular arc of length equal to 20.32 cm and a height of 1.905 cm protruding from a cylinder of 15.24 cm in diameter. A boundary layer of approximately 1 cm is formed incident on the bump. A normal shock is formed off the trailing edge of the bump, and a separation region is produced in its wake.

### 3.3 Computational Grid

The main validation grid is based off of the grid used by Iyer et al.[8]. The grid used does not exactly meet the same metrics as used by Iyer. Further grid refinement could allow for a closer match to their results. The unstructured grid contains approximately 55 million cells. The grid is axisymmetric with uniform spacing in the azimuthal direction containing 180 cells across a 30 degree rotation.

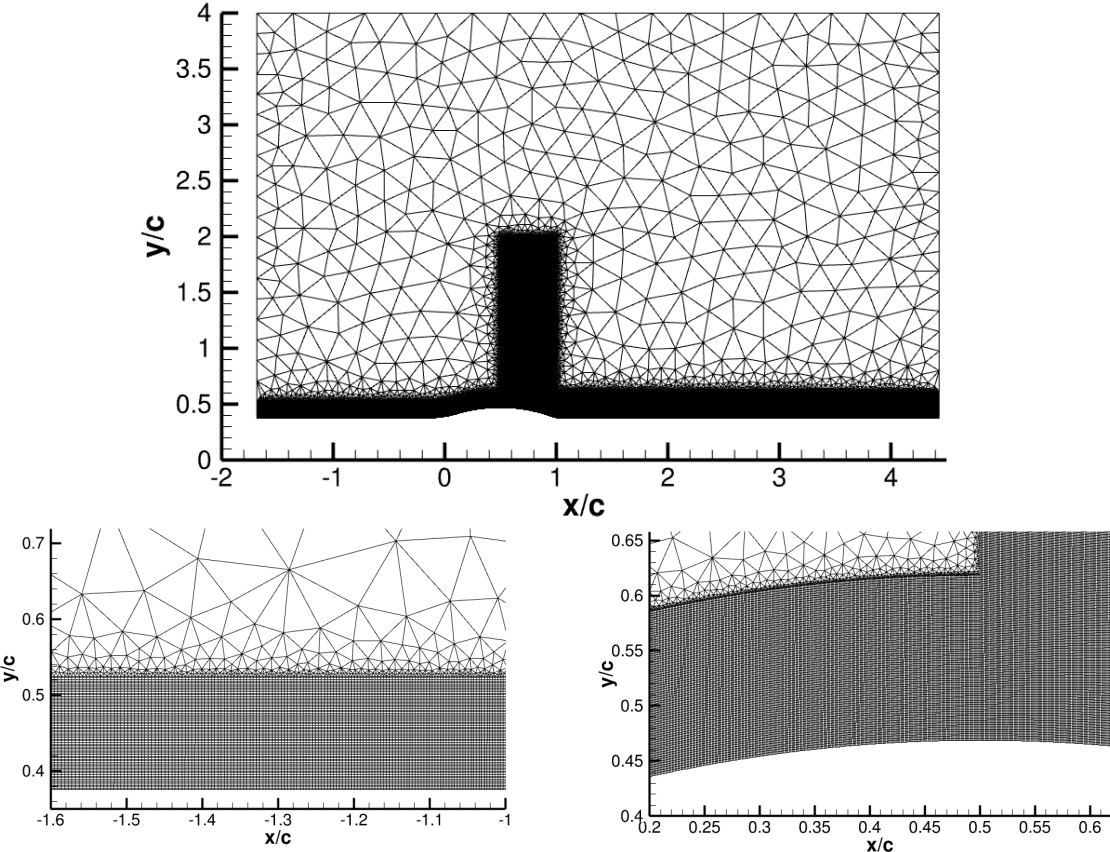


Figure 3.4 Grid used for the in the simulation of the axisymmetric bump.

*Table 3.1* Grid spacings at various locations used in the simulation of the axisymmetric bump. Boundary layer thickness  $\delta$  is assumed to be 1 cm, and  $c = 20.32$  cm.

$x/c$	$\Delta x/c$	$\Delta y/c$	$\Delta z/c$	$\Delta x+$	$\Delta y+$	$\Delta z+$	$Nx/\delta$	$Ny/\delta$	$Nz/\delta$
-1.5	4.9E-03	2.0E-03	1.1E-03	532	220	119	10	24	45
-0.5	2.5E-03	2.0E-03	1.1E-03	273	220	119	20	24	45
0	1.7E-03	2.0E-03	1.1E-03	180	220	119	30	24	45
1	1.2E-03	3.3E-03	1.1E-03	134	353	119	40	15	45

A Synthetic Eddy Method (SEM) inflow boundary condition is used at the inflow section close to the wall as originally described by Jarrin et al. [18] and implemented into Eagle3D by Kopper et al. [17]. The profile entered was determined by a flat plate RANS simulation with height chosen such that the resulting boundary layer is approximately 1cm tall when impinging on the bump. Above the SEM region there is a velocity inlet set with a  $U_\infty = 303$  m/s. The upper boundary of the mesh has a boundary condition of pressure farfield, and the back portion uses a pressure outlet. The Bump and cylinder bottom section utilizes the adiabatic wall model described in previous section. Results of this validation are found in the Results section 5



## 4 Grid Generation for Wall Modeled LES

### 4.1 Problem Introduction

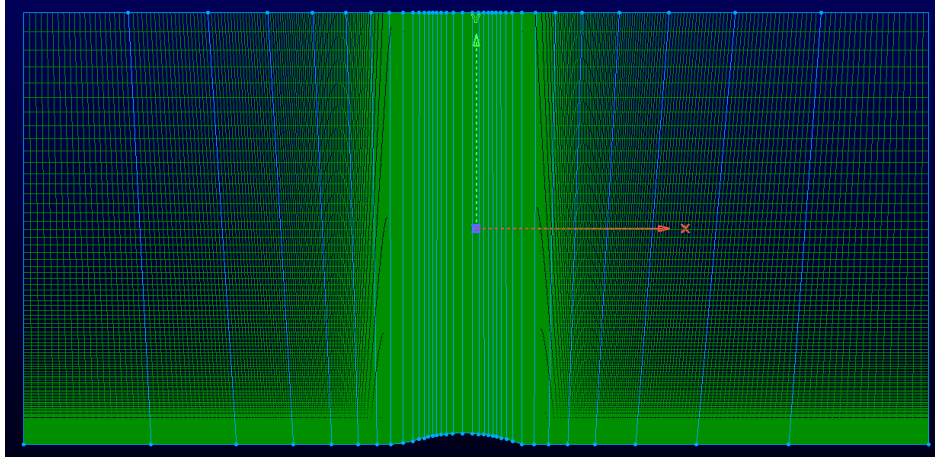
One of the challenges of LES is its reliance on a good grid. Many problems that benefit with the accuracy of LES require a large amount of grid to fully resolve flow structures and the effect of real turbulence. Many of these grids take a long time to develop, with the time taken to iterate through new grids quickly scaling with the size of the case.

It is common for many LES cases to be initialized using a RANS solution in order to speed up the computation time needed to get a starting solution. The RANS solution, while lacking aspects of the flow that LES can capture, has enough information contained within it to infer a grid for LES. If a grid could be generated for wall modeled LES based off of a RANS solution, LES may become more attainable to industry. The automation of the LES grid generation would greatly cut down on computational costs, as well as time spent developing a lean grid for WMLES. The use of the wall model would additionally greatly cut down on computation time and grid requirements as compared to wall resolved LES. As well, the methods to be explained could theoretically be used to generate grid suitable for wall resolved LES, or even for Direct Numerical Simulation (DNS).

### 4.2 RANS Case

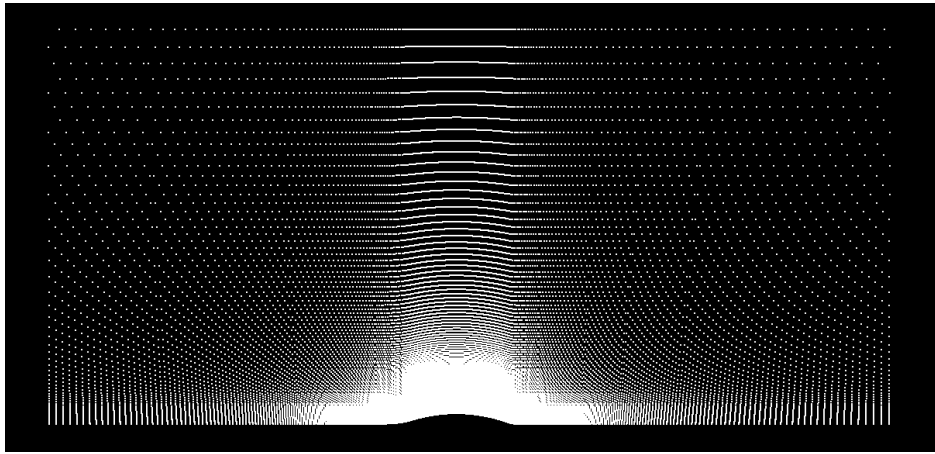
The case being demonstrated on is the same Transonic Bump problem described in previous sections for validation of the Eagle3D wall model. The RANS grid consists of 230,400 cells and spacing off the wall has a  $Y^+ < 1$ . The Grid is one cell thick in the z direction and structured in the x and y direction with 1,400 and 320 cells in each respective dimension. The RANS grid extends from approximately  $-3 x/c$  to  $4.5 4x/c$  in x and  $0 x/c$  to  $4 x/c$  in the y direction, which is larger domain as run on the Eagle3D LES validation case. The grid is non-dimensional as defined by a unit of one equal to the length of the bump.

From this RANS case, the grid is read into the grid generator, along with the primitive vector of  $\langle \rho, u, v, w, k, \omega \rangle$ . For each cell a point is generated at the respective cell center as shown in the comparison of figure 4.1 to figure 4.2. For each point a spacing is to be



*Figure 4.1* RANS grid used for LES grid generation

assigned from which a grid can be generated. The spacing will be defined based off of a number of factors including acoustic considerations, local Kolmogorov scale, distance from the nearest wall, and distance and relation to a shock.



