



# A high-order multi-zone cut-stencil method for numerical simulations of high-speed flows over complex geometries



Patrick T. Greene\*, Jeff D. Eldredge, Xiaolin Zhong, John Kim

Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, 420 Westwood Plaza, Los Angeles, CA, 90095, USA

## ARTICLE INFO

### Article history:

Received 2 June 2014

Received in revised form 3 February 2016

Accepted 16 April 2016

Available online 20 April 2016

### Keywords:

Cut-stencil

Multi-zone refinement

High-order

Complex geometry

Cartesian grid

## ABSTRACT

In this paper, we present a method for performing uniformly high-order direct numerical simulations of high-speed flows over arbitrary geometries. The method was developed with the goal of simulating and studying the effects of complex isolated roughness elements on the stability of hypersonic boundary layers. The simulations are carried out on Cartesian grids with the geometries imposed by a third-order cut-stencil method. A fifth-order hybrid weighted essentially non-oscillatory scheme was implemented to capture any steep gradients in the flow created by the geometries and a third-order Runge–Kutta method is used for time advancement. A multi-zone refinement method was also utilized to provide extra resolution at locations with expected complex physics. The combination results in a globally fourth-order scheme in space and third order in time. Results confirming the method's high order of convergence are shown. Two-dimensional and three-dimensional test cases are presented and show good agreement with previous results. A simulation of Mach 3 flow over the logo of the Ubuntu Linux distribution is shown to demonstrate the method's capabilities for handling complex geometries. Results for Mach 6 wall-bounded flow over a three-dimensional cylindrical roughness element are also presented. The results demonstrate that the method is a promising tool for the study of hypersonic roughness-induced transition.

© 2016 Elsevier Inc. All rights reserved.

## 1. Introduction

The ability to accurately predict the location of laminar to turbulent transition in boundary layers is of great importance to the design of hypersonic vehicles [1]. In spite of its importance, the mechanisms leading to the transition of hypersonic boundary layers are still poorly understood. Predicting the location of transition is further complicated by the presence of surface roughness. Although a great deal of research has been done to study the effects of roughness on hypersonic transition, most of the data was for correlations to be used in the design of hypersonic vehicles. Very little is actually known about the exact physics behind hypersonic boundary-layer transition with roughness elements. The goal of this work was to develop a set of numerical tools for studying the effects of roughness elements on the transition of hypersonic boundary-layer flow. There are two major difficulties when simulating high-speed flow over roughness elements: imposing the roughness geometry and resolving the large variety of length scales present in the problem. The methods presented in this paper seek to address these difficulties.

\* Corresponding author. Currently at the Lawrence Livermore National Laboratory.

E-mail address: [green@ucla.edu](mailto:green@ucla.edu) (P.T. Greene).

### 1.1. Cartesian grid methods

The conventional approach to imposing geometries in direct numerical simulations (DNS) is to use a body-fitted grid [2]. In body-fitted grids, at least one of the edges of the computational domain will conform to the geometry. This makes the application of boundary conditions relatively simple. However, the generation of a body-fitted grid for complex geometries can be a very difficult and time-consuming process. As an alternative to body-fitted grids, an assortment of Cartesian grid methods have been developed [2]. The commonality between all the methods is that they can impose complex geometries on simple Cartesian grids. Most Cartesian grid methods fall into one of two categories: immersed boundary methods or sharp-interface methods.

The immersed boundary method was originally developed by Peskin [3,4]. Peskin used the immersed boundary method to study blood flow in a beating heart. The blood flow was simulated on a Cartesian grid while the elastic walls were discretized on a Lagrangian grid. The effect of the elastic wall was imposed on the fluid through a source term added to the equations governing the fluid flow. The source term would have an approximate Dirac  $\delta$  function associated with it, which limits its area of influence to just near the walls. The use of a forcing term added to the governing equations is the defining characteristic of immersed boundary methods.

Goldstein et al. [5] extended the immersed boundary method for flows with immersed rigid walls. Goldstein et al. designed the forcing term to have a magnitude and direction opposing the local flow. This would drive the fluid to zero velocity at the immersed boundary. This method, and others like it, are usually referred to as continuous forcing methods. Mohd-Yusof [6] derived an alternative to this called the discrete forcing method. Mohd-Yusof changed the forcing term so that it would impose zero velocity at the boundary for all time.

The primary advantage of the immersed boundary method is the ease of its implementation. Since the body is imposed through a source term, the method can be easily incorporated into an existing code without having to alter the underlining solver. The numerical method used in the solver is simply applied everywhere, including inside any immersed bodies and across their surfaces. This has led to the immersed boundary method being used to simulate a large variety of problems such as modeling the cochlea [7], flow over an idealized road vehicle [8], and studying the dynamics of vesicles [9]. Although used successfully for many problems, the immersed boundary method has its disadvantages. Most applications of the immersed boundary method have been at low to moderate Reynolds numbers. At high Reynolds numbers, the spreading effect of the approximate  $\delta$  function has a more detrimental effect on the flow since the local solution has greater importance [2]. In addition, the method is only locally first order, which may not have the required accuracy for some applications such as the numerical simulation of hypersonic boundary-layer transition.

While the immersed boundary method uses a forcing term added to the governing equation, sharp-interface methods usually alter the discretization of the governing equation near the body. The alteration allows boundary conditions to be applied just at the edge of the body. The methods generate precise locations for the body instead of smearing the interface as the immersed boundary method does. The two most common sharp-interface approaches are ghost fluid methods and cut-cell methods.

The ghost fluid method was developed by Fedkiw et al. [10,11]. The method begins by creating a layer of ghost fluid inside the immersed body. The values at the ghost points are extrapolated from the real fluid points in such a way that the boundary conditions are imposed at the interface exactly. Similar to the immersed boundary method, the interior discretization scheme is used across the body boundary, but now the values on the inside of the body are imposed by the method instead of advanced in time by the governing equations.

The primary motivation of the cut-cell method was to ensure global and local conservation of mass and momentum. To achieve this, the original formulation of the cut-cell method was made in a finite-volume discretization. Any cell that the edge of the body intersects is divided along the edge. The portion of the cells that are inside the body are discarded. If the center of the original cell is outside the body, then the remaining portion of the cell is used as the new cell. If the center is inside the body, then the remaining portion is merged into a neighboring cell. Unlike the previously mentioned methods, no values from inside the immersed body are ever used in the calculations. The cut-cell method was originally developed by Clarke et al. [12] for inviscid flow. Udaykumar et al. [13–15] and Ye et al. [16] extended the method to viscous flow.

Compared to the immersed boundary methods, sharp-interface methods tend to be more complex to implement. Sharp-interface methods are also victim to the “small cell problem” [17]. The small cell problem occurs when a grid cell becomes much smaller than its neighbors due to it being near the body’s edge. For time dependent problems, this can severely limit the size of the permitted time step. Most sharp-interface methods have addressed this issue and provide a solution. The added complexity for sharp-interface methods is usually outweighed by the ability to precisely enforce the location of the immersed body’s edge. This can be seen in the wide variety of problems the methods have been applied to such as shallow water flows [18], flapping flight [19], and magnetohydrodynamic flows [20].

### 1.2. Adaptive mesh refinement

Simulations of flow over roughness elements tend to involve a wide range of length scales. This is especially true for high-speed flows over roughness elements that have a height comparable to the boundary-layer thickness. When the flow encounters the roughness element, a new boundary layer is created along the roughness element. The thickness of this newly formed boundary layer can be more than an order of magnitude smaller than the thickness of the incoming boundary

layer. In addition, the streamwise length of the domain can be an order of magnitude larger than the incoming boundary-layer thickness. Problems such as this can benefit greatly from the adaptive mesh refinement method.

The adaptive mesh refinement method was originally developed by Berger et al. [21–23]. The method begins by covering the entire computational domain with a coarse grid. Then smaller refinement grids are added at locations where extra resolution is required. Refinement grids can be recursively nested until the desired level of accuracy is obtained. In addition to the increase in spatial resolution, the refinement grids are also refined in time. This is necessary to maintain the stability of explicit schemes. As the solution advances in time, the location of the refinement grids can move with the complex physics. Refinement grids can also be removed if no longer needed or added if more resolution is required. Berger et al. used Richardson extrapolation to select locations where extra refinement was needed. The method has also been paired with various Cartesian grid methods [24–27].

### 1.3. Current study

In our work, we develop a method capable of performing uniformly high-order numerical simulations of high-speed flows over complex geometries. The primary goal of the method is to simulate hypersonic boundary-layer flow over arbitrary roughness geometries. The new method is subsequently applied to the unsteady compressible Navier–Stokes equations for a calorically perfect gas. The inviscid terms are computed with a fifth-order hybrid weighted essentially non-oscillatory scheme and a third-order Runge–Kutta scheme is used for time advancement. A third-order cut-stencil method based on the work of Duan et al. [28] is utilized to impose geometries. It was found that the original implementation by Duan et al. could be numerically unstable for roughness elements with a height approximately equal to or greater the boundary-layer thickness. To address this instability, a different treatment of the wall pressure and temperature boundary conditions is developed in this work. In addition, the method is reformulated in conservative form, which permits it to be easily integrated with the interior scheme. A multi-zone refinement method is also used to precisely control grid resolution. The multi-zone refinement method is based on the high-order adaptive mesh refinement method of Shen et al. [29]. In the current implementation, the refinement zones are stationary in time instead of adaptive as in Shen et al. The combination of the interior scheme, the cut-stencil method, and the multi-zone refinement method results in a solver that is globally fourth order in space and third order in time, which is capable of simulating high-speed flows over complex geometries.

The remainder of this paper is structured as follows. In Section 2, we present the numerical methodology. This begins with a description of the governing equations and the numerical method used for the interior of the computational domain. Then the cut-stencil method is described and an overview of the multi-zone method is provided after that. Section 3 provides results confirming the order of convergence for the numerical methods. In Section 4, a comparison of the cut-stencil method and the immersed boundary method is presented. Section 5 contains the results for various validation and test cases. The conclusions of our work are presented in Section 6.

## 2. Methodology

The governing equations are the three-dimensional unsteady compressible Navier–Stokes equations for a calorically perfect gas with density  $\rho$ , pressure  $p$ , temperature  $T$ , total energy per unit mass  $E$ , and velocity  $(u_1, u_2, u_3) = (u, v, w)$ . The gas used in all the simulations is air so the gas constant,  $R$ , has a value of  $287.04 \text{ m}^2/\text{s}^2 \text{ K}$ , the ratio of the specific heats,  $\gamma$ , is 1.4, and the Prandtl number,  $Pr$ , is 0.72. The fluid viscosity,  $\mu$ , is computed using Sutherland's law.

The flow fields are obtained through direct numerical simulation of the governing equations. Before solving, the governing equations are transformed from the physical  $(x, y, z)$  space to  $(\xi, \eta, \zeta) = (\xi_1, \xi_2, \xi_3)$  computational space. The grid in the computational space is uniformly spaced in  $\xi$ ,  $\eta$ , and  $\zeta$  with each varying from 0 to 1. Even though the cut-stencil method allows simulations to be performed on Cartesian grids, the equations are still transformed into computational space to facilitate the use of grid point clustering. After the transformation, the governing equations become

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_j}{\partial \xi_j} = \frac{\partial \mathbf{F}_{v,j}}{\partial \xi_j}, \quad j = 1, 2, 3 \quad (1)$$

where  $\mathbf{U}$  is a vector of the conservative variables,  $[\rho, \rho u, \rho v, \rho w, E]^T / J$ .  $J$  is the determinant of the Jacobian matrix.  $\mathbf{F}_j$  and  $\mathbf{F}_{v,j}$  are the vectors of the inviscid and viscous fluxes, respectively, in the  $j$ th direction of the computational space. The high-order solver being presented here is comprised of three components: the Navier–Stokes solver, the cut-stencil method, and the multi-zone refinement method. Each component will be covered separately, starting with the Navier–Stokes solver.

### 2.1. Navier–Stokes solver

Most of the problems considered in this work involve supersonic flow over objects so a Navier–Stokes solver with shock-capturing capabilities was required. This motivated the selection of a fifth-order finite-difference hybrid scheme that combines a weighted essentially non-oscillatory (WENO) scheme with a low-dissipation scheme for the inviscid terms in the governing equations. The selection of the interior scheme does not have a large impact on the cut-stencil method or the multi-zone refinement method, which permits other schemes to be used instead provided they are at least globally fourth order.

The hybrid scheme begins by spiting the inviscid terms using local Lax–Friedrichs flux splitting for upwinding.

$$\mathbf{F}_j = \mathbf{F}_j^+ + \mathbf{F}_j^- \tag{2}$$

The derivatives are computed in conservative form. As an example, the derivative of the inviscid flux in the  $\xi$  direction with respect to  $\xi$  at the point  $\xi_i$  is computed as

$$\frac{\partial \mathbf{F}_\xi}{\partial \xi}(\xi_i) = \frac{\hat{\mathbf{F}}_\xi^+(\xi_{i+1/2}) - \hat{\mathbf{F}}_\xi^+(\xi_{i-1/2})}{\Delta \xi} + \frac{\hat{\mathbf{F}}_\xi^-(\xi_{i+1/2}) - \hat{\mathbf{F}}_\xi^-(\xi_{i-1/2})}{\Delta \xi}. \tag{3}$$

$\hat{\mathbf{F}}_\xi^\pm(\xi_{i+1/2})$  is an approximation to the numerical flux function (see Jiang and Shu [30] for definition) at the interface  $\xi_{i+1/2}$ .  $\hat{\mathbf{F}}_\xi^\pm(\xi_{i+1/2})$  is a linear combination of a fifth-order WENO scheme ( $\hat{\mathbf{F}}_\xi^{\pm\text{WENO}}$ ) and a fifth-order low-dissipation scheme ( $\hat{\mathbf{F}}_\xi^{\pm\text{LD}}$ ).

$$\hat{\mathbf{F}}_\xi^\pm(\xi_{i+1/2}) = (1 - \sigma_{i+1/2}) \hat{\mathbf{F}}_\xi^{\pm\text{WENO}}(\xi_{i+1/2}) + \sigma_{i+1/2} \hat{\mathbf{F}}_\xi^{\pm\text{LD}}(\xi_{i+1/2}) \tag{4}$$

where  $\sigma_{i+1/2}$  is the hybrid switch and is based on the work of Ren et al. [31]. The switch used by Ren et al. was computed from the grid point fluxes and each component of the flux had a different switch value. In the current work, a single value for the switch is used for all the components of the flux and its value is computed from density. In addition, the number of points used to compute the switch was increased to accommodate the larger stencils of the schemes used here. The switch is computed as

$$\sigma_{i+1/2} = \begin{cases} 0 & \text{if } r_{i+1/2} < r_c \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

where  $r_{i+1/2}$  is the smoothness indicator and  $r_c$  is a smoothness cutoff. In this work,  $r_c$  was set to 0.8. The smoothness indicator is

$$r_{i+1/2} = \min(r_{i-1}, r_i, r_{i+1}, r_{i+2}) \tag{6}$$

where

$$r_i = \frac{|2\Delta\rho_{i+1/2}\Delta\rho_{i-1/2}| + \epsilon_r}{(\Delta\rho_{i+1/2})^2 + (\Delta\rho_{i-1/2})^2 + \epsilon_r}, \tag{7}$$

and  $\Delta\rho_{i+1/2} = \rho_{i+1} - \rho_i$ .  $\epsilon_r$  is a small number used to prevent the denominator from going to zero. The value used is

$$\epsilon_r = \frac{0.9r_c}{1 - 0.9r_c} \nu^2 \rho_{\text{ref}}^2 \tag{8}$$

where  $\rho_{\text{ref}}$  is the reference density for the problem and  $\nu$  is  $10^{-3}$ .

The WENO flux is computed using the mapped WENO scheme of Hendrick et al. [32]. The low-dissipation flux is computed using a conservative version of the fifth-order scheme of Zhong [33]. The low-dissipation flux for the positive eigenvalue fluxes is computed as

$$\hat{\mathbf{F}}_\xi^{+\text{LD}}(\xi_{i+1/2}) = \sum_{j=-2}^3 a_j \mathbf{F}_\xi^+(\xi_{i+j}) \tag{9}$$

where the constants  $a_j$  are given in Table 1. The constants for the negative eigenvalue fluxes can be found from symmetry. The  $\alpha$  in the constants controls the scheme's dissipation and stability, and is equivalent to the  $\alpha$  from Zhong [33]. In this work,  $\alpha$  was set to  $-6$ . All the derivatives in the viscous terms are computed using a sixth-order central finite difference. The governing equations were advanced in time by a third-order TVD Runge–Kutta scheme [30].

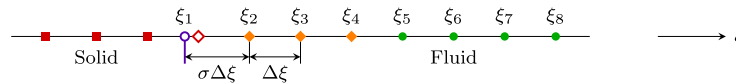
### 2.2. Cut-stencil method

The cut-stencil<sup>1</sup> method used in this work is based on the method of Duan et al. [28]. Duan et al. developed a globally fourth-order method for simulating flows with non-body-conforming grids. The main idea behind the cut-stencil method of Duan et al. is to switch to a non-uniform one-sided finite-difference stencil whenever the normally used stencil will

<sup>1</sup> In Duan et al., this method is referred to as the cut-cell method. This can be a source of confusion since the name cut-cell is usually associated with the finite-volume cut-cell method. In the finite-volume approach, the cells of the grid are cut along the immersed surface. Duan et al. applied a similar idea in a finite-difference formulation. Instead of cutting cells, Duan et al. cut the finite-difference stencils along the immersed surface and place a grid point on the surface. To help distinguish the two different methods, we refer to the finite-difference formulation as the cut-stencil method.

**Table 1**  
Low-dissipation scheme constants for positive eigenvalue numerical flux function.

$j$	$a_j$
-2	$1 - \frac{1}{12}\alpha$
-1	$-8 + \frac{5}{12}\alpha$
0	$37 - \frac{5}{6}\alpha$
1	$37 + \frac{5}{6}\alpha$
2	$-8 - \frac{5}{12}\alpha$
3	$1 + \frac{1}{12}\alpha$



**Fig. 1.** Point classification for sample  $\xi$  grid line. Immersed boundary —; Regular point ●; Irregular point ◆; Boundary point ○; Dropped point ■; Dropped point in  $\xi$  direction only ◇.

cross the immersed boundary. The current implementation has a number of differences compared to the original method of Duan et al. The calculation of the wall pressure and the treatment of the adiabatic boundary condition has been changed. The changes were needed to address some numerical instabilities seen in simulations of roughness elements with a height approximately equal to the boundary-layer thickness. The method for computing the shear stress along the wall and at a grid point very close to the wall has also been changed. In addition, the method was reformulated into conservative form so that it can be easily combined with the interior conservative scheme. A summary of the method will be presented here with the differences between the current implementation and the original implementation of Duan et al. pointed out.

