

Code Verification and Numerical Accuracy Assessment for Finite Volume CFD Codes

Subrahmanya P. Veluri

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy  
In  
Aerospace Engineering

Christopher J. Roy, Chair  
William Mason  
Leigh S. McCue-Weil  
Danesh K. Tafti

August 6, 2010  
Blacksburg, Virginia

Keywords: Code Verification, Method of Manufactured Solutions, Discretization Error, Order of Accuracy, Computational Fluid Dynamics

Copyright 2010, Subrahmanya P. Veluri

# Code Verification and Numerical Accuracy Assessment for Finite Volume CFD Codes

Subrahmanya P. Veluri

## ABSTRACT

A detailed code verification study of an unstructured finite volume Computational Fluid Dynamics (CFD) code is performed. The Method of Manufactured Solutions is used to generate exact solutions for the Euler and Navier-Stokes equations to verify the correctness of the code through order of accuracy testing. The verification testing is performed on different mesh types which include triangular and quadrilateral elements in 2D and tetrahedral, prismatic, and hexahedral elements in 3D. The requirements of systematic mesh refinement are discussed, particularly in regards to unstructured meshes. Different code options verified include the baseline steady state governing equations, transport models, turbulence models, boundary conditions and unsteady flows. Coding mistakes, algorithm inconsistencies, and mesh quality sensitivities uncovered during the code verification are presented.

In recent years, there has been significant work on the development of algorithms for the compressible Navier-Stokes equations on unstructured grids. One of the challenging tasks during the development of these algorithms is the formulation of consistent and accurate diffusion operators. The robustness and accuracy of diffusion operators depends on mesh quality. A survey of diffusion operators for compressible CFD solvers is conducted to understand different formulation procedures for diffusion fluxes. A patch-wise version of the Method of Manufactured Solutions is used to test the accuracy of selected diffusion operators. This testing of diffusion operators is limited to cell-centered finite volume methods which are formally second order accurate. These diffusion operators are tested and compared on different 2D mesh topologies to study the effect of mesh quality (stretching, aspect ratio, skewness, and curvature) on their numerical accuracy. Quantities examined include the numerical approximation errors and order of accuracy associated with face gradient reconstruction. From the analysis, defects in some of the numerical formulations are identified along with some robust and accurate diffusion operators.

## Acknowledgements

I would like to express my sincere gratitude to Dr. Christopher J. Roy for giving me an opportunity to work on this research. I am thankful to him for his guidance, encouragement, and his continuous financial support during my PhD. I thank my committee members Dr. William Mason, Dr. Leigh S. McCue-Weil, and Dr. Danesh K. Tafti for serving on my committee.

This work is supported by the National Aeronautic and Space Administration's Constellation University Institutes Program (CUIP) with Claudia Meyer and Jeffrey Ryback of NASA Glenn Research Center serving as program managers and Kevin Tucker and Jeffrey West of NASA Marshall Space Flight Center serving as technical monitors. I would like to thank Edward Luke from Mississippi State University for sharing his ideas and helping me during my research.

I thank my parents, brother, and sister for encouraging me and giving me the confidence required to complete my PhD successfully.

I thank Ravi Duggirala for his thoughtful ideas and useful research discussions whenever needed during my research.

I also want to thank system administrators James Clark, Shannon Price from Auburn University and Steve Edwards from Virginia Polytechnic Institute and State University for their help when required.

Finally, I want to thank all my fellow research students at Auburn University and Virginia Polytechnic Institute and State University for their valuable suggestions and providing me with ideal work environment. I thank all my friends for understanding me and providing the necessary help.

# Table of Contents

Nomenclature .....	viii
List of Figures.....	x
List of Tables .....	xv
<b>1 Introduction.....</b>	<b>1</b>
<b>1.1 Motivation .....</b>	<b>1</b>
<b>1.2 Literature Review .....</b>	<b>2</b>
<b>1.2.1 Code Verification.....</b>	<b>2</b>
<b>1.2.2 Diffusion Operators.....</b>	<b>8</b>
<b>2 Basic Concepts of Verification.....</b>	<b>14</b>
<b>2.1 Verification.....</b>	<b>14</b>
<b>2.2 Code Verification.....</b>	<b>14</b>
<b>2.3 Order of Accuracy .....</b>	<b>15</b>
<b>2.3.1 Formal Order of Accuracy .....</b>	<b>16</b>
<b>2.3.2 Observed Order of Accuracy.....</b>	<b>16</b>
<b>2.4 Method of Manufactured Solutions .....</b>	<b>17</b>
<b>2.4.1 Procedure for MMS.....</b>	<b>17</b>
<b>2.4.2 Attributes of a Good Manufactured Solution .....</b>	<b>18</b>
<b>2.4.3 Advantages of Method of Manufactured Solutions.....</b>	<b>19</b>
<b>3 Mesh Generation.....</b>	<b>20</b>
<b>3.1 2D Mesh Topologies.....</b>	<b>20</b>
<b>3.1.1 2D Structured Meshes .....</b>	<b>20</b>
<b>3.1.2 2D Unstructured Meshes.....</b>	<b>21</b>
<b>3.1.3 2D Hybrid Meshes .....</b>	<b>23</b>
<b>3.2 3D Mesh Topologies.....</b>	<b>23</b>
<b>3.2.1 3D Structured Meshes .....</b>	<b>23</b>

3.2.2	3D Unstructured Meshes.....	24
3.2.3	3D Hybrid Meshes .....	26
3.3	Systematic Mesh Refinement.....	26
<b>4</b>	<b>Verification of a Finite Volume CFD Code .....</b>	<b>29</b>
4.1	Finite Volume Code.....	29
4.2	Verification of Baseline Governing Equations.....	30
4.2.1	Flow Equations .....	30
4.2.2	Verification of the Euler Equations .....	31
4.2.3	Verification of the Navier-Stokes Equations.....	34
4.2.4	Effect of Mesh Quality .....	39
4.3	Verification of Transport Models .....	43
4.4	Verification of Boundary Conditions.....	44
4.4.1	Mesh Topologies .....	46
4.4.2	No-slip Wall.....	46
4.4.2.1	No-slip Adiabatic Wall .....	46
4.4.2.2	No-slip Isothermal Wall .....	49
4.4.3	Slip Wall .....	51
4.4.4	Isentropic Inflow.....	55
4.4.5	Outflow .....	56
4.4.6	Extrapolation .....	58
4.4.7	Farfield .....	60
4.5	Verification of Turbulence Models .....	61
4.6	Verification of Time Accuracy for Unsteady Flows .....	67
4.7	Issues Uncovered During Code Verification .....	71
4.7.1	Coding Mistakes/Algorithm Inconsistencies.....	71

4.7.2	Grid Sensitivities.....	71
<b>5</b>	<b>Finite Volume Diffusion Operators.....</b>	<b>73</b>
5.1	Robustness and Accuracy .....	73
5.2	Testing Framework .....	75
5.3	Cell-Centered Finite Volume Diffusion Operators.....	76
5.3.1	Node Averaging Method .....	77
5.3.2	Unweighted and Weighted Least Squares.....	77
5.3.3	Loci-CHEM Diffusion Operator .....	77
5.3.4	FAMLSQ Method with Compact Stencil .....	78
5.4	Mesh Topologies .....	79
5.5	Effect of Mesh Quality .....	82
5.5.1	2D Structured Meshes.....	82
5.5.1.1	Uniform Mesh .....	82
5.5.1.2	Aspect Ratio.....	83
5.5.1.3	Stretching.....	84
5.5.1.4	Skewness .....	85
5.5.1.5	Curvature .....	86
5.5.2	2D Unstructured Meshes.....	87
5.5.2.1	Mesh with Isotropic Cells.....	88
5.5.2.2	Skewness with Aspect Ratio.....	90
5.5.2.3	Skewness with Curvature.....	92
<b>6</b>	<b>Conclusions.....</b>	<b>98</b>
6.1	Summary of Results.....	98
6.2	Main Contributions of this Work.....	99
6.3	Future Work and Recommendations .....	100
	References.....	101

<b>Appendix A</b> .....	<b>108</b>
<b>Appendix B</b> .....	<b>109</b>
<b>Appendix C</b> .....	<b>110</b>

## Nomenclature

$C_p, C_v$	=	specific heats, J/Kg · K
$DE$	=	discretization error
$e$	=	energy, J
$g_p$	=	coefficient of the leading error term
$h$	=	normalized grid spacing; enthalpy, J/Kg · K
$h_f$	=	heat of formation, J/Kg · K
$F$	=	blending function
$f$	=	general solution variable
$k$	=	turbulent kinetic energy
$p$	=	spatial order of accuracy; pressure, N/m <sup>2</sup>
$q$	=	temporal order of accuracy
$q_L$	=	laminar heat flux
$q_T$	=	turbulent heat flux
$R$	=	gas constant, J/Kg · K
$r$	=	refinement factor
$T$	=	temperature, K
$t$	=	time, sec
$V_i$	=	Volume of cell 'i'
$u, v, w$	=	Cartesian velocity components, m/s
$x, y, z$	=	Cartesian coordinates, m

### *Greek Symbols*

$\varepsilon$  = error

$\mu$  = viscosity,  $\text{N} \cdot \text{sec}/\text{m}^2$

$\mu_T$  = turbulent viscosity,  $\text{N} \cdot \text{sec}/\text{m}^2$

$\rho$  = density,  $\text{Kg}/\text{m}^3$

$\omega$  = turbulent dissipation rate

### *Superscripts*

$n$  = time level

### *Subscripts*

*exact* = exact continuum value

$k$  = mesh level, 1, 2, 3, etc.; fine to coarse

*ref* = reference value

## List of Figures

Figure 1: 2D structured meshes: a) Cartesian, b) stretched Cartesian, c) curvilinear, and d) skewed curvilinear .....	21
Figure 2: 2D unstructured meshes: a) general unstructured, b) uni-directional diagonal, and c) alternate diagonal .....	22
Figure 3: 2D hybrid meshes: a) skewed hybrid and b) highly skewed hybrid .....	23
Figure 4: 3D structured meshes: a) Cartesian and b) skewed curvilinear.....	24
Figure 5: 3D unstructured meshes: a) unstructured mesh with tetrahedral cells, unstructured mesh with prismatic cells, c) highly skewed unstructured mesh with tetrahedral cells, and d) highly skewed unstructured mesh with prismatic cells .....	25
Figure 6: 3D hybrid meshes: a) skewed hybrid and b) highly skewed hybrid .....	26
Figure 7: a) Cube split into five tetrahedral cells, b) 3D unstructured mesh with tetrahedral cells generated from a skewed and stretched curvilinear mesh .....	28
Figure 8: a) Manufactured Solution of u-velocity and b) mass source term .....	33
Figure 9: Order of accuracy results for Euler equations on a 3D skewed hybrid mesh using a) $L_1$ norm of the discretization error, b) $L_2$ norm of the discretization error, and c) $L_\infty$ norm of the discretization error .....	34
Figure 10: Order of accuracy results for Navier-Stokes equations on a 2D hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	36
Figure 11: Order of accuracy results for Navier-Stokes equations on a 3D hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	36
Figure 12: Order of accuracy results using $L_2$ norm of the discretization error for Navier-Stokes equations on a 2D skewed curvilinear mesh using a) original diffusion operator and b) new diffusion operator.....	38
Figure 13: Order of accuracy results using $L_2$ norm of the discretization error for Navier-Stokes equations on a 2D alternate diagonal unstructured mesh using a) old diffusion operator and b) modified diffusion operator .....	39
Figure 14: Order of accuracy results for Navier-Stokes equations on a 2D highly skewed hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	40

Figure 15: Order of accuracy results for Navier-Stokes equations on a 3D highly skewed hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	41
Figure 16: Comparison of $L_2$ norm discretization error on 3D skewed hybrid mesh and 3D highly skewed hybrid mesh .....	42
Figure 17: Order of accuracy results for Sutherland's law of viscosity on a 3D skewed hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	44
Figure 18: Wavy surface used as a well-defined boundary for 3D meshes.....	45
Figure 19: Meshes used for boundary condition verification a) 2D hybrid mesh and b) 3D hybrid mesh.....	46
Figure 20: Temperature contours when the bottom boundary is defined as a no-slip adiabatic wall.....	48
Figure 21: Order of accuracy results for adiabatic no-slip wall boundary on a 3D hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	49
Figure 22: Temperature contours when the bottom boundary is defined as a no-slip isothermal wall.....	50
Figure 23: Order of accuracy results for isothermal no-slip wall boundary on a 3D hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	51
Figure 24: Order of accuracy results for slip wall boundary with the Euler equations on a 3D skewed hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	54
Figure 25: Order of accuracy results for slip wall boundary with the Navier-Stokes equations on a 3D hybrid mesh with straight boundaries using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error.....	54
Figure 26: Order of accuracy results for isentropic inflow boundary on a 3D hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error.....	56

Figure 27: Order of accuracy results for subsonic outflow boundary on a 3D hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error.....	58
Figure 28: Order of accuracy results for extrapolate boundary on a 3D hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	60
Figure 29: Order of accuracy results for farfield boundary on a 3D skewed hybrid mesh using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error.....	61
Figure 30: Ratio of production term to the destruction term in turbulent kinetic energy equation in a 2D rectangular domain .....	63
Figure 31: Order of accuracy results on the 3D skewed hybrid mesh for k- $\omega$ turbulence model (F=1) using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	64
Figure 32: Order of accuracy results on the 3D skewed hybrid mesh for k- $\epsilon$ turbulence model (F=0) using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	65
Figure 33: Order of accuracy results on the 2D hybrid mesh for k- $\epsilon$ turbulence model (F=0) using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error.....	66
Figure 34: Order of accuracy results on the 3D highly skewed curvilinear (i.e., structured) mesh with hexahedral cells for k- $\epsilon$ turbulence model (F=0) using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error.....	66
Figure 35: Order of accuracy results on the 3D unstructured mesh with tetrahedral cells for k- $\epsilon$ turbulence model (F=0) using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error.....	67
Figure 36: Order of accuracy results for time accuracy of the unsteady flows on the 2D hybrid grid using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	70
Figure 37: Order of accuracy results for time accuracy of the unsteady flows on the 3D hybrid grid using a) $L_1$ norm of the discretization error and b) $L_\infty$ norm of the discretization error .....	70

Figure 38: Shrinking of a mesh over a focal point for a 2D unstructured mesh.....	76
Figure 39: An illustration of the reconstruction of a centered control volume about face $F_c$ as shown in the shaded region.....	78
Figure 40: Structured mesh topologies in 2D: a) uniform mesh, b) mesh with aspect ratio cells, c) stretched mesh, and d) mesh with skewed cells .....	80
Figure 41: 2D structured mesh curvilinear mesh.....	81
Figure 42: Unstructured mesh topologies in 2D: a) mesh with isotropic cells, b) mesh with aspect ratio cells, and c) curvilinear mesh .....	81
Figure 43: Order of accuracy results for gradients at face center on a 2D structured uniform mesh using a) node averaging method and b) weighted least squares method.....	83
Figure 44: Order of accuracy results for gradients at face center on a 2D structured mesh with aspect ratio using a) node averaging method and b) weighted least squares method.....	84
Figure 45: Order of accuracy results for gradients at face center on a 2D structured mesh with stretched cells using a) node averaging method and b) weighted least squares method.....	85
Figure 46: Order of accuracy results for gradients at face center on a 2D structured mesh with skewed cells using a) node averaging method and b) weighted least squares method.....	86
Figure 47: Order of accuracy results for gradients at face center on a 2D structured mesh with skewed cells using a) node averaging method and b) weighted least squares method.....	87
Figure 48: Order of accuracy results for gradients at face center on a 2D unstructured mesh with isotropic triangular cells using a) node averaging method and b) weighted least squares method .....	88
Figure 49: Order of accuracy results for gradients at face center on a 2D unstructured mesh with isotropic triangular cells using Loci-CHEM approach.....	89
Figure 50: Comparison of error in the gradients at the face center for the tested numerical methods on a 2D unstructured mesh with isotropic triangular cells.....	90

Figure 51: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and aspect ratio cells using a) node averaging method and b) weighted least squares method .....	91
Figure 52: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and aspect ratio cells using Loci-CHEM approach .....	92
Figure 53: Comparison of error in the gradients at the face center for the tested numerical methods on a 2D unstructured triangular mesh with skewed and aspect ratio cells....	92
Figure 54: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and curved cells using a) node averaging method and b) weighted least squares method .....	94
Figure 55: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and curved cells using Loci-CHEM approach.....	95
Figure 56: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and curved cells using FAMLSQ approach.....	96
Figure 57: Comparison of error in the gradients at the face center for the tested numerical methods on a 2D unstructured triangular mesh with skewed and curved cells .....	96
Figure A1: Verification of different options in the finite volume Loci-CHEM CFD code.....	108

## List of Tables

Table 1. Different mesh levels and mesh types .....	28
Table 2. Order of accuracy of the governing equations on highly skewed cells in 3D .....	43
Table B1. Quantification of cell quality.....	109
Table C1. Constants for 2D Manufactured Solutions.....	110
Table C2. Constants for 3D Manufactured Solutions.....	111

# 1. Introduction

## 1.1 Motivation

Computational Fluid Dynamics, CFD, is playing an ever-increasing role in the design, analysis, and optimization of many engineering problems. CFD is used across all disciplines where fluid flow is important. Apart from the CFD applications in aerospace engineering, other applications of CFD are in automobile engineering, industrial manufacturing, civil engineering, environmental engineering, naval architecture and many more. It is therefore very important that decision makers have confidence in the correctness of the CFD predictions. With increases in computational power, CFD practitioners often focus on solving more complex and difficult problems rather than demonstrating accuracy of their current problems, which can lead to a decrease in the quality of the simulations.