*Figure 4.2* Point cloud Generated from RANS grid.

### 4.3 Shock Sensing

The first step taken by the grid generator is to identify which cells contain a shock, and to flag them. Shock sensing is performed by the method developed by Ducros et. al. [16] similar to as described in previous section, where velocity dilatation is compared to the vorticity present in each cell as is shown below.

$$D = \frac{(\nabla \cdot \vec{u})^2}{(\nabla \cdot \vec{u})^2 + (|\nabla \times \vec{u}|)^2 + \epsilon} \quad (4.1)$$

Where  $\epsilon$  is a small value to prevent divide by zero. The grid generator uses the value of  $D$  to determine which cells get flagged for containing shocks. For the case of this example problem, cells with a value of  $D \geq 0.99$  are flagged as shown in figure 4.3.

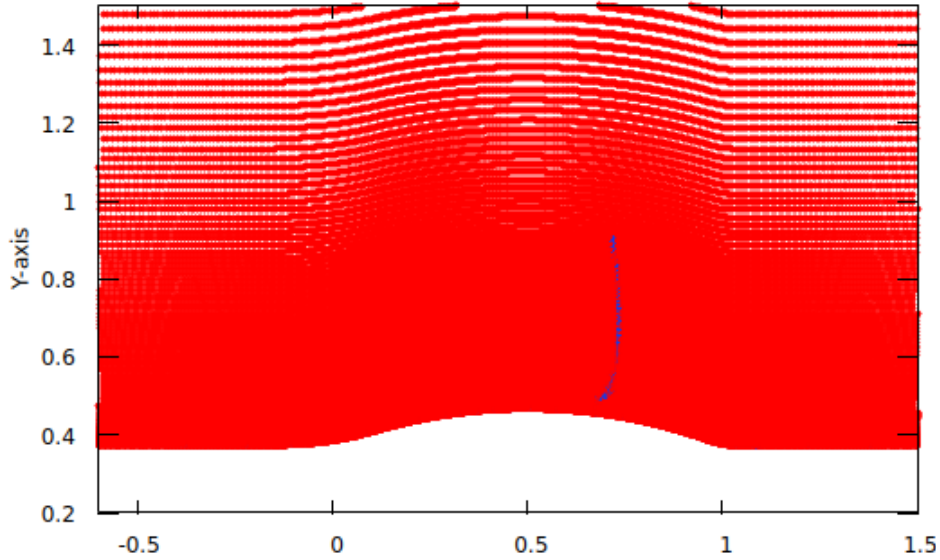


Figure 4.3 Cloud points flagged by the ducros sensor in blue.

The cells flagged are assigned a point spacing specified by the user. For this example case, the corresponding points are assigned a spacing of 0.005 non-dimensional units.

#### 4.4 Near-Field/Background Acoustic Sizing

The next step taken by the grid generator is to evaluate spacing based on acoustic near field and background resolution. The user specifies four values for the acoustic spacing, a desired near field frequency, a back ground frequency, and the number of points used to resolve each frequency. For this example case the near field was set to resolve acoustic waves of 10 kHz using 15 points per wavelength, and the background resolution was set to resolve 400 Hz using 15 points per wavelength. The following equation is used to calculate the grid

spacing each point:

$$S_{NF} = \frac{\sqrt{\gamma RT}}{f_{NF} * N_{NF}} \quad (4.2)$$

$$S_{BK} = \frac{\sqrt{\gamma RT}}{f_{BK} * N_{BK}} \quad (4.3)$$

Whether the background or near field spacing is used is determined by the turbulent kinetic energy (TKE)  $k$ . A threshold value is set, for the case of this example  $k = 1$  is used, to check if the TKE at each cell is above or below the threshold. If it is below the threshold the spacing for the corresponding point is set to the background spacing, if it is above, the near field spacing is used. Currently there is no blending or smoothing performed. Future work suggested would include modifying the generation technique to assure that the grid does not grow faster than a specified growth rate of 1.2 or 1.15, for example.

#### 4.5 Grid Sizing from Turbulence Parameters

The next step taken is to size the grid based on the turbulence present in the solution. The spacing gets determined based off a user specified multiple of the local Kolmogorov scale. The Kolmogorov scale can be estimated from the  $k - \omega$  RANS model using the following equations:

$$\varepsilon = 0.09 * k\omega \quad (4.4)$$

$$\eta = \left( \frac{\mu^3}{\varepsilon} \right)^{\frac{1}{4}} \quad (4.5)$$

$$S_{kolm} = C_{kolm} * \eta \quad (4.6)$$

The value of  $C_{kolm}$  for this example case is set to 100. This roughly lines up such that the minimum value of  $S_{kolm}$  is equal to the minimum value of the height corresponding to a  $Y^+ = 40$  present in the grid. The same TKE check used for the near field/background

spacing is applied for the Kolmogorov spacing. If the cell is determined to be in a near field region, the code will use the spacing that is smaller between the near field determined spacing, and the Kolmogorov spacing.

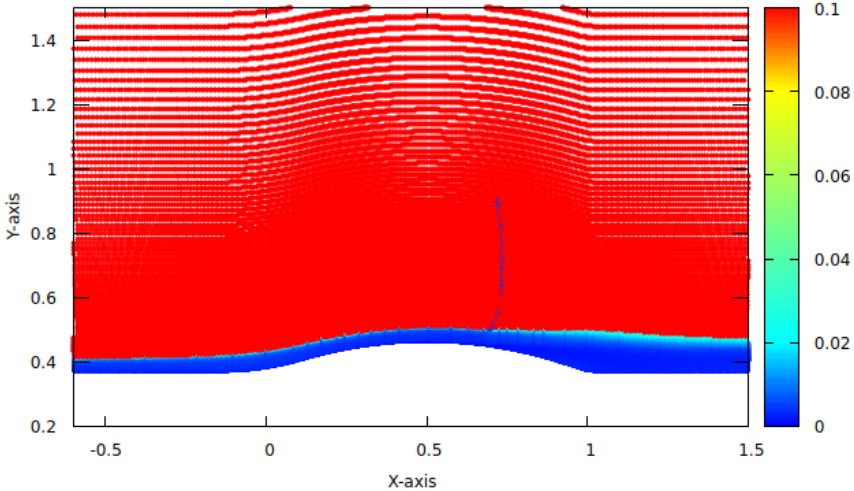


Figure 4.4 Cloud points heat map based on assigned spacing.

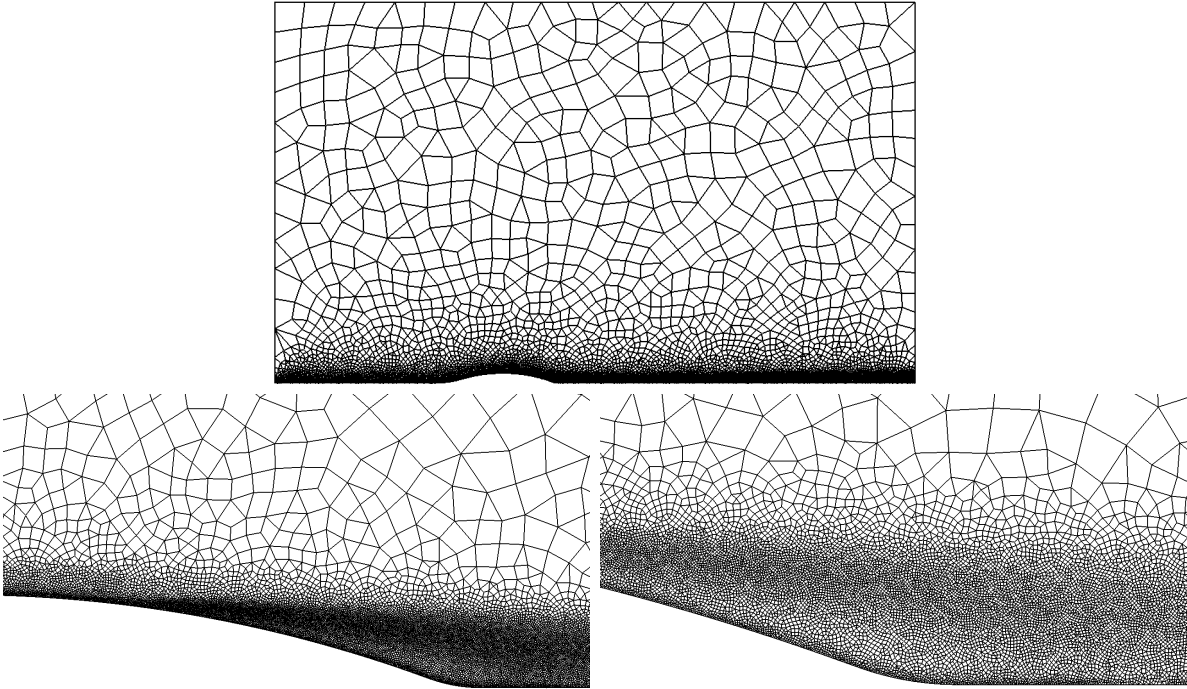


Figure 4.5 Grid generated by near field/background spacing and Kolmogorov scale.

## 4.6 Wall Spacing

Wall spacing is an important consideration for WMLES. A large amount of the computation time savings come from a larger smallest cell volume, and therefore a larger time step. The grid generation procedure needs to take into account the wall spacing for this reason and also because the wall spacing affects the results generated by the wall model. To do this, a user specifies a target  $Y^+$  (in this case a target of  $Y^+ = 40$  is used). The code will iterate through each wall face looking for cells normal to the current face, and the  $y^+$  of each cell center is calculated based on RANS flow solution. If the calculated  $y^+$  is less than the  $y^+$  target, the code will increase the wall spacing value to the larger  $y^+$  target. Based on the  $y^+$  values, the code calculates a corresponding wall spacing value. This new “wall spacing” becomes a new constraint on the minimum cell center spacing to be compared to the constraints given in previous steps.