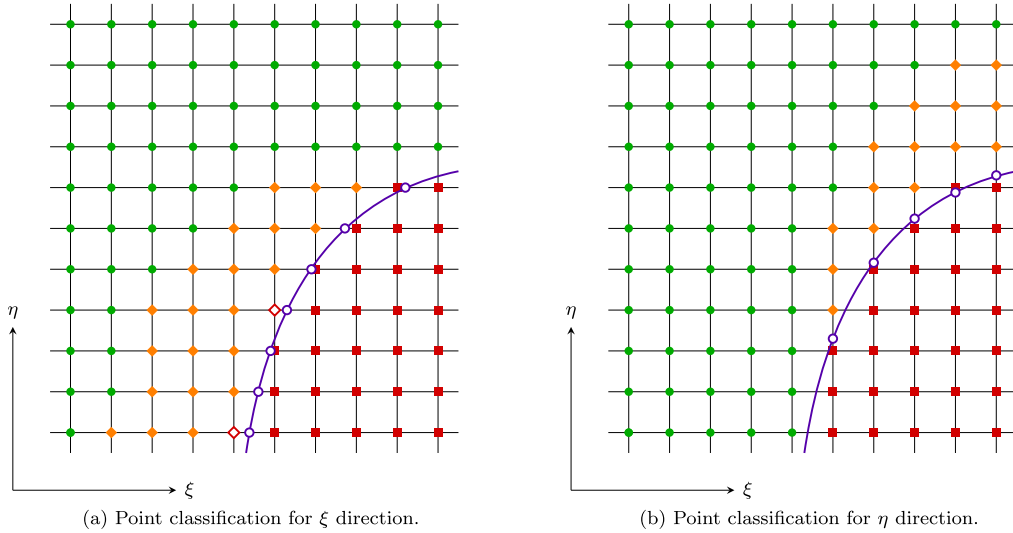
### 2.2.1. Point classification

The cut-stencil method begins by classifying all the grid points as either a regular point, an irregular point, a dropped point, or a boundary point. Fig. 1 shows the classification for a sample  $\xi$  grid line. The immersed boundary intersects the grid line at the point  $\xi_1$ . The solid is assumed to be to the left of  $\xi_1$  and the fluid to the right. The point  $\xi_1$  is classified as a boundary point. Boundary points are created anywhere the immersed surface crosses a grid line. Boundary points are not part of the original Cartesian grid. All the points to the left of  $\xi_1$  are classified as dropped points. Dropped points are usually points within the immersed surface and should not be used in any of the calculations. The first point to the right of  $\xi_1$  is classified as dropped in the  $\xi$  direction only. Although this point is in the fluid, the point is treated as a dropped point while performing computations in the  $\xi$  direction. The point was dropped because it was deemed too close to the boundary point and its inclusion would severely limit the time step size. The criteria for determining when a point is too close is based on a specified parameter  $\Theta$  and will be explained later.  $\xi_2$ ,  $\xi_3$ , and  $\xi_4$  are all classified as irregular points. Irregular points are any points whose finite-difference stencil would normally include a dropped point. The points  $\xi_5$  to  $\xi_8$  are regular points. Regular points are any points whose finite-difference stencil does not include any dropped points. When applying the method to two-dimensional and three-dimensional problems, a grid point may have a different classification in each of the computational directions. Fig. 2 shows the point classification for a sample two-dimensional domain. The treatment for regular, irregular, boundary, and dropped points will be covered one at a time.

### 2.2.2. Dropped points

There are two different types of dropped points. The first type is when a point is dropped in all directions of the computational domain. In this case, the point is never used in the calculations and therefore no special treatment is needed. All the points inside the immersed surface are of this type. The second type is when a point is dropped in at least one computational direction but not all of them. If the value of  $\sigma$  along a grid line is less than the specified value  $\Theta$ , then the grid point is dropped in the direction of that grid line. Duan et al. recommend a value of 1.0 for  $\Theta$  in all directions of the computational domain. For the problems under consideration here, it was found that a value of 1.0 for  $\Theta$  would cause numerical instabilities to form. In this work, separate  $\Theta$ 's were used in each direction. For all the two-dimensional problems done in this work,  $\Theta$  for the  $x$  direction was set 0.25 and  $\Theta$  for the  $y$  direction was set to 0.75. For three-dimensional problems, the same values were used in the  $x$  and  $y$  directions and the  $\Theta$  in the  $z$  direction was set to 0.25. These values were found to maintain numerical stability while not limiting the time step too severely.

For the second type of dropped point, the point will not be used in stencils in the dropped direction(s) but can be used in the remaining direction(s). Due to this, the values at these points must be computed but they cannot be advanced in time like points that are regular or irregular in all the directions of the domain. Duan et al. interpolated the conservative flow variables at these points using a fourth-order interpolation along the grid line in one of the dropped directions. The interpolation stencil always includes the boundary point adjacent to the dropped point. In this work, the conservative flow variables at these points are computed using a second-degree least-squares polynomial in  $x$ ,  $y$ , and  $z$  of the form



**Fig. 2.** Point classification for sample two-dimensional computational space. Immersed boundary —; Regular point ●; Irregular point ◆; Boundary point ○; Dropped point in both directions ■; Dropped point in specified direction only ◇.

$$\mathbf{U}(x, y, z) = c_1 + c_2 \Delta x + c_3 \Delta y + c_4 \Delta z + c_5 \Delta x \Delta y + c_6 \Delta x \Delta z + c_7 \Delta y \Delta z + c_8 \Delta x^2 + c_9 \Delta y^2 + c_{10} \Delta z^2 \quad (10)$$

where

$$\Delta x = x - x_d, \quad \Delta y = y - y_d, \quad \Delta z = z - z_d, \quad (11)$$

and  $x_d$ ,  $y_d$ , and  $z_d$  are the  $x$ ,  $y$ , and  $z$  values of the dropped point. For three-dimensional simulations, the constants are computed from the 35 closest points by index to the dropped point. For two-dimensional flows, the terms with  $\Delta z$  were removed and only the closest 10 points are used. The number of points needed for the least squares are problem and grid dependent. We found these values maintained numerical stability for all the problems we considered. The points can include regular, irregular, and boundary points. Other dropped points cannot be included in the points. Including them in the interpolation will cause the flow variables at one dropped point to be related to the flow variables at all the other dropped points resulting in a large implicit system. By not including the other dropped points, the flow variables at each point may be computed individually. In order to obtain a globally fourth-order method, a third-degree least-squares polynomial is actually needed. The order of convergence calculation done in this work used a third-degree polynomial. The second-degree polynomial was used in the remaining simulations as a method of decreasing the runtime.

In addition to computing the flow variables at points dropped in one or more directions, the shear stresses and heat fluxes must also be computed at these points. Duan et al. omitted how they compute the shear stresses and heat fluxes at these points. In this work, they are computed using a second-degree least-squares polynomial and the same set of grid points as the conservative flow variables.

### 2.2.3. Irregular points

Irregular points are any points whose normal finite-difference stencil includes dropped points. Since dropped points should not be used in the calculations, the finite-difference stencil for irregular points must be changed. For the irregular points  $\xi_2$ ,  $\xi_3$ , and  $\xi_4$  from Fig. 1, Duan et al. change the stencil for the inviscid terms to a third-order or fourth-order one-sided non-uniform upwind finite-difference stencil. The stencils can include the boundary point at  $\xi_1$ . The inclusion of the boundary point is what makes the stencil non-uniform since the spacing between the boundary point and  $\xi_2$  is  $\sigma \Delta \xi$  instead of just  $\Delta \xi$ . In this work, the derivatives are first cast in conservative form and then the numerical flux function is interpolated using a non-uniform stencil. The derivative of the inviscid flux in the  $\xi$  direction with respect to  $\xi$  is computed as

$$\frac{\partial \mathbf{F}_\xi}{\partial \xi}(\xi_i) = \frac{\hat{\mathbf{F}}_\xi^+(\xi_{i+1/2}) - \hat{\mathbf{F}}_\xi^+(\xi_{i-1/2})}{\Delta \xi} + \frac{\hat{\mathbf{F}}_\xi^-(\xi_{i+1/2}) - \hat{\mathbf{F}}_\xi^-(\xi_{i-1/2})}{\Delta \xi}. \quad (12)$$

$\hat{\mathbf{F}}_\xi^\pm(\xi_{i+1/2})$  is an approximation to the numerical flux function at the interface  $\xi_{i+1/2}$ . The numerical flux function at the half grid points is computed as

$$\hat{\mathbf{F}}_\xi^\pm(\xi_{i+1/2}) = \sum_{j=1}^5 a_{i,j}^\pm \mathbf{F}_\xi^\pm(\xi_j) \quad (13)$$

**Table 2**  
One-sided non-uniform interpolation constants for inviscid terms at irregular points.

	$a_{i,1}^+$	$a_{i,2}^+$	$a_{i,3}^+$	$a_{i,4}^+$	$a_{i,5}^+$
$i = 1$	$\frac{3}{2\sigma(\sigma+1)(\sigma+2)}$	$\frac{22\sigma-9}{12\sigma}$	$\frac{2-7\sigma}{6(\sigma+1)}$	$\frac{4\sigma-1}{12(\sigma+2)}$	0
$i = 2$	$\frac{-1}{2\sigma(\sigma+1)(\sigma+2)}$	$\frac{4\sigma+3}{12\sigma}$	$\frac{5\sigma+2}{6(\sigma+1)}$	$\frac{-(2\sigma+1)}{12(\sigma+2)}$	0
$i = 3$	$\frac{1}{2\sigma(\sigma+1)(\sigma+2)}$	$\frac{-(2\sigma+3)}{12\sigma}$	$\frac{5\sigma+8}{6(\sigma+1)}$	$\frac{4\sigma+5}{12(\sigma+2)}$	0
	$a_{i,1}^-$	$a_{i,2}^-$	$a_{i,3}^-$	$a_{i,4}^-$	$a_{i,5}^-$
$i = 1$	0	$\frac{25}{12}$	$-\frac{23}{12}$	$\frac{13}{12}$	$-\frac{3}{12}$
$i = 2$	0	$\frac{3}{12}$	$\frac{13}{12}$	$-\frac{5}{12}$	$\frac{1}{12}$

**Table 3**  
One-sided non-uniform finite-difference constants for viscous terms at irregular and boundary points.

$i$	$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$a_{i,4}$	$a_{i,5}$	$a_{i,6}$
1	$\frac{-2(2\sigma+3)(\sigma^2+3\sigma+1)}{\sigma(\sigma+1)(\sigma+2)(\sigma+3)}$	$\frac{(\sigma+1)(\sigma+2)(\sigma+3)}{6\sigma}$	$\frac{-\sigma(\sigma+2)(\sigma+3)}{2(\sigma+1)}$	$\frac{\sigma(\sigma+1)(\sigma+3)}{2(\sigma+2)}$	$\frac{-\sigma(\sigma+1)(\sigma+2)}{6(\sigma+3)}$	0
2	$\frac{-6}{\sigma(\sigma+1)(\sigma+2)(\sigma+3)}$	$\frac{6-11\sigma}{6\sigma}$	$\frac{3\sigma}{(\sigma+1)}$	$\frac{-3\sigma}{2(\sigma+2)}$	$\frac{\sigma}{3(\sigma+3)}$	0
3	$\frac{2}{\sigma(\sigma+1)(\sigma+2)(\sigma+3)}$	$\frac{-2(\sigma+1)}{6\sigma}$	$\frac{1-\sigma}{2(\sigma+1)}$	$\frac{\sigma+1}{(\sigma+2)}$	$\frac{-(\sigma+1)}{6(\sigma+3)}$	0
4	0	$\frac{5}{60}$	$-\frac{40}{60}$	0	$\frac{40}{60}$	$-\frac{5}{60}$

where the constants are given in Table 2.  $\mathbf{F}_\xi^\pm(\xi_j)$  is the grid point fluxes after applying local Lax–Friedrichs flux splitting. The numerical flux function at  $\xi_{1+1/2}$  is not computed at the half way point between  $\xi_1$  and  $\xi_2$ . To maintain the high order of convergence, the spacing between the locations where the numerical flux function is calculated must be equal to  $\Delta\xi$ . Therefore,  $\hat{\mathbf{F}}_\xi^\pm(\xi_{1+1/2})$  is evaluated at  $\xi_2 - \Delta\xi/2$  instead of  $\xi_2 - \sigma\Delta\xi/2$ .  $\hat{\mathbf{F}}_\xi^+(\xi_{4+1/2})$  and  $\hat{\mathbf{F}}_\xi^-(\xi_{4+1/2})$  are computed using the interior hybrid scheme since there are sufficient uniform points to form the hybrid stencil.  $\hat{\mathbf{F}}_\xi^-(\xi_{3+1/2})$  is computed using just WENO since there are enough uniform grid points for the WENO stencil but not enough for the hybrid stencil.

To compute the viscous terms, Duan et al. use a fourth-order one-sided non-uniform finite-difference stencil at  $\xi_2$  and  $\xi_3$  and a fifth-order one-sided non-uniform finite-difference stencil at  $\xi_4$ . In this work, the same stencil is used for  $\xi_2$  and  $\xi_3$  but a uniform fourth-order central finite-difference stencil is used at point  $\xi_4$ . The derivative of the viscous flux in the  $\xi$  direction with respect to  $\xi$  at all three points can be expressed as

$$\frac{\partial \mathbf{F}_{v,\xi}}{\partial \xi}(\xi_i) = \frac{1}{\Delta\xi} \sum_{j=1}^6 a_{i,j} \mathbf{F}_{v,\xi}(\xi_j) \tag{14}$$

where the constants are given in Table 3. The table also includes the constants for computing the derivative at the boundary point  $\xi_1$ . These values will be used in the next section.

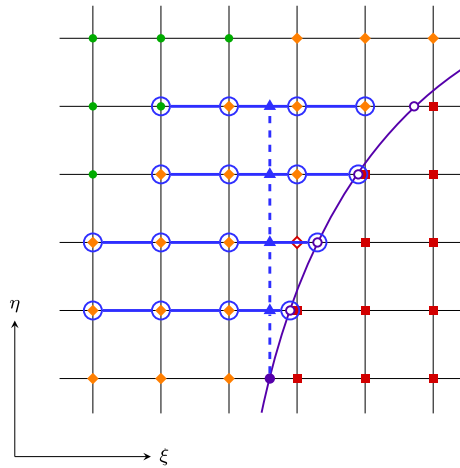
### 2.2.4. Boundary points

Boundary points are created anywhere a grid line crosses the immersed surface. The boundary conditions for the flow will be imposed at the boundary points. Since the immersed surface is always treated as a wall in this work and the flow is viscous, the no-slip boundary condition is imposed there. The temperature at the wall will depend on the temperature boundary condition for the problem and will be either isothermal or adiabatic in this work. The pressure at the boundary points will be extrapolated from the flow.

In Duan et al., a fourth-order extrapolation in the direction of the grid line that created the boundary point is used to compute the wall pressure. For adiabatic walls, only the temperature gradient in the direction of the grid line that created the boundary point is set to zero. In the current work, it was found that this method works for shorter roughness elements but can cause numerical instabilities for taller roughness elements. Taller roughness elements tend to generate steep gradients in the pressure near the roughness element. If two boundary points that were created by different grid lines were located next to each other, interpolating along the grid lines could result in very different wall pressure values at the two points. To rectify this, a least-squares polynomial was used instead. This allows values at boundary points to be computed from all near by grid points and the values would not depend on the direction of the grid line which created the boundary point.

The pressure at boundary points is computed using a second-degree least-squares polynomial in  $x$ ,  $y$ , and  $z$  of the form

$$p(x, y, z) = c_1 + c_2\Delta x + c_3\Delta y + c_4\Delta z + c_5\Delta x\Delta y + c_6\Delta x\Delta z + c_7\Delta y\Delta z + c_8\Delta x^2 + c_9\Delta y^2 + c_{10}\Delta z^2 \tag{15}$$



**Fig. 3.** Points used to compute a derivative in the  $\eta$  direction for a  $\xi$  boundary point. Point classification for the  $\xi$  direction shown. Immersed boundary —; Regular point ●; Irregular point ◆; Boundary point ○; Dropped point in  $\xi$  direction only ◇; Boundary point where derivative is being computed ●; Interpolated point for finite-difference stencil ▲; Stencil used to interpolate points for finite-difference stencil ○—○; Finite-difference stencil used to compute derivative —▲.

where

$$\Delta x = x - x_b, \quad \Delta y = y - y_b, \quad \Delta z = z - z_b, \tag{16}$$

and  $x_b$ ,  $y_b$ , and  $z_b$  are the  $x$ ,  $y$ , and  $z$  values of the boundary point. For three-dimensional simulations, the constants are computed using 25 regular or irregular points near the boundary point. For two-dimensional simulations, the terms with a  $\Delta z$  are removed and only 15 points are used. Similarly to the drop point least squares, these values are problem and grid dependent. The first three points along the grid line that created the boundary point should be included in the set of points. Including these points reduces the likelihood of obtaining a singular matrix when computing the constants. The remaining points are simply the closest points by index to the boundary point. Other boundary points cannot be included. Including them would result in a large implicit system. In addition, points dropped in any direction cannot be included. The least-squares polynomial used for points dropped in one or two directions includes boundary points and therefore must be updated after the boundary points. As with the drop point least-squares polynomial, a third-degree polynomial was used during the order of convergence calculations and the above second-degree polynomial for the remaining simulations.