With the numerical problems involving fluid flows getting more complicated, mistakes made by the developers in building a CFD code for solving those numerical problems will increase. Hatton<sup>1</sup> conducted a study on the reliability of the scientific software. According to Hatton, approximately 40% of the software failures are caused due to static faults which are mistakes made by the developers during building a code. Some examples of static faults include dependency on uninitialized variables, too many arguments passed into a subroutine, use of non-local variables in functions, etc. Hatton<sup>1</sup> classified his study into two parts which are collectively called the “T Experiments”. In the first part of the study, he examined codes from a wide range of scientific disciplines using static testing where testing can be done without running a code and in the second part of the study he examined codes in a single discipline using dynamic testing. During the static testing, Hatton examined 100 different codes in 40 different application areas written in C or FORTRAN. He observed that the C codes contained approximately 8 serious static faults for every 1000 lines of executable code, while the FORTRAN codes contained approximately 12 serious faults per 1000 lines which can lead to the failure of software meaning software giving a wrong solution. In his dynamic testing, where the testing is done by actually running the code, Hatton examined codes in the area of seismic data processing. In this study, he examined 9 independent, mature commercial codes developed independently by different companies which employed the same algorithm, the same programming language, the same user

defined parameters, and the same input data and the agreement between the codes was only within 100% (i.e., a factor of two). These experiments done by Hatton provide motivation to work towards building confidence in CFD solutions. So it is necessary to find any coding mistakes or bugs that may have been made during the development stage of a code. Engineers and other CFD practitioners who rely on the numerical solutions from CFD codes need assurance that the codes are free from coding mistakes. The procedure of finding mistakes in the code is called code verification.

## **1.2 Literature Review**

Topics investigated include the code verification of finite volume CFD codes using the Method of Manufactured Solutions and also the robustness and accuracy of diffusion operators. The numerical formulations of the diffusion terms in the Navier-Stokes equations are termed as diffusion operators. During the code verification process of an unstructured finite volume CFD code, it was observed that the formulation of an accurate and robust viscous operator is a challenging task which motivated the study and testing of different diffusion operators.

### **1.2.1 Code Verification**

Verification of CFD codes has been the subject of many studies in recent years. Abanto et al.<sup>2</sup> demonstrated an approach to test the accuracy of some of the most widespread commercial codes. They presented grid convergence studies on atypical CFD cases using some commercial CFD packages. Their verification test cases include an incompressible laminar Poiseuille flow, a manufactured incompressible laminar boundary layer flow, an incompressible re-circulating flow and an incompressible annular flow. Different types of structured and unstructured meshes were used during the study. They observed non-monotonic grid convergence for all their test cases. Iterative convergence of the discrete equations to machine zero did not guaranty accurate flow field predications which meant that the codes converged to wrong solutions. From their study, they recommended that users perform the verification of commercial CFD codes and be cautious when using the commercial codes on industrial problems.

Kleb and Wood<sup>3</sup> pointed out that the computational simulation community is not routinely publishing independently verifiable tests to accompany new models or algorithms. They mentioned the importance of conducting component-level verification tests before attempting system-level verification and also publishing them when introducing a new

component algorithm. They proposed a protocol for the introduction of new methods and physical models that would provide the computational community with a credible history of documented, repeatable verification tests that would enable independent replication.

Roache<sup>4</sup> discussed the verification of codes and calculations along with some definitions and descriptions related to confidence building in computational fluid dynamics. Verification was described as solving the equations right and validation as solving the right equations. Different aspects discussed in the paper include the distinction between code verification and validation, grid convergence and iterative convergence, truncation error and discretization error. Also discussed were verification of calculations, error taxonomies, code verification via systematic grid convergence testing, the Grid Convergence Index (GCI) and sensitivity of grid convergence testing. According to the author, verification does not include all aspects of code quality assurance like the important concerns of version control or archiving of input data.

In the books by Roache<sup>5</sup> and Knupp and Salari,<sup>6</sup> the authors comprehensively discussed code verification, the Method of Manufactured Solutions (MMS) used to obtain exact solutions for code verification purposes, and order of accuracy verification. Recently, Oberkampf and Roy<sup>7</sup> also in their book discussed in detail the concepts of verification. Their detailed focus was on fundamental concepts of code verification and software engineering, solution verification, validation, and predictive capability of scientific computing.

The first application of MMS for code verification was by Roache and Steinberg in 1984.<sup>8</sup> In their work, they used the MMS approach to verify a code for generating three-dimensional transformations for elliptic partial differential equations. Additional discussions of the MMS procedure for code verification have been presented by Roache.<sup>5,9</sup> More recently, Oberkampf and Roy<sup>7</sup> discussed the use of MMS for generating exact solutions along with the order of accuracy testing for code verification.

In prior work, MMS has been used to verify two compressible CFD codes<sup>10-12</sup>: Premo<sup>13</sup> and WIND.<sup>14</sup> The Premo code employed a node-centered approach using unstructured meshes and the Wind code employed a similar scheme on structured meshes. Both codes used Roe's upwind method with MUSCL extrapolation for the convective terms and central differences for the diffusion terms making the codes second-order accurate. The form of Manufactured Solution

was chosen to be a smooth, infinitely differentiable and chosen such that all the terms in the governing equations were exercised. The Manufactured Solutions were tested on different grid levels and both the codes demonstrated second-order special accuracy as the mesh was refined, thus giving a high degree of confidence that the codes were free from coding mistakes. The MMS was found to be an invaluable tool for finding coding mistakes. In their work, the authors successfully verified both the inviscid Euler equations and the laminar Navier-Stokes equations. An alternative statistical approach to MMS was proposed by Hebert and Luke<sup>15</sup> for the Loci-CHEM combustion CFD code.<sup>16</sup> In their approach, they employed a single mesh level which was shrunk down and used to statistically sample the discretization error in different regions of the domain of interest. Their work successfully verified the Loci-CHEM CFD code for the 3D, multi-species, laminar Navier-Stokes equations using both statistical and traditional MMS. Another similar approach to statistical MMS was the downscaling approach to order verification<sup>17,18</sup> which also employed a single mesh which was scaled down about a single point in the domain instead of statistically sampling the smaller meshes in the domain of interest. Both the statistical MMS and the downscaling approaches have an advantage of being relatively inexpensive because they do not require mesh refinement. But one of the disadvantages of using these methods is that they neglect the possibility of discretization error transport into the scaled-down domain. Thus, it is possible to pass a code order of accuracy verification test using statistical MMS or the downscaling approach for a case that would fail the test using traditional MMS.

MMS was also applied to verify the boundary conditions. Bond et al.<sup>19,20</sup> documented the development of a Manufactured Solution capable of testing the governing equations and boundary conditions commonly implemented in CFD codes. Along with the Euler, Navier-Stokes, and Reynolds-averaged Navier-Stokes equations, verification of boundary conditions including slip wall, no-slip wall (adiabatic and isothermal) boundary and outflow (subsonic, supersonic and mixed) boundary conditions was performed. The derived Manufactured Solution was applicable not only to Premo for which the Manufactured Solution was used, but also to general CFD codes. The verification of the Premo code was done on skewed, non-uniform, three-dimensional meshes and the sequence of meshes were designed to obtain asymptotic results with reasonable computational cost. The Manufactured Solution used had identified a number of formulation weaknesses with boundary conditions and gradient reconstruction

methods. In their study, the reason for the inconsistency in the order of accuracy results for the Euler slip wall boundary was found to be not because of the coding mistakes but instead caused by a problem with the weak enforcement of the slip condition. A newly implemented characteristic formulation eliminated the problems and second order convergence was observed. The implementation of Green-Gauss gradient calculation in Premo was found to be strictly valid only for tetrahedral element types because the implementation neglected some additional support necessary for non-Cartesian hexahedral element types and hence showed poor order performance.

There have been three coordinated efforts to apply MMS to turbulent flows. Pelletier and co-workers have summarized their work on 2D incompressible turbulent shear layers using a finite element code with a focus on a logarithmic form of the  $k$ - $\epsilon$  two-equation RANS model.<sup>21,22</sup> They employed Manufactured Solutions which mimic turbulent shear flows, with the turbulent kinetic energy and the turbulent eddy viscosity as the two quantities specified in the Manufactured Solution. For the cases examined, they were able to verify the code by reproducing the formal order of accuracy. More recently, Eca and co-workers have published a series of papers on Manufactured Solutions for the 2D incompressible turbulent Navier-Stokes equations.<sup>23-25</sup> They also employed physically-based Manufactured Solutions, in this case mimicking wall-bounded turbulent flow. This group looked at both finite-difference and finite-volume discretizations, and examined a number of turbulence models including the Spalart-Allmaras one-equation model<sup>26</sup> and two two-equation models: Menter's baseline (BSL) version  $k$ - $\omega$  model<sup>27</sup> and Kok's turbulent/non-turbulent  $k$ - $\omega$  model.<sup>28</sup> While successful in some cases, their physically-based Manufactured Solution often led to numerical instabilities, a reduction in the observed mesh convergence rate, or even inconsistency of the numerical scheme (i.e., the discretization error did not decrease as the mesh was refined). In order to independently test different aspects of the governing equations, in some cases they replaced certain discretized terms (or even whole equations) with the analytic counterpart from the Manufactured Solution. For the Spalart-Allmaras model they specified the turbulent viscosity, while for the two equation models they specified both the turbulent eddy viscosity and the turbulent kinetic energy. The cases they examined employed a Reynolds number of  $10^6$  and used Cartesian meshes which were clustered in the  $y$ -direction towards the wall.

Roy et al.<sup>29</sup> presented a different approach for verifying RANS turbulence models in CFD codes. Their approach used smooth, non-physical Manufactured Solutions which were chosen such that they provided contributions from all the terms in the turbulence transport equations including convection, diffusion, production, and destruction terms. The turbulence model verified was the baseline version of Menter's  $k-\omega$  model and the code used was the Loci-CHEM CFD code. Special attention was paid to the blending function which allowed the model to switch between a  $k-\omega$  and a transformed  $k-\epsilon$  model. In their approach, Manufactured Solutions were selected such that they activate only one branch of the min and max functions in the turbulence model. Certain terms were turned off in both the numerical code and the Manufactured Solution to focus on different parts of the turbulence equations. The turbulence transport equations along with the RANS equations were verified in the Loci-CHEM CFD code by computing the observed order of accuracy on a series of consistently refined two-dimensional meshes. For the structured mesh topologies examined, i.e. Cartesian and skewed curvilinear meshes, the observed order matched the formal order of two. But for the unstructured meshes examined, the observed order of accuracy was found to be first order. Testing the Euler equations and Navier-Stokes equations on the unstructured meshes suggested that the source for the reduced order of accuracy was the diffusion operator on the unstructured meshes.

Thomas et al.<sup>18</sup> presented a new methodology for verification of finite volume computational methods using unstructured meshes. The discretization-order properties were studied in computational windows, constructed within a collection of meshes or a single mesh and tests were performed within each window to address a combination of problem-, solution-, and discretization/mesh-related features affecting discretization error convergence. A computational window was used to improve the verification of unstructured mesh computational methods intended for large-scale applications and the integral norms do not provide sufficient information to isolate the source of errors. A downscaling technique was also used, in addition to a traditional mesh refinement. Meshes within the windows were constrained to be consistently refined, enabling a meaningful assessment of asymptotic error convergence on unstructured grids. Two types of finite volume schemes were considered: node centered schemes and cell centered schemes. Demonstration of the methodology was shown which included a comparative accuracy assessment of commonly used schemes on general mixed grids and the identification of local accuracy deterioration at boundary intersections. The authors provided recommendations

on choosing relevant tests to verify a code for large-scale computations. They recommended that each problem-related feature like boundary conditions etc. to be addressed by choosing an appropriate computational window and each solution related features like shocks, boundary layer, flow separation etc. to be addressed by choosing an appropriate Manufactured Solution.

Proper mesh refinement between different mesh levels used for code verification is important to achieve correct order of accuracy. Salas<sup>30</sup> presented the necessary conditions to properly establish mesh convergence. He demonstrated his ideas using a theoretical model and a numerical example and showed that anomalously low or high observed rates can be exhibited by otherwise well-behaved algorithms because of improper use of mesh refinement ratios in different directions. In his study, he showed that as the mesh aspect ratio increased, finer meshes are required to enter the asymptotic range.

Etienne et al.<sup>31</sup> demonstrated three approaches to verification of temporal accuracy of unsteady time accurate flow solvers. When higher order time-stepping schemes are used with second order spatial discretization, verifying the time-stepping schemes requires very fine meshes that are often impractical to perform calculations on due to limits in memory size and CPU times. The three approaches presented to verify time accuracy in a flow solver included the direct approach in which both the mesh size and time step were refined simultaneously in a consistent and coherent fashion, the decoupled approach in which the mesh size was so small that the spatial discretization error could be neglected so that only a time-step refinement study was done and the Iterated Richardson Extrapolation method which allowed verification of the time-integrators by time step refinement on a coarse mesh. In the direct approach with a higher order time-stepping scheme, every time the time step was refined by a factor of two, the mesh needs to be refined by a higher factor of  $2^{n/p}$  where  $n$  and  $p$  are the formal orders of time and space, respectively, requiring this approach to use very fine meshes for verifying higher order time-stepping schemes. In the case of decoupled approach, a mesh size sufficiently small was used so that the spatial contribution to the error was neglected compared to the temporal contribution and again this approach required a very fine mesh which makes this approach along with the direct approach very expensive or impractical for code verification. In the case of the Iterated Richardson Extrapolation method, for a fixed mesh size, a time step refinement study was performed, and Richardson extrapolation was applied repeatedly to the data to yield an

approximation to the exact solution to the ordinary differential equations in time resulting from spatial discretization with the given fixed mesh size. Performing Richardson extrapolation decoupled the time step refinement from the mesh size refinement for estimating the temporal contribution to the global error in the final time. If the Richardson extrapolation in time was feasible, then time-step refinement on any mesh however coarse was sufficient to achieve verification of time integrators of arbitrarily high order of accuracy.

Tremblay et al.<sup>32</sup> presented the use of MMS for fluid-structure interactions code verification and debugging. MMS was used to generate exact solutions and then test the code using systematic mesh refinement. Manufactured Solutions were presented for two-dimensional incompressible flow strongly coupled with an isotropic structure undergoing large displacements. Their results illustrated the power and cost-effectiveness of the approach.

### **1.2.2 Diffusion Operators**

Since the challenge of formulating a diffusion operator for an unstructured mesh is more complicated when compared to a structured mesh, the literature survey is concentrated mostly on the diffusion operator formulations on unstructured meshes. Knight<sup>33</sup> developed an implicit algorithm for the two-dimensional, compressible, laminar, Navier-Stokes equations using an unstructured grid consisting of triangular cells. A cell centered data structure was employed with the flow variables stored at the cell centroids. For the viscous fluxes, the values on the faces were calculated by applying Gauss's theorem to the quadrilateral defined by the cell centroids of the cells adjacent to the face and the two nodes defining the end points. The solution variables at the nodes were obtained by the second order interpolation of the conserved variables from those cells sharing the node. A similar formulation of viscous fluxes on the cell faces was used by Ollivier-Gooch et al.<sup>34</sup> They presented a new approach for a high-order-accurate finite-volume discretization for diffusive fluxes that was based on the gradients computed during solution reconstruction. In their analysis, their schemes based on linear and cubic reconstruction achieved second and fourth order accuracy, respectively, while the schemes based on quadratic reconstruction were second order accurate. They examined both vertex centered and cell centered control volumes for linear, quadratic and cubic reconstructions. For cell centered control volumes, gradients were calculated by using Green-Gauss integration around a diamond connecting the end points of an edge and the centroids of the cells that share the edge. The

solution at the end points of the edge was estimated by averaging data in incident control volumes. The numerical experiments conducted showed that nominal accuracy was attained in all cases for the advection-diffusion problems.

Vaassen et al.<sup>35</sup> presented a finite volume cell centered scheme for the solution of the three-dimensional Navier-Stokes equations where they used a conservative and consistent discretization approach for the diffusive terms based on an extended version of Coirier's diamond path.<sup>36</sup> A Gauss formula was employed to perform the integration of fluxes on the cell surfaces. To obtain a consistent discretization of the diffusion terms regardless of the irregularity of the mesh, a second order approximation of the gradient was used. For the calculation of gradients on the face centers, a diamond path was built by connecting each vertex of the face to the left and right neighboring cell centroids, forming a polyhedron on which a Green-Gauss formula was applied. Numerical results using the solver were shown to have good accuracy even on highly distorted meshes.