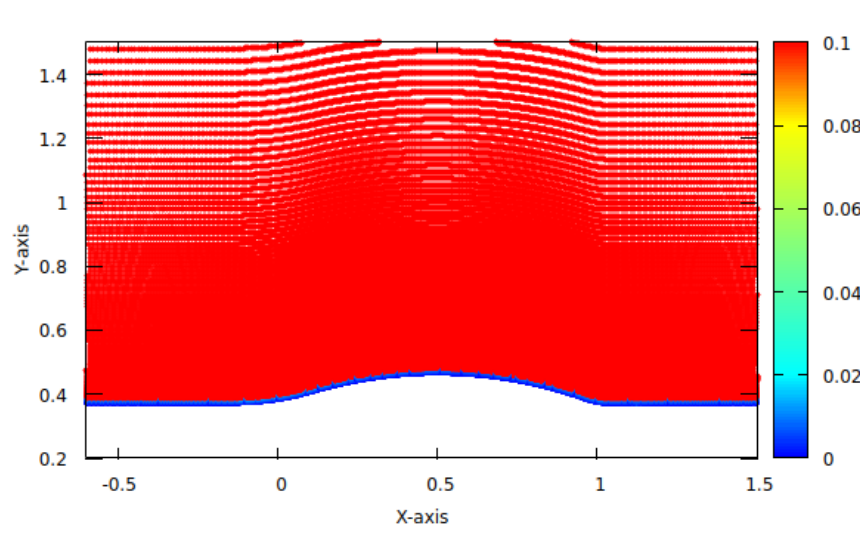
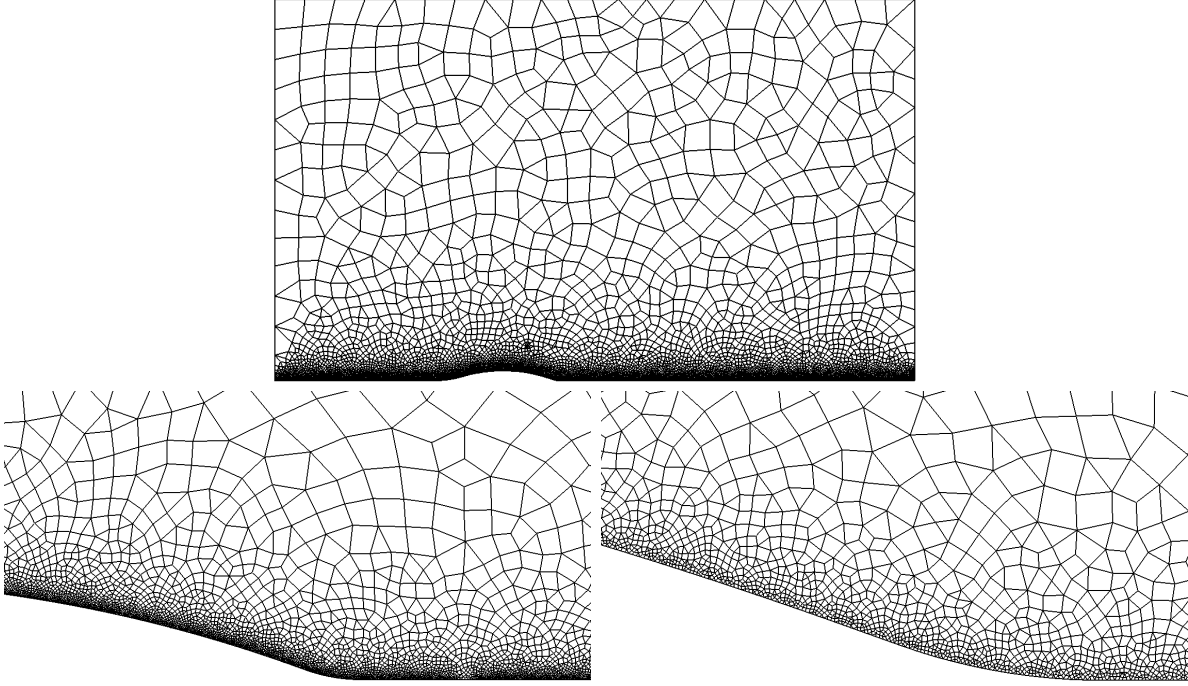


Figure 4.6 Spacing generated by only considering wall  $y^+$ .



*Figure 4.7* Grid generated only considering wall  $y+$ .

#### 4.7 Smoothing of Grid Around Shocks

The final step that the code applies is the smoothing of spacing around shocks. The smoothing is based off of the distance of each cell to the nearest Ducros flagged cell. A radius is set by the user to determine the distance around the shock high resolution cells should be generated, as well as the order of the decay function used. The following equation determines the spacing:

$$S_i = (S_{ff,i} - S_{sh} + 1)^{\frac{x_i^n}{R_{sh}^n}} + S_{sh} - 1 \quad (4.7)$$

Where the user specifies  $n$  the exponent magnitude,  $R_{sh}$  the aforementioned shock radius,  $S_{sh}$  the spacing assigned to the shock. Parameters found by the program include  $x_i$  the distance from the  $i$ th cell to the nearest Ducros flagged cell, and  $S_{ff,i}$  the background spacing calculated at the  $i$ th cell.

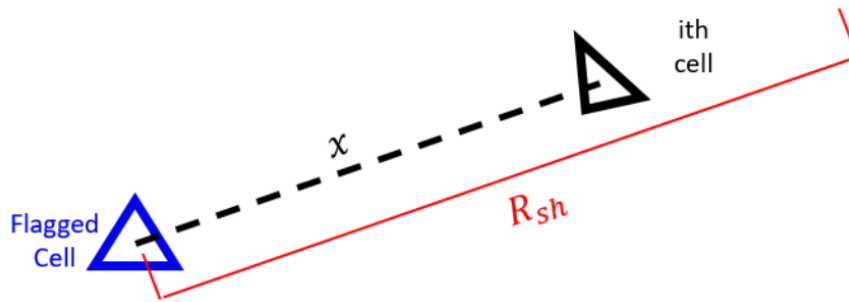


Figure 4.8 A diagram showing the distance from a Ducros flagged cell and the  $i$ th cell

The user selected exponent  $n$  determines the shape of the smoothing function used as shown by figure 4.9. The smoothing performed is independent of the cell order.

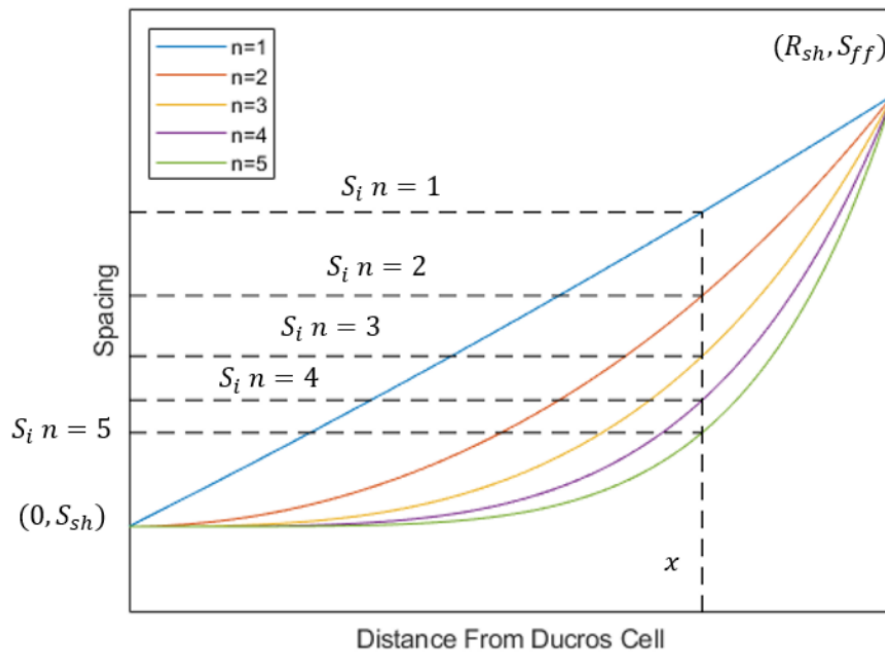


Figure 4.9 The shape functions of the grid smoothing dependent on  $n$ .

The final grid is a combination of all the rules from the previous sections. The final spacing and grid is shown in figures 4.10 and 4.11.

The smoothing effect produced by equation 4.7 in the boundary layer as shown in figure 4.10 is a consequence of a bug in the Ducros sensor code. This bug created Ducros flagged cells along the wall such that the smoothing algorithm was executed along the wall. For this grid, the spacing assigned to the shock was larger than the spacing determined by the



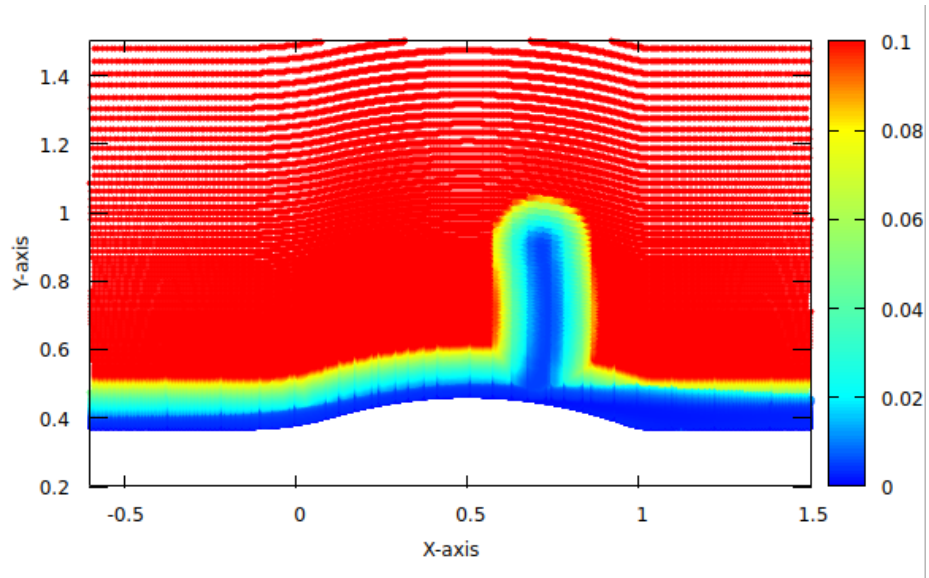


Figure 4.10 Cloud points of final spacing.