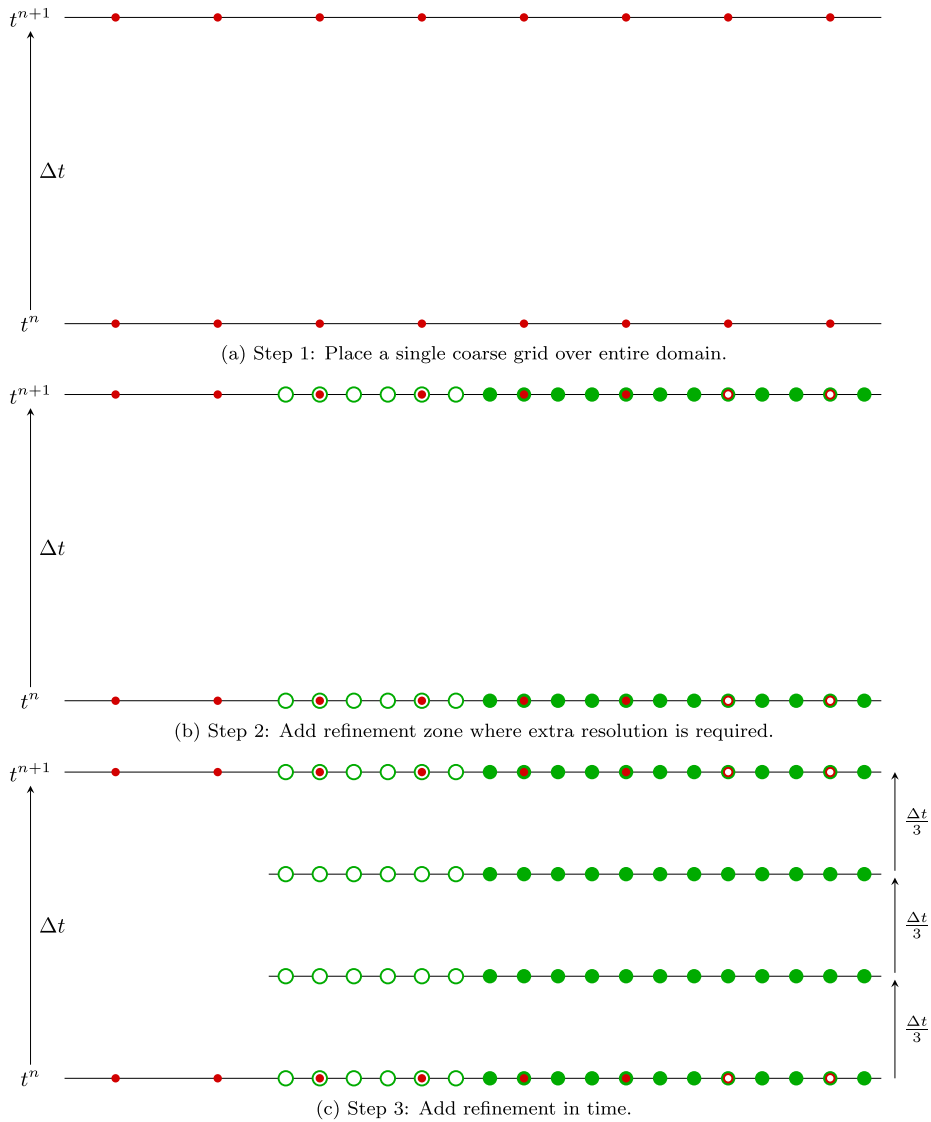
A similar method is used to enforce the adiabatic boundary condition. A second-degree polynomial for the temperature is created using the same points as the pressure polynomial. The polynomial is evaluated at four evenly spaced points in the wall normal direction. These values are then used along with a uniform one-sided fourth-order finite-difference stencil to enforce a zero temperature gradient at the wall.

In addition to imposing the boundary conditions on the flow variables, the shear stresses must be computed at the boundary points. To compute the shear stresses, derivatives are required in all the directions of the computational domain. To compute the required derivatives in the direction of the grid line that created the boundary point, Duan et al. use a fourth-order non-uniform finite-difference stencil. The same stencil is used in this work and is evaluated using Equation (14) with the constants coming from Table 3. However, Duan et al. overlooked to explain how they compute the derivatives in the other directions of the computational domain. In this work, these derivatives are computed using a combination of interpolation and a uniform one-sided finite-difference stencil. Fig. 3 shows how the derivative in the  $\eta$  direction is computed for a boundary point defined by a  $\xi$  grid line in a two-dimensional domain. Four points above the boundary point are first computed by interpolating values in the  $\xi$  direction. The interpolation stencils should not include any points dropped in the direction of the grid line that the boundary point lies on (e.g. the  $\xi$  grid line in Fig. 3). The interpolation stencil may include boundary points. These interpolated values are then used in a uniform one-side finite-difference stencil to compute the derivative.

### 2.3. Multi-zone refinement

The multi-zone refinement method utilizes overlapping grids to precisely control grid resolution. Since the numerical schemes being used in this work are high order, it is important that the refinement method maintain the spatial and temporal accuracy of the schemes. The method used in this work comes from Shen et al. [29]. Shen et al. developed a high-order finite-difference WENO version of the adaptive mesh refinement method. Since their method was adaptive, the refinement zones would move with the flow physics. In our work, the locations needing increased resolution are fixed in time. Therefore, a stationary refinement grid version was used in this work. An overview of the method will be provided here. The details can be found in Shen et al. [29].





**Fig. 4.** Multi-zone refinement grid generation and point classification. Coarse grid point ●; coarse grid ghost point ○; refinement grid point ●; refinement grid ghost point ○.

Fig. 4 depicts the process for generating a one-dimensional domain with a single refinement zone. The method begins by placing a single coarse grid over the entire flow field. Fig. 4a shows the coarse grid at the  $t^n$  and  $t^{n+1}$  time steps. Next smaller grids with increased resolution are placed on top of the coarse grid at locations where complex physics are expected. Fig. 4b shows a refinement grid with a factor of three increase in resolution placed over the coarse grid. Shen et al. recommend setting the refinement ratio to an odd value as this will allow the coarse grid points and coarse grid flux locations to have a corresponding value on the refinement grid. Six ghost points are attached to the edge of the refinement grid. The flow variables at the ghost points are computed from the coarse grid using fifth-order interpolation. In Shen et al., WENO interpolation was used. In our work, simple fifth-order interpolation was found to be sufficient. Most of the coarse grid points that are overlapped by refinement grid points become coarse grid ghost points. The flow variables at these points are simply set equal to the corresponding refinement grid point values. The exception to this is the first two overlapped coarse points. These two points are allowed to advance in time with the rest of the coarse grid points. This is needed to keep the interpolation used at the refinement grid ghost points numerically stable. As noted by Shen et al., the simple interpolation and point value replacement procedure used in this method does not guarantee conservation. Shen et al. demonstrated that the conservation errors are very small for smooth solutions. Since our refinement zones are stationary, we cannot guarantee that the solution will be smooth across the refinement boundaries. We attempt to mitigate the error by judiciously placing the refinement zones so that the shock has been significantly weakened before crossing a refinement boundary.

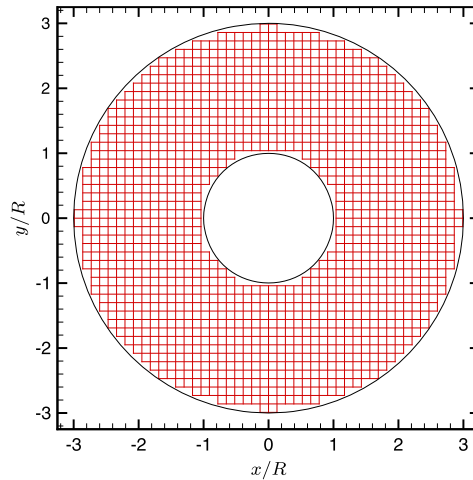


Fig. 5. Sample grid for circular Couette flow with  $50 \times 50$  points. Points inside the cylinders are not shown.

In addition to the spatial refinement, the temporal resolution of the refinement grid is also increased by the same factor. Fig. 4c shows the additional time steps taken by the refinement grid. The flow variables at the refinement grid ghost points for the additional time steps are computed from the ghost points at steps  $t^n$  and  $t^{n+1}$  using fourth-order Hermite interpolation. This requires that the coarse grid be advanced in time before advancing the refinement grid since coarse grid values at  $t^n$  and  $t^{n+1}$  are required to compute the refinement grid ghost points.

For all the problems considered in this work, the solid interfaces are contained completely within the finest refinement zone. This was done to ease the implementation of the cut-stencil and mesh refinement methods. Aside from the increased coding effort, the authors know of no reason the solid interface cannot cross the refinement boundary.

### 3. Order of convergence

The order of convergence was computed to ensure that the target high-order of convergence was achieved. Two test cases were used to find the order of convergence. The simulations tested the cut-stencil and mesh refinement methods individually.

#### 3.1. Cut-stencil method

The order of convergence for the cut-stencil portion of the code was tested using circular Couette flow. The problem consists of two concentric cylinders with a viscous fluid between them. For the test case, the gap between the cylinders is twice the inner cylinder radius ( $R$ ). The inner cylinder rotated with a Mach number of one while the outer cylinder was stationary. Both walls were isothermal. The inner cylinder was set to a temperature of 300 K and the outer cylinder was set to 400 K. The Reynolds number based on the inner cylinder radius was 500. Since the steady-state solution for this problem is only dependent on the pressure gradient, the pressure may fluctuate significantly during the simulation. To prevent this, the pressure at the inner wall was set to 200 Pa. The simulation was performed assuming a constant viscosity. This permitted the comparison to an analytical exact solution.

The simulation was performed on a uniform Cartesian grid measuring  $6.5R \times 6.5R$ . Both cylinders were imposed by the cut-stencil method. Fig. 5 shows a sample grid using  $50 \times 50$  grid points. The simulation was carried out on four grids:  $100 \times 100$ ,  $200 \times 200$ ,  $400 \times 400$ , and  $800 \times 800$ . Fig. 6 shows the Mach contours for the solution computed on the  $200 \times 200$  grid. Despite being simulated on a Cartesian grid, the contours are circular as the exact solution predicts. Fig. 7 shows the pressure, temperature, and Mach number profiles on the  $200 \times 200$  grid. All the profiles match the exact solution.

Fig. 8 shows the  $L^2$ ,  $L^1$ , and  $L^\infty$  errors for the conservative variables from each of the grid resolutions. The number next to each line segment is the slope of that segment, which is equal to the order of convergence. In some of the plots, the  $L^2$  error is not visible because it is approximately equal to the  $L^1$  error for that variable. The cut-stencil method uses third-order differencing near the immersed surface. When combined with the fifth-order interior scheme, the expected global order of convergence is four. The  $L^2$  and  $L^1$  errors show that the order of convergence is near the desired value of four. The  $L^\infty$  error is showing a value of three since it is being computed from the maximum error, which is located at the third-order stencils used in the cut-stencil method.

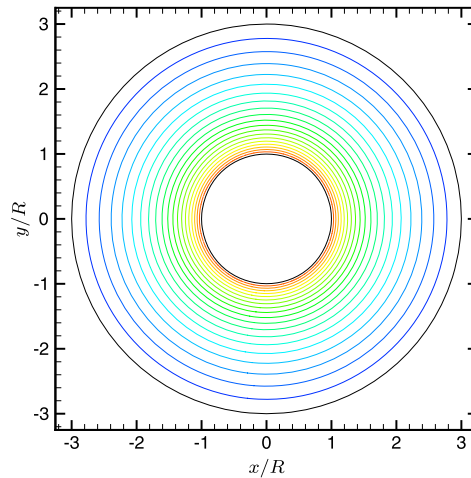


Fig. 6. Mach contour for circular Couette flow. Solution computed on  $200 \times 200$  grid. 21 evenly spaced levels from 0.0 to 1.0.

### 3.2. Multi-zone refinement method

The order of convergence for the multi-zone refinement portion of the code was tested by simulating the propagation of a low-amplitude density wave. The wave was sinusoidal in shape with an amplitude of 10% of the mean density. The wave propagates at an angle of approximately  $26^\circ$  with the  $x$  axis. The viscous terms were turned off for this simulation.

Fig. 9 shows the density contour for the simulation. The figure also shows the locations of the first and second refinement zones. The first zone increases the resolution by a factor of three compared to the coarse grid and the second increases the resolution by another factor of three. Periodic boundary conditions are used along the coarse grid edges.

The simulation was performed on a uniform Cartesian grid. In order to compute the spatial and temporal order of convergence, two sets of simulations were performed. In the first, the CFL number was fixed at 0.1 while the number of grid points was changed. Four simulations were carried out with the coarse grid having  $100 \times 50$ ,  $200 \times 100$ ,  $400 \times 200$ , and  $800 \times 400$  points. The results from this set of simulations were used to compute the spatial order of convergence. Fig. 10 shows the  $L^2$ ,  $L^1$ , and  $L^\infty$  errors for density from each of the grid resolutions. The error computed on the coarse grid and each refinement zone are shown separately. Since the cut-stencil method is not used in this case and the domain is periodic, the optimal order of the interior scheme should be obtained. Since a fifth-order hybrid scheme is being used, the expected order of convergence is five. As all the errors show, the desired value of five is obtained.

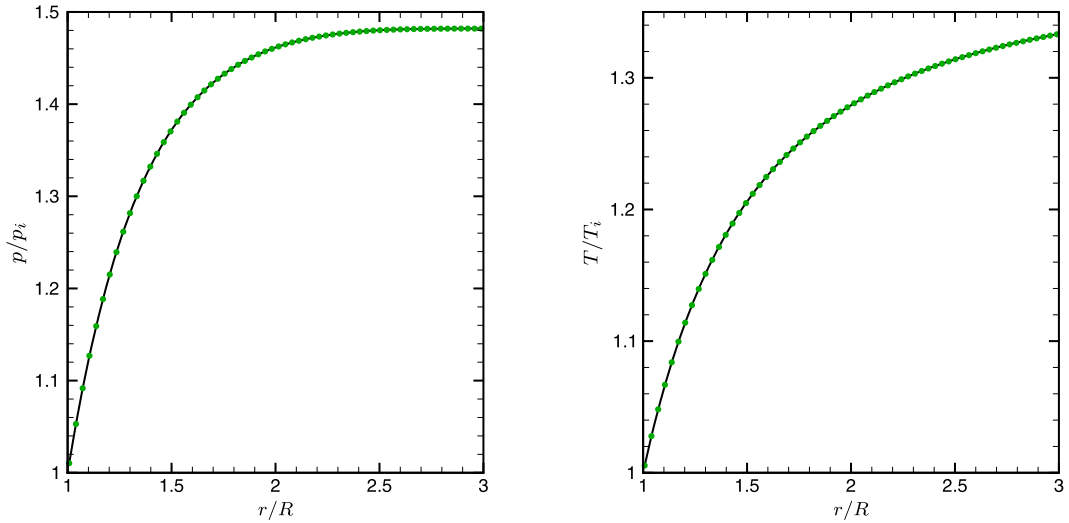
In the second set of simulations, the grid was kept constant while the CFL number was changed. Four simulations were performed with a CFL number of 0.9, 0.81, 0.729, and 0.6561. The results from this set of simulations were used to compute the temporal order of convergence. Fig. 11 shows the  $L^2$ ,  $L^1$ , and  $L^\infty$  errors for density from each of the CFL values. The error computed on the coarse grid and each refinement zone are shown separately. Since a third-order Runge–Kutta scheme is used to advance the simulation in time and the multi-zone refinement uses a fourth-order Hermite interpolation in time, the expected temporal order of convergence is three. All three errors show the expected order of convergence.

## 4. Cut-stencil and immersed boundary method comparison

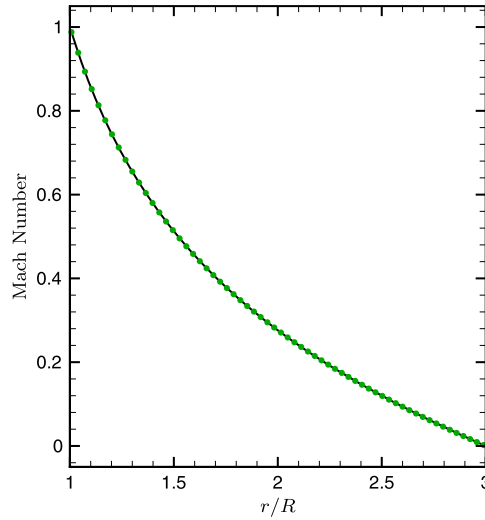
To help evaluate the performance of the cut-stencil method, an immersed boundary method was also implemented for comparison. The immersed boundary method used here comes from Marxen and Iaccarino [34] and von Terzi et al. [35]. This version of the immersed boundary method was selected because Marxen and Iaccarino have demonstrated that the method works for supersonic flow over isolated roughness elements. A two-dimensional test case was used to evaluate and compare the cut-stencil method and the immersed boundary method. The test case is Mach 4.8 boundary-layer flow over an isolated roughness element constructed from hyperbolic tangents. This simulation was also performed by Marxen and Iaccarino. The use of a smooth roughness geometry allows this simulation to also be performed on a body-fitted grid. The results from the body-fitted grid matched very well with results from Marxen and Iaccarino and is assumed to be correct. The body-fitted results are used as a basis for comparison for the cut-stencil and immersed boundary methods.

A diagram of the test case is shown in Fig. 12 where flow is from left to right. The problem consists of boundary-layer flow over a flat plate. The  $x$  axis is aligned with the flow direction and the  $y$  axis is normal to the plate. An isolated roughness element constructed from hyperbolic tangent curves is located on the flat plate. The roughness height to undisturbed boundary-layer thickness ratio ( $k/\delta$ ) is 0.547. The equation for the roughness geometry is

$$h(x) = \frac{k}{2} \left[ \tanh(S_r(x + L_r)) - \tanh(S_r(x - L_r)) \right] \quad (17)$$



(a) Pressure profile. Non-dimensionalized by pressure at inner wall ( $p_i$ ). (b) Temperature profile. Non-dimensionalized by temperature at inner wall ( $T_i$ ).



(c) Mach profile.

**Fig. 7.** Pressure, temperature, and Mach number profiles for circular Couette flow. Solution computed on  $200 \times 200$  grid. Computed Solution  $\bullet$ ; Exact Solution  $—$ .