Grismer et al.<sup>37</sup> developed an implicit, unstructured Euler/Navier-Stokes finite volume solver which is based on a cell centered scheme. In their procedure for the calculation of viscous fluxes, a piecewise linear reconstruction was used to approximate the solution variables within cells. The linear reconstruction was derived from the cell values and gradients by applying a second-order Taylor-series expansion. The gradient at the cell centroid was evaluated by minimizing the weighted error between the reconstructed function and the neighboring cell values. The gradient on the face was shown as a vector sum of the components normal and tangential to the face. The tangential component of the gradient vector at the cell face was the average of tangential components of the gradients in the two cells sharing the face. Luke<sup>39</sup> used the same formulation for the viscous fluxes with some changes in the normal gradient calculation. The changes were made after the formulation for the diffusive fluxes was shown to be inconsistent on a skewed curvilinear mesh.<sup>38</sup> The diffusion operator formulations used by Grismer<sup>37</sup> and Luke<sup>39</sup> are different from the approaches discussed earlier. Instead of directly calculating the gradients at the face centers, the gradients were calculated at the cell centroid and then interpolated to the face centers.

Mavriplis<sup>40</sup> examined the accuracy of the various gradient reconstruction techniques on unstructured meshes. He demonstrated that the unweighted least-squares gradient construction

severely under-estimated normal gradients for highly stretched meshes in the presence of surface curvature. It was shown that the above behavior could be expected for vertex based discretizations and cell-centered discretizations operating on triangular and quadrilateral meshes (tetrahedral and prismatic meshes in 3D). The use of inverse distance weighting in the least-squares reconstruction could be used to recover good accuracy for the vertex and cell-centered discretizations on quadrilateral meshes and for vertex discretizations on triangular meshes; however, and the technique was shown to be ineffective for cell-centered discretizations on triangular meshes. The Green-Gauss construction technique produced adequate gradient estimate in all the cases. The author concluded that the use of inverse distance weighted least-squares gradients and Green-Gauss gradients in the discretization of convective and viscous terms was a prudent strategy, but the approaches were shown to be less robust than upwind schemes and often required gradient limiting to achieve stable solutions.

Delanaye et al.<sup>41</sup> presented a second-order finite volume cell-centered technique for computing steady-state solutions of the full Euler and Navier-Stokes equations on unstructured meshes. The scheme was designed such that its accuracy was only weakly sensitive to mesh distortions. An original quadratic reconstruction with a fixed stencil and high-order flux integration by the Gauss quadrature rule was employed to compute the advective term of the equations. For the diffusive term, the derivatives were obtained at the midpoint of an edge by using a linear interpolation between the right and left neighbors. The scheme's weak sensitivity to mesh distortions was demonstrated for inviscid flow calculation with the computation of Ringleb's flow and for the viscous flow calculation with the computation of the viscous flow around an isothermal flat plate. Since the high-order scheme produces oscillations in the vicinity of discontinuities leading to instability, the full quadratic reconstruction was used in smooth flows and the scheme was automatically switched to a linear reconstruction at the vicinity of discontinuities. Two inviscid flow computations over the NACA0012 airfoil, a supersonic flow at a free stream Mach number of 1.2 with zero angle of attack and a transonic flow at free stream Mach number of 0.8 with angle of attack of  $1.25^\circ$  were performed to illustrate the application of their quadratic reconstruction scheme. They also investigated the supersonic flow over a compression ramp by using a hybrid grid.

Haselbacher et al.<sup>42,43</sup> presented an upwind flow solution method for mixed unstructured meshes in two dimensions consisting of triangular and quadrilateral meshes. The discretization of the viscous fluxes on general unstructured meshes was studied and a positive discretization was developed for Laplace's equation and extended to Navier-Stokes equations. A positive discretization in this case was defined as the discretization of the Laplacian where the weights in the discretized equation were positive. It is desirable for any discretization technique to ensure positivity since turbulence models are extremely sensitive to non-positive weights. An approximate form of the viscous fluxes was devised that was independent of the cell topology, an approach they referred to as mesh transparent. The authors constructed control volumes from dual cells and the solution variables were stored at mesh vertices. The authors investigated the viscous fluxes to obtain a positive and second order accurate scheme on arbitrary mixed meshes that was compatible with the edge based data structure along with mesh transparency. The investigation was simplified by studying Laplace's equation. They presented numerical results for transonic inviscid flow, laminar flow about an airfoil with large separation and turbulent transonic flow over an airfoil. A mesh refinement study was conducted to assess the accuracy of different control constructions and the influence of triangular, quadrilateral and mixed meshes for viscous flows. Median-dual control volumes on triangular meshes resulted in higher numerical error than the containment dual control volumes on quadrilateral and triangular meshes.

Diskin et al.<sup>44</sup> compared the node-centered and cell-centered schemes for unstructured finite-volume discretization of Poisson's equation as a model of the viscous fluxes. Accuracy and efficiency were studied for six nominally second-order accurate schemes which include a node-centered scheme, cell-centered node-averaging schemes with and without clipping, and cell-centered schemes with un-weighted, weighted and approximately mapped least-squares face gradient reconstruction. Among the considered schemes, the node centered scheme has the lowest complexity and the cell-centered node-averaging scheme has the highest complexity. They tested the schemes on grids which ranged from structured regular grids to irregular grids composed of arbitrary mixtures of triangles and quadrilaterals, including random perturbations of the grid points to examine the worst possible behavior of the solution. Two classes of tests were considered. The first class of tests involved smooth manufactured solution on both isotropic and highly anisotropic grids with discontinuous metrics, typical of those encountered in unstructured

grid adaptation and the second class of tests were performed on consistently refined stretched grids generated around a curved body representative of high Reynolds number turbulent flow simulations. From the first class of tests, they observed that the face least-square methods, node-averaging method without clipping, and the node centered method demonstrated second-order convergence of discretization errors. From the second class of tests, they observed that the node-centered scheme was always second order accurate and cell-centered node-averaging schemes were less accurate and also failed to converge to the exact solution when clipping of the node-averaged values was used. Cell-centered node averaging schemes are second-order accurate and stable when the coefficients of the pseudo-Laplacian operator are close to 1 and on highly stretched and deformed grids, some coefficients of pseudo-Laplacian become negative or larger than 2, which has a detrimental effect on stability and robustness (Barth<sup>45</sup>). Holmes and Connell<sup>46</sup> proposed to enforce stability by clipping the coefficients between 0 and 2. In this paper, it was shown that clipping seriously degrades the solution accuracy. The cell-centered schemes using least square face gradient reconstruction had more compact stencils with a complexity similar to that of the node centered schemes. For simulations on highly anisotropic curved grids, the least square methods had to be amended by modifying the scheme stencil to reflect the direction of strong coupling. The authors concluded that the accuracies of the node-centered and best cell-centered schemes were comparable at equivalent number of degrees of freedom.

Many authors also employ finite-element methods for computing viscous flows governed by the Navier-Stokes equations. Sun et al.<sup>47</sup> extended a spectral volume method to handle viscous flows. Using the spectral volume method high-order accuracy was achieved through high-order polynomial reconstructions within spectral volumes. They developed a formulation similar to the Local Discontinuous Galerkin (LDG) approach to discretize the viscous fluxes. Gauss's theorem was used to integrate the gradients in the control volume. Kannan et al.<sup>48</sup> improved the Navier-Stokes solver developed by Sun et al.<sup>47</sup> based on the spectral volume method with the use of a new viscous flux formulation. Instead of a LDG-type approach, a penalty approach based on the first method of Bassi and Rebay<sup>49</sup> was used. The advantage of the penalty approach over the LDG approach was the speed up of the convergence with the implicit method and indicated that the approach had a great potential for 3D flow problems. Fidkowski and Darmofal<sup>50</sup> also used a finite-element approach for high order discretizations of the

compressible Navier-Stokes equations. The viscous flux terms were discretized using the second form of Bassi and Rebay.<sup>49</sup>

## 2. Basic Concepts of Verification

### 2.1 Verification

Verification addresses the mathematical correctness of numerical simulations and it plays an important role in building confidence in CFD solutions. There are two fundamental aspects to verification: code verification and solution verification. Code verification is the process of ensuring, that there are no mistakes (bugs) in a computer code or inconsistencies in the numerical algorithm. Solution verification is the process of estimating the three types of numerical error that can occur in numerical simulations: round-off error, iterative error, and discretization error.

### 2.2 Code Verification

Code verification<sup>7</sup> is the process of ensuring, that there are no mistakes (bugs) in a computer code or inconsistencies in the numerical algorithm. By performing code verification, a code can be tested for mistakes and can be followed by code debugging to identify and remove the mistakes. During the development of scientific software, there is a high possibility of making a number of coding mistakes. These mistakes need to be removed before using the code on real applications.

1. Coding mistake: A simple example of a coding mistake in a FORTRAN program is shown below.

FORTRAN Code:

```
TEMPNORM = 0
DO J = 2, JMAX - 1
    DO I = 2, JMAX - 1
        RES = RES + ((T(I,J) - TOLD(I,J))/DT)**2
        TEMPNORM = TEMPNORM + T(I,J)**2
    ENDDO
ENDDO
```

The underlined part of the code is a mistake in FORTRAN code shown here. The wrong term 'JMAX' is used instead of the right term 'IMAX'. This kind of coding mistake is a

possibility while developing a code. In the process of code verification, mistakes like the one shown above can be identified and removed.

2. Algorithm Inconsistency: For a consistent numerical algorithm, the discretized equations must approach the original partial differential equations in the limit as mesh size  $(\Delta x, \Delta t)$  approaches zero. A famous example of an inconsistent algorithm is the DuFort-Frankel<sup>51</sup> differencing of a 1D unsteady heat equation.<sup>52</sup>

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = 0 \quad (2.1)$$

For the 1D unsteady heat equation shown above, the DuFort-Frankel differencing is given by

$$\frac{T_j^{n+1} - T_j^{n-1}}{2\Delta t} = \frac{\alpha}{(\Delta x)^2} (T_{j+1}^n - T_j^{n+1} - T_j^{n-1} + T_{j-1}^n) \quad (2.2)$$

for which the leading terms in the truncation error are shown below.

$$TE = \frac{\alpha}{12} \frac{\partial^4 T}{\partial x^4} \Big|_{n,j} (\Delta x)^2 - \alpha \frac{\partial^2 T}{\partial x^2} \Big|_{n,j} \left( \frac{\Delta t}{\Delta x} \right)^2 - \frac{1}{6} \frac{\partial^3 T}{\partial x^3} \Big|_{n,j} (\Delta t)^2 \quad (2.3)$$

If  $\Delta x$  and  $\Delta t$  approach zero at the same rate such that  $\Delta t/\Delta x = \beta$ , then the discretized equations will not approach original partial difference equations, i.e., the 1D unsteady heat equation, instead they converge to a different partial difference equation shown below which is a hyperbolic equation.

$$\frac{\partial T}{\partial t} + \alpha\beta^2 \frac{\partial^2 T}{\partial t^2} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (2.4)$$

In the process of code verification, numerical inconsistencies can also be identified and corrected.

### 2.3 Order of Accuracy

Order of accuracy<sup>5,6,7</sup> is the rate of decrease of discretization error with mesh refinement. Order of accuracy test, which is equivalent to testing whether the formal order of accuracy matches the observed order of accuracy, is the most exercising code verification test. This test not only determines whether the solution is converging, but also whether or not the discretization error is reduced at the expected rate.

### 2.3.1 Formal Order of Accuracy

The formal order of accuracy of a numerical scheme is the rate at which the discretized equations approach the original partial differential equations. For all discretization approaches (finite difference, finite volume, finite element, etc.) the formal order of accuracy is obtained from a truncation error analysis of the discrete algorithm. The formal order will be the power of  $\Delta x$  or  $\Delta t$  in the leading terms in the truncation error. Any partial differential equation can be written as the sum of the finite difference equation and the truncation error. As an example, for a 1D unsteady heat equation, applying Taylor series and discretizing the 1D unsteady heat equation with a forward difference in time and a central difference in space, it can be written as:

$$\begin{aligned} \frac{T_i^{n+1} - T_i^n}{\Delta t} - \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2} \\ = \frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} + \left[ \frac{1}{2} \frac{\partial^2 T}{\partial t^2} \right] \Delta t + \left[ -\frac{\alpha}{12} \frac{\partial^4 T}{\partial t^4} \right] (\Delta x)^2 \\ + O((\Delta t)^2, (\Delta x)^4) \end{aligned} \quad (2.5)$$

From the above expression, the formal order of accuracy of the finite difference scheme is first order in time and second order in space since the leading terms in the truncation error contain the factors  $\Delta t$  and  $(\Delta x)^2$ , respectively.

### 2.3.2 Observed Order of Accuracy

The observed order of accuracy is computed directly from the code output for a given set of simulations on systematically refined grids. The observed order of accuracy will not match the formal order of accuracy due to mistakes in the computer code, defective numerical algorithms, and when numerical solutions are not in the asymptotic grid convergence range. For calculating the observed order of accuracy using two mesh levels, it is required to have an exact solution to find the discretization error.

Assuming that exact solution to the partial differential equations is known, let us now consider the method for calculating the observed order of accuracy. The discretization error is formally defined as the difference between the exact solution to the discrete equations and the exact solution to the governing partial differential equations. Since the exact solution to the discrete equations (which will be different on different mesh levels) is generally not known, the numerical solution on the same mesh level is substituted in its place, thus neglecting iterative

error and round-off error. The observed order of accuracy can be evaluated either locally within the solution domain or globally by employing a norm of the discretization error or for a global system response quantity. Consider a series expansion of the discretization error in terms of  $h_k$ , a measure of the element size on the mesh level  $k$ ,

$$DE_k = f_k - f_{exact} = g_p h_k^p + HOT \quad (2.6)$$

where  $f_k$  is the numerical solution on mesh  $k$ ,  $g_p$  is the coefficient of the leading error term, and  $p$  is the formal order of accuracy. Neglecting the higher order terms, we can write the discretization error equation for a fine mesh ( $k=1$ ) and a coarse mesh ( $k=2$ ) in terms of the observed order of accuracy  $\hat{p}$  as

$$\begin{aligned} DE_1 &= f_1 - f_{exact} = g_p h_1^{\hat{p}} \\ DE_2 &= f_2 - f_{exact} = g_p h_2^{\hat{p}} \end{aligned} \quad (2.7)$$

Since the exact solution is known, these two equations can be solved for the observed order of accuracy  $\hat{p}$ . Introducing  $r$ , the ratio of coarse to fine mesh element spacing ( $r = h_2/h_1 > 1$ ), the observed order of accuracy becomes

$$\hat{p} = \frac{\ln\left(\frac{DE_2}{DE_1}\right)}{\ln(r)} \quad (2.8)$$

Thus, when the exact solution is known, only two solutions are required to obtain the observed order of accuracy.

## 2.4 Method of Manufactured Solutions

Code verification using order of accuracy testing requires an exact solution to the governing equations which are tested in the code. Traditional exact solutions exist only for simple governing equations. For complex governing equations which can handle complex physics, complex geometries, and significant nonlinearities, it is difficult to find exact solutions. The Method of Manufactured Solutions, (MMS) is a general approach for obtaining exact solutions for code verification purposes.

### 2.4.1 Procedure for MMS

The procedure for applying MMS with order of accuracy verification is demonstrated below for a simple example problem.<sup>7</sup>

1. Choose the specific form of the governing equations; here we choose the linear 1D heat equation.

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = 0 \quad (2.9)$$

2. Choose the Manufactured Solution.

$$T(x, t) = T_0 \exp(t/t_0) \sin(\pi x/L) \quad (2.10)$$

3. Operate the governing equations on the chosen solution, resulting in analytic source terms.

$$\begin{aligned} \frac{\partial T}{\partial t} &= T_0 \sin(\pi x/L) \frac{1}{t_0} \exp(t/t_0) \\ \frac{\partial^2 T}{\partial x^2} &= -T_0 \exp(t/t_0) (\pi/L)^2 \sin(\pi x/L) \end{aligned} \quad (2.11)$$

4. Solve the modified governing equation (original equation plus source terms) on various mesh levels.

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = \left[ \frac{1}{t_0} + \alpha \left( \frac{\pi}{L} \right)^2 \right] T_0 \sin(\pi x/L) \frac{1}{t_0} \exp(t/t_0) \quad (2.12)$$

5. Compute the observed order of accuracy and compare it with the formal order of accuracy.

#### 2.4.2 Attributes of a Good Manufactured Solution

The chosen Manufactured Solution should be smooth, analytic functions with smooth derivatives. The choice of smooth derivatives will allow the formal order of accuracy to be achieved on relatively coarse meshes. Trigonometric and exponential functions are selected as Manufactured Solutions since they are smooth and infinitely differential. There is an advantage of using trigonometric functions as they can be adjusted to include only a fraction of an oscillation period over the domain which makes it easier to achieve the asymptotic range. The Manufactured Solutions need to be selected such that no derivatives vanish, including cross derivatives if they are present in the governing equations. Even though the Manufactured Solution is not required to be physically realistic, since we are only testing the mathematics, they should be chosen to give realizable physical states. For example, if the code requires that the

temperature to be positive for the calculation of speed of sound, then the Manufactured Solution should be chosen such that the temperature values are only positive.

A Manufactured Solution should be selected such that all the terms in the governing equation have similar magnitudes meaning one term in the governing equation should not dominate other terms. For example, a Manufactured Solution should be selected such that the convective terms and diffusive terms in Navier-Stokes equations have the same order of magnitude. It can be checked whether all the terms in the governing equations are roughly the same order of magnitude, by examining the ratios of those terms in the considered domain.