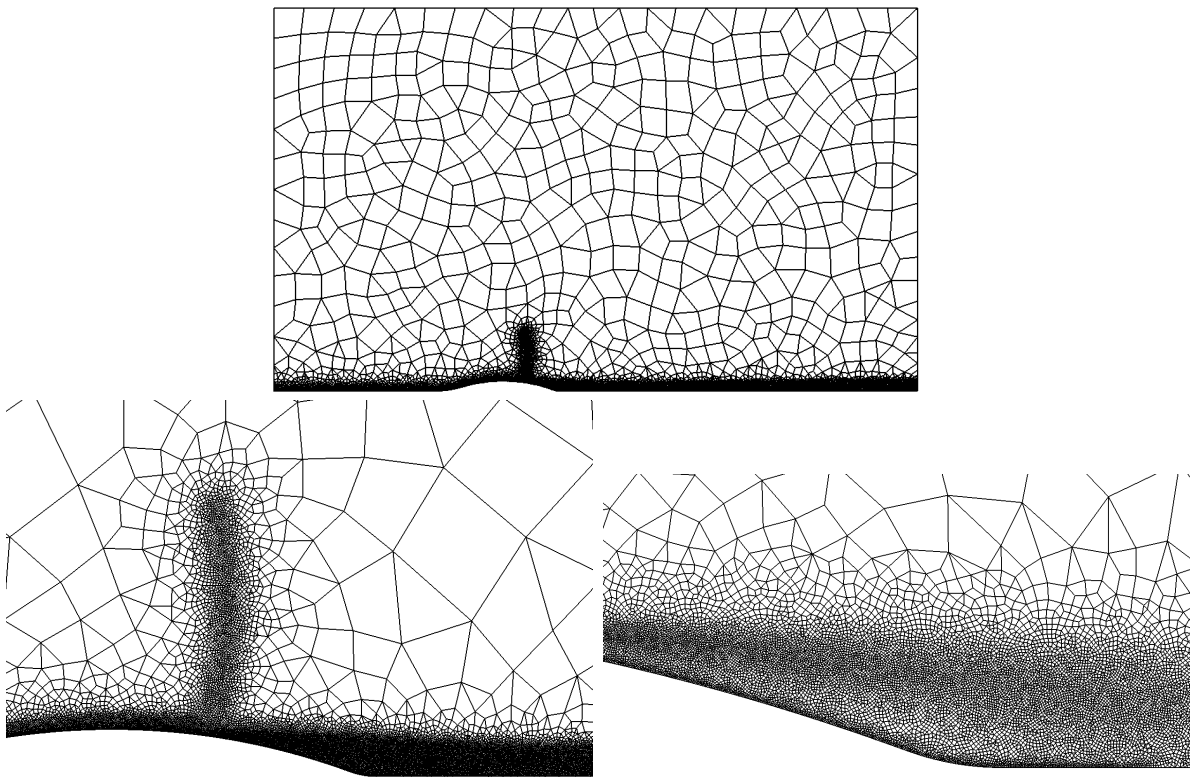


Figure 4.11 Final grid generation.

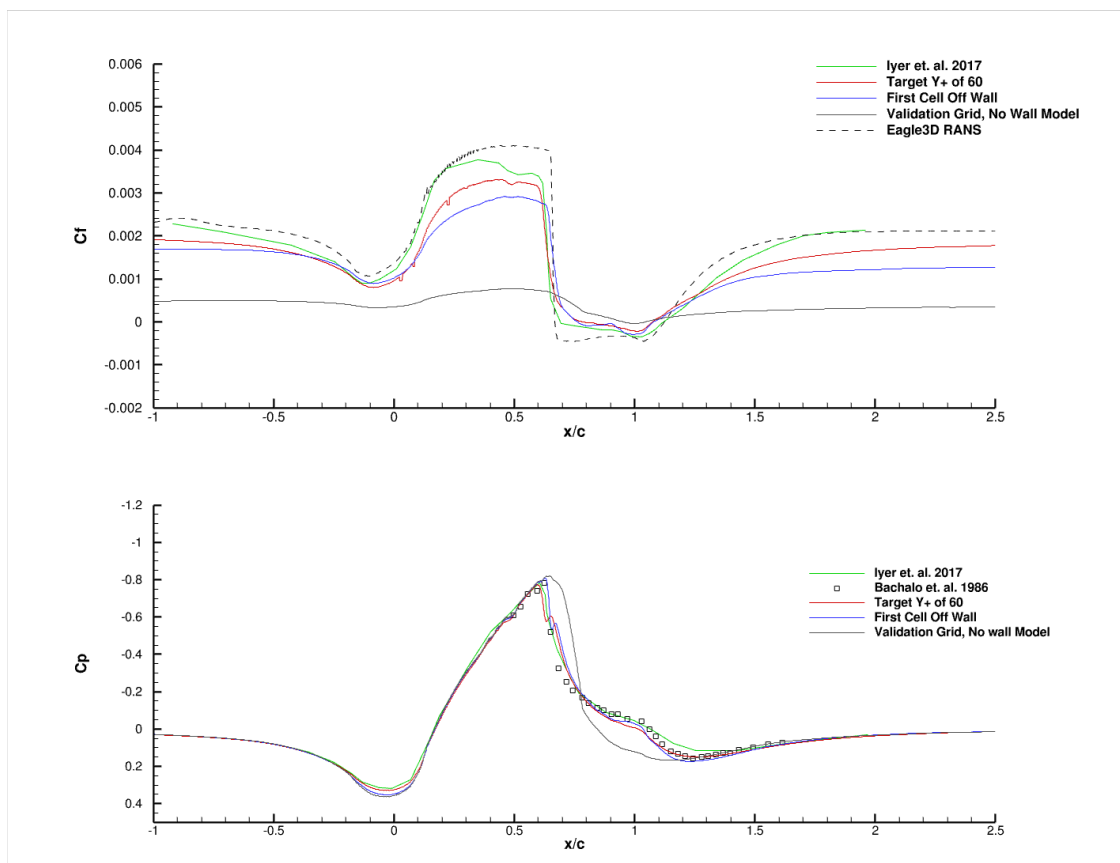
Kolmogorov scale, so the boundary layer resolution is still determined without effect of the shock smoothing. The bug has since been fixed.

## 5 Results

### 5.1 Validation Results

Several cases were run to determine the effects of the wall model. The first case run chose a  $Y^+$  target of 60 from which to sample. The second case samples the face opposite the wall face within the first cell off the wall, and the third was with the same grid with no wall model active. These results were compared with a RANS case run in Eagle3D, as well as the results generated from Iyer et al. [8] and the experiment performed by Bachalo et al. [19] as shown in figure 5.1.

All LES runs done in Eagle3D were run transient for six flow through times, then averaged temporally and axi-symmetrically for two flow through times.



*Figure 5.1* Skin friction and pressure coefficient plots for validation cases.

There is no experimental data for the coefficient of friction, so the results generated by Iyer et al. are taken as a baseline. The shape of the  $C_f$  for the runs utilizing the wall model

are consistent with Iyer et al. However, the values are highly dependent on the incoming boundary layer and the height chosen by the wall model as shown by the difference between the red and blue line. More iterations of incoming boundary layer and  $Y^+$  target are needed to dial in a matching result. Even with a poor match of boundary layers the  $C_p$  agrees well.

Present on the red curve are small regions of discontinuous dropouts. This is due to an issue with the algorithm for determining sample cells and the partition of the mesh. The issue has since been resolved in the code and the dropouts do not effect the result to a high degree.

For most cells, a  $Y^+$  of 60 is somewhere between the wall and the height of the first cell. This means that the wall model is interpolating lower for the red curve than the blue line. This seems to suggest that sampling at  $Y^+ > 60$  targets drive the  $C_f$  lower. More cases and evaluation need to be run to come to this conclusion.

Additionally, no wall model yields highly dissipated results, as expected. Interestingly, with the SGS and wall model turned on, similar results are yielded both in  $C_f$  and  $C_p$ . A guess as to why would be that the grid is course enough that the SGS determines a large  $\mu_t$  to dissipate the mesh enough that it overcomes the wall model.

Figures 5.2 through 5.5 show results from the  $Y^+$  target of 60 run. Figure 5.2 shows a representation of Q criterion of the turbulence impinging, on, and due to the separation generated by the bump. The Q criterion isosurfaces are colored by the velocity nondimensionalized by the freestream. Along with this, a visualization of the shock structure is superimposed onto the bump. This visual is made from the derivative of density. A clear lambda structure is visible as well as regions of negative velocity showing separation occurring downstream of the shock.

Figure 5.3 shows the acceleration caused by the geometry of the bump and the resulting velocity jump and separation caused by the shock. A more detailed view is present in figure 5.4 in the form of Mach number.