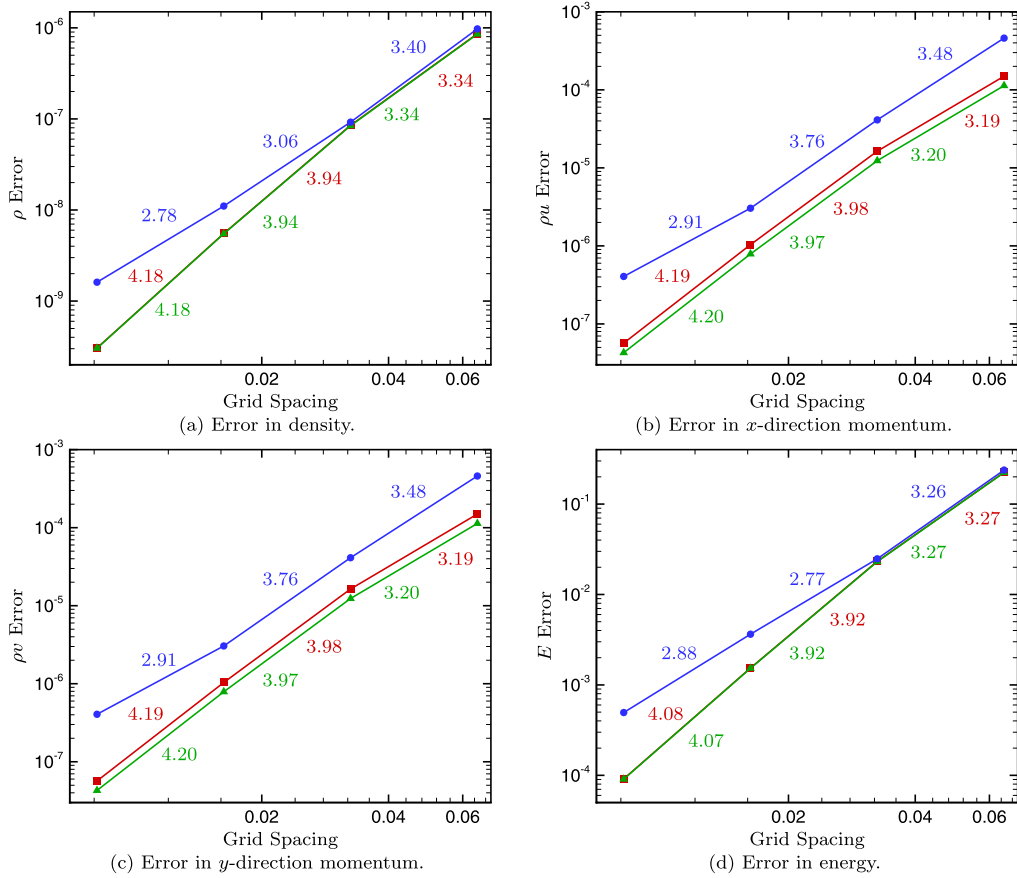
where  $S_r = 20$  and  $L_r = 0.2$ . The freestream conditions for the flow are summarized in Table 4 where  $Re_\infty = \rho_\infty u_\infty L_{ref} / \mu_\infty$ .

The leading edge of the flat plate is not included in the simulation. Instead, a compressible similarity solution is imposed at the inlet of the numerical domain and held fixed with time. The outlet of the domain is mostly supersonic so the conservative variables are extrapolated from the interior of the domain. At the top of the domain, the  $u$  velocity component and temperature are set to their freestream values while the derivative of the  $v$  velocity component and pressure with respect to  $y$  was set to zero. In addition, a sponge layer [36] is used at the top of the domain to prevent any flow structures created by the roughness from reflecting back into the domain. The flat plate and the roughness are no-slip adiabatic walls.

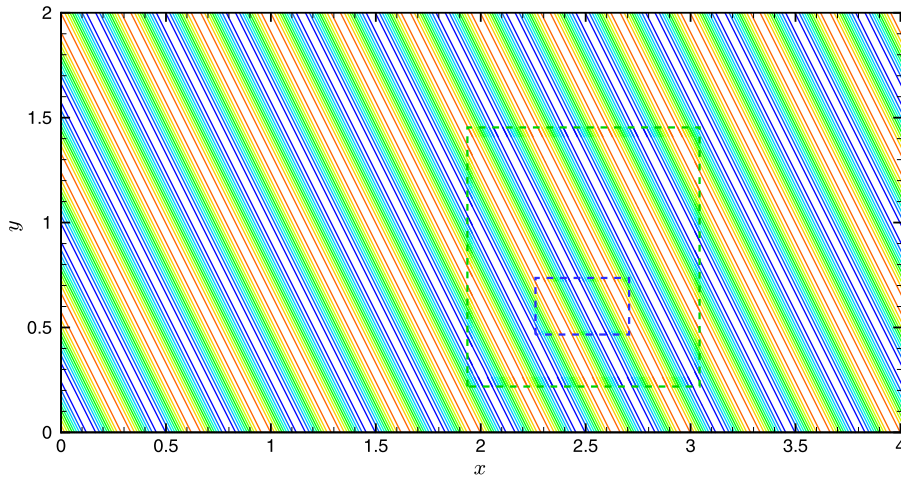
The grids in the vicinity of the roughness element for the body-fitted and cut-stencil method are shown in Fig. 13. The grids use 800 points in the  $x$  direction and 400 points in the  $y$  direction. The grids were generated by the following equations:

$$x = (x_c - x_{in}) \left( 1 + \frac{\sinh(a_1)}{\sinh(a_2)} \right) + x_{in}, \tag{18}$$

$$y = (H - B(x)) \frac{\sinh(S_\eta \eta)}{\sinh(S_\eta)} + B(x). \tag{19}$$



**Fig. 8.** Error in the conservative variables versus grid spacing for circular Couette flow. The number next to each line segment is the order of convergence computed using the line segment end points.  $L^\infty$  error  $\bullet$ ;  $L^1$  error  $\blacktriangle$ ;  $L^2$  error  $\blacksquare$ .

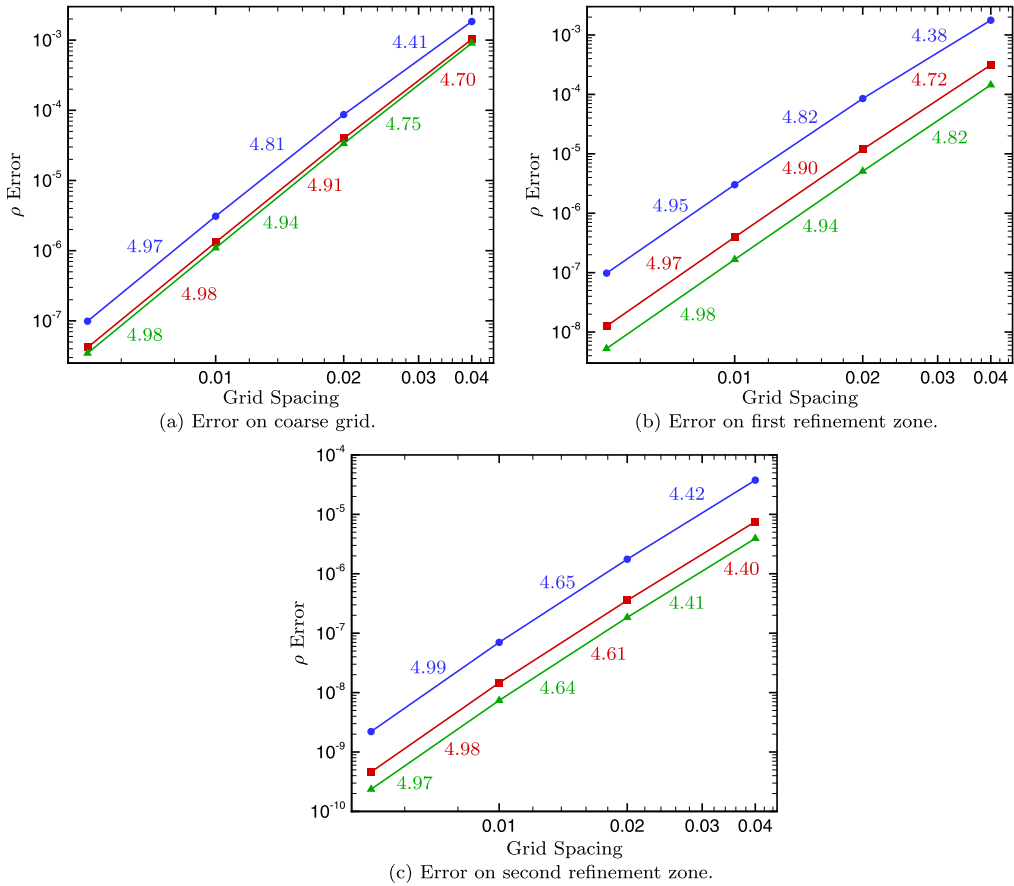


**Fig. 9.** Density contour for propagating density wave problem. First refinement zone edge  $- - -$ ; Second refinement zone edge  $- - -$ .

Where

$$a_1 = S_\xi \xi - \frac{1}{2} \log \left( \frac{b_+}{b_-} \right),$$

$$a_2 = \frac{1}{2} \log \left( \frac{b_+}{b_-} \right), \tag{20}$$



**Fig. 10.** Error in density for propagating density wave problem with different grid spacing. The number next to each line segment is the order of convergence computed using the line segment end points.  $L^\infty$  error — blue circles;  $L^1$  error — green triangles;  $L^2$  error — red squares.

**Table 4**  
Flow conditions for hyperbolic tangent roughness simulation.

Freestream Mach number	$M_\infty$	4.8
Freestream velocity	$u_\infty$	720 m/s
Freestream pressure	$p_\infty$	239,426 Pa
Freestream temperature	$T_\infty$	55.4 K
Freestream density	$\rho_\infty$	$1.51 \times 10^{-2}$ kg/m <sup>3</sup>
Freestream Reynolds number	$Re_\infty$	$10^5$
Reference length	$L_{ref}$	$3.365 \times 10^{-2}$ m
Roughness height	$k$	$0.1L_{ref}$
Ratio of roughness height to boundary-layer thickness	$k/\delta$	0.547

$$b_{\pm} = 1 + (\exp(\pm S_{\xi}) - 1) \frac{x_c - x_{in}}{x_{out} - x_{in}}.$$

$\xi$  and  $\eta$  are the computational domain variables and vary from 0 to 1.  $x_{in}$  and  $x_{out}$  are the  $x$  locations of the domain inlet and outlet, respectively.  $x_c$  is the center of the clustering in the  $x$  direction.  $H$  is the domain height.  $B(x)$  is the function that defines the shape of the bottom edge of the domain. For the body-fitted grid,  $B(x)$  is equal to the function governing the roughness geometry,  $h(x)$ . For the cut-stencil method and the immersed boundary method,  $B(x)$  is simply zero.  $S_{\xi}$  and  $S_{\eta}$  control the amount of clustering in the  $x$  and  $y$  directions, respectively. The grids were generated with  $S_{\xi} = 10$ ,  $S_{\eta} = 3.6$ , and  $x_c/\delta = 0.0$ . The numerical domain starts at  $x/\delta = -50.0$  and ends at  $x/\delta = 83.0$ . The top of the domain is located at  $y/\delta = 8.62$ . The grid used for the immersed boundary method is the same as the cut-stencil method grid except 600 points are used in the  $y$  direction.

Fig. 14 shows contour plots of pressure, density, and  $v$  velocity in the vicinity of the roughness element for the three methods. The results from the body-fitted grid and the cut-stencil method match exactly. Both results were confirmed to be grid converged by increasing the grids points in the  $x$  direction to 1200 and the points in the  $y$  direction to 600. There were no visible changes in the results. The results from the immersed boundary method appear similar to the body-fitted

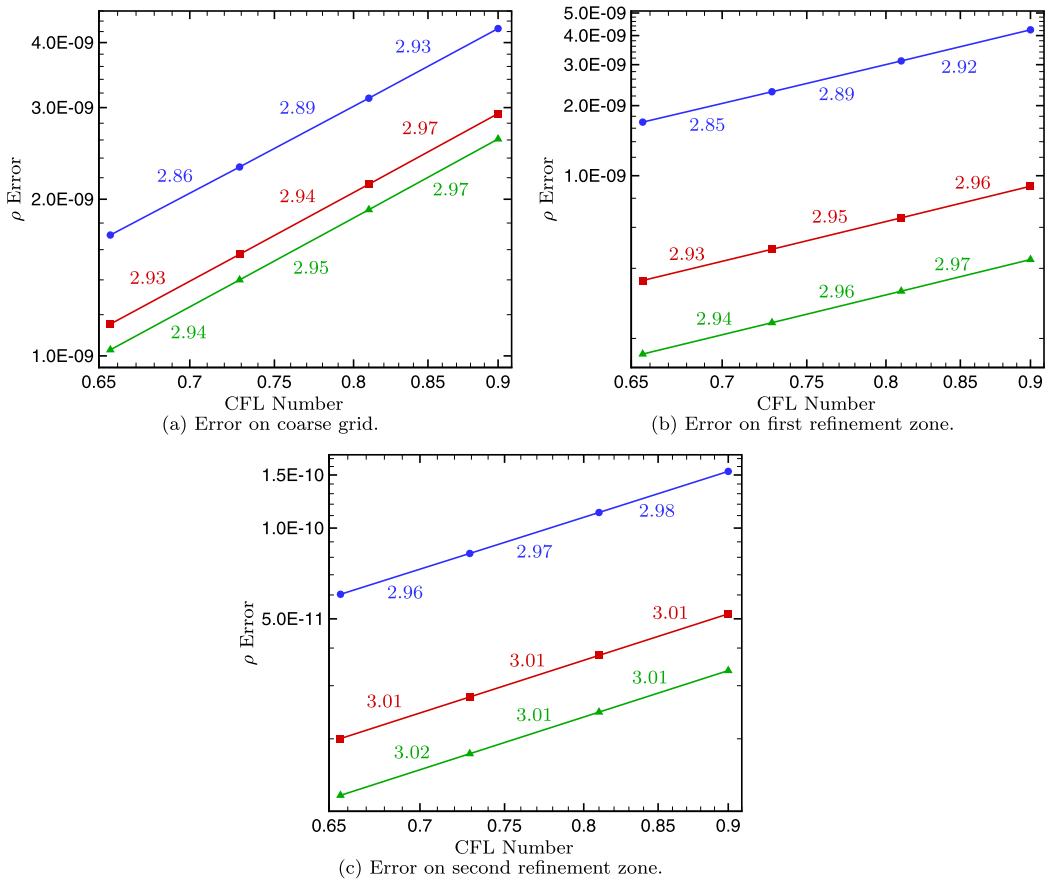


Fig. 11. Error in density for propagating density wave with different time step size. The number next to each line segment is the order of convergence computed using the line segment end points.  $L^\infty$  error —●—;  $L^1$  error —▲—;  $L^2$  error —■—.

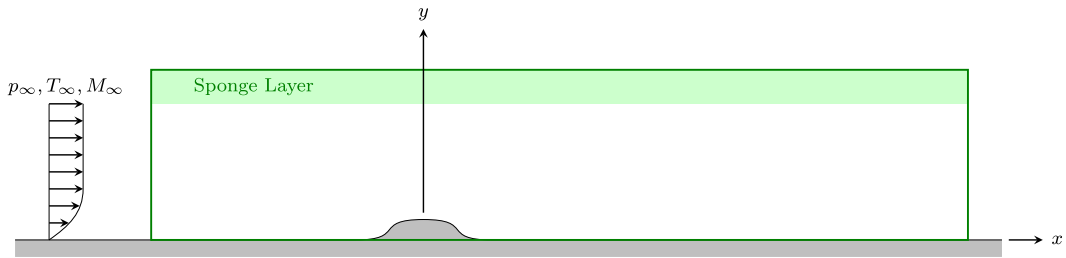


Fig. 12. Diagram for boundary-layer flow over two-dimensional hyperbolic tangent roughness. Numerical domain edge shown as —■—.

results away from the roughness but there are significant differences near the roughness. The immersed boundary method results were not checked for grid convergence due to the method’s low spatial convergence rate.

Fig. 15 shows  $u$  velocity and pressure profiles at various  $x/\delta$  locations for the three methods. The profiles are at the upstream curved portion of the roughness ( $x/\delta = -1.0$ ), the center of the roughness ( $x/\delta = 0.0$ ), and the downstream curved portion of the roughness ( $x/\delta = 1.0$ ). Just as with the contours, the profiles for the body-fitted grid and cut-stencil method match. There are noticeable differences between the body-fitted and immersed boundary method, which are greatest near the roughness.

Despite the immersed boundary simulation being performed on a slightly denser grid, the cut-stencil method was able to reproduce the body-fitted results while the immersed boundary results still showed significant differences. It may be possible to improve the immersed boundary results. There are many slight variations of the immersed boundary method but only one was tested here. However, all the variations will use a  $\delta$  function to apply the forcing. This will always extend the influence of the body force away from the wall. The cut-stencil method does not suffer from this since the boundary conditions are applied only at the boundary.

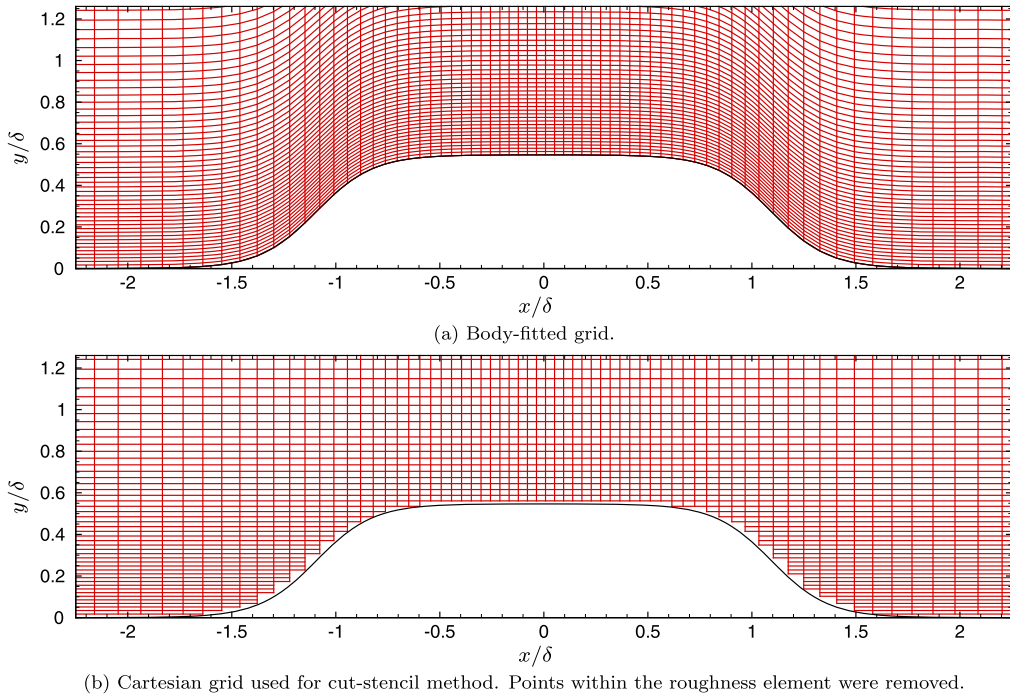


Fig. 13. Grids in vicinity of the two-dimensional hyperbolic tangent roughness element. Only every fourth point in both directions shown.