### **2.4.3 Advantages of Method of Manufactured Solutions**

There are several advantages of using MMS. This procedure can be applied to all discretization schemes (finite-difference, finite-volume, and finite element). MMS can also be applied without trouble when dealing with nonlinear equations and multiple equations like the Navier-Stokes equations. Roache<sup>5</sup> showed that by using MMS for code verification, very sensitive mistakes in the discretization can be determined. The sensitivity of MMS procedure for code verification is explained with this example where during the testing of a compressible Navier-Stokes code, the reason for non-convergence of global norms of the discretization error was found to be because of a small discrepancy in the 4<sup>th</sup> significant digit for the thermal conductivity between the governing equations and the numerical implementation in the code.<sup>29</sup>

### 3. Mesh Generation

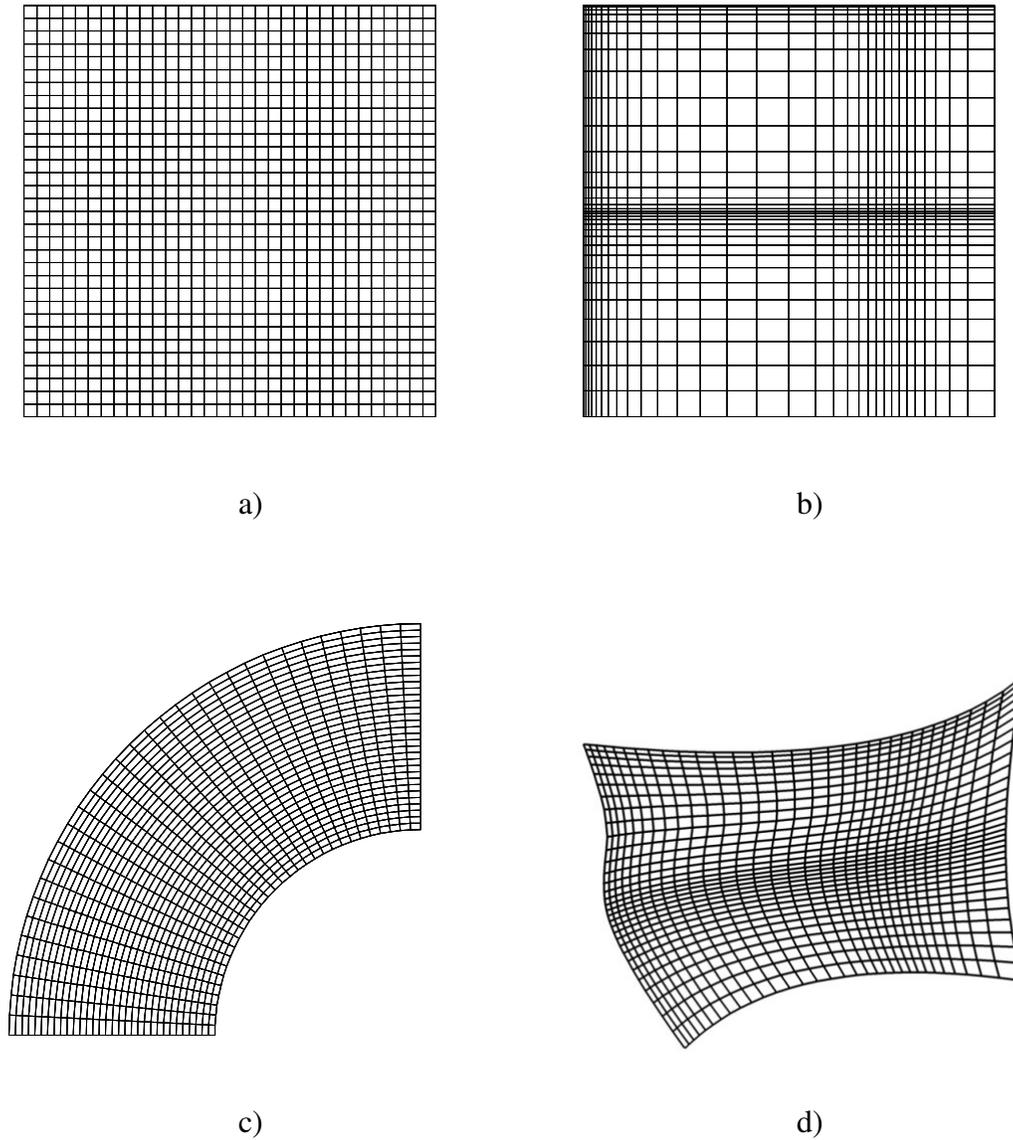
A finite volume CFD code is verified on different mesh topologies in the verification process. To verify all mesh transformations are coded correctly, the code needs to be run on the most general mesh types<sup>29,38</sup> which include meshes with mild skewness, aspect ratio, curvature, and stretching. The most general mesh types are termed as hybrid meshes which contain different mesh topologies. However, the code needs to be run on simpler meshes if the code verification fails on the general meshes. Simpler meshes include meshes with isolated mesh topologies and mesh qualities. The different mesh topologies considered for code verification in both 2D and 3D are classified as structured, unstructured, and hybrid meshes which are a combination of structured and unstructured meshes.

#### 3.1 2D Mesh Topologies

Different 2D structured and unstructured mesh topologies are used during the code verification of the finite volume CFD code. The most general mesh topology used for 2D verification is a 2D hybrid mesh which includes quadrilateral and triangular cells with curvilinear boundaries, skewed cells, and stretched cells.

##### 3.1.1 2D Structured Meshes

The 2D structured meshes used for code verification are the Cartesian mesh, the stretched Cartesian mesh, the curvilinear mesh, and the skewed curvilinear mesh. The 2D structured meshes are shown in Figure 1. The 2D structured meshes considered here contain quadrilateral cells. By testing the code on these meshes, the behavior of the code on the quadrilateral cell topology along with its cell quality effect can be studied. All these 2D structured meshes are generated using the mesh generation tool GRIDGEN.<sup>57</sup> The stretched Cartesian mesh can be used to isolate the effects of grid stretching and aspect ratio, the curvilinear mesh can be used to test the effects of curved boundaries without the presence of skewness or stretching, and the skewed curvilinear mesh tests all the effects on a single structured mesh type.

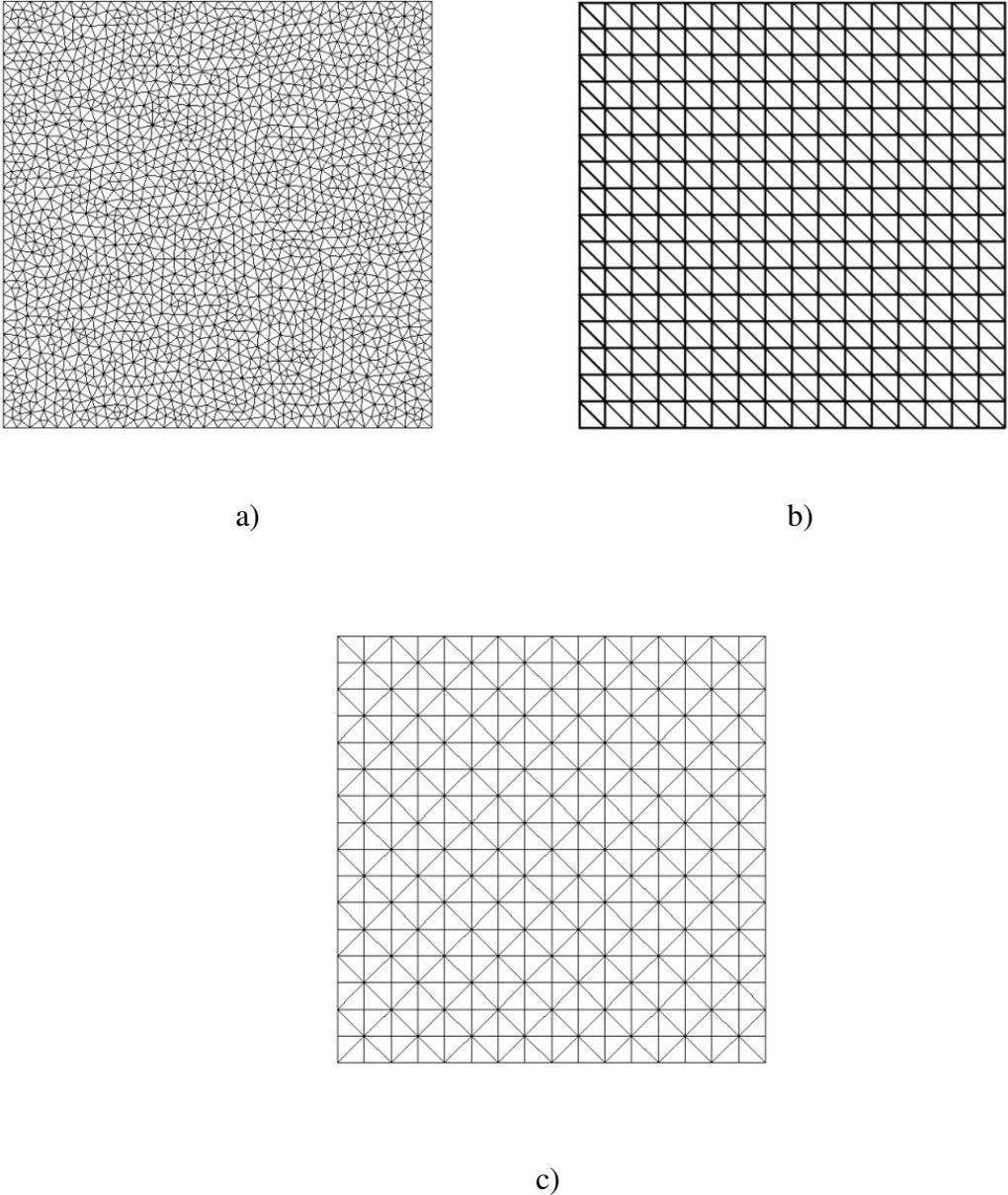


**Figure 1: 2D structured meshes: a) Cartesian, b) stretched Cartesian, c) curvilinear, and d) skewed curvilinear**

### 3.1.2 2D Unstructured Meshes

Selected 2D unstructured meshes used for code verification are shown in Figure 2. The 2D unstructured meshes considered here contain triangular cells. The general unstructured mesh is generated by automatic mesh generation in GRIDGEN. Other unstructured meshes, i.e., the unidirectional diagonal and bidirectional diagonal unstructured meshes shown in Figure 2, are generated by starting from a structured mesh and then adding diagonals in the quadrilateral cells.

The 2D unstructured meshes generated by adding diagonals in a structured mesh are used in the verification study to achieve a uniform and consistent refinement between different mesh levels. By testing the code on these meshes, the behavior of the code on the triangular cell topology along with its cell quality effect can be studied. The concept of uniform and consistent mesh refinement and its necessity for code verification purposes are discussed in Section 3.3.



**Figure 2: 2D unstructured meshes: a) general unstructured, b) uni-directional diagonal, and c) alternate diagonal**

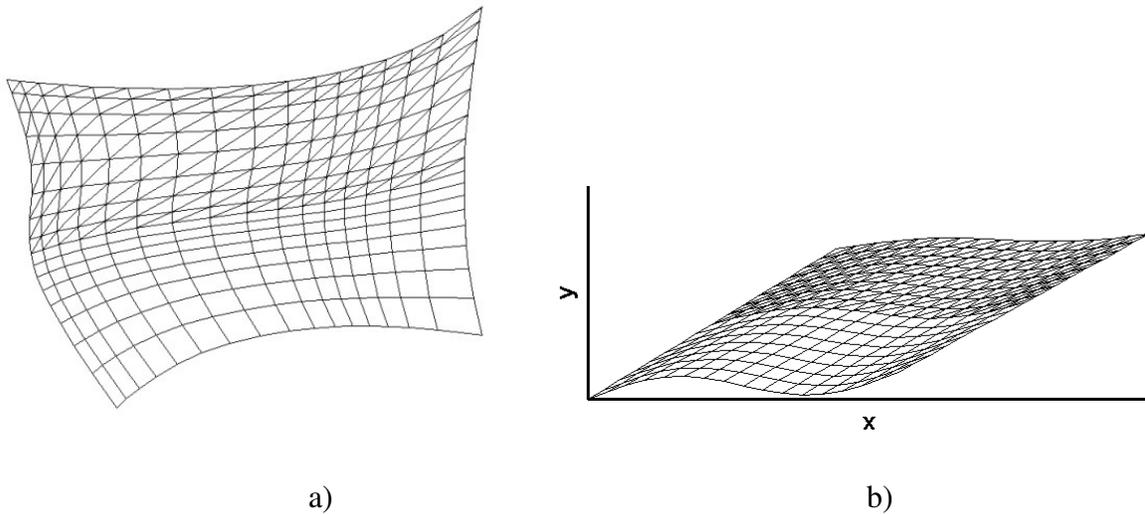
### 3.1.3 2D Hybrid Meshes

A 2D hybrid mesh contains both quadrilateral and triangular cell topologies. The 2D skewed hybrid mesh and the highly skewed hybrid mesh considered for code verification are shown

in a)

b)

Figure 3. The 2D hybrid meshes are also generated using the mesh generation tool GRIDGEN. By performing code verification on these meshes, the behavior of the code on both the triangular and quadrilateral cell topologies with all the different cell quality effects is tested.



**Figure 3: 2D hybrid meshes: a) skewed hybrid and b) highly skewed hybrid**

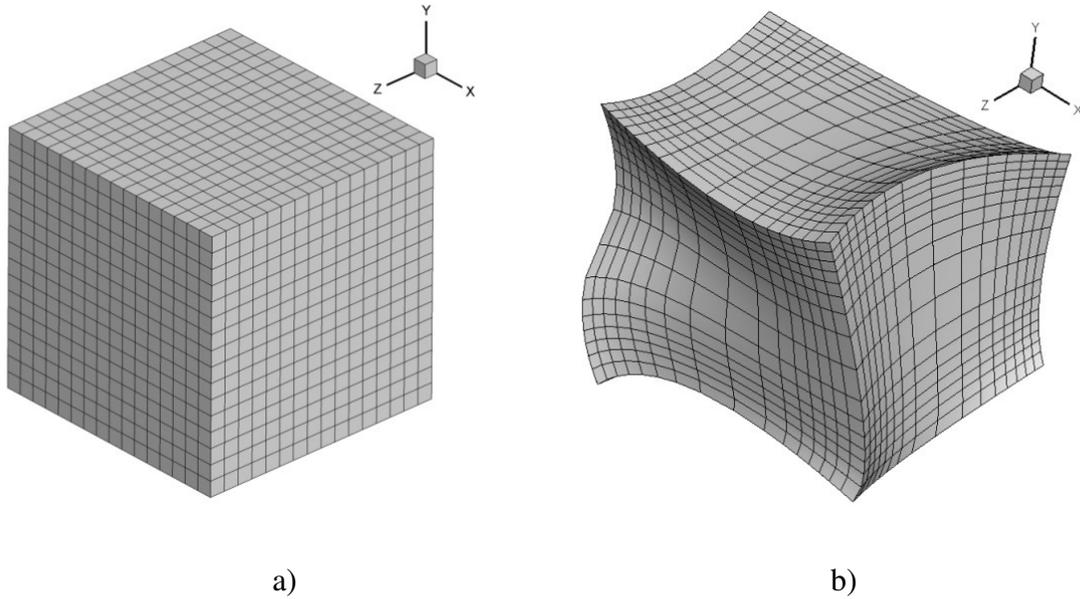
## 3.2 3D Mesh Topologies

Different 3D structured and unstructured mesh topologies are used during the code verification of the finite volume CFD code. The most general mesh topology used for 3D verification is a 3D hybrid mesh which includes hexahedral cells, tetrahedral cells, and prismatic cells with curvilinear boundaries, skewed cells, and stretched cells.

### 3.2.1 3D Structured Meshes

The 3D structured meshes used during code verification include the Cartesian mesh and the skewed curvilinear mesh. The 3D structured meshes are shown in Figure 4. The 3D structured meshes contain hexahedral cells. By testing the code on these meshes, the behavior of the code on the hexahedral cell topology along with its cell quality effect can be studied. All the

3D structured meshes are also generated using GRIDGEN. The skewed curvilinear mesh is generated to test the effects of aspect ratio, skewness, stretching and the effect of curved boundaries on the code.