In figure 5.5 the lambda shock structure is shown in more detail. The experiment predicts

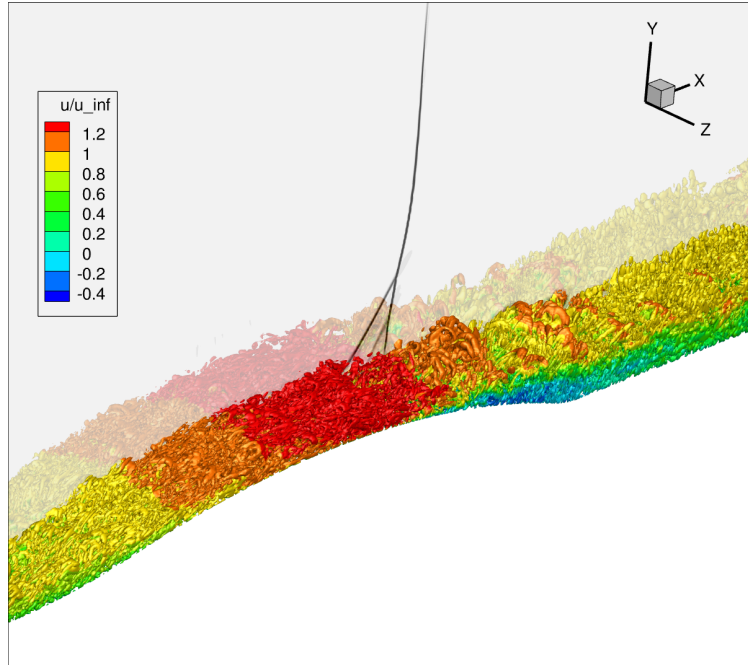


Figure 5.2 Q criterion visualization of the  $Y^+$  target of 60 run

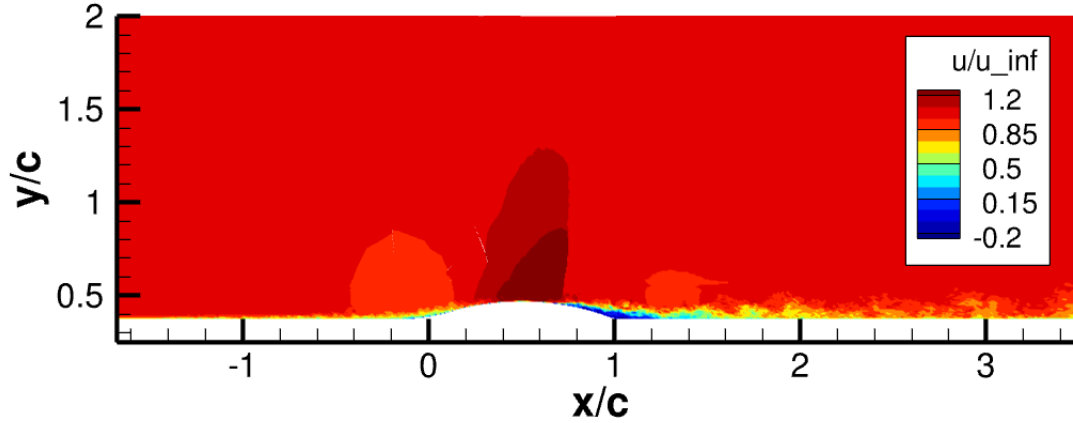


Figure 5.3 Nondimensional velocity plot of the  $Y^+$  target of 60 run

shock impingement location at a value of  $x/c = 0.7$  which is slightly ahead of the shock location shown in figure 5.5. This could be due to the natural unsteadiness of the shock as well as the fact that the incoming boundary layer has not been exactly tuned to the experiment.

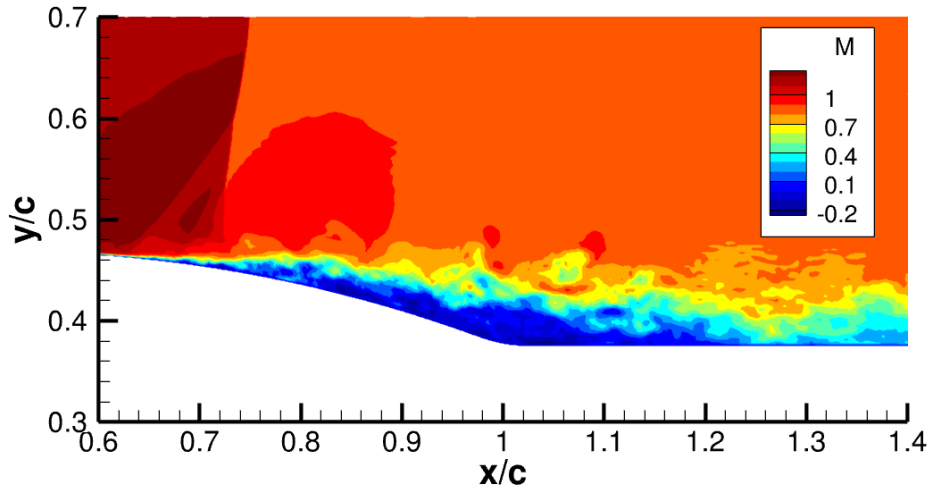


Figure 5.4 Mach number of the  $Y^+$  target of 60 run

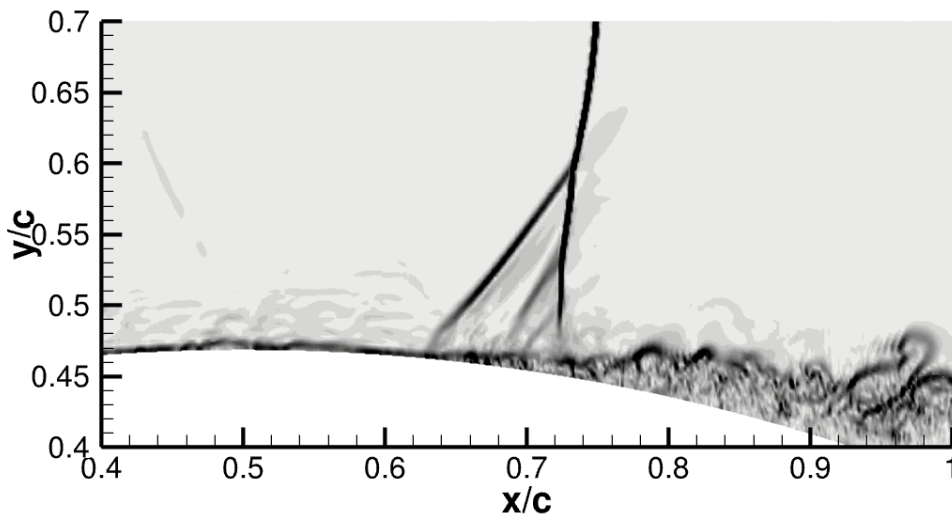
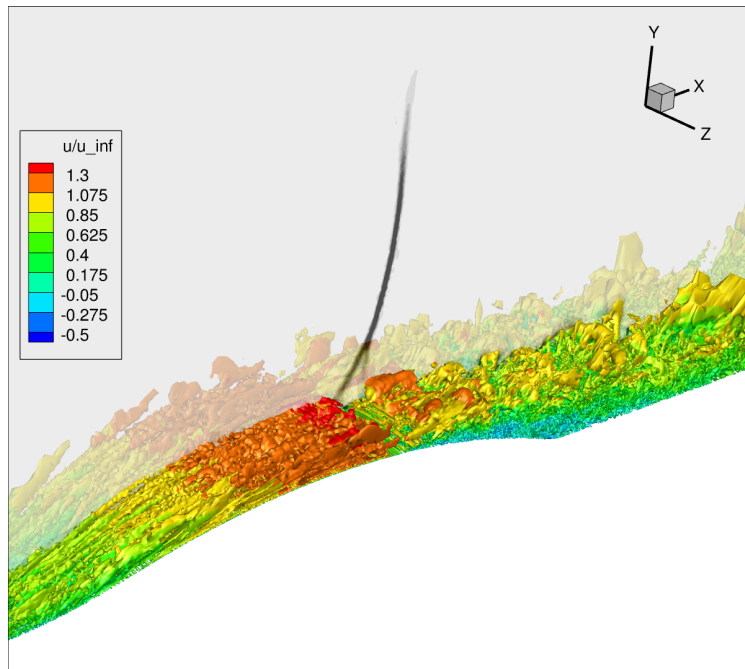


Figure 5.5 Gradient of density of the  $Y^+$  target of 60 run

## 5.2 Validation of WMLES Results for Generated Grid

The validation of the grid generated case is still in preliminary steps and, due to time constraints, has only been run to ten percent of a flow through time. Further work will be performed in a future paper to address the problems stated with the current generated grid and it will be run completely to fully analyze the validity of the method. This limited run time should be considered for this evaluation and little legitimacy given to the results as

they are not fully indicative of the true solution. However, for the purpose of this thesis, these early results show enough promise to warrant the further research into this area.



*Figure 5.6* Q criterion visualization of the  $Y^+$  target of 60 run with generated grid

Figure 5.6 shows a flow visualization of the turbulent structures as well as a visualization of the shock. As compared to figure 5.2, the turbulent structures in the upper part of the boundary layer seem significantly larger, but the lower layers of turbulence are much better resolved. This is thought to be due to the harsh change in cell size, and as eddies convect out of the generated resolution, they are dissipated by the courser resolution. A view of this can be found in figure 5.7. In a later version of the grid generation code smoothing is preformed to insure more grid is present in region above the average boundary layer to capture these transient effects. Despite this, figure 5.7 shows clearly the difference in grid size between the validation case and the generated grid.