Table 5  
Flow conditions for subsonic flow over a cylinder.

Freestream Mach number	$M_\infty$	0.2
Freestream velocity	$u_\infty$	31.0 m/s
Freestream pressure	$p_\infty$	400.0 Pa
Freestream temperature	$T_\infty$	60.0 K
Freestream density	$\rho_\infty$	$2.32 \times 10^{-2}$ kg/m <sup>3</sup>
Reynolds number	$Re_D$	150

## 5. Results

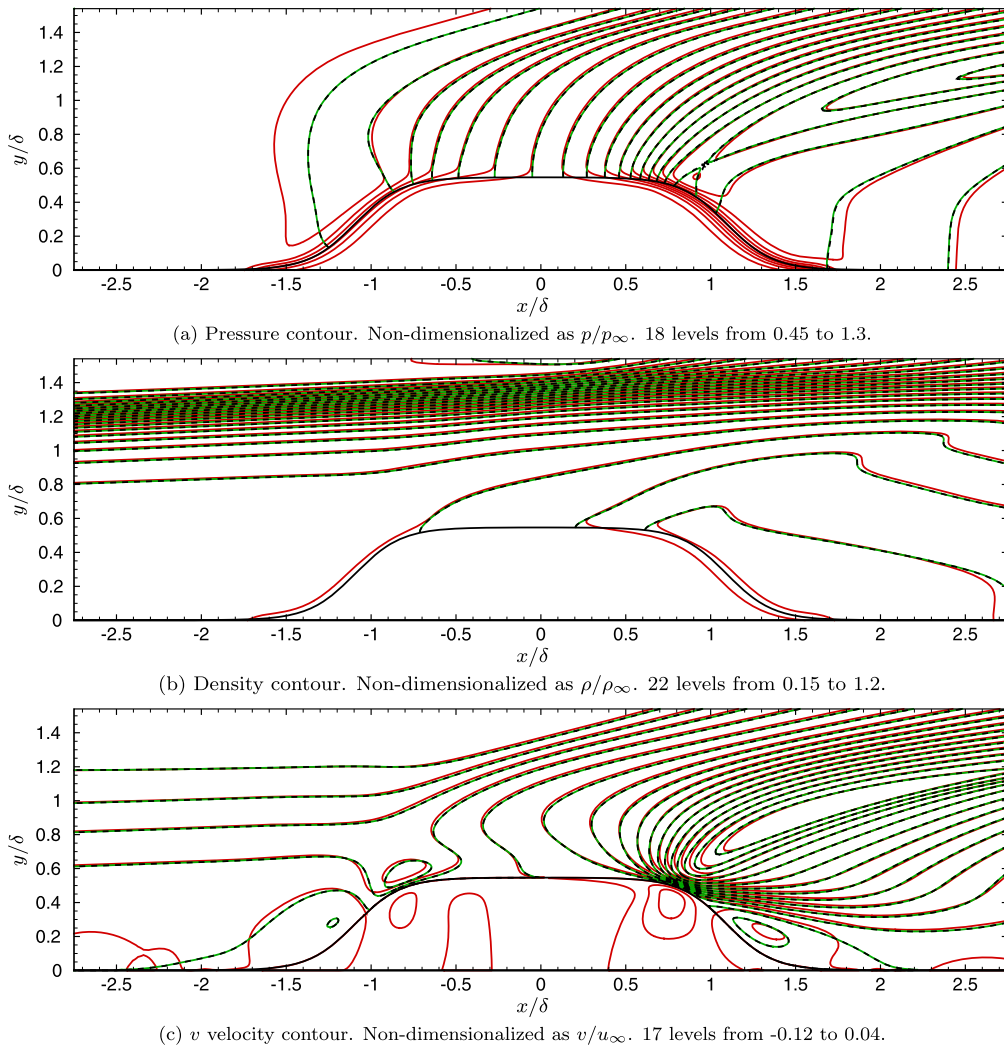
### 5.1. Subsonic flow over a cylinder

The first set of results being presented is for Mach 0.2 flow over a cylinder. The purpose of this simulation was to test the cut-stencil and multi-zone refinement methods on an unsteady problem. Since the multi-zone method requires interpolation in time, it was important to test the method on an unsteady problem. Although we solve the compressible Navier–Stokes equations, this incompressible problem was selected due to its widely known solution. The Reynolds number for the flow based on the cylinder diameter ( $D$ ) is 150. This places the solution in the periodic vortex shedding regime. The freestream conditions are provided in Table 5.

Fig. 16 shows the grid used in the simulation. The red grid is the coarse grid, which covers the entire domain. The grid is uniformly spaced in both directions. The coarse grid has 300 grid points in the  $x$  direction and 150 grid points in the  $y$  direction. The first refinement zone is shown in green and the second in blue. Each refinement zone increases the resolution by a factor of three. The figure only shows a portion of the domain near the cylinder. In the  $x$  direction, the domain starts at  $x/D = -5$  and ends at  $x/D = 15$ . The domain has a length of  $10D$  in the  $y$  direction and is centered at  $y/D = 0$ . Non-reflective boundary conditions [37] are used along all edges of the domain. A sponge layer [36] was also used at the inlet with the freestream conditions used as the target. This ensured that the non-reflective inlet did not deviate from the desired freestream conditions. The cylinder has no-slip adiabatic walls.

Fig. 17a shows contours of vorticity for the flow after reaching a periodic state. The results show that the vortices are able to pass across the refinement zone boundaries unimpeded. Fig. 17a also shows the location of four probes placed in the flow. Once the solution reached a periodic state, the pressure at each probe location was recorded as a function of time. The dominant frequency of the pressure history was computed using a fast Fourier transform (FFT). The results of the FFT are shown in Fig. 17b. The different line colors correspond to the same colored probe in Fig. 17a. Fig. 17b shows that all the probes measured the same dominant frequency. This suggests that the multi-zone refinement did not have a noticeable





**Fig. 14.** Contours in vicinity of the two-dimensional hyperbolic tangent roughness element. Immersed boundary method —; Cut-stencil method —; Body-fitted grid - - -.

negative effect on the solution. The first peak measured by all the probes is at a Strouhal number of 0.185. The remaining peaks are harmonics of the first.

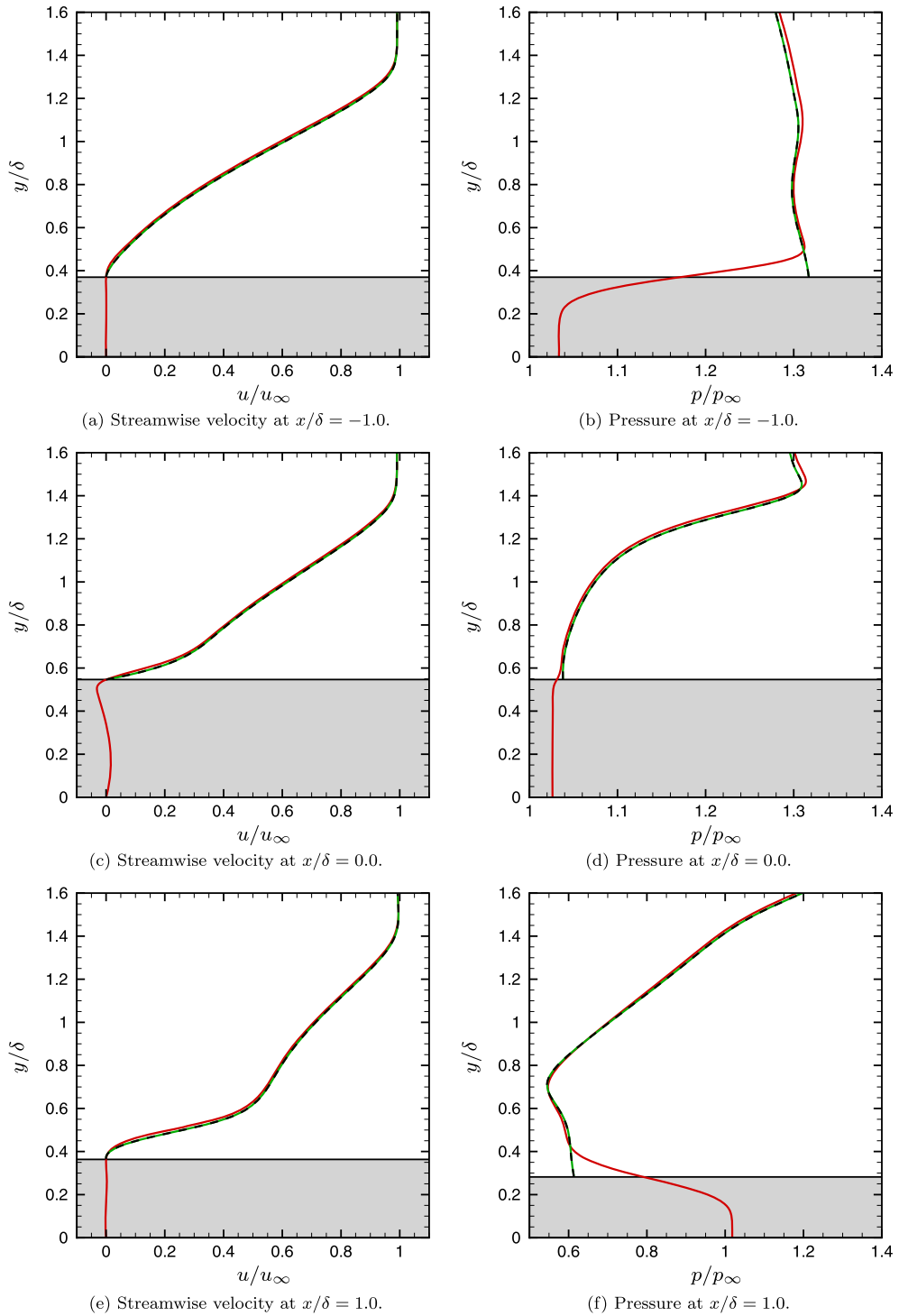
Fig. 18 is a plot taken from the cylinder wake review paper by Williamson [38]. The plot shows the Strouhal number versus Reynolds number for the current problem. The data comes from a number of different experiments. The Strouhal number from the current simulation was added to the plot. The current results agree very well with the experimental data. These results suggest that the cut-stencil method and the multi-zone refinement are working correctly.

## 5.2. Hyperbolic tangent roughness

The second set of results is for Mach 4.8 boundary-layer flow over a three-dimensional isolated roughness element constructed from hyperbolic tangent curves. The purpose of this simulation was to test the code on a problem closer to the problem of interest, hypersonic flow over an isolated roughness element. This simulation is a three-dimensional version of the case used to compare the cut-stencil and immersed boundary methods in Section 4. The  $x$  and  $y$  axis are still in the flow direction and normal to the wall, respectively. The  $z$  axis is in the spanwise direction. The roughness height to undisturbed boundary-layer thickness ratio ( $k/\delta$ ) is still 0.547. The equation for the three-dimensional roughness geometry is

$$h(x, z) = \frac{k}{4} h_x(x) h_z(z) \quad (21)$$

where

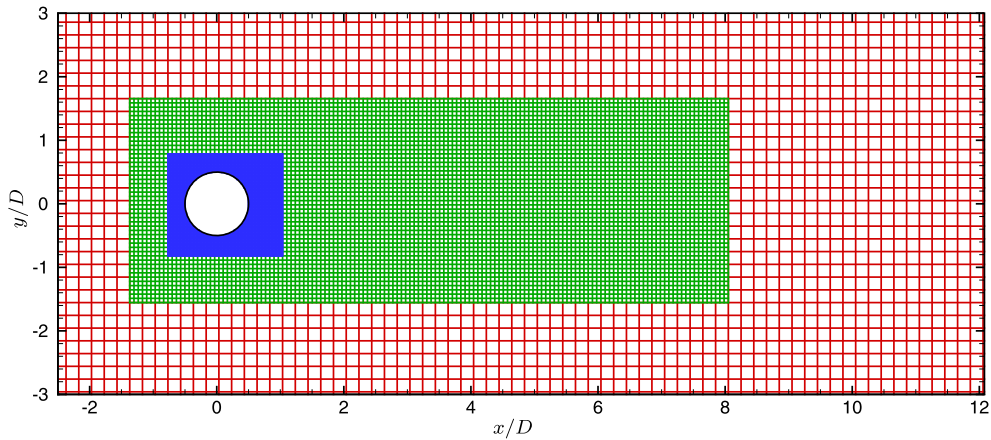


**Fig. 15.** Streamwise velocity and pressure profiles for two-dimensional hyperbolic tangent roughness element. Roughness element shown in gray. Immersed boundary method —; Cut-stencil method —; Body-fitted grid - - -.

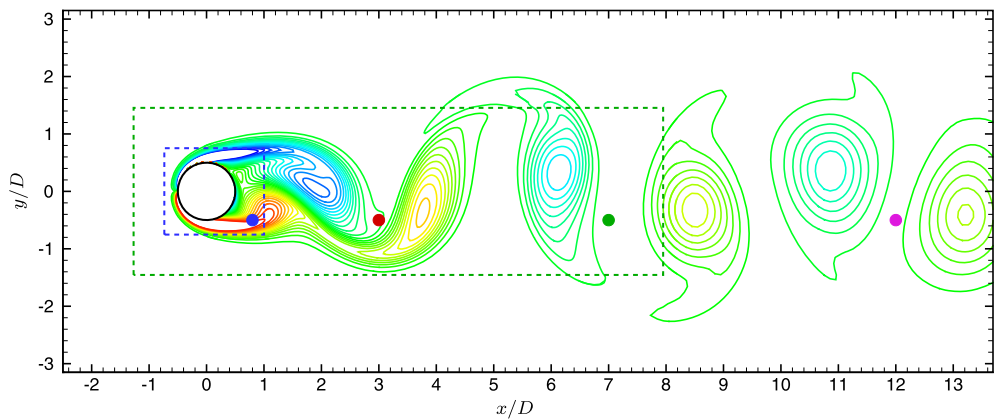
$$h_x(x) = \tanh(S_r(x + L_r)) - \tanh(S_r(x - L_r)), \tag{22}$$

$$h_z(z) = \tanh(S_r(z + L_r)) - \tanh(S_r(z - L_r)). \tag{23}$$

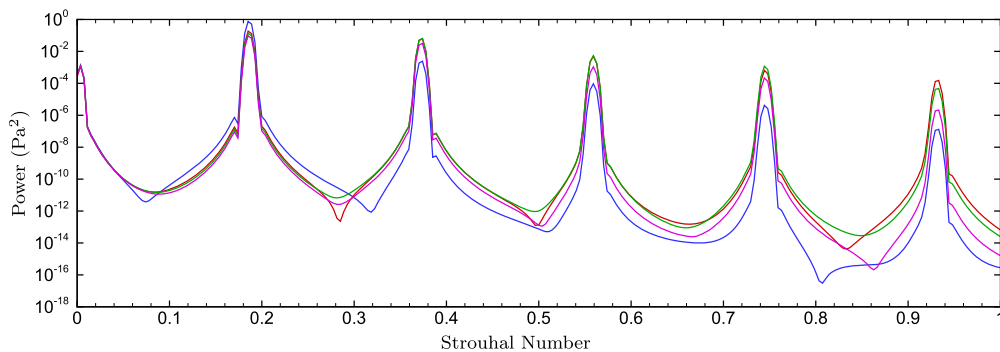
The same values for  $S_r$  and  $L_r$  are used here ( $S_r = 20$ ,  $L_r = 0.2$ ). The same freestream conditions are used in this simulation and are summarized in Table 4.



**Fig. 16.** Grid in vicinity of cylinder. Points inside the cylinder are not shown. Coarse grid shown in red. First refinement zone with three times the resolution shown in green. Second refinement zone with an additional three times increase in resolution shown in blue. Only every second point shown in both directions. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)



(a) Vorticity contour. Non-dimensionalized as  $\omega D/u_\infty$  with 30 levels from -4.0 to 4.0. Colored dots denote the location of pressure probes. First refinement zone edge ---; Second refinement zone edge - - -.



(b) FFT of the pressure history at each probe. The line color corresponds to the probe color in Figure (a).

**Fig. 17.** Results from subsonic flow over a cylinder.

The boundary conditions used in this simulation are similar to the ones used in Section 4. A compressible similarity solution is imposed at the inlet of the numerical domain and held fixed with time. At the outlet, the conservative variables are extrapolated from the interior of the domain. At the top of the domain, the  $u$  velocity component and temperature are set to their freestream values while the derivative of the  $v$  velocity component,  $w$  velocity component, and pressure with respect to  $y$  was set to zero. The same sponge layer is also used. The flat plate and the roughness are no-slip adiabatic walls. Symmetry boundary conditions are used at both edges on the domain in the  $z$  direction.

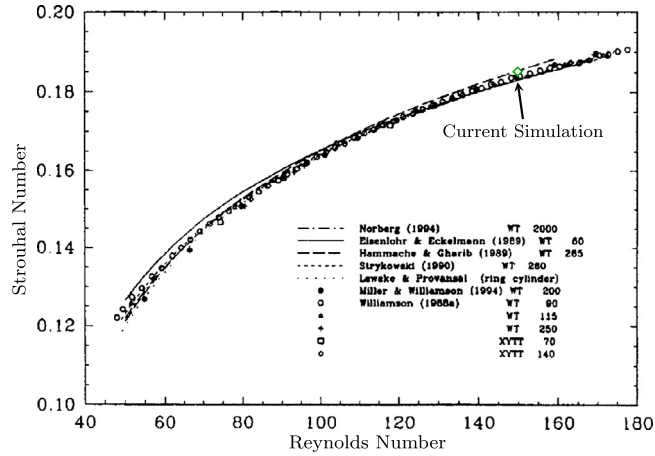


Fig. 18. Strouhal number versus Reynolds number for incompressible flow over a cylinder. Taken from Williamson [38]. Results from current simulation shown as  $\diamond$ .