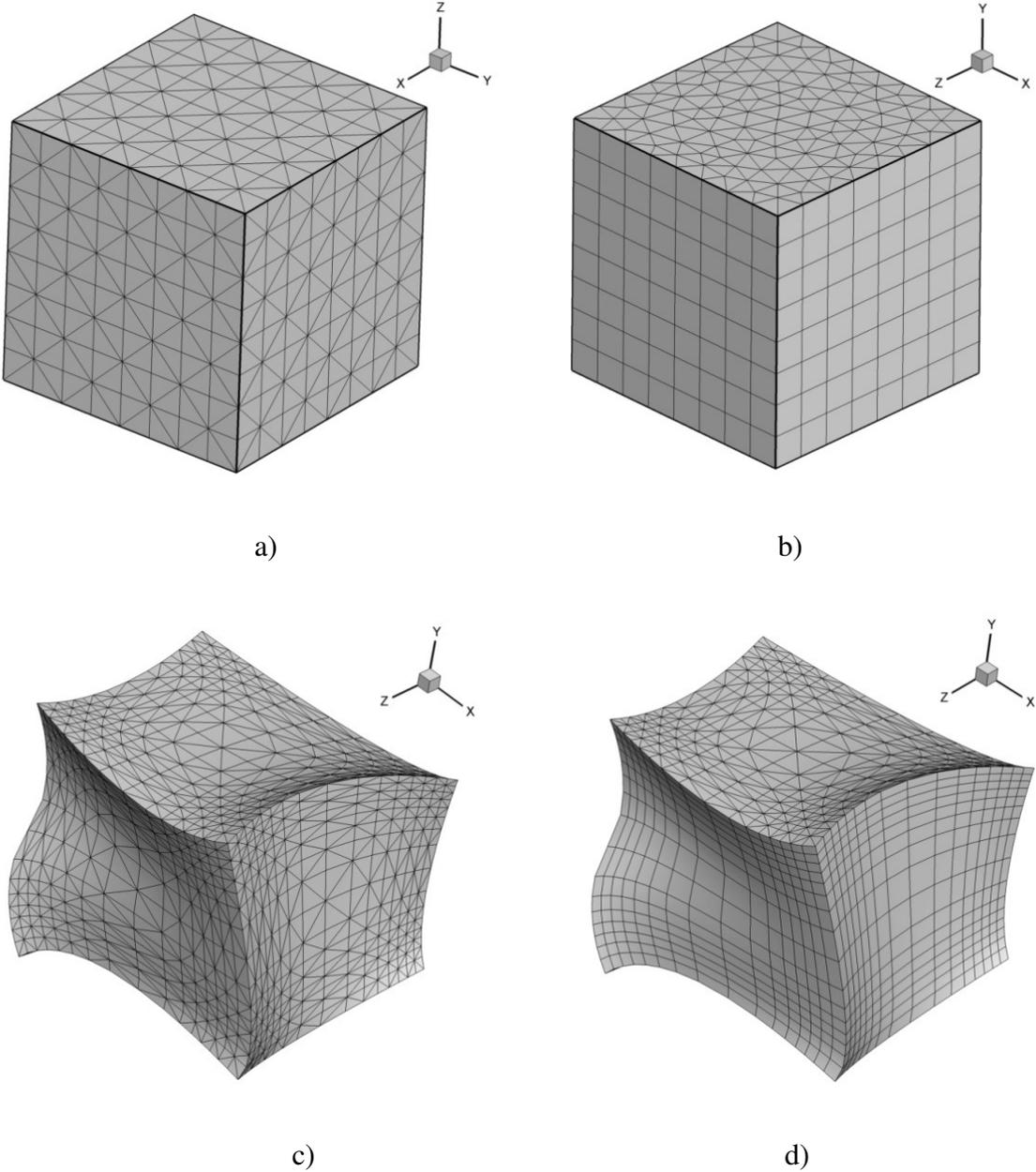


**Figure 4: 3D structured meshes: a) Cartesian and b) skewed curvilinear**

### 3.2.2 3D Unstructured Meshes

Different 3D unstructured meshes used for code verification are shown in Figure 5. The 3D unstructured meshes considered here contain tetrahedral cells and prismatic cells. By testing the code on these meshes, the behavior of the code on the tetrahedral and prismatic cell topologies along with their cell quality effects can be studied. The 3D unstructured meshes used during code verification include the unstructured meshes with tetrahedral and prismatic cells as shown in Figure 5a and Figure 5b, respectively, and these meshes have cells close to isotropic cells. The highly skewed unstructured meshes with tetrahedral and prismatic cells are shown in Figure 5c and Figure 5d, respectively, and these meshes contain highly skewed and stretched cells which can be used to study the cell quality effect while testing the code. The unstructured mesh with prismatic cells is generated in GRIDGEN by starting with an unstructured 2D domain and projecting in the third direction normal to the 2D domain. For generating all the other 3D unstructured meshes, particularly for the unstructured tetrahedral meshes, a mesh generation

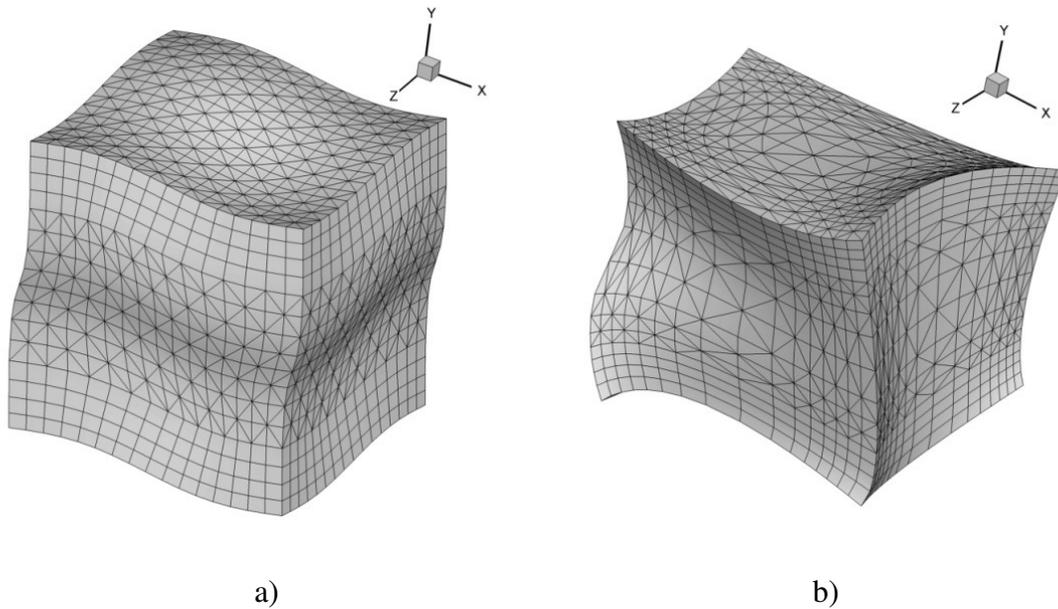
code is developed using Fortran. The reason for developing a Fortran code to generate these 3D unstructured meshes is discussed in Section 3.3.



**Figure 5: 3D unstructured meshes: a) unstructured mesh with tetrahedral cells, unstructured mesh with prismatic cells, c) highly skewed unstructured mesh with tetrahedral cells, and d) highly skewed unstructured mesh with prismatic cells**

### 3.2.3 3D Hybrid Meshes

The skewed 3D hybrid mesh and the highly skewed 3D hybrid mesh used for code verification are shown in Figure 6. The 3D hybrid meshes considered contain hexahedral, tetrahedral, and prismatic cells. By testing the code on these meshes, the behavior of the code on the hexahedral, tetrahedral and prismatic cell topologies along with their cell quality effect can be studied. To isolate the cell quality effects, 3D hybrid meshes which have cells close to isotropic can be used to test the code. Again, for generating all the 3D hybrid meshes, a mesh generation code is developed using Fortran.

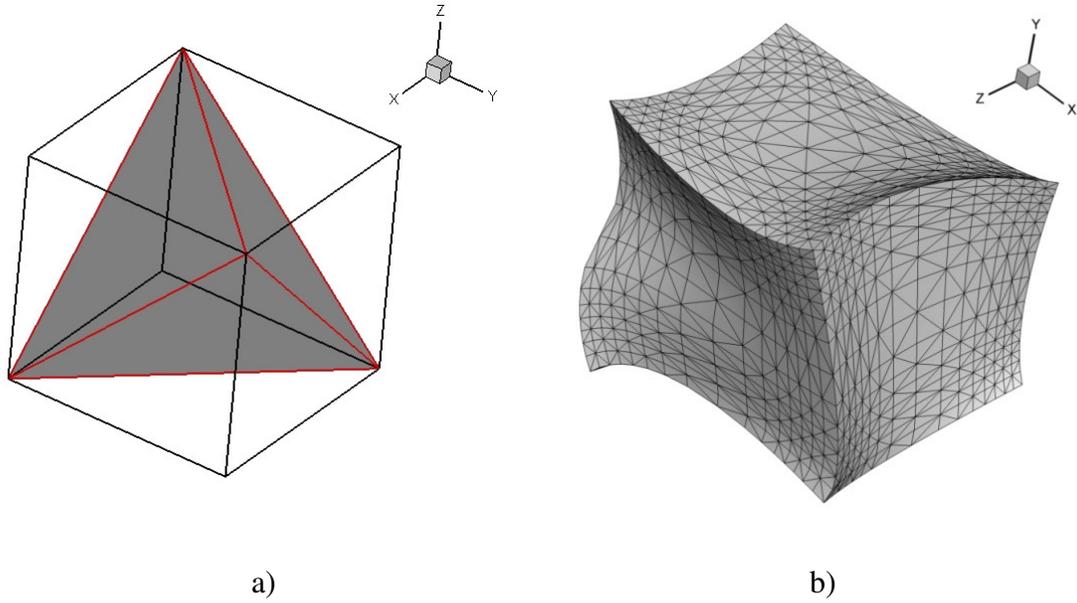


**Figure 6: 3D hybrid meshes: a) skewed hybrid and b) highly skewed hybrid**

### 3.3 Systematic Mesh Refinement

Systematic mesh refinement<sup>7</sup> is defined as uniform and consistent refinement over a spatial domain. A mesh is said to be uniformly refined if the mesh is refined in all the coordinate directions equally and it is said to be consistently refined if the mesh quality stays constant or improves with mesh refinement. For the purpose of code verification, it is necessary to have a systematic mesh refinement.<sup>7</sup> In the case of structured meshes, refinement/coarsening of the meshes for the verification purpose are straightforward. A coarse mesh is generated from a fine mesh by removing every alternate mesh point or mesh line to produce mesh levels with a refinement factor of two. In this process, the mesh quality is maintained for the structured

meshes. But in the case of unstructured meshes refinement/coarsening of meshes with a uniform refinement factor throughout the domain preserving the mesh quality is more challenging, particularly in 3D. In the 2D case, systematic mesh refinement can be achieved by generating an unstructured mesh from a structured mesh by splitting quadrilaterals into triangles using diagonals.<sup>38</sup> To the author's knowledge, generation of 3D unstructured meshes with uniform refinement preserving the mesh quality has not yet been achieved using commercial software. Therefore, for code verification purposes, a mesh generation code is developed to generate 3D unstructured meshes from a 3D structured mesh with hexahedral elements. In the process, a cube will be split into five tetrahedral cells as shown in Figure 7a. The central tetrahedral cell, which does not share a surface boundary with the parent cube, is isotropic and the other four tetrahedral cells have the same topology with good cell quality. An unstructured mesh with tetrahedral elements generated using the mesh generation code is shown in Figure 7b. By generating unstructured meshes in this fashion, a uniform and consistent refinement can be achieved by a uniform refinement factor and maintaining the cell quality between the mesh levels. Based on this concept of generating an unstructured mesh from a structured mesh, another code for generating 3D hybrid meshes which contain hexahedral, tetrahedral, and prismatic cells with proper connectivity between different cell types was also developed. The prismatic cells in the 3D hybrid mesh are generated by diagonally splitting a hexahedral cell into two prismatic cells. To obtain a hybrid mesh from a structured mesh of hexahedral cells, 25 percent of the hexahedral cells are split into prismatic cells, 50 percent of the hexahedral cells are split into tetrahedral cells and the other 25 percent are left as hexahedral cells. The 3D hybrid meshes are generated to satisfy the uniform and consistent refinement criteria between the mesh levels required for code order of accuracy verification.



**Figure 7: a) Cube split into five tetrahedral cells, b) 3D unstructured mesh with tetrahedral cells generated from a skewed and stretched curvilinear mesh**

The different mesh levels and mesh types used for code verification are given in Table 1. A maximum of seven mesh levels are used for 2D mesh topologies and a maximum of five mesh levels are used for 3D mesh topologies with systematic mesh refinement between consecutive mesh levels.

**Table 1. Different mesh levels and mesh types**

2D Mesh Topologies			3D Mesh Topologies		
Structured	Unstructured	Hybrid	Structured	Unstructured	Hybrid
8×8	128	96	8×8×8	320	1664
16×16	512	384	16×16×16	2560	13312
32×32	1024	1536	32×32×32	20480	106496
64×64	4096	6144	64×64×64	163840	851968
128×128	16384	24576	128×128×128	1310720	6815744
256×256	65536	98304			
512×512	262144	393216			

## 4. Verification of a Finite Volume CFD Code

During the verification of a finite volume CFD code, different code options need to be tested which include baseline steady-state governing equations, Sutherland's law for viscosity, equation of state, boundary conditions, turbulence models, time accuracy of unsteady flows, etc. The options in the code are verified by comparing the observed order of accuracy calculated for the CFD solutions for multiple systematically-refined meshes to the formal order of accuracy of the numerical method. An option in the code is considered fully verified if it passes the order of accuracy test on the 2D hybrid mesh and 3D hybrid mesh with all cell topologies which include skewness, aspect ratio, curvature, and mesh stretching. When a verification test fails on the hybrid meshes, then the governing equations are tested on other simpler meshes to find whether the discrete formulation of the governing equations is inconsistent on a particular mesh topology or due to the cell quality attributes or coding mistakes.

During the code verification process, the observed order of accuracy is calculated using the  $L_2$ ,  $L_1$  and  $L_\infty$  norms of the discretization error. Normally the observed order of accuracy calculated using  $L_2$  and  $L_1$  norms of the discretization error showed similar behavior, but the observed order of accuracy calculated using  $L_\infty$  norm of the discretization error asymptoted to the formal order of accuracy at a slower rate requiring more mesh levels to see the asymptotic behavior. The observed order of accuracy results are shown for the different options verified on different meshes using  $L_1$  and  $L_\infty$  norms of the discretization error. Also, the verification test results of different options in the finite volume code are shown mostly on 2D and 3D hybrid meshes. The verification test results on other meshes are shown for some interesting cases when the code options had problems during testing. A summary of the options verified in the finite volume CFD code is shown in Appendix A.

### 4.1 Finite Volume Code

The code verification procedure can be applied to any scientific computing code in general, but in the current work, the finite volume CFD code verified is Loci-CHEM.<sup>16,58</sup> Loci-CHEM was developed at Mississippi State University using the Loci framework<sup>59,60</sup> and can simulate three-dimensional flows of turbulent, chemically-reacting mixtures of thermally perfect gases. The Loci framework provides a high-level programming environment for numerical

methods that is automatically parallel and utilizes a logic-based strategy to detect or prevent common software faults (such as errors in loop bounds or errors caused by subroutine calling sequences being inconsistent with data dependencies). The code uses an unstructured, edge based method with a formal order of accuracy of two.

## 4.2 Verification of Baseline Governing Equations

The baseline governing equations include the 3D steady state Euler and the Navier-Stokes equations. Removing the viscous terms from the 3D Navier-Stokes equations lead to 3D Euler equations and removing the z-direction variables ( $\partial/\partial z = 0$  and  $w = 0$ ) converts the equations in three dimensions to two dimensions.

### 4.2.1 Flow Equations

The 3D, steady state, Farve-averaged Navier-Stokes equations<sup>61</sup> can be written as

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \quad (4.1)$$

$$\frac{\partial(\rho u^2 + p - t_{xx} - \tau_{xx})}{\partial x} + \frac{\partial(\rho uv - t_{xy} - \tau_{xy})}{\partial y} + \frac{\partial(\rho uw - t_{xz} - \tau_{xz})}{\partial z} = 0 \quad (4.2)$$

$$\frac{\partial(\rho vu - t_{xy} - \tau_{xy})}{\partial x} + \frac{\partial(\rho v^2 + p - t_{yy} - \tau_{yy})}{\partial y} + \frac{\partial(\rho vw - t_{yz} - \tau_{yz})}{\partial z} = 0 \quad (4.3)$$

$$\frac{\partial(\rho wu - t_{xz} - \tau_{xz})}{\partial x} + \frac{\partial(\rho wv - t_{yz} - \tau_{yz})}{\partial y} + \frac{\partial(\rho w^2 + p - t_{zz} - \tau_{zz})}{\partial z} = 0 \quad (4.4)$$

$$\begin{aligned} & \frac{\partial(\rho u h_t - u(t_{xx} + \tau_{xx}) - v(t_{xy} + \tau_{xy}) - w(t_{xz} + \tau_{xz}) + q_{L_x} + q_{T_x})}{\partial x} \\ & + \frac{\partial(\rho v h_t - u(t_{xy} + \tau_{xy}) - v(t_{yy} + \tau_{yy}) - w(t_{yz} + \tau_{yz}) + q_{L_y} + q_{T_y})}{\partial y} \\ & + \frac{\partial(\rho w h_t - u(t_{xz} + \tau_{xz}) - v(t_{yz} + \tau_{yz}) - w(t_{zz} + \tau_{zz}) + q_{L_z} + q_{T_z})}{\partial z} = 0 \end{aligned} \quad (4.5)$$

where  $t_{ij}$  is the laminar stress tensor given by

$$\begin{aligned} t_{xx} &= \frac{2}{3}\mu \left( 2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right), & t_{yy} &= \frac{2}{3}\mu \left( 2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right), \\ t_{zz} &= \frac{2}{3}\mu \left( 2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \\ t_{xy} &= \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), & t_{yz} &= \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right), & t_{xz} &= \mu \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \end{aligned} \quad (4.6)$$

and  $\tau_{ij}$  is the turbulent stress tensor given by

$$\begin{aligned}\tau_{xx} &= \frac{2}{3}\mu_T \left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}\right), & \tau_{yy} &= \frac{2}{3}\mu_T \left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z}\right), \\ \tau_{zz} &= \frac{2}{3}\mu_T \left(2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right), & & \\ \tau_{xy} &= \mu_T \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), & \tau_{yz} &= \mu_T \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right), & \tau_{xz} &= \mu_T \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right).\end{aligned}\tag{4.7}$$

The turbulent and laminar heat flux terms are

$$\begin{aligned}q_{T_x} &= -\frac{\mu_T}{Pr_T} c_p \frac{\partial T}{\partial x}, & q_{T_y} &= -\frac{\mu_T}{Pr_T} c_p \frac{\partial T}{\partial y}, & q_{T_z} &= -\frac{\mu_T}{Pr_T} c_p \frac{\partial T}{\partial z} \\ q_{L_x} &= -\frac{\mu}{Pr} c_p \frac{\partial T}{\partial x}, & q_{L_y} &= -\frac{\mu}{Pr} c_p \frac{\partial T}{\partial y}, & q_{L_z} &= -\frac{\mu}{Pr} c_p \frac{\partial T}{\partial z}\end{aligned}\tag{4.8}$$

and the total energy and enthalpy are

$$e_t = e + \frac{1}{2}(u^2 + v^2 + w^2) \quad \text{and} \quad h_t = e_t + \frac{P}{\rho}\tag{4.9}$$

where

$$e = nRT + h_f \quad \text{and} \quad h = (n+1)RT + h_f.\tag{4.10}$$

The perfect gas equation of state is assumed as

$$P = \rho RT\tag{4.11}$$

and the heat of formation and excited energy mode parameter can be calculated as

$$h_f = h_{ref} - (n+1)RT_{ref} \quad \text{and} \quad n = \frac{c_v}{R}\tag{4.12}$$

where  $h_{ref} = 0$ ,  $T_{ref} = 298\text{K}$ , and  $n = 5/2$  are used.