The preliminary skin friction and pressure coefficients is shown in figure 5.8. These results are to be minimally acknowledge due to the small run time. The baseline experiment and reference case by Iyer et al. [8] are compared against two test cases and the validation cases. The first grid generation case uses the same grid as the second with the only difference being

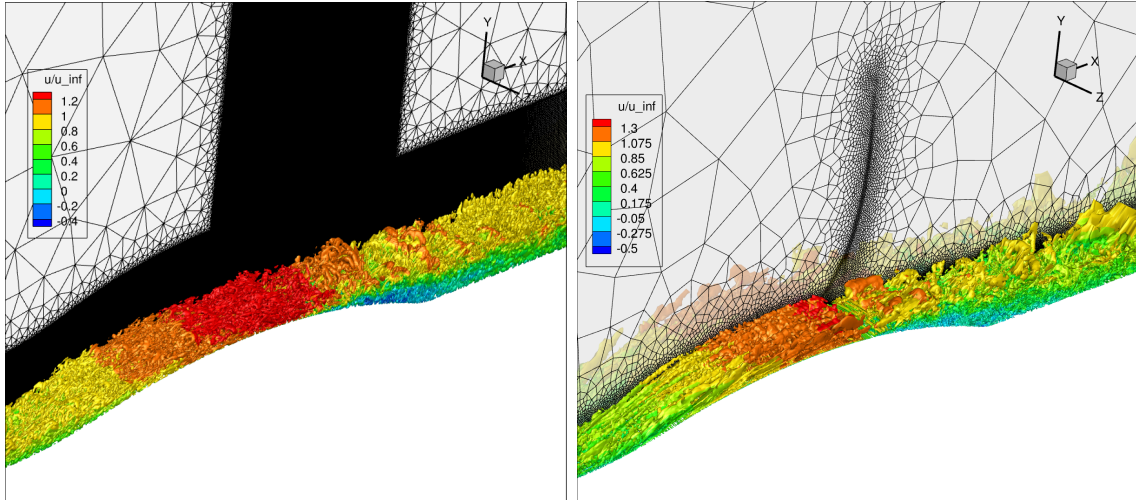


Figure 5.7 Q criterion visualization comparison with grid between validation grid (left) and generated grid (right).

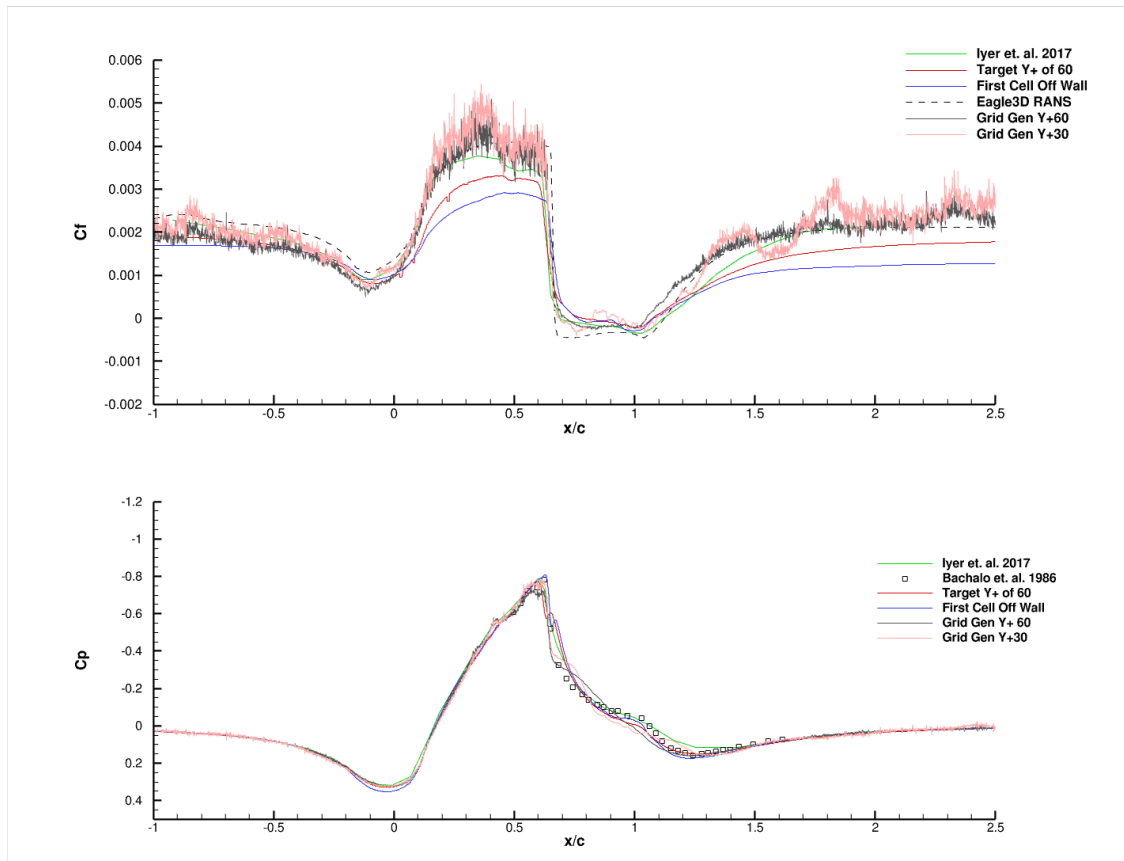
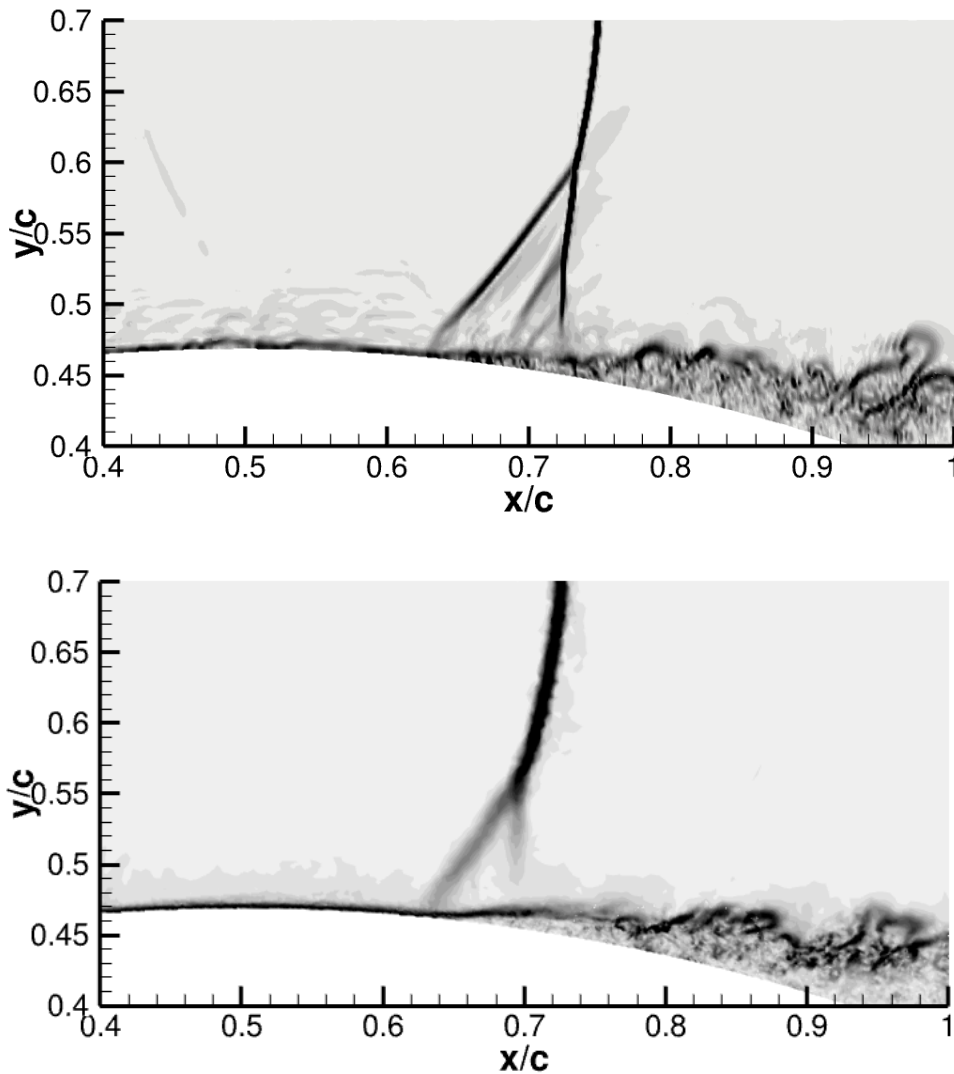


Figure 5.8 Skin friction and pressure coefficient plots for grid generation case.

the selected  $Y^+$  target used on the wall model. The  $Y^+ = 60$  case was run to 10% of a flow through while the  $Y^+ = 30$  case was run closer to 5%.





*Figure 5.9* Gradient of density comparison of validation case (top) vs grid generation case (bottom).

Another point of note is the significantly lower resolution capturing the normal shock and its lambda structure. This is a result of the input settings chosen. A finer resolution grid will be present in future work. Figure 5.9 shows a comparison between the density gradients of the validation case and the generated grid. The shock location and height of interaction differ slightly, which is also most likely due to the lack of resolution. Although the shock is under resolved, the turbulence in the separation region is more resolved with a finer grid present than on the validation case.

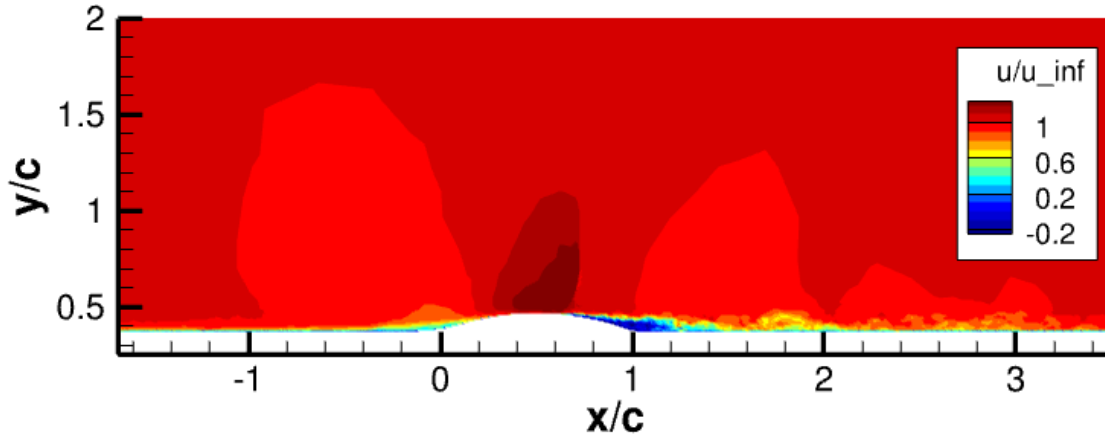


Figure 5.10 Nondimensional velocity plot of the  $Y^+$  target of 60 run with generated grid.