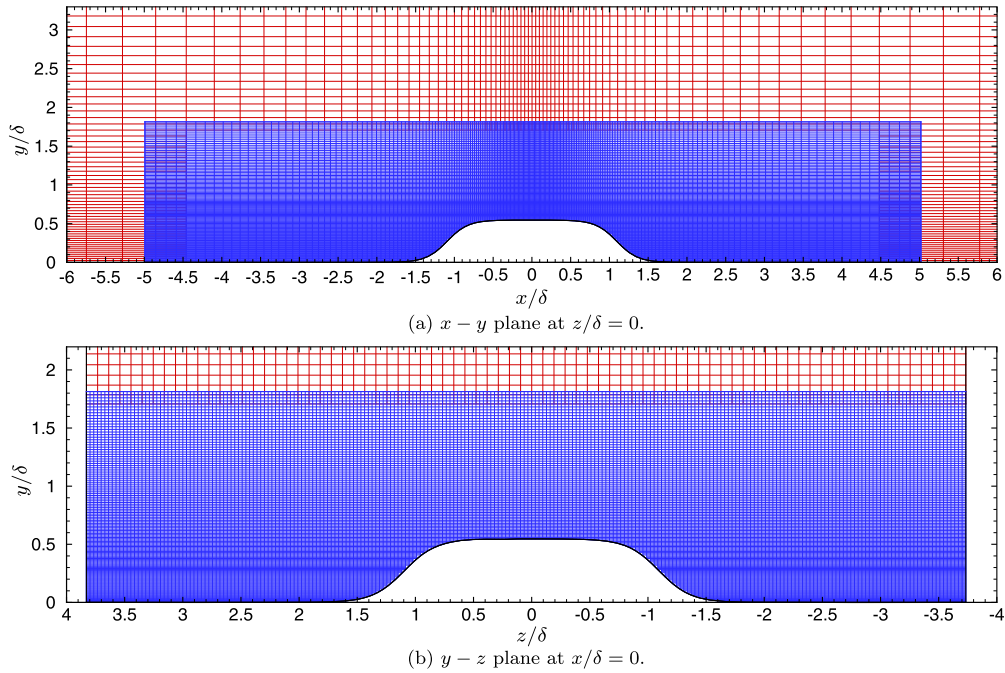
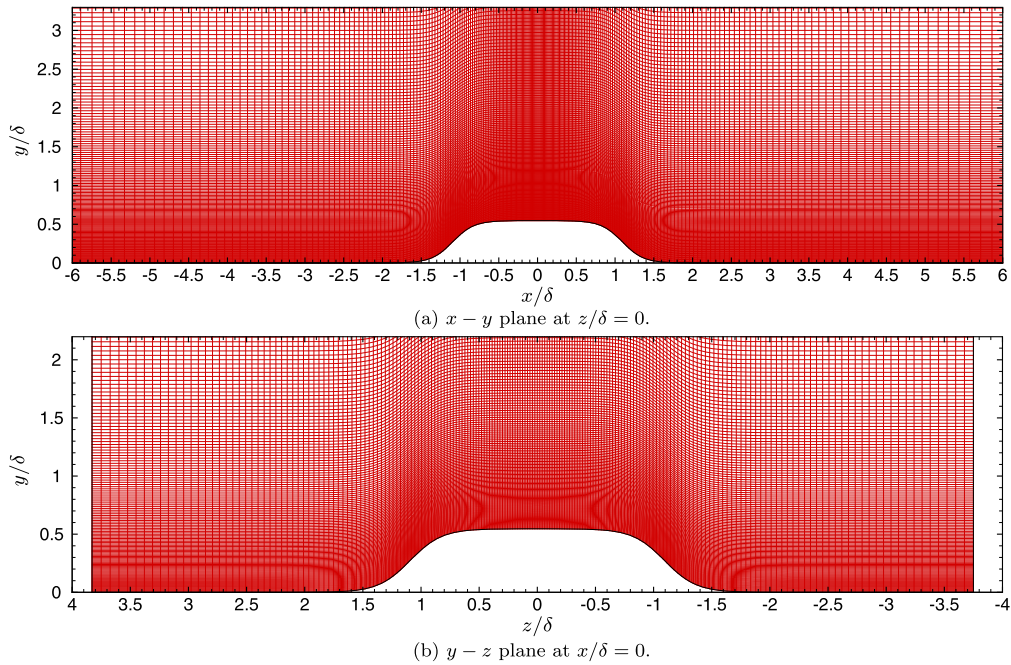


Fig. 19. Grid in the vicinity of the three-dimensional hyperbolic tangent roughness element used in multi-zone cut-stencil simulation. Grid points inside the roughness were removed. Coarse grid shown in red. Refinement zone with three times the resolution shown in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The grid in the vicinity of the roughness is shown in Fig. 19. The coarse grid has 120 points in the  $x$  direction, 80 points in the  $y$  direction, and 80 points in the  $z$  direction. The grids are clustered near the roughness element in the  $x$  direction and near the wall in the  $y$  direction. The grid is uniformly spaced in the  $z$  direction. The grid was generated using Equations (18) and (19) for the  $x$  and  $y$  coordinates, respectively, with  $S_\xi = 10$ ,  $S_\eta = 3.4$ , and  $x_c/\delta = 0.0$ . Only the domain around the roughness element is shown in the figure. The full numerical domain starts at  $x/\delta = -24.0$  and ends at  $x/\delta = 72.0$ . The top of the domain is located at  $y/\delta = 8.61$  and the domain half width is  $W/\delta = 3.82$ . A single refinement zone was used in the simulation, which increased the resolution by a factor of three.

The simulation was also performed on a body-fitted grid for comparison. The body-fitted grid in the vicinity of the roughness is shown in Fig. 20. The grid has 400 points in the  $x$  direction, 200 points in the  $y$  direction, and 200 points in the  $z$  direction. The  $x$  and  $y$  point distribution is given by Equations (18) and (19) with  $S_\xi = 10.0$ ,  $S_\eta = 3.6$ , and  $x_c/\delta = 0.0$ . In Equation (19),  $B(x)$  is now a function of  $x$  and  $z$  and is equal to  $h(x, z)$ . The  $z$  distribution is given by

$$z = W \frac{\sinh(S_\zeta (2k/KL - 1))}{\sinh(S_\zeta)}, \quad k = 1, \dots, KL \tag{24}$$



**Fig. 20.** Body-fitted grid in the vicinity of the three-dimensional hyperbolic tangent roughness element.

**Table 6**

Comparison of runtime for different grid methods. Times are non-dimensionalized by body-fitted grid results.

Body-fitted grid	Cut-stencil method with single grid	Multi-zone cut-stencil method
1.0	0.82	0.38

**Table 7**

Freestream conditions for flow over the Ubuntu logo.

Mach number	$M_\infty$	3.0
Velocity	$u_\infty$	601.4 m/s
Pressure	$p_\infty$	400.0 Pa
Temperature	$T_\infty$	100.0 K
Reynolds number	$Re_D$	5000

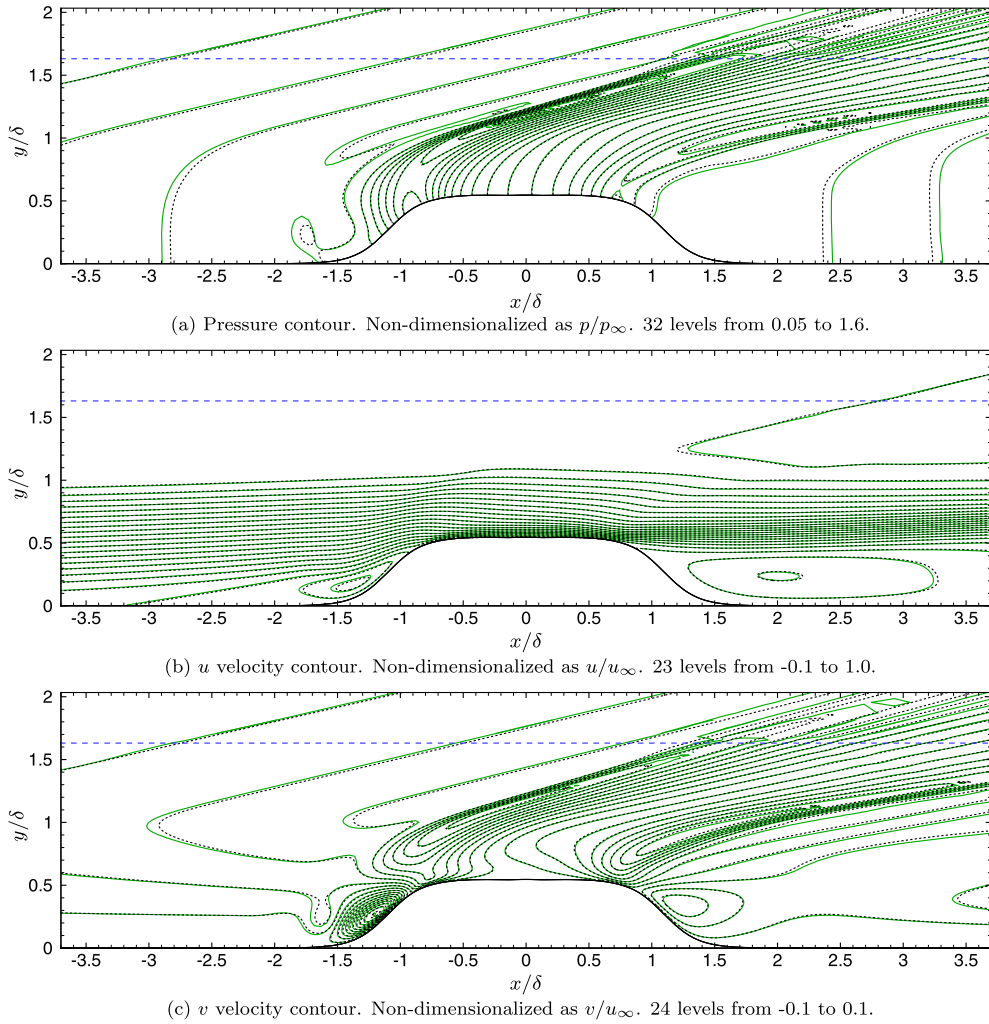
where  $KL$  is the total number of grid points in the  $z$  direction and  $S_\zeta = 2.0$ . The extents of the domain are the same as the multi-zone cut-stencil grid.

The results from the body-fitted grid were compared to results presented by Marxen and Iaccarino [34]. The results matched very well with Marxen and Iaccarino's results and are assumed to be correct. Contours of pressure,  $u$  velocity, and  $v$  velocity near the roughness at the  $z/\delta = 0$  plane are shown in Fig. 21. The results from the two methods agree very well. Only small differences between the body-fitted results and the multi-zone cut-stencil results are visible. Neither set of results were confirmed to be grid converged. Increasing the grid resolution may remove the differences. The results suggest that the code is working properly and can simulate high-speed flow over isolated roughness elements.

Table 6 shows a comparison of the runtime for the body-fitted simulation and the multi-zone cut-stencil simulation. Also included in the table is the runtime for a simulation performed on a single grid with the cut-stencil method. The grid for this simulation has the same number of grid points as the body-fitted grid and was generated using the same equations and constants. Just using the cut-stencil method reduced the runtime of the simulation by 18%. When the multi-zone method was also used, a 62% reduction in runtime compare to the single body-fitted grid method was observed.

### 5.3. Supersonic flow over the Ubuntu logo

The test cases presented so far contained rather simple geometries. To test the method on a more complex geometry, supersonic flow over the logo of the Ubuntu Linux distribution [39] was simulated. The Ubuntu logo is shown in Fig. 22. The freestream flow was set to Mach 3 with a Reynolds number based on the outer diameter of the logo's ring ( $D$ ) of 5000. The freestream conditions are given in Table 7.



**Fig. 21.** Contours in vicinity of three-dimensional hyperbolic tangent roughness element at  $z/\delta = 0$  plane. Multi-zone cut-stencil simulation —; Refinement zone edge - - -; Body-fitted simulation - - -.



**Fig. 22.** Logo for the Ubuntu Linux distribution [39].

Fig. 23a shows the grid used for the simulation. The simulation used a coarse grid and two refinement zones. Both refinement zones increase the resolution by a factor of three. The coarse grid has 300 points in the  $x$  direction and 150 points in the  $y$  direction. The figure only shows the grid near the logo. The coarse grid has a total length of  $40D$  and is centered about the logo. The coarse grid has a height of  $20D$  and is also centered about the logo. Fig. 23b shows the grid near the upstream circle of the logo. The grid was generated by the following equations:

$$x = \frac{L}{2} \left( A_\xi + |2\xi - 1|^{P_\xi} (B_\xi - A_\xi) \right), \tag{25}$$

**Table 8**  
Grid constants for flow over the Ubuntu logo.

$\ell_\xi$	$\ell_\eta$	$S_\xi$	$S_\eta$	$P_\xi$	$P_\eta$
0.195	0.195	5.0	5.0	4	4

$$y = \frac{H}{2} \left( A_\eta + |2\eta - 1|^{P_\eta} (B_\eta - A_\eta) \right), \quad (26)$$

where

$$A_\xi = \ell_\xi (2\xi - 1), \quad (27)$$

$$A_\eta = \ell_\eta (2\eta - 1), \quad (28)$$

$$B_\xi = \frac{\sinh(S_\xi (2\xi - 1))}{\sinh(S_\xi)}, \quad (29)$$

$$B_\eta = \frac{\sinh(S_\eta (2\eta - 1))}{\sinh(S_\eta)}. \quad (30)$$

$\xi$  and  $\eta$  are the computational domain variables and vary from 0 to 1.  $L$  and  $H$  are the total length and height of the domain, respectively. The remaining constants are used to control the grid point clustering and the values are provided in Table 8.

Since the flow is supersonic, the freestream conditions are imposed at the inlet of the domain and held constant with time. The outlet is supersonic so the conservative variables are extrapolated from the interior of the domain. At the top and bottom of the domain, the  $u$  velocity component and temperature are set to their freestream values while the derivative of the  $v$  velocity component and pressure with respect to  $y$  was set to zero. Sponge layers [36] are placed at the top and bottom of the domain to prevent any shocks from reflecting back into the domain. The logo has no-slip adiabatic walls.

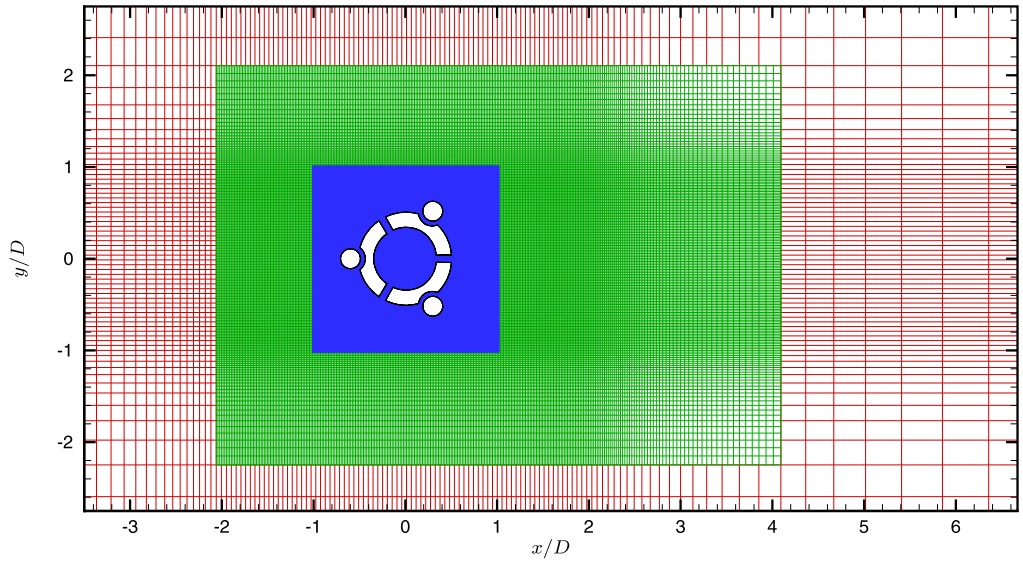
Contours of vorticity and Mach number are shown in Fig. 24. The dominant features of the flow are the bow shock created upstream of the logo and the unsteady wake. The results shown in the figure have reached a periodic state. The figure also demonstrates one of the weaknesses of the current multi-zone refinement method. As the shock passes across a refinement zone boundary, the thickness of the shock increases abruptly. A weak reflected shock can sometimes also be propagated back into the refinement zone. The difference in the shock width and the strength of the reflected shock appears to be proportional to the strength of the shock. In addition, spurious vorticity can be created as the shock crosses the refinement boundary. This is visible in the vorticity contours as the shock passes from the second to the first refinement zone. Generally this is not an issue since the spurious vorticity will be much smaller in magnitude than the physical vorticity generated in the problem. Fig. 25 shows contours of vorticity and Mach number in the vicinity of the logo. Fig. 26 shows the streamlines near the logo. Although there are no results to compare with for this flow, the results appear reasonable. This suggests the code is capable of handling complex geometries.