#### 4.2.2 Verification of the Euler Equations

In our current work, we adhere to the philosophy that code verification is simply a mathematical test to ensure the numerical solution truly represents the solution to the continuum mathematical equations that are being solved. As such, we specifically choose the Manufactured Solutions which are not physically realistic, but which are simple, smooth, and exercise all terms in the governing equations. The steady Manufactured Solutions employed take the following general form

$$\begin{aligned}
\phi(x, y, z) = & \phi_0 + \phi_x f_s \left( \frac{\alpha_{\phi x} \pi x}{L} \right) + \phi_y f_s \left( \frac{\alpha_{\phi y} \pi y}{L} \right) + \phi_z f_s \left( \frac{\alpha_{\phi z} \pi z}{L} \right) \\
& + \phi_{xy} f_s \left( \frac{\alpha_{\phi xy} \pi xy}{L^2} \right) + \phi_{yz} f_s \left( \frac{\alpha_{\phi yz} \pi yz}{L^2} \right) \\
& + \phi_{zx} f_s \left( \frac{\alpha_{\phi zx} \pi zx}{L^2} \right)
\end{aligned} \tag{4.13}$$

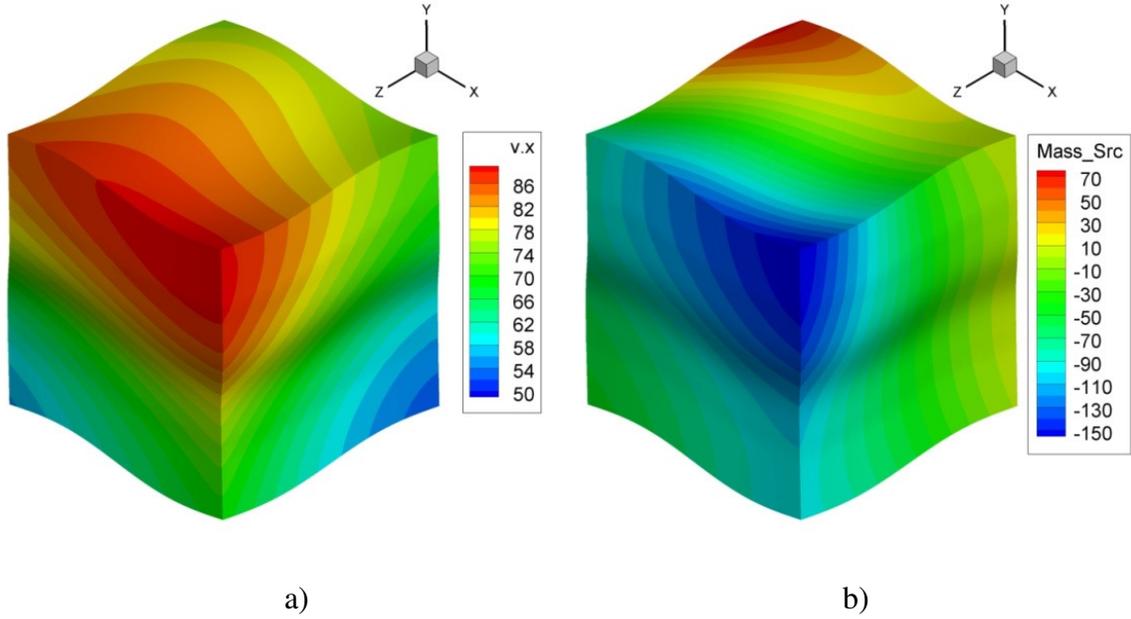
where  $\phi = [\rho, u, v, w, p, k, \omega]^T$  represents any of the primitive variables and the  $f_s(\cdot)$  functions represent sine or cosine functions. After selecting the constants in the above equation, the 2D Manufactured Solution used for the verification of Euler equations is given as

$$\begin{aligned}
\rho &= 1 + 0.15 \cos(0.75\pi x) - 0.1 \sin(\pi y) + 0.08 \cos(1.25\pi xy) \\
u &= 70 + 5 \sin(1.5\pi x) - 15 \cos(1.5\pi y) + 7 \cos(0.6\pi xy) \\
v &= 90 - 5 \cos(1.5\pi x) + 10 \cos(\pi y) - 11 \sin(0.9\pi xy) \\
p &= 100000 + 20000 \cos(\pi x) - 50000 \sin(1.25\pi y) \\
&\quad - 25000 \sin(0.75\pi xy)
\end{aligned} \tag{4.14}$$

The 3D Manufactured Solution used for the verification of Euler equations is given as

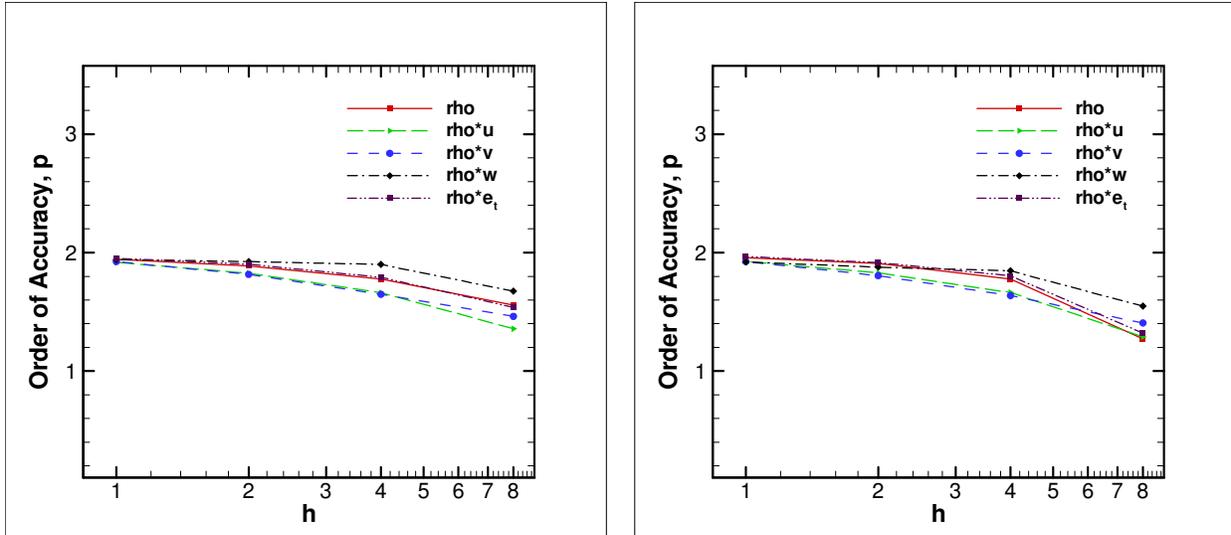
$$\begin{aligned}
\rho &= 1 + 0.15 \cos(0.75\pi x) - 0.1 \sin(0.45\pi y) + 0.1 \sin(0.8\pi z) \\
&\quad + 0.08 \cos(0.65\pi xy) + 0.05 \sin(0.75\pi yz) \\
&\quad + 0.12 \cos(0.5\pi zx) \\
u &= 70 + 5 \sin(0.5\pi x) - 15 \cos(0.85\pi y) - 10 \cos(0.4\pi z) \\
&\quad + 7 \cos(0.6\pi xy) + 4 \sin(0.8\pi yz) - 4 \cos(0.9\pi zx) \\
v &= 90 - 5 \sin(0.8\pi x) + 10 \cos(0.8\pi y) + 5 \cos(0.5\pi z) \\
&\quad - 11 \cos(0.9\pi xy) - 5 \sin(0.4\pi yz) + 5 \cos(0.6\pi zx) \\
w &= 80 - 10 \cos(0.85\pi x) + 10 \sin(0.9\pi y) + 12 \cos(0.5\pi z) \\
&\quad - 12 \sin(0.4\pi xy) + 11 \sin(0.8\pi yz) + 5 \cos(0.75\pi zx) \\
p &= 100000 + 20000 \cos(0.4\pi x) - 50000 \cos(0.45\pi y) \\
&\quad + 20000 \sin(0.85\pi z) - 25000 \cos(0.75\pi xy) \\
&\quad - 10000 \sin(0.7\pi yz) + 10000 \cos(0.8\pi zx)
\end{aligned} \tag{4.15}$$

These Manufactured Solutions are smooth and the source terms generated after applying them to the governing equations also vary smoothly over the considered domain. As an example, a Manufactured Solution for the x-component of velocity is shown in Figure 8a and a smooth analytic source term in 3D domain is shown in Figure 8b.



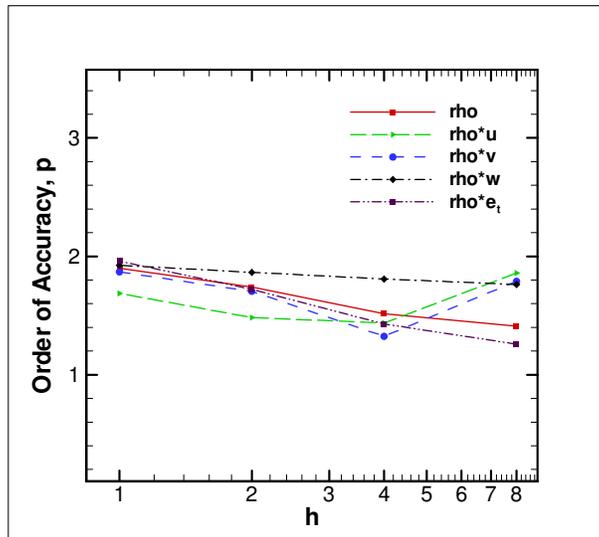
**Figure 8: a) Manufactured Solution of u-velocity and b) mass source term**

The Euler equations are discretized and solved on different mesh levels to calculate the observed order of accuracy. The code is then tested on the structured, unstructured and hybrid meshes in both 2D and 3D and is verified successfully to be second order accurate. The order of accuracy results for the verification of Euler equation are shown for the 3D hybrid case only. The observed order of accuracy results calculated using  $L_2$ ,  $L_1$  and  $L_\infty$  norms of the discretization error on the 3D skewed hybrid mesh are shown in Figure 9. In the plots shown, the observed order of accuracy  $p$  is calculated for all the conserved variables and is plotted on the y-axis against the normalized mesh size  $h$  on the x-axis. Five mesh levels are used for the verification of Euler equations in 3D with a uniform mesh refinement of two between consecutive mesh levels. The coarsest skewed hybrid mesh is generated starting from a structured mesh containing  $8 \times 8 \times 8$  cells and the finest skewed hybrid mesh is generated starting from a structured mesh containing  $128 \times 128 \times 128$  cells. The value of  $h = 1$  means the finest mesh and a higher  $h$  value represents a coarser mesh. It is seen in the plot that as the mesh is refined, the observed order of accuracy asymptotes to two which is the formal order of the code. As mentioned previously, the observed order of accuracy results using  $L_2$  and  $L_1$  norms of the discretization error are similar and for the subsequent verification results, only the observed order of accuracy results calculated using  $L_1$  and  $L_\infty$  norms are presented.



a)

b)



c)

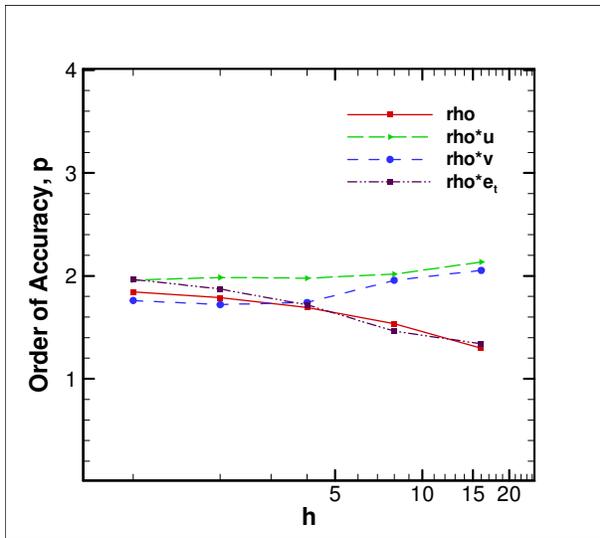
**Figure 9: Order of accuracy results for Euler equations on a 3D skewed hybrid mesh using a)  $L_1$  norm of the discretization error, b)  $L_2$  norm of the discretization error, and c)  $L_\infty$  norm of the discretization error**

### 4.2.3 Verification of the Navier-Stokes Equations

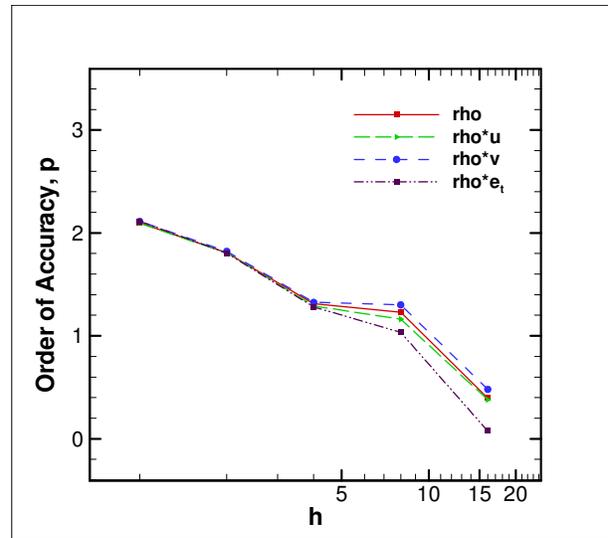
Successful verification of the Euler equations on a particular mesh and failure of the order of accuracy test for the Navier-Stokes equations means that there is a problem with the formulation of the diffusion operator on that particular mesh, i.e., an algorithm inconsistency, a

coding mistake, or simply mesh quality sensitivity. The Navier-Stokes equations are tested on the structured, unstructured and hybrid meshes in both 2D and 3D and in the process of testing, several issues in the formulation of the diffusion operator in the code were uncovered and corrected. The Manufactured Solution used for the Euler equations is also used for the Navier-Stokes equations.

In the process of selecting the Manufactured Solution for code verification purposes, it is required that different terms in the governing equations be roughly same order of magnitude. This prevents the larger magnitude errors from masking errors in other terms of smaller magnitude. In the current work, during the verification of Navier-Stokes equations, a constant viscosity value of  $10 \text{ N/m}^2$  is used such that there is an approximately equal contribution from the inviscid and viscous terms. The Navier-Stokes equations are verified successfully to be second order accurate on the 2D hybrid and 3D hybrid meshes after correcting the formulation of the diffusion operator each time the code failed on a particular mesh. The observed order of accuracy results calculated using  $L_1$  and  $L_\infty$  norms of the discretization error for the Navier-Stokes equations on the 2D hybrid mesh are shown in Figure 10. The observed order of accuracy results calculated using  $L_1$  and  $L_\infty$  norms of the discretization error for the Navier-Stokes equations on the 3D skewed hybrid mesh are shown in Figure 11.