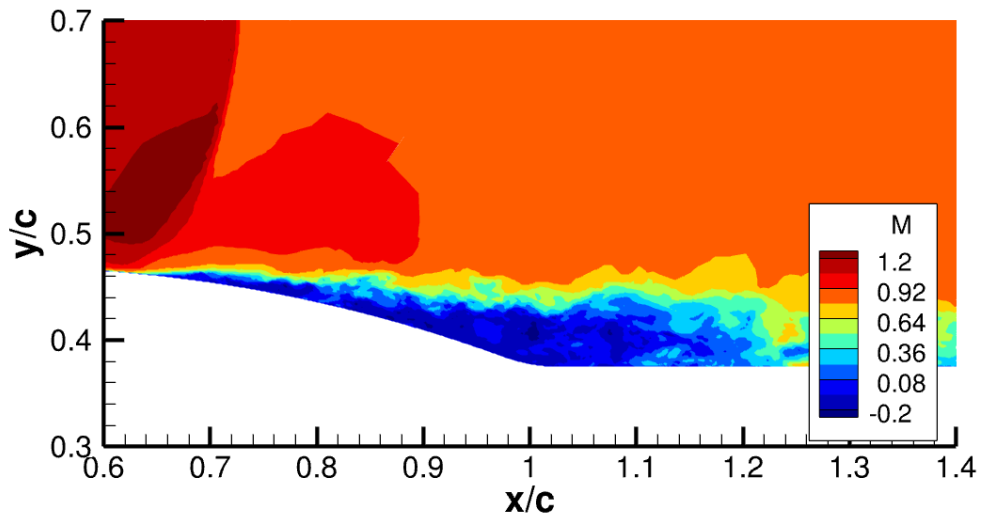


Figure 5.11 Mach number of the  $Y^+$  target of 60 run with generated grid.

Figures 5.10 and 5.11 show contours of the velocity field and Mach number. The shock and separation region are still present. The solution needs to further develop to extract more useful information. However, the bulk flow showing similar features to the validation case indicate that the generated grid may still be able to resolve the proper features.

## 6 Conclusion and Future Work

Eagle3D was found to meet the state of the art for WMLES with the addition of an equilibrium wall model, comparable flux schemes to other standard solvers, and the integration of SEM. The capabilities of the solver were grown past the state of the art as methods of grid generation were preliminary explored. While the early results on the generated grid show promise, more research and run time is required to draw accurate conclusions as to the usefulness of the method.

Going forward, several cases are proposed to cover more aspects of the grid generation method and validation of the Eagle3D solver. Firstly, the generated grid case needs to be run to completion. Secondly, the effect of  $Y^+$  target should be studied to determine the sensitivity of the results to the chosen target. A grid generation sensitivity study should also be run with the effect of Kolmogorov scale coefficient, the chosen acoustic spacing, the shock spacing, and the effect of tetrahedral vs hexahedral grid should be weighed and studied.

An additional feature of grid smoothing should be added to the grid generation. This should ensure that a growth rate of chosen value is enforced throughout the grid to account for abrupt changes in the grid.

Lastly, it is suggested that cases involving heat transfer should be explored with the wall model. One of the possible advantages of the wall model is to capture heat transfer due to the convection of the eddies, an aspect lost with RANS calculations causing poor agreement to experiment for complex flow fields.

With more validation of the capability and accuracy of grid generation for WMLES the technique could be more seriously considered as a method useful to industry and for lowering the cost of LES in general.

## REFERENCES

- [1] Morris, C., “Studies of Inviscid Flux Schemes for Acoustics and Turbulence Problems,” 2013. <https://doi.org/10.2514/6.2013-75>, URL <https://arc.aiaa.org/doi/abs/10.2514/6.2013-75>.
- [2] Piomelli, U., and Balaras, E., “WALL-LAYER MODELS FOR LARGE-EDDY SIMULATIONS,” *Annual Review of Fluid Mechanics*, Vol. 34, No. 1, 2002, pp. 349–374. <https://doi.org/10.1146/annurev.fluid.34.082901.144919>, URL <https://doi.org/10.1146/annurev.fluid.34.082901.144919>.
- [3] Piomelli, U., “Wall-layer models for large-eddy simulations,” *Progress in Aerospace Sciences*, Vol. 44, No. 6, 2008, pp. 437–446. <https://doi.org/https://doi.org/10.1016/j.paerosci.2008.06.001>, URL <https://www.sciencedirect.com/science/article/pii/S037604210800047X>, large Eddy Simulation - Current Capabilities and Areas of Needed Research.
- [4] Larsson, J., Kawai, S., Bodart, J., and Bermejo-Moreno, I., “Large eddy simulation with modeled wall-stress: recent progress and future directions,” *Mechanical Engineering Reviews*, Vol. 3, No. 1, 2016, pp. 15–00418–15–00418. <https://doi.org/10.1299/mer.15-00418>.
- [5] Lyrantzis, A. S., and Coderoni, M., “Overview of the Use of Large-Eddy Simulations in Jet Aeroacoustics,” *AIAA Journal*, Vol. 58, No. 4, 2020, pp. 1620–1638. <https://doi.org/10.2514/1.J058498>, URL <https://doi.org/10.2514/1.J058498>.
- [6] Aikens, K. M., Blaisdell, G. A., and Lyrantzis, A. S., “Analysis of Converging-Diverging Beveled Nozzle Jets Using Large Eddy Simulation with a Wall Model,” 2015. <https://doi.org/10.2514/6.2015-0509>, URL <https://arc.aiaa.org/doi/abs/10.2514/6.2015-0509>.
- [7] Mettu, B. R., and Subbareddy, P. K., “Wall-Modeled Large Eddy Simulation of High

- Speed Flows,” *AIAA Journal*, Vol. 60, No. 7, 2022, pp. 4302–4324. <https://doi.org/10.2514/1.J061501>, URL <https://doi.org/10.2514/1.J061501>.
- [8] Prahladh S. Iyer, G. I. P., and Malik, M. R., “Wall-Modeled Large Eddy Simulation of Transonic Flow over an Axisymmetric Bump with Shock-Induced Separation,” *23rd AIAA Computational Fluid Dynamics Conference*, 2017.
- [9] Spalart, P. R., “Detached-Eddy Simulation,” *Annual Review of Fluid Mechanics*, Vol. 41, No. 1, 2009, pp. 181–202. <https://doi.org/10.1146/annurev.fluid.010908.165130>, URL <https://doi.org/10.1146/annurev.fluid.010908.165130>.
- [10] Choi, J.-I., Edwards, J., and Baurle, R., “Compressible Boundary Layer Predictions at High Reynolds Number using Hybrid LES/RANS Methods,” 2008. <https://doi.org/10.2514/6.2008-4175>, URL <https://arc.aiaa.org/doi/abs/10.2514/6.2008-4175>.
- [11] Giesecking, D. A., Choi, J.-I., Edwards, J. R., and Hassan, H. A., “Compressible-Flow Simulations Using a New Large-Eddy Simulation/Reynolds-Averaged Navier-Stokes Model,” *AIAA Journal*, Vol. 49, No. 10, 2011, pp. 2194–2209. <https://doi.org/10.2514/1.J051001>, URL <https://doi.org/10.2514/1.J051001>.
- [12] Shur, M. L., Spalart, P. R., Strelets, M. K., and Travin, A. K., “A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities,” *International Journal of Heat and Fluid Flow*, Vol. 29, No. 6, 2008, pp. 1638–1649. <https://doi.org/10.1016/j.ijheatfluidflow.2008.07.001>, URL <https://www.sciencedirect.com/science/article/pii/S0142727X08001203>.
- [13] Soshi Kawai, J. L., “Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy,” *Physics of Fluids*, Vol. 24, 2012.
- [14] Park, G. I., and Moin, P., “An improved dynamic non-equilibrium wall-model for large eddy simulation,” *Physics of Fluids*, Vol. 26, No. 1, 2014, p. 015108. <https://doi.org/10.1063/1.4861069>, URL <https://doi.org/10.1063/1.4861069>.

- [15] Toro, E. F., “The HLLC Riemann solver,” *Shock Waves*, Vol. 29, No. 8, 2019, pp. 1065–1082. <https://doi.org/10.1007/s00193-019-00912-4>, URL <https://doi.org/10.1007/s00193-019-00912-4>.
- [16] Ducros, F., Ferrand, V., Nicoud, F., Weber, C., Darracq, D., Gacherieu, C., and Poinso, T., “Large-Eddy Simulation of the Shock/Turbulence Interaction,” *Journal of computational physics*, Vol. 152, No. 2, 1999, pp. 517–549.
- [17] Kopper, P., “Implementation and Verification of a Synthetic Eddy Method (SEM) in the Eagle3d Compressible Flow Solver,” Master’s thesis, Embry-Riddle Aeronautical University, 2017. Master’s thesis.
- [18] Jarrin N., L. D. . P. R., Benhamadouche S., “A synthetic-eddy method for generating inflow conditions for large-eddy simulations,” *International Journal of Heat and Fluid Flow*, Vol. 27, 2006, pp. 585 – 593.
- [19] Bachalo, W. D., and Johnson, D. A., “Transonic, turbulent boundary-layer separation generated on an axisymmetric flow model,” *AIAA Journal*, Vol. 24, 1986, pp. 437–443.