#### 5.4. Cylindrical roughness element

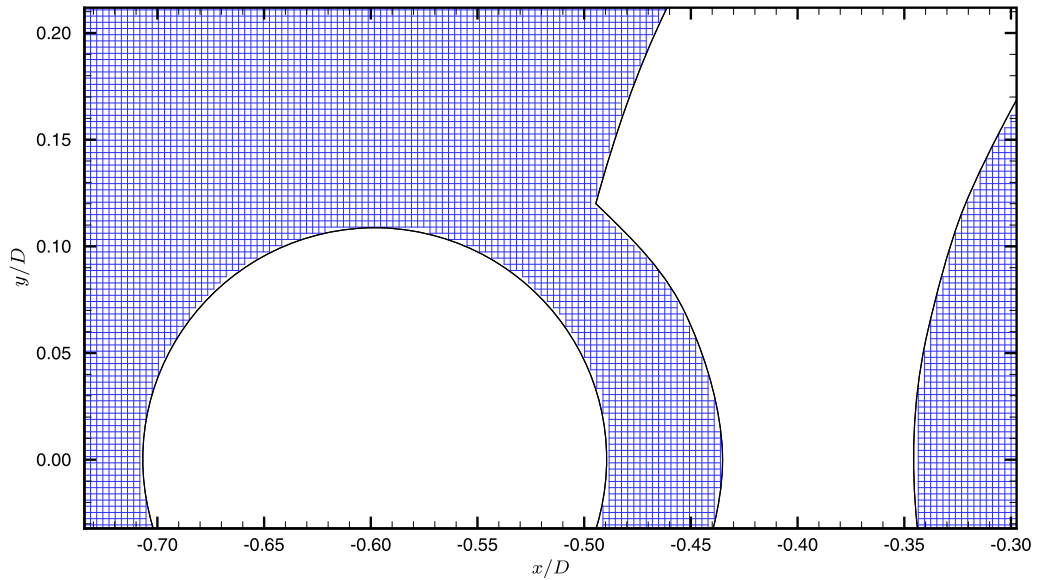
The final set of results being presented is for supersonic flow over a cylindrical roughness element. This simulation is based on experiments performed in the Boeing/AFOSR Mach 6 quiet wind tunnel at Purdue University. The wind tunnel is a Ludwig tube capable of generating quiet Mach 6 flow and is maintained by Professor Steven Schneider [40]. Wheaton and Schneider [41] have recently measured a 21 kHz instability in Pitot probe and hot-wire measurements for Mach 6 flow over a cylindrical roughness element with a roughness height to boundary-layer thickness ratio ( $k/\delta$ ) of 1.10 and a diameter to boundary-layer thickness ratio ( $D/\delta$ ) of 0.647. The roughness Reynolds number ( $Re_k = \rho u k / \mu$ ) was 57,600. This flow has been simulated by Bartkovic et al. [42] and Wheaton et al. [43]. The goal of the current simulation is to reproduce these results. Due to difficulties in simulating the entire wind tunnel, the tunnel geometry was modeled as a flat plate. Although the geometry has changed, the same flow conditions as the Purdue wind tunnel have been used. The flow conditions used in the simulation come from Wheaton and Schneider and are given in Table 9.

The setup for this simulation is very similar to the hyperbolic tangent roughness simulations. The domain starts at  $x/\delta = -20$  and ends at  $x/\delta = 25$ . The top of the domain is located at  $y/\delta = 5$  and the domain half width is  $W/\delta = 7.5$ . The same boundary conditions and sponge layer as the three-dimensional hyperbolic tangent roughness are used in this simulation except the flat plate and roughness are now isothermal. A coarse grid and two refinement zones were used for this simulation. The grid in the vicinity of the roughness is shown in Fig. 27. The coarse grid has 200 points in the  $x$  direction, 140 points in the  $y$  direction, and 121 points in the  $z$  direction. The grid is clustered near the front of the roughness in the  $x$  direction, near the top of the roughness in the  $y$  direction, and near the center of the roughness in the  $z$  direction. The grid was generated with the following equations:

$$x = (x_c - x_{in}) \left( 1 + \frac{\sinh(a_1)}{\sinh(a_2)} \right) + x_{in}, \quad (31)$$



(a) Grid near logo. Only every second point shown in both directions.



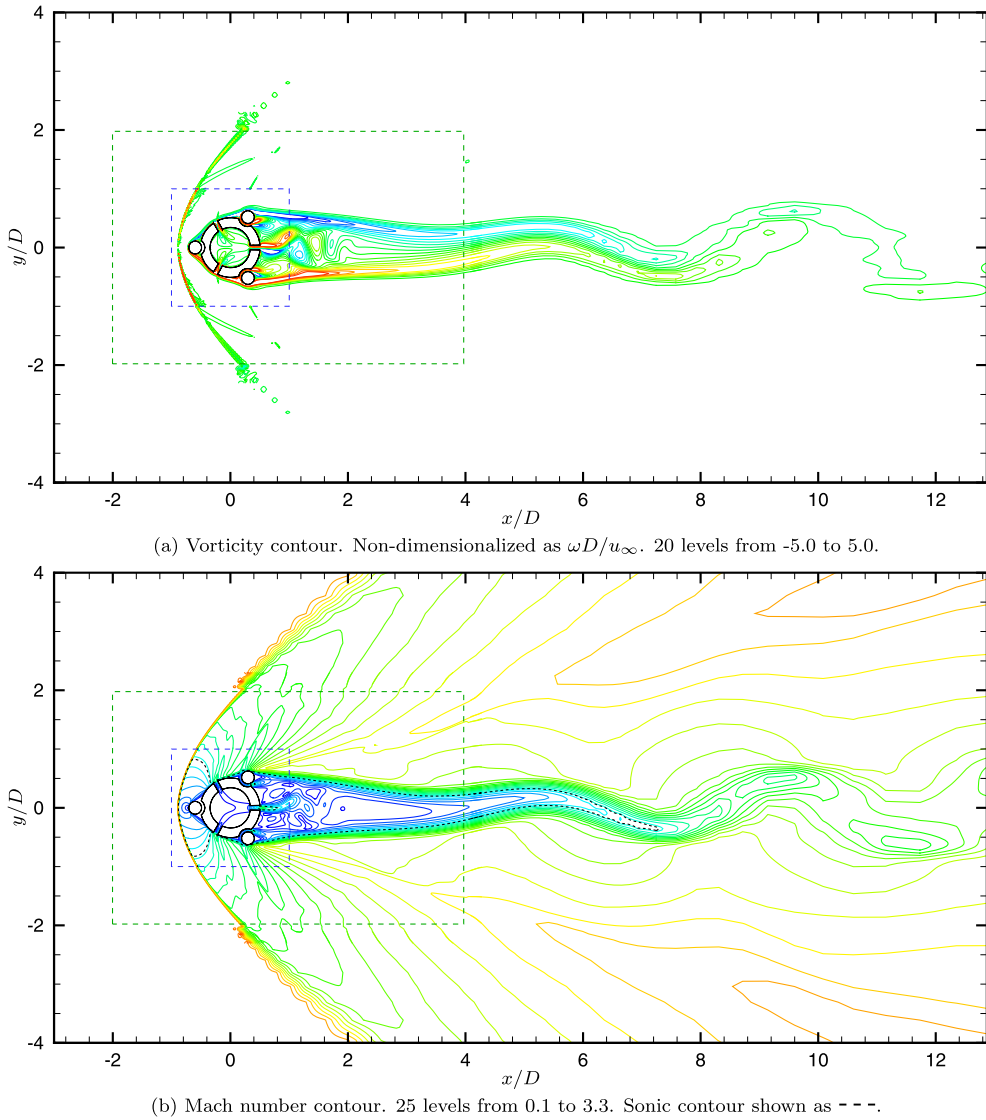
(b) Grid near upstream circle of logo. Every point shown.

**Fig. 23.** Grid for flow over the Ubuntu logo. Points inside the logo are not shown. Coarse grid shown in red. First refinement zone with three times the resolution shown in green. Second refinement zone with an additional three times increase in resolution shown in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 9**  
Flow conditions for cylindrical roughness simulation.

Freestream Mach number	$M_\infty$	5.93
Freestream velocity	$u_\infty$	873 m/s
Freestream pressure	$p_\infty$	0.37 kPa
Freestream temperature	$T_\infty$	53.9 K
Stagnation pressure	$p_0$	551.58 kPa
Stagnation temperature	$T_0$	433 K
Wall temperature	$T_{wall}$	298.2 K
Freestream unit Reynolds number	$Re_\infty$	$5.94 \times 10^6 \text{ m}^{-1}$
Undisturbed boundary-layer thickness	$\delta$	9.23 mm





**Fig. 24.** Vorticity and Mach number contours for flow over the Ubuntu logo. First refinement zone edge ---. Second refinement zone edge ----.

$$y = H \left( A_\eta + |\eta|^{P_\eta} (B_\eta - A_\eta) \right), \quad (32)$$

$$z = W \frac{\sinh(S_\zeta(2\zeta - 1))}{\sinh(S_\zeta)}, \quad (33)$$

where

$$A_\eta = \ell_\eta \eta, \quad (34)$$

$$B_\eta = \frac{\sinh(S_\eta \eta)}{\sinh(S_\eta)}, \quad (35)$$

$$a_1 = S_\xi \xi - \frac{1}{2} \log \left( \frac{b_+}{b_-} \right), \quad (36)$$

$$a_2 = \frac{1}{2} \log \left( \frac{b_+}{b_-} \right), \quad (37)$$

$$b_\pm = 1 + (\exp(\pm S_\xi) - 1) \frac{x_c - x_{in}}{x_{out} - x_{in}}. \quad (38)$$

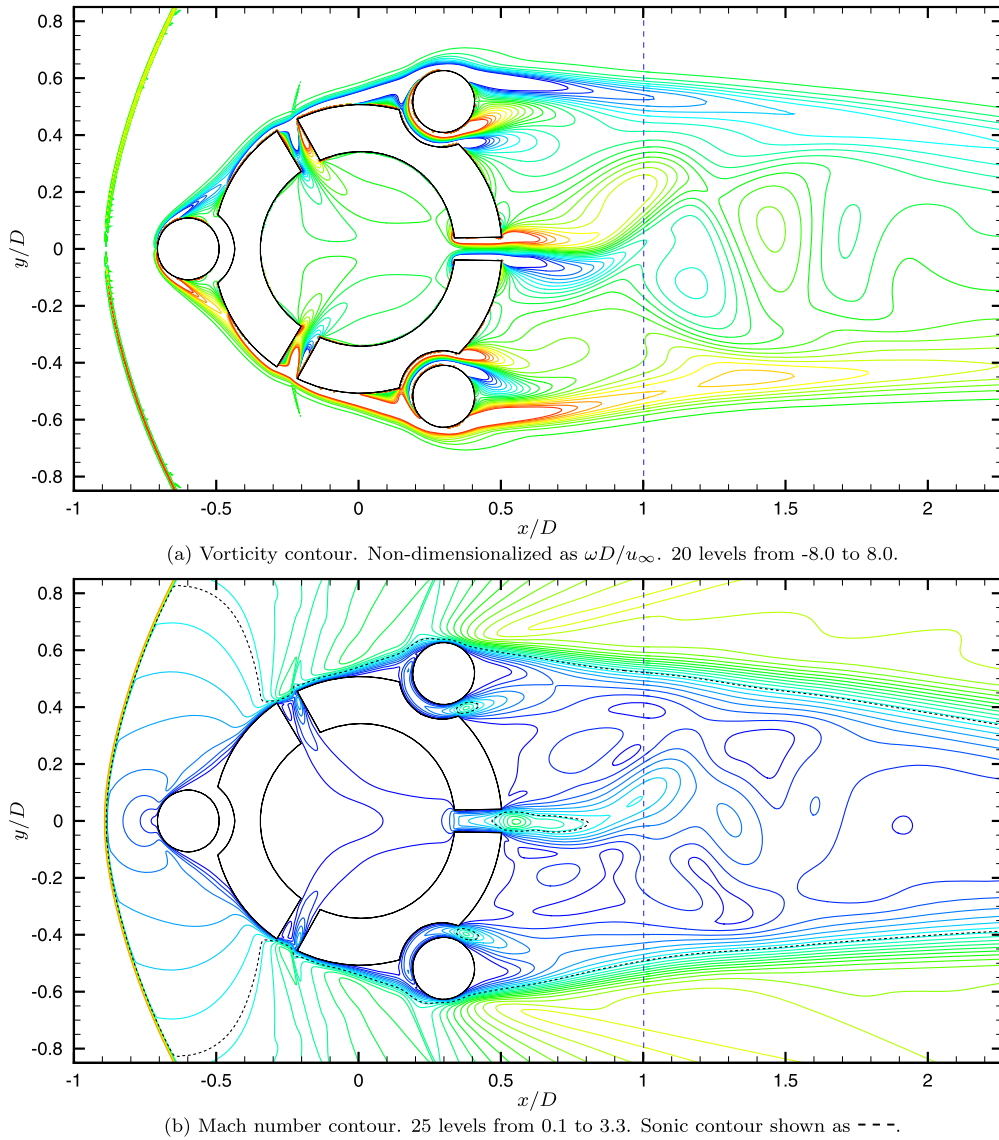


Fig. 25. Vorticity and Mach number contours near the Ubuntu logo. Second refinement zone edge - - -.

Table 10  
Grid constants for cylindrical roughness.

$S_\xi$	$S_\eta$	$S_\zeta$	$\ell_\eta$	$P_\eta$
9.2	4.099	3.5	0.71	1.07136061

The value for the constants used in the expressions are given in Table 10.

Fig. 28 shows an isosurface of the Q-criterion. The surface is colored by the streamwise velocity. The flow field has become unsteady, however it has not reached a statistically stationary state. The flow downstream of the roughness element is still changing significantly with time. Since the entire flow field has not reached a statistically stationary state, a frequency analysis of the instability has not been performed. However, the flow near the roughness element is no longer undergoing large scale changes. This allows preliminary results to be compared with published work from other researchers.

Fig. 29 shows a comparison of the streamlines in front of the roughness element for the current simulation and a simulation performed at similar conditions by Wheaton et al. [43]. The vortical structures from the two simulations seem to match very well. Fig. 30 shows the surface streamlines and wall pressure for the current simulation and a simulation performed by Bartkovicz et al. [42]. The streamlines show the separation and re-attachment lines created by the vortices shown in Fig. 29. Upstream of the roughness, the results in Fig. 30 appear very similar. To the sides and downstream of the

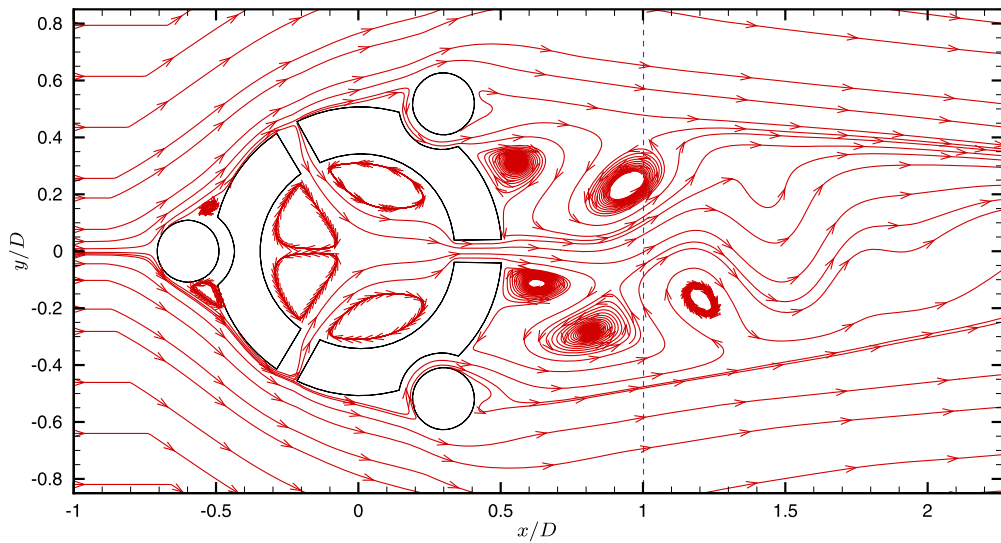
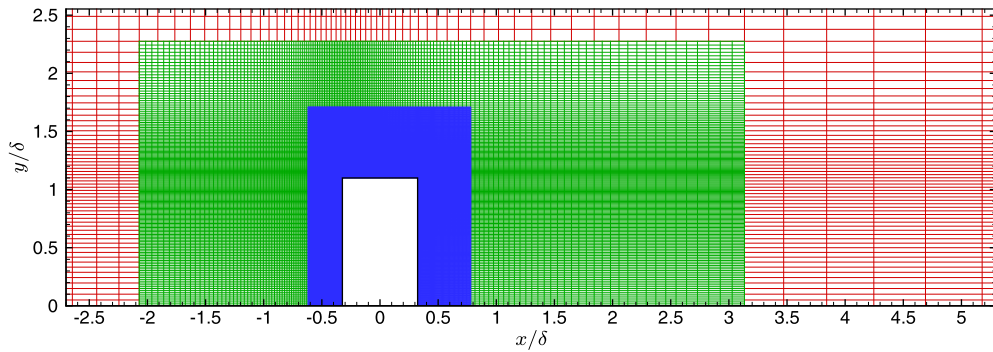
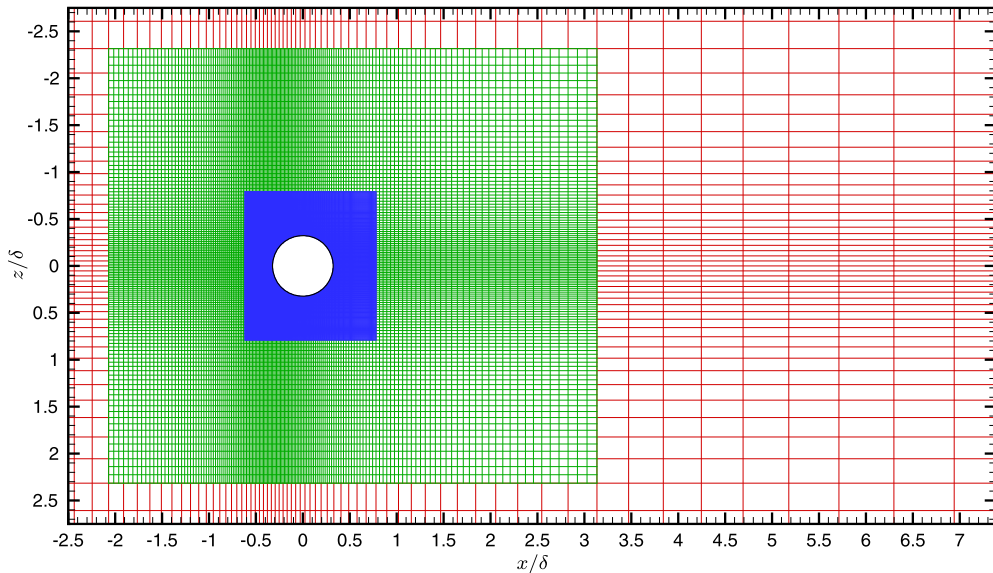


Fig. 26. Streamlines near the Ubuntu logo. Second refinement zone edge - - -.



(a)  $x - y$  plane at  $z/\delta = 0$ .



(b)  $x - z$  plane at  $y/\delta = 0$ .