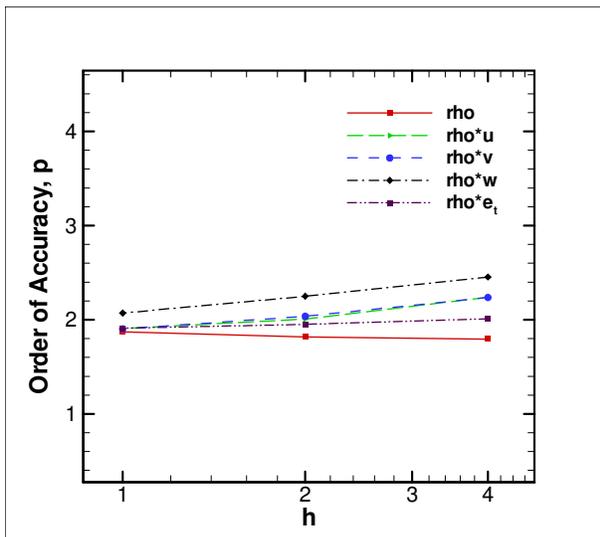


a)

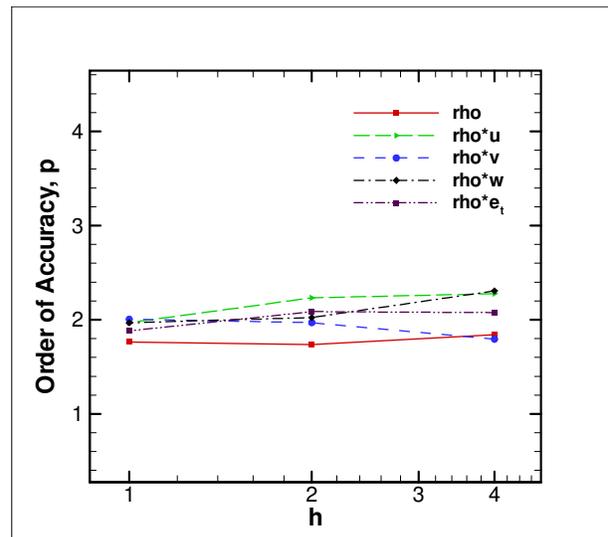


b)

**Figure 10: Order of accuracy results for Navier-Stokes equations on a 2D hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**



a)



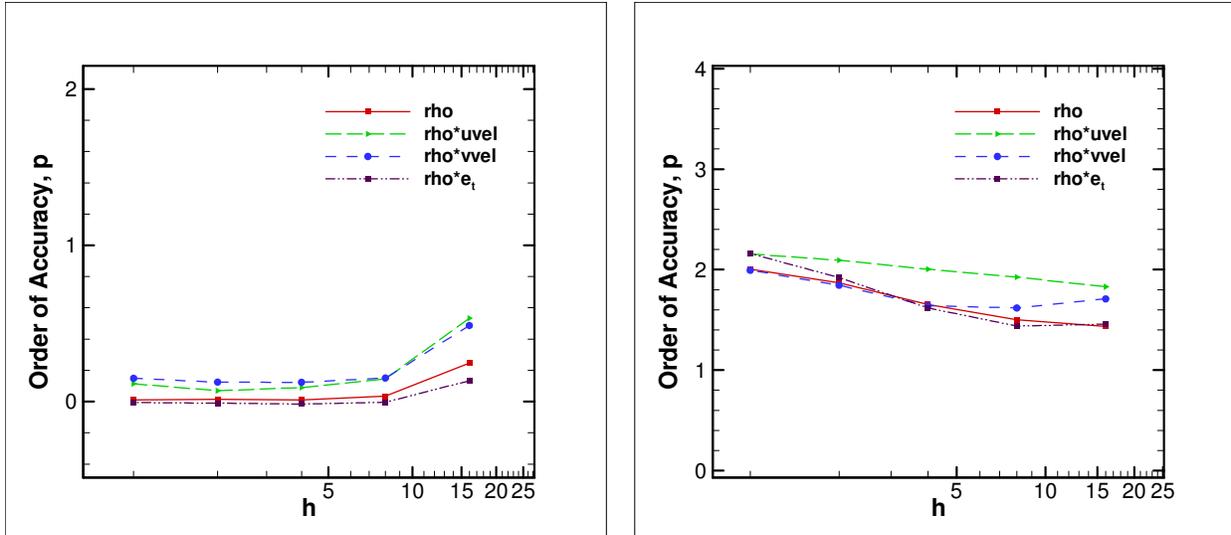
b)

**Figure 11: Order of accuracy results for Navier-Stokes equations on a 3D hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

The code was tested on different mesh types and failed initially on simpler meshes. Each time the code failed on a particular mesh, a problem with the numerical formulation of diffusion operator was detected and a correction in the numerical formulation was made by the code

developers. Initially, the Navier-Stokes equations were successfully verified on 2D rectangular Cartesian and stretched Cartesian meshes, but on the skewed curvilinear meshes, the discretization error did not decrease with mesh refinement. Since the code was already verified when Euler equations were tested, the problem or the error was found to be in the diffusion operator formulation. A modification was made to the diffusion operator by Luke<sup>39</sup> and the new diffusion operator rectified the problem. An observed order of accuracy of two was attained with mesh refinement after the modification. The code verification results on the skewed curvilinear mesh with the original diffusion operator and the new diffusion operator are shown in Figure 12.

The improvement in the observed order of accuracy results with the use of modified diffusion operator for the structured meshes is explained here. In the cell-centered finite volume diffusion operator, gradients are required at the cell faces to compute the viscous fluxes. Since unstructured finite-volume CFD codes typically compute and store gradients at the cell centers a mechanism for obtaining gradients at the faces is required. The Loci-CHEM code calculates the gradient at the face by computing both normal and tangential components of the gradient. The original formulation for calculating the normal component of the face gradient utilized the strategy suggested by Strang et al.<sup>62</sup> for the Cobalt 60 code. This approach effectively neglects a term in the normal gradient which can lead to stencils with negative weights (which affects the code's stability). Luke<sup>39</sup> modified the normal gradient calculation such that a limiter is applied to the offending term that both maintains second order accuracy in smooth regions of the flow and ensures a positive stencil.



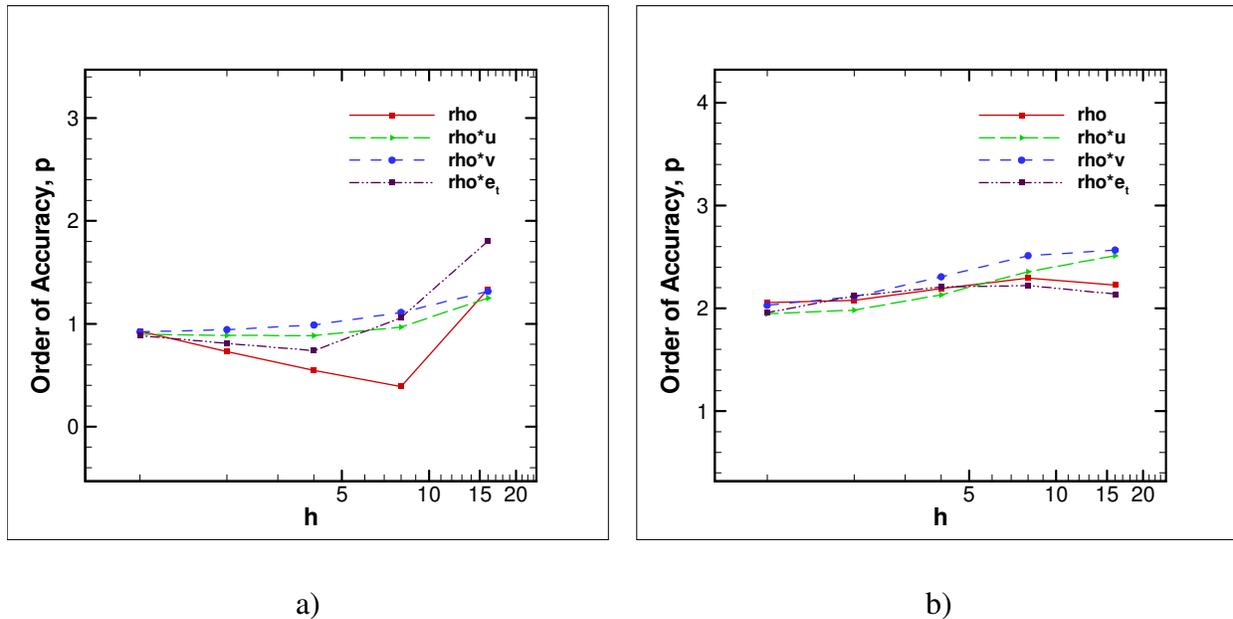
a)

b)

**Figure 12: Order of accuracy results using  $L_2$  norm of the discretization error for Navier-Stokes equations on a 2D skewed curvilinear mesh using a) original diffusion operator and b) new diffusion operator**

Later on 2D unstructured mesh with triangular cells, preliminary investigations showed that the observed order of accuracy was approaching one with mesh refinement. To determine the reason for this reduction in order of accuracy, the Navier-Stokes equations were tested on different types of unstructured meshes (triangular cells generated from a structured topology). On all these 2D unstructured meshes with triangular cells, the behavior was the same. The Euler equation were successfully verified to be second order accurate on these 2D unstructured meshes which explained that the problem was with the diffusion operator formulation on the 2D unstructured meshes. Again, Luke<sup>63</sup> came up with a modification in the formulation of the diffusion operator which rectified the problem. The estimation of gradients at the face centers is the cornerstone of the diffusion flux computations which are required to be second order accurate. In the old diffusion operator used in Loci-CHEM, gradients at face centers were computed by splitting the gradients into tangential and normal components and the diffusion operator was using the inviscid operator (gradients at cell centers). While the inviscid operator retained second order accuracy, the diffusion operator degenerated to first order as stencils departed symmetry. In order to more reliably obtain second order gradients at the face centers, a modified method was considered. In this method, a new control volume centered about the face

was constructed and a second order function reconstruction was used to compute the nodal values of a control volume centered about the face center. Greens theorem was employed to compute second order gradients at the face center in the control volume (diamond cell). This form of reconstruction retained second order gradients at face centers even when the gradients at cell centers degenerate to first order. The modified diffusion operator formulation was successfully verified to be second order accurate on all 2D unstructured meshes with triangular cells. The code verification results on the 2D alternate diagonal unstructured mesh with the old diffusion operator and the modified diffusion operator are shown in Figure 13.



**Figure 13: Order of accuracy results using  $L_2$  norm of the discretization error for Navier-Stokes equations on a 2D alternate diagonal unstructured mesh using a) old diffusion operator and b) modified diffusion operator**

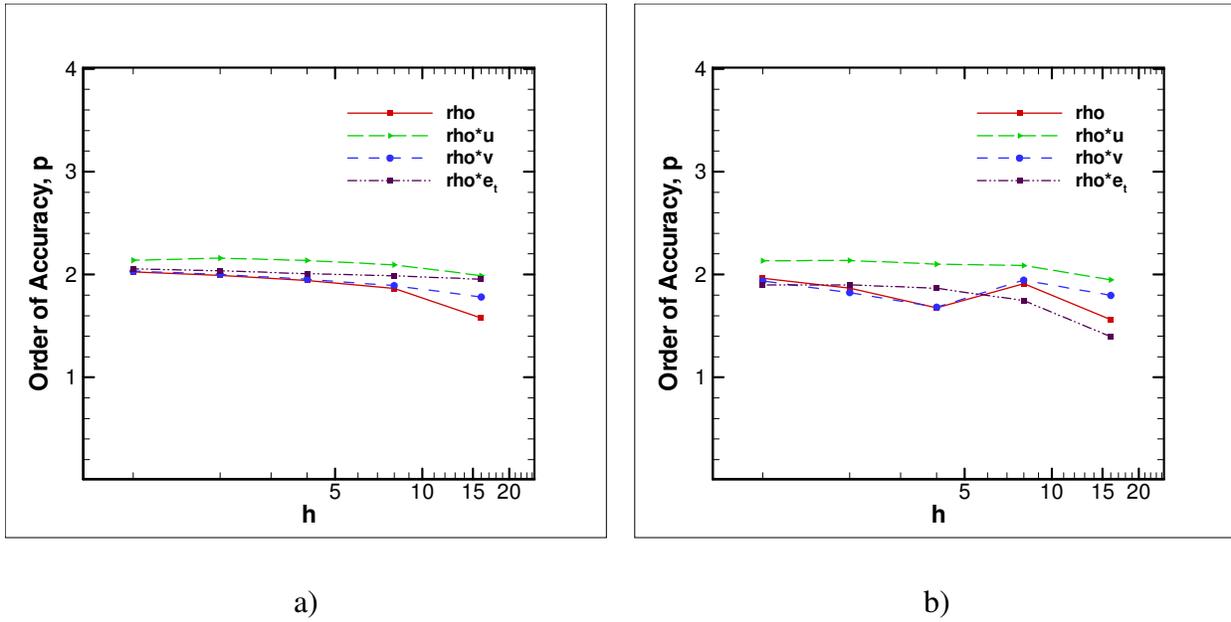
These two modifications in the formulation of the diffusion operator enabled the Navier-Stokes equations to be successfully verified as second order accurate on the 2D hybrid mesh and the 3D skewed hybrid mesh, thus covering the effect of all cell topologies which include some skewness, aspect ratio, curvature, and mesh stretching.

#### 4.2.4 Effect of Mesh Quality

The Navier-Stokes equations are also tested on the 2D highly skewed hybrid meshes (a)

b)

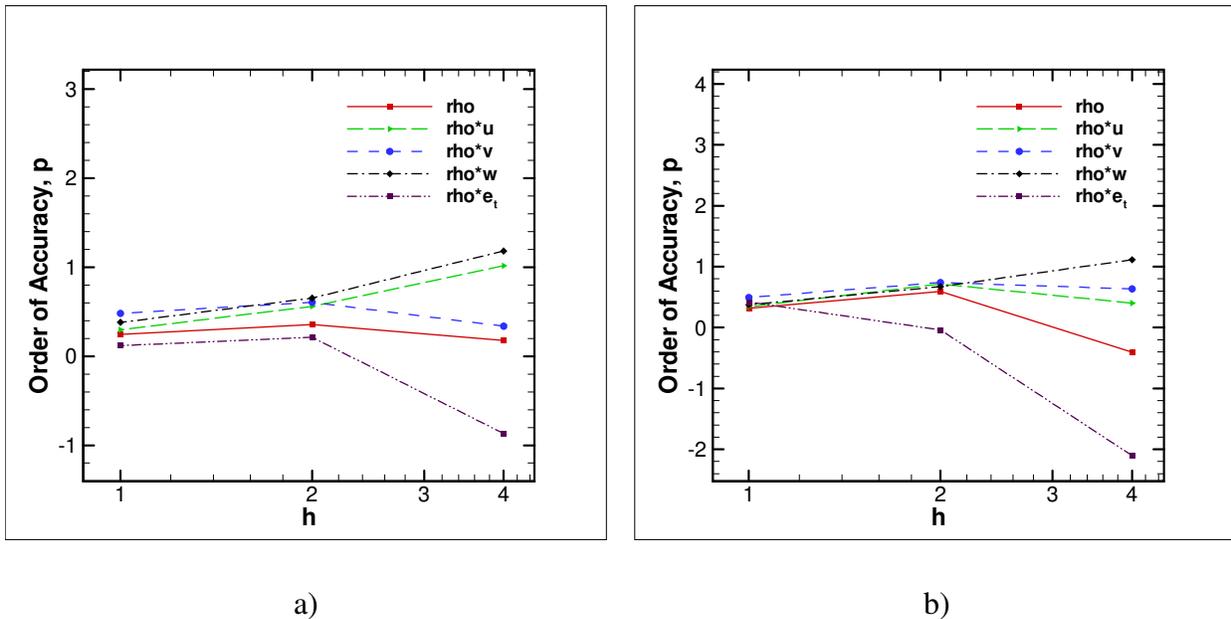
Figure 3b) to look at the effect of cell quality of quadrilateral cells and triangular cells on the discrete formulation of the governing equations. On this mesh, the code is successfully verified and the observed order of accuracy approaches two with mesh refinement. The observed order of accuracy results calculated using  $L_1$  and  $L_\infty$  norms of the discretization error for the Navier-Stokes equations on the 2D highly skewed hybrid mesh are shown in Figure 14. This shows that the finite volume code works fine on highly skewed quadrilateral and triangular cells in 2D.



**Figure 14: Order of accuracy results for Navier-Stokes equations on a 2D highly skewed hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

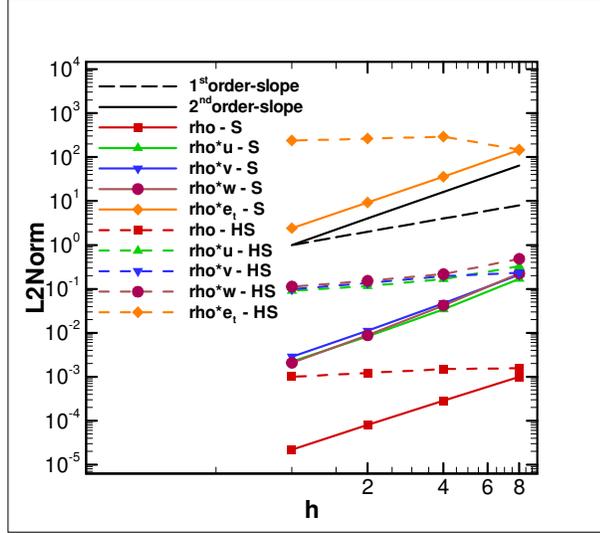
The Navier-Stokes equations are also tested on a 3D highly skewed hybrid mesh (Figure 6b) to look at the effect of cell quality of hexahedral, prismatic, and tetrahedral cells on the discrete formulation of the governing equations. The 3D highly skewed hybrid mesh is generated starting from a 3D skewed curvilinear mesh shown in Figure 4b and using the mesh generation code which generates the hybrid mesh (containing hexahedral, tetrahedral, and prismatic cells) from the structured mesh. The Navier-Stokes equations are successfully verified to be second order accurate on the 3D skewed curvilinear mesh with hexahedral cells shown in Figure 4b, but the verification test failed on the 3D highly skewed hybrid mesh. The observed order of accuracy results calculated using  $L_1$  and  $L_\infty$  norms of the discretization error for the Navier-Stokes

equations on the 3D highly skewed hybrid mesh are shown in Figure 15. From the plot, the observed order of accuracy seems to approach a value less than one with mesh refinement. The difference between the 3D skewed hybrid mesh (Figure 6a) and the 3D highly skewed hybrid mesh (Figure 6b) is only the quality of the cells in the mesh; otherwise both the meshes have the same mesh topology and connectivity. This indicates a problem in the discrete formulation of the governing equations when the cells have a comparatively lower quality. The cell quality is quantified with some parameters and quantification of cell quality of the 3D skewed hybrid mesh and the 3D highly skewed hybrid mesh is shown in Table B1 in Appendix B.