## A Appendix

```

1  subroutine wmls
2      use thermochemmod , only : nspec
3      use cgnsgrid , only : myzone, nbocos, bocos_per_zone, boco_name, faces_per_boco,
boco_name, boundary_conditions
4      use gridmetrics , only : bndfacemap, unorm_face
5      use casereadmod , only : neqn, ip, WM_cell_limit, WM_yplus_target, ref_shearw,
ref_R, ref_gam
6      use sharedthermomod
7
8      implicit none
9      ! Declare input variables and output from the subroutine
10     real*4 :: u_cells, mu_cells, t_cells, h_cells, rho_cells, p_cells, cp_cells,
r_cells, tw_out
11     ! number of grid points and iteration number and other integers
12     integer :: ny, it, i, j, twsign
13     integer :: nz, ifcid, nboco, ibc, bndid
14     ! declare convergence criteria, convergence check, and new variables
15     real*4 :: convcheck, conv = 1.e-03, twold, twnorm, tw, p, r, cp, ust, thermkl
16     ! define constants needed for the calc heat transfer coeff, turbulent prandlt
number, laminar prandlt number, aplus coeff used in model
17     real*4 :: k = 0.41, prt = 0.9, pr = 0.71, apls = 17.0
18     ! declare all variables used in calculation of size of grid points
19     real*4, dimension(30) :: yscl, y, u, temp, rho, mu, mut, upls, yplus, d, a_mom,
b_mom, c_mom, a_eng, b_eng, c_eng, d_mom, d_eng
20     !outward unit vector normal of face
21     real(4), dimension(3) :: unitf, unit_cells
22
23     ! set number of points for WM grid
24     ny = 30
25
26     ! use reference wall shear stress to normalize wall shear for convergence check
27     twnorm = ref_shearw
28
29     ! create a scalable profile for y
30     yscl(1) = 0.0
31     do i = 2, ny
32         yscl(i) = 0.6*(1.016**i-1)/(1.016-1)
33     end do
34
35     ! Check for appropriate no-slip wall faces within BOCO list
36     do nz=myzone,myzone
37         do nboco=1,bocos_per_zone(nz)
38             if((boco_name(nz,nboco) == 'Wall Viscous Isoth').or.(boco_name(nz,nboco) ==
'Wall Viscous Adiab')) then
39                 do ibc=1, faces_per_boco(nz,nboco)
40                     ! Establish LES provided values at interpolation location for this face
41                     ifcid = boundary_conditions(nz,nboco,ibc,1)
42                     unitf(:) = unorm_face(nz,ifcid,:)
43                     bndid = bndfacemap(ifcid)
44                     h_cells = LES_interp(nz,bndid,1)
45                     u_cells = LES_interp(nz,bndid,2)
46                     t_cells = LES_interp(nz,bndid,3)
47                     rho_cells = LES_interp(nz,bndid,4)
48                     unit_cells(1:3) = LES_interp(nz,bndid,5:7)
49                     r_cells = ref_R
50                     p_cells = rho_cells*r_cells*t_cells
51                     cp_cells = ref_gam*r_cells/(ref_gam - 1.E+0)
52                     ! call sutherland(mu_cells,thermkl,t_cells,1)
53                     mu_cells = 1.716e-5*(t_cells/273.)**1.5*(273.+111.)/(t_cells+111.)
54
55                     ! set up boundary values of and constants
56                     twsign = u_cells/abs(u_cells)
57                     u(ny) = abs(u_cells)
58                     temp(ny) = t_cells

```



```

59     p = p_cells
60     cp = cp_cells
61     mu(ny) = mu_cells
62     rho(ny) = rho_cells
63     r = r_cells
64
65     ! Arbitrary value to check convergece
66     convcheck = 1.0
67
68     ! Rescale the y and apply U
69     y(1:ny) = yscl(1:ny)/yscl(ny)*h_cells
70     u(1:ny) = u(ny)/h_cells*y(1:ny)
71     temp(1:ny) = temp(ny)
72
73     ! call subroutines for rho, mu, tw, and mut
74     ! do j=1,ny
75     !     call sutherland(mu(j),thermkl,temp(j),1)
76     ! end do
77     mu(1:ny) = 1.716e-5*(temp(1:ny)/273.)**(1.5)*(273.+111.)/(temp(1:ny)
+111.)
78     rho(1:ny) = p/r/temp(1:ny)
79     call stressupdt(ny, mu, u, rho, y, apls, k, tw, mut, d, yplus, upls,
ust)
80     twold = tw
81
82     ! do while loop until convergence
83     it = 0
84     do while (convcheck > conv)
85         it = it + 1
86         !print *, 'wmles iteration: ', it, nz, u(ny), y(ny)
87         ! set up thomas algorithm coefficients for momentum equation
88         do i=2,ny-1
89             a_mom(i) = (mu(i)+mut(i))/(y(i)-y(i-1))
90             b_mom(i) = -(mu(i)+mut(i))/(y(i)-y(i-1)) - (mu(i+1)+mut(i+1))/
(y(i+1)-y(i))
91             c_mom(i) = (mu(i+1)+mut(i+1))/(y(i+1)-y(i))
92             d_mom(i) = 0
93         end do
94
95         ! boundary conditions for thomas algorithm
96         b_mom(1) = 1
97         c_mom(1) = 0
98         b_mom(ny) = 1
99         a_mom(ny) = 0
100        d_mom(ny) = u(ny)
101
102        ! solve thomas algorithm for velocity
103        call thmsalg(ny,a_mom,b_mom,c_mom,d_mom,u)
104        ! update the stress terms
105        call stressupdt(ny, mu, u, rho, y, apls, k, tw, mut, d, yplus, upls,
ust)
106
107        ! set up thomas algorithm matrix for energy equation
108        do i= 2, ny-1
109            d_eng(i) = (mu(i+1)+mut(i+1))*u(i+1)*(u(i+1)-u(i))/(y(i+1)-y(i)) &
- (mu(i)+mut(i))*u(i)*(u(i)-u(i-1))/(y(i)-y(i-1))
110            a_eng(i) = -cp*(mu(i)/pr+mut(i)/prt)/(y(i)-y(i-1))
111            c_eng(i) = -cp*(mu(i+1)/pr+mut(i+1)/prt)/(y(i+1)-y(i))
112            b_eng(i) = -(a_eng(i)+c_eng(i))
113        end do
114
115
116        ! bc for energy equation
117        ! neumann boundary condition for temp at the wall
118        c_eng(1) = 1

```

```

119         b_eng(1) = -1
120         d_eng(1) = 0
121
122         ! dirichlet boundary condition for temp at hwm
123         d_eng(ny) = temp(ny);
124         b_eng(ny) = 1;
125         a_eng(ny) = 0;
126
127         ! thomas alg for energy equation
128         call thmsalg(ny, a_eng,b_eng,c_eng,d_eng,temp)
129
130         ! call subroutines for rho, mu, tw, and mut
131         ! do j=1,ny
132         !     call sutherland(mu(j),thermkl,temp(j),1)
133         ! end do
134         mu(1:ny) = 1.716e-5*(temp(1:ny)/273.)**(1.5)*(273.+111.)/(temp(1:ny)
+111.)
135         rho(1:ny) = p/r/temp(1:ny)
136         call stressupdt(ny, mu, u, rho, y, apls, k, tw, mut, d, yplus, upls,
ust)
137
138         ! check convergence
139         tw_out = tw
140         if(it == 1) twnorm = abs(twold - tw)
141         if(twnorm > 1.E-20) then
142             convcheck = abs(twold-tw)/twnorm
143         else
144             convcheck = conv/10.
145         end if
146
147         !print *, nz, tw, twold, twnorm
148
149         if (it > 1000) then
150             write(*, *) 'WARNING COULD NOT CONVERGE WALL MODEL ZONE: ', nz,
'WITH VELOCITY: ', u(ny), 'WITH HEIGHT: ', y(ny), 'WITH TW TW OLD AND TW NORM: ', tw,
twold, twnorm
151             convcheck = conv/10.
152             end if
153
154             twold = tw
155
156
157         end do !:while loop for convergence
158
159         ! convert final wall shear stress into wall force per unit area (i.e.,
vector)
160         flux_vis(nz,ifcid,nspec+1:nspec+3) = -tw_out*unit_cells(1:3)
161         ! write(*,*) 'tw = ', tw_out, ' it = ', it
162         ! if(it > 2000) write(*,'(I6,1x,I4,1x,I2,1x,I8,1x,E14.5,1x,E14.5)') 'it
nz nboco ibc t V: ', it, nz, nboco, ibc, t_cells, u_cells
163         if(it > 2000) write(*,'(A40,1x,I8,1x,I4,1x,I2,1x,I8)') 'WMLES
iterations is large:', it, nz, nboco, ibc
164         end do !:ibc
165         end if !:check for no-slip wall
166         end do !:nboco
167         end do !:nz
168
169     end subroutine wmls
170
171
172     ! wall stress subroutine
173     subroutine stressupdt(ny, mu, u, rho, y, apls, k, tw, mut, d, yplus, upls, ust)
174         implicit none
175         integer :: ny, i

```

```

176     real*4, dimension(ny) :: mu, u, rho, y, mut, d, yplus, upls
177     real*4 :: tw, apls, k, ust
178     ! tw calculated only at the wall with laminar assumption
179     tw = mu(1)*(u(2)-u(1))/(y(2)-y(1))
180     ! find u*
181     ust = sqrt(tw/rho(1))
182     do i = 1, ny
183         ! find u+ at all points
184         upls(i) = u(i)/ust
185         ! find yplus for all points
186         yplus(i) = y(i)*ust*rho(i)/mu(i)
187         ! damping function
188         d(i) = (1-exp(-yplus(i)/apls))**2
189         ! calculate turbulent viscosity
190         mut(i) = k*rho(i)*y(i)*sqrt(tw/rho(i))*d(i)
191     end do
192 end subroutine stressupdt
193
194
195     ! thomas algorithm subroutine
196 subroutine thmsalg(n,a,b,c,d,x)
197     implicit none
198     integer :: n, i
199     real*4 :: w
200     real*4, dimension(n) :: a, b, c, d, x
201
202     do i = 2, n
203         w = a(i)/b(i-1)
204         b(i) = b(i)-w*c(i-1)
205         d(i) = d(i) - w*d(i-1)
206     end do
207     x(n) = d(n)/b(n)
208     do i=(n-1), 1, -1
209         x(i) = (d(i)-c(i)*x(i+1))/b(i)
210     end do
211 end subroutine thmsalg

```