Fig. 27. Grid in the vicinity of the three-dimensional cylindrical roughness element. Points inside the roughness element are not shown. Coarse grid shown in red. First refinement zone with three times the resolution shown in green. Second refinement zone with an additional three times increase in resolution shown in blue. Only every second point shown in all directions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

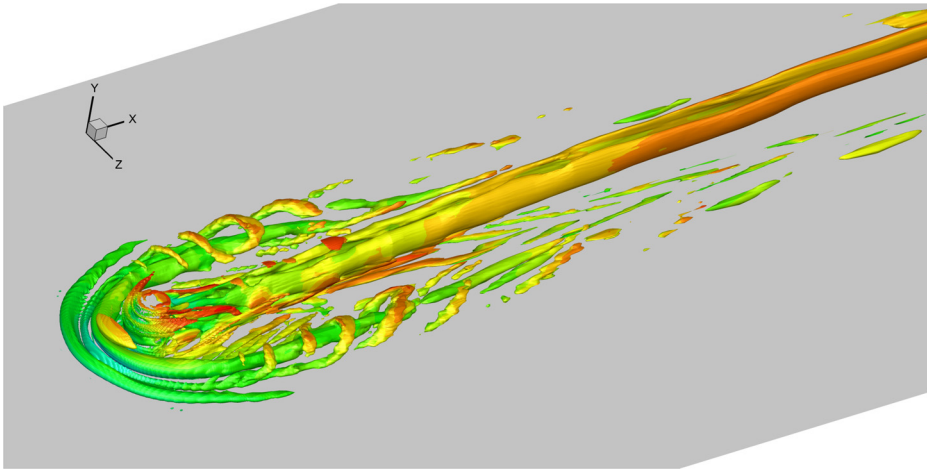


Fig. 28. Isosurface of Q-criterion for flow over cylindrical roughness. Surface for  $Q \delta^2 / u_\infty^2 = 0.2$  shown. Colored by  $u / u_\infty$  with 15 levels from  $-0.5$  to  $0.9$ .

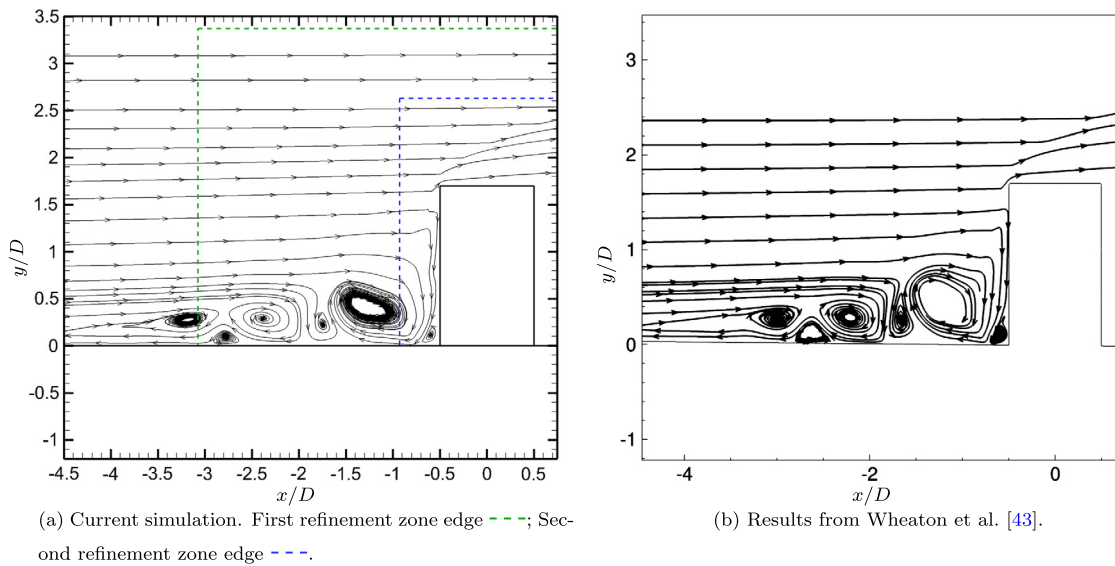


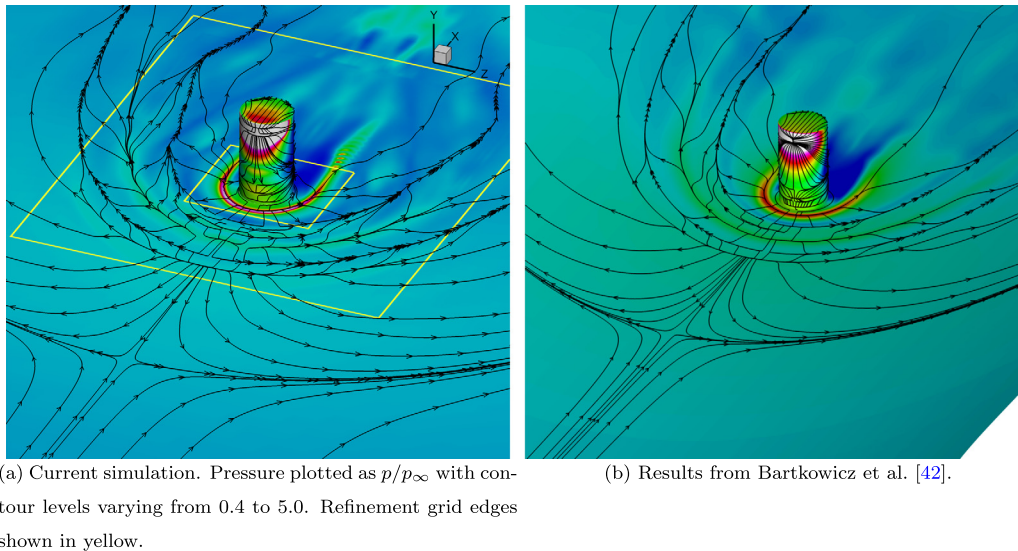
Fig. 29. Comparison of streamlines in  $z/\delta = 0$  plane for flow over cylindrical roughness.

roughness, the streamlines begin to show differences. This may be due to transient effects from the initial condition for the current simulation.

### 6. Conclusion

In this paper, we presented a high-order cut-stencil method for simulating high-speed flows over arbitrary geometries. Simulations were carried out on Cartesian grids with the geometries imposed by the cut-stencil method based on the work of Duan et al. [28]. A different treatment of the pressure and temperature boundary conditions was utilized. The modification addressed an instability observed in the original formulation when simulating flow over taller roughness elements. In addition, the method was reformulated in conservative form. A multi-zone refinement method was also utilized to provide extra control over the placement of grid points. The cut-stencil method was confirmed to provide globally fourth-order convergence when combined with the fifth-order hybrid WENO interior scheme. It was also shown that the multi-zone method does not reduce the order of the interior scheme. The combination of the cut-stencil and multi-zone refinement methods should result in a globally fourth-order scheme.

A simulation of Mach 4.8 flow over a two-dimensional hyperbolic tangent roughness was used to compare the cut-stencil method to the immersed boundary method. Away from the roughness element, the results from both methods matched well with results from a simulation on a body-fitted grid. Near the roughness, however, there were significant differences between the immersed boundary results and the body-fitted results. The cut-stencil results matched the body-fitted results all the way to the roughness wall, demonstrating the advantage of the cut-stencil method in terms of accuracy.



**Fig. 30.** Comparison of surface streamlines and pressure contours for flow over cylindrical roughness.

Results were presented for applying the new method to the numerical simulation of subsonic flow over a cylinder and supersonic flow over a hyperbolic tangent roughness. Both simulations compared well with previous results published by other researchers. For the hyperbolic tangent roughness, a 62% reduction in the runtime compared to a simulation performed on a single body-fitted grid was observed. To demonstrate the method's capability to handle complex geometries, Mach 3 flow over the logo of the Ubuntu Linux distribution was simulated. There are no other published results to compare with, but the current results appear reasonable. Preliminary results for Mach 6 flow over a cylindrical roughness element were also presented. The flow near the roughness qualitatively compared very well with results from similar simulations performed by Bartkowicz et al. [42] and Wheaton et al. [43].

The results presented here have demonstrated the method's ability to handle flows over complex geometries and that it is a promising high-order tool for the study of roughness-induced transition in hypersonic boundary layers. The method's properties also make it well suited for related problems, such as high-speed turbulent flows over rough walls and for aerodynamic sound generation from complex bodies. Work is currently underway on applying the method to such problems.

## Acknowledgements

The authors gratefully acknowledge support by the NASA Fundamental Aeronautics Program, under cooperative agreement NNX07AC39A, monitored by Dr. Meelan Choudhari and the partial support of the AFOSR/NASA National Center for Hypersonic Research in Laminar-Turbulent Transition headed by Professor W. Saric at Texas A&M University. The computer time for this work was provided by the NASA Ames Research Center Advanced Supercomputing Division and the Extreme Science and Engineering Discovery Environment (XSEDE) supported by the National Science Foundation.

## References

- [1] X. Zhong, X. Wang, Direct numerical simulation on the receptivity, instability, and transition of hypersonic boundary layers, *Annu. Rev. Fluid Mech.* 44 (2012) 527–561, <http://dx.doi.org/10.1146/annurev-fluid-120710-101208>.
- [2] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261, <http://dx.doi.org/10.1146/annurev.fluid.37.061903.175743>.
- [3] C.S. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (1972) 252–271, [http://dx.doi.org/10.1016/0021-9991\(72\)90065-4](http://dx.doi.org/10.1016/0021-9991(72)90065-4).
- [4] C.S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (1977) 220–252, [http://dx.doi.org/10.1016/0021-9991\(77\)90100-0](http://dx.doi.org/10.1016/0021-9991(77)90100-0).
- [5] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* 105 (1993) 354–366, <http://dx.doi.org/10.1006/jcph.1993.1081>.
- [6] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, in: *Annual Research Briefs 1997, Center for Turbulence Research*, 1997, pp. 317–327, <http://ctr.stanford.edu/ResBriefs/ARB97.html>.
- [7] R. Beyer, A computational model of the cochlea using the immersed boundary method, *J. Comput. Phys.* 98 (1992) 145–162, [http://dx.doi.org/10.1016/0021-9991\(92\)90180-7](http://dx.doi.org/10.1016/0021-9991(92)90180-7).
- [8] R. Verzicco, M. Fatica, G. Iaccarino, B. Khalighi, P. Moin, Large eddy simulation of a road vehicle with drag-reduction devices, *AIAA J.* 40 (2002) 2447–2455, <http://dx.doi.org/10.2514/2.1613>.
- [9] W.-F. Hu, Y. Kim, M.-C. Lai, An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in Navier–Stokes flows, *J. Comput. Phys.* 257 (2014) 670–686, <http://dx.doi.org/10.1016/j.jcp.2013.10.018>.
- [10] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), *J. Comput. Phys.* 152 (1999) 457–492, <http://dx.doi.org/10.1006/jcph.1999.6236>.

- [11] R.P. Fedkiw, T. Aslam, S. Xu, The ghost fluid method for deflagration and detonation discontinuities, *J. Comput. Phys.* 154 (1999) 393–427, <http://dx.doi.org/10.1006/jcph.1999.6320>.
- [12] D.K. Clarke, H.A. Hassan, M.D. Salas, Euler calculations for multielement airfoils using Cartesian grids, *AIAA J.* 24 (1986) 353–358, <http://dx.doi.org/10.2514/3.9273>.
- [13] H.S. Udaykumar, R. Mittal, W. Shyy, Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids, *J. Comput. Phys.* 153 (1999) 535–574, <http://dx.doi.org/10.1006/jcph.1999.6294>.
- [14] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, *J. Comput. Phys.* 174 (2001) 345–380, <http://dx.doi.org/10.1006/jcph.2001.6916>.
- [15] H.S. Udaykumar, R. Mittal, P. Rampunggoon, Interface tracking finite volume method for complex solid–fluid interactions on fixed meshes, *Commun. Numer. Methods Eng.* 18 (2002) 89–97, <http://dx.doi.org/10.1002/cnm.468>.
- [16] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* 156 (1999) 209–240, <http://dx.doi.org/10.1006/jcph.1999.6356>.
- [17] M.J. Berger, R.J. LeVeque, Stable boundary conditions for Cartesian grid calculations, *Comput. Syst. Eng.* 1 (1990) 305–311, [http://dx.doi.org/10.1016/0956-0521\(90\)90016-E](http://dx.doi.org/10.1016/0956-0521(90)90016-E).
- [18] D. Causon, D. Ingram, C. Mingham, G. Yang, R. Pearson, Calculation of shallow water flows using a Cartesian cut cell approach, *Adv. Water Resour.* 23 (2000) 545–562, [http://dx.doi.org/10.1016/S0309-1708\(99\)00036-6](http://dx.doi.org/10.1016/S0309-1708(99)00036-6).
- [19] R. Mittal, Y. Urturkar, H. Udaykumar, Computational modeling and analysis of biomimetic flight mechanisms, in: 40th AIAA Aerospace Sciences Meeting & Exhibit, 2002, AIAA Paper 2002-0865, <http://dx.doi.org/10.2514/6.2002-865>.
- [20] J. Zhang, M.-J. Ni, A consistent and conservative scheme for MHD flows with complex boundaries on an unstructured Cartesian adaptive system, *J. Comput. Phys.* 256 (2014) 520–542, <http://dx.doi.org/10.1016/j.jcp.2013.08.004>.
- [21] M.J. Berger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984) 484–512, [http://dx.doi.org/10.1016/0021-9991\(84\)90073-1](http://dx.doi.org/10.1016/0021-9991(84)90073-1).
- [22] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989) 64–84, [http://dx.doi.org/10.1016/0021-9991\(89\)90035-1](http://dx.doi.org/10.1016/0021-9991(89)90035-1).
- [23] M.J. Berger, R.J. LeVeque, Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems, *SIAM J. Numer. Anal.* 35 (1998) 2298–2316, <http://dx.doi.org/10.1137/S0036142997315974>.
- [24] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, *J. Comput. Phys.* 153 (1999) 509–534, <http://dx.doi.org/10.1006/jcph.1999.6293>.
- [25] B.E. Griffith, R.D. Hornung, D.M. McQueen, C.S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, *J. Comput. Phys.* 223 (2007) 10–49, <http://dx.doi.org/10.1016/j.jcp.2006.08.019>.
- [26] S.K. Sambasivan, H. Udaykumar, Sharp interface simulations with local mesh refinement for multi-material dynamics in strongly shocked flows, *Comput. Fluids* 39 (2010) 1456–1479, <http://dx.doi.org/10.1016/j.compfluid.2010.04.014>.
- [27] P. Lee, B.E. Griffith, C.S. Peskin, The immersed boundary method for advection–electrodiffusion with implicit timestepping and local mesh refinement, *J. Comput. Phys.* 229 (2010) 5208–5227, <http://dx.doi.org/10.1016/j.jcp.2010.03.036>.
- [28] L. Duan, X. Wang, X. Zhong, A high-order cut-cell method for numerical simulation of hypersonic boundary-layer instability with surface roughness, *J. Comput. Phys.* 229 (2010) 7207–7237, <http://dx.doi.org/10.1016/j.jcp.2010.06.008>.
- [29] C. Shen, J.-M. Qiu, A. Christlieb, Adaptive mesh refinement based on high order finite difference WENO scheme for multi-scale simulations, *J. Comput. Phys.* 230 (2011) 3780–3802, <http://dx.doi.org/10.1016/j.jcp.2011.02.008>.
- [30] G. Jiang, C. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1996) 202–228, <http://dx.doi.org/10.1006/jcph.1996.0130>.
- [31] Y.-X. Ren, M. Liu, H. Zhang, A characteristic-wise hybrid compact–WENO scheme for solving hyperbolic conservation laws, *J. Comput. Phys.* 192 (2003) 365–386, <http://dx.doi.org/10.1016/j.jcp.2003.07.006>.
- [32] A.K. Henrick, T.D. Aslam, J.M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, *J. Comput. Phys.* 207 (2005) 542–567, <http://dx.doi.org/10.1016/j.jcp.2005.01.023>.
- [33] X. Zhong, High-order finite-difference schemes for numerical simulation of hypersonic boundary-layer transition, *J. Comput. Phys.* 144 (1998) 662–709, <http://dx.doi.org/10.1006/jcph.1998.6010>.
- [34] O. Marxen, G. Iaccarino, An immersed boundary method for numerical simulations of boundary layers with roughness, in: Annual Research Briefs 2008, Center for Turbulence Research, 2008, pp. 89–100, <http://ctr.stanford.edu/ResBriefs/ARB08.html>.
- [35] D.A. von Terzi, M.N. Linnick, J. Seidel, H.F. Fasel, Immersed boundary techniques for high-order finite-difference methods, in: 15th AIAA Computational Fluid Dynamics Conference, 2001, AIAA Paper 2001-2918, <http://dx.doi.org/10.2514/6.2001-2918>.
- [36] S. Xu, M. Martin, Assessment of inflow boundary conditions for compressible turbulent boundary layers, *Phys. Fluids* 16 (2004) 2623–2639, <http://dx.doi.org/10.1063/1.1758218>.
- [37] T. Poinso, S. Lele, Boundary conditions for direct simulations of compressible viscous flows, *J. Comput. Phys.* 101 (1992) 104–129, [http://dx.doi.org/10.1016/0021-9991\(92\)90046-2](http://dx.doi.org/10.1016/0021-9991(92)90046-2).
- [38] C. Williamson, Vortex dynamics in the cylinder wake, *Annu. Rev. Fluid Mech.* 28 (1996) 477–539, <http://dx.doi.org/10.1146/annurev.fluid.28.1.477>.
- [39] Canonical Ltd, Ubuntu, <http://www.ubuntu.com/>, 2014.
- [40] S.P. Schneider, Fabrication and testing of the Purdue Mach-6 quiet-flow Ludwig tube, in: 38th Aerospace Sciences Meeting & Exhibit, 2000, AIAA Paper 2000-0295, <http://dx.doi.org/10.2514/6.2000-295>.
- [41] B.M. Wheaton, S.P. Schneider, Roughness-induced instability in a laminar boundary layer at Mach 6, in: 48th AIAA Aerospace Sciences Meeting, 2010, AIAA Paper 2010-1574, <http://dx.doi.org/10.2514/6.2010-1574>.
- [42] M.D. Bartkovicz, P. Subbareddy, G. Candler, Numerical simulations of roughness induced instability in the Purdue Mach 6 wind tunnel, in: 40th Fluid Dynamics Conference and Exhibit, 2010, AIAA Paper 2010-4723, <http://dx.doi.org/10.2514/6.2010-4723>.
- [43] B.M. Wheaton, M.D. Bartkovicz, P. Subbareddy, S. Schneider, G. Candler, Roughness-induced instabilities at Mach 6: a combined numerical and experimental study, in: 41st AIAA Fluid Dynamics Conference and Exhibit, 2011, AIAA Paper 2011-3248, <http://dx.doi.org/10.2514/6.2011-3248>.