**Figure 15: Order of accuracy results for Navier-Stokes equations on a 3D highly skewed hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

The  $L_2$  norms of the discretization error for both the 3D skewed hybrid mesh and the 3D highly skewed hybrid mesh are compared and it is observed that the error is higher for the highly skewed hybrid meshes relative to the skewed hybrid meshes for the same number of cells and similar mesh structure. The comparison of the  $L_2$  norm of the discretization error is shown in Figure 16. In the plot, errors shown in the solid lines correspond to the 3D skewed hybrid mesh and the errors shown in dashed lines correspond to the 3D highly skewed hybrid mesh. This plot gives the effect of cell quality on the error in the solution. The error in the solution either decreases slowly or does not decrease with mesh refinement for lower quality meshes.



**Figure 16: Comparison of  $L_2$  norm discretization error on 3D skewed hybrid mesh and 3D highly skewed hybrid mesh**

To further study the discrete formulations of the inviscid and viscous terms in the governing equations, the Euler and Navier-Stokes equations are tested separately on meshes with a particular mesh topology, i.e., either highly skewed tetrahedral cells or highly skewed prismatic cells alone. The above analysis explains how MMS along with the order of accuracy test can be used to find sensitivities to mesh quality by testing on different meshes with different cell topologies and different cell quality. From the study, it is observed that the code is successfully verified to be second order accurate while testing Euler and Navier-Stokes equations on highly skewed hexahedral and prismatic cells but failed the order of accuracy test while testing Euler equations and Navier-Stokes equations on highly skewed tetrahedral cells and hence on the 3D highly skewed hybrid mesh. The Euler equations are only first order accurate and the Navier-Stokes equations are less than first order accurate on highly skewed tetrahedral cells which led to the same effect on 3D highly skewed hybrid mesh. The order of accuracy of the governing equations on the highly skewed cells in 3D is tabulated in Table 2.

The reason for failure of code verification on the 3D highly skewed hybrid mesh is because of the instability in the inviscid operator. Testing only the heat equation (similar to testing the diffusion operator) on the highly skewed 3D hybrid mesh produced second order accuracy. On the highly skewed mesh, a limiter is needed to achieve iterative convergence. Running the Euler equations with the limiter, the code converges to only first order because of a

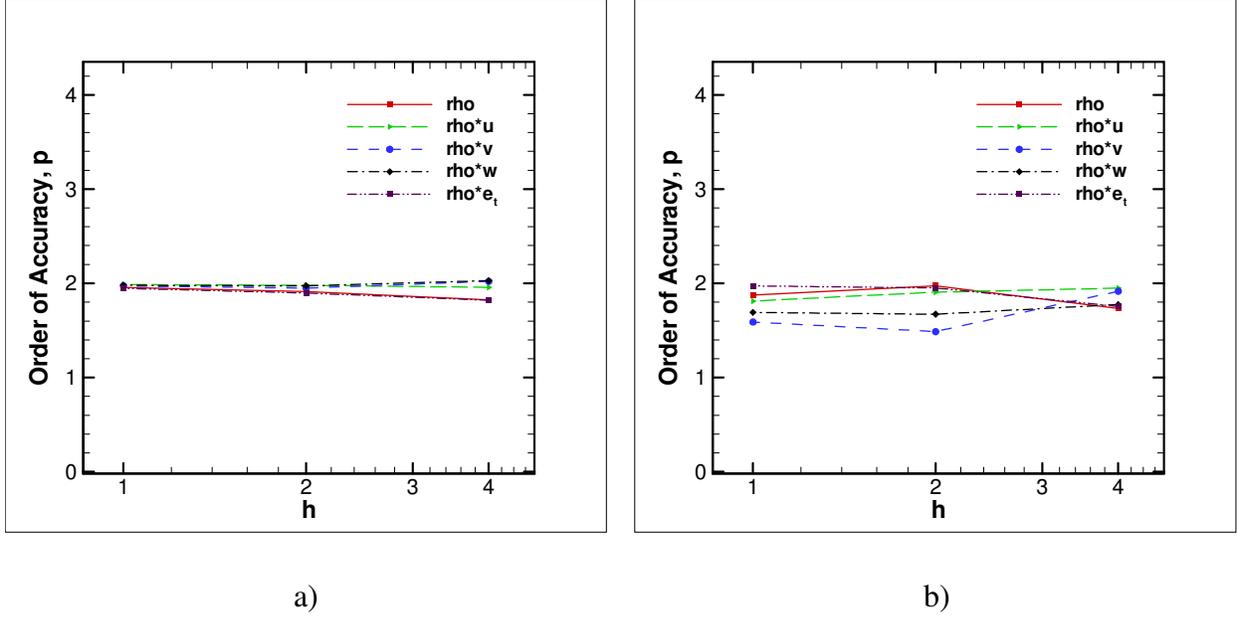
stability problem caused by the geometry of the gradient stencil at the boundary. Since there is instability in the inviscid operator, the limiter is forced to be active and it affects the diffusion operator which also incorporates the limiter. The diffusion operator incorporates the limiter to ensure that it does not introduce new solution extrema.

**Table 2. Order of accuracy of the governing equations on highly skewed cells in 3D**

	Euler Equations	Navier-Stokes Equations
Hexahedral Cells	2 <sup>nd</sup> Order	2 <sup>nd</sup> Order
Prismatic Cells	2 <sup>nd</sup> Order	2 <sup>nd</sup> Order
Tetrahedral Cells	1 <sup>st</sup> Order	Less than 1 <sup>st</sup> Order
Hybrid Cells	1 <sup>st</sup> Order	Less than 1 <sup>st</sup> Order

### 4.3 Verification of Transport Models

Other equations verified in the finite volume code are the equation of state, thermally perfect thermodynamic model and Sutherland’s law of viscosity. Verifying the Navier-Stokes equations in the Loci-CHEM code automatically verifies equation of state. For verifying the thermally perfect thermodynamic model and Sutherland’s law of viscosity, thermal conductivity and viscosity are defined as functions of temperature. Both the equations are tested in 3D skewed hybrid mesh and are successfully verified to be second order accurate with the observed order of accuracy approaching two with mesh refinement. The Manufactured Solution used for the verification of the transport equations is same as the Manufactured Solution used for the Euler and Navier-Stokes equations. The order of accuracy results for the Sutherland’s law of viscosity on a 3D skewed hybrid grid are shown in Figure 17.



**Figure 17: Order of accuracy results for Sutherland's law of viscosity on a 3D skewed hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.4 Verification of Boundary Conditions

In order to verify the implementation of a boundary condition in a code, the Manufactured Solution can be tailored to exactly satisfy a given boundary condition on a domain boundary. A general approach for tailoring Manufactured Solution to ensure that a given boundary condition is satisfied along a domain boundary was developed by Bond et al.<sup>20</sup> The approach is explained with a simple example in 2D.<sup>7</sup> The standard form of the Manufactured Solution for the 2D steady-state solution can be written as

$$\phi(x, y) = \phi_0 + \phi_1(x, y) \quad (4.16)$$

where

$$\phi_1(x, y) = \phi_x f_s \left( \frac{a_{\phi x} \pi x}{L} \right) + \phi_y f_s \left( \frac{a_{\phi y} \pi y}{L} \right) + \phi_{xy} f_s \left( \frac{a_{\phi xy} \pi xy}{L^2} \right) \quad (4.17)$$

A boundary in 2D can be represented by a general curve  $F(x, y) = C$ , where  $C$  is a constant. The new Manufactured Solution for verifying boundary conditions can be found by multiplying the  $\phi_1(x, y)$  term with the function  $[C - F(x, y)]^m$  as shown below:

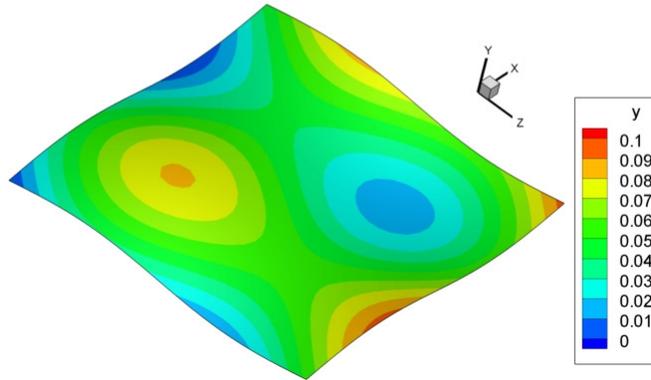
$$\phi_{BC}(x, y) = \phi_0 + \phi_1(x, y)[C - F(x, y)]^m. \quad (4.18)$$

This procedure will ensure that the Manufactured Solution is equal to a constant  $\phi_0$  satisfying a Dirichlet boundary condition  $\phi(x,y) = \phi_0$  along the specified boundary for  $m = 1$ . For  $m = 2$ , it ensures that the Manufactured Solution will satisfy both Dirichlet and Neumann (zero normal gradient) boundary conditions along the specified boundary.

To test a boundary condition option where the boundary is curved, a well-defined curved boundary for 2D meshes, i.e.,  $F(x,y) = C$  and a well defined curved surface for 3D meshes, i.e.,  $F(x,y,z) = C$  are considered as one side of a domain in 2D and 3D, respectively, and the mesh is built along with that boundary. The analytic definition of the curved boundaries considered for the boundary condition verification in both 2D and 3D are defined as given below.

$$\begin{aligned}
 F(x, y) &= y - x\text{Sin}(5\pi/180) - 0.05\text{Sin}(2\pi x) = 0 \\
 F(x, y, z) &= 2y - x\text{Sin}(5\pi/180) - z\text{Sin}(6\pi/180) - 0.06\text{Sin}(2\pi x) \\
 &\quad - 0.05\text{Sin}(2\pi z) = 0
 \end{aligned}
 \tag{4.19}$$

These boundaries are used both in the Manufactured Solution and the grid generation. The curved surface used for the verification of boundary conditions in 3D is shown in Figure 18. The boundary conditions are also tested when the boundary is a straight boundary (straight line in 2D and a flat surface in 3D) instead of a curved boundary.

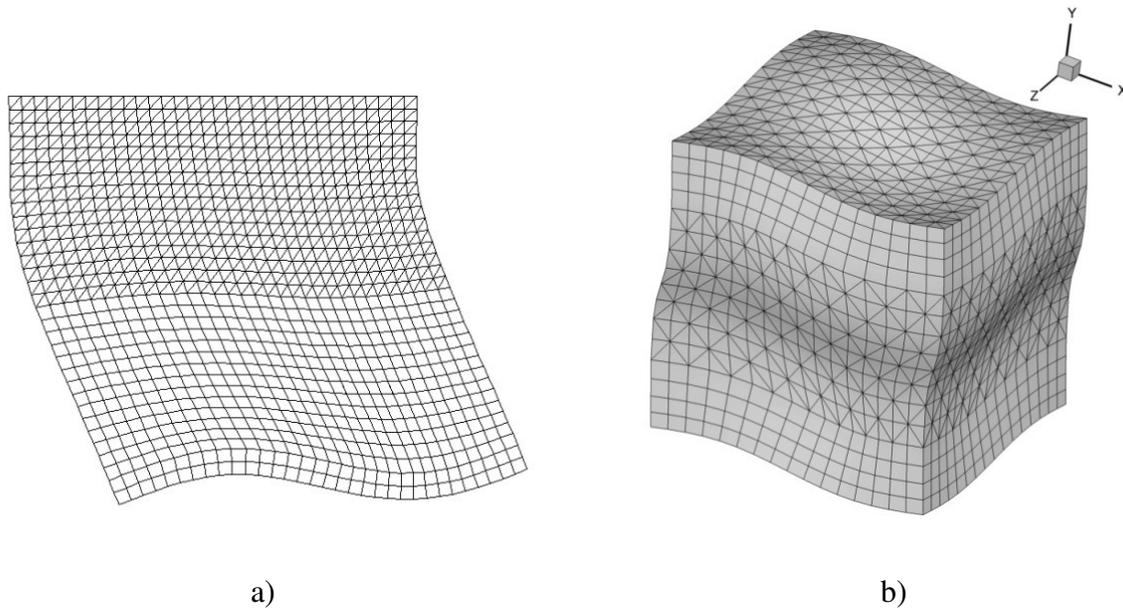


**Figure 18: Wavy surface used as a well-defined boundary for 3D meshes**

The different boundary condition options verified in the finite volume code include the no-slip adiabatic wall, no-slip isothermal wall, slip wall with the Euler equations and the Navier-Stokes equations, isentropic inflow, outflow, extrapolation, and farfield.

#### 4.4.1 Mesh Topologies

The meshes used for verification of boundary conditions are different and a separate set of meshes are generated for boundary condition verification in both 2D and 3D. The 2D hybrid mesh and the 3D hybrid mesh with a curved boundary used for the boundary condition verification are shown in Figure 19. In the figure, for both meshes the bottom boundaries are the well defined curved boundaries which are tested for different boundary condition options. While generating these meshes for boundary condition verification, the mesh is generated to be normal to the tested boundary.



**Figure 19: Meshes used for boundary condition verification a) 2D hybrid mesh and b) 3D hybrid mesh**

#### 4.4.2 No-slip Wall

The no-slip wall boundary is verified as an adiabatic boundary and an isothermal boundary. In both the cases, the no-slip wall boundary is tested as a straight wall boundary and also as a curved wall boundary.

##### 4.4.2.1 No-slip Adiabatic Wall

By testing the no-slip wall as an adiabatic boundary, a Neumann boundary condition for temperature ( $dT/dn = 0$ ) is verified along with the no-slip condition ( $\vec{V} = 0$ ) on a particular boundary. The no-slip wall is tested as an adiabatic boundary on meshes with both straight and

curved boundaries in 2D and 3D. The 2D Manufactured Solution used for the verification of no-slip wall as an adiabatic boundary is of the form

$$\begin{aligned}
\rho &= \rho_0 + (F(x, y))^2 (\phi_\rho(x, y)) \\
u &= (F(x, y))(u_0 + \phi_u(x, y)) \\
v &= (F(x, y))(v_0 + \phi_v(x, y)) \\
p &= p_0 + (F(x, y))^2 (\phi_p(x, y))
\end{aligned} \tag{4.20}$$

and the function  $F(x, y)$  is defined for a straight boundary and a curved boundary as

$$\begin{aligned}
F(x, y) &= y = 0 \\
F(x, y) &= y - x\sin(5\pi/180) - 0.05\sin(2\pi x) = 0
\end{aligned} \tag{4.21}$$

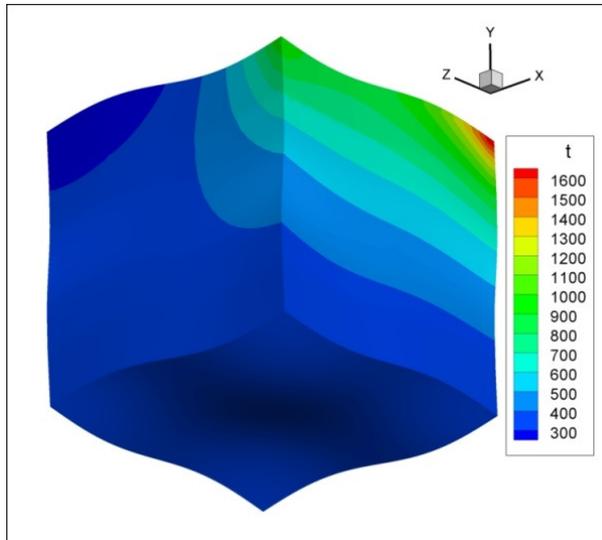
The 3D Manufactured Solution used for the verification of no-slip wall as an adiabatic boundary is of the form

$$\begin{aligned}
\rho &= \rho_0 + (F(x, y, z))^2 (\phi_\rho(x, y, z)) \\
u &= (F(x, y, z))(u_0 + \phi_u(x, y, z)) \\
v &= (F(x, y, z))(v_0 + \phi_v(x, y, z)) \\
w &= (F(x, y, z))(w_0 + \phi_w(x, y, z)) \\
p &= p_0 + (F(x, y, z))^2 (\phi_p(x, y, z))
\end{aligned} \tag{4.22}$$

and the function  $F(x, y, z)$  is defined for a straight boundary and a curved boundary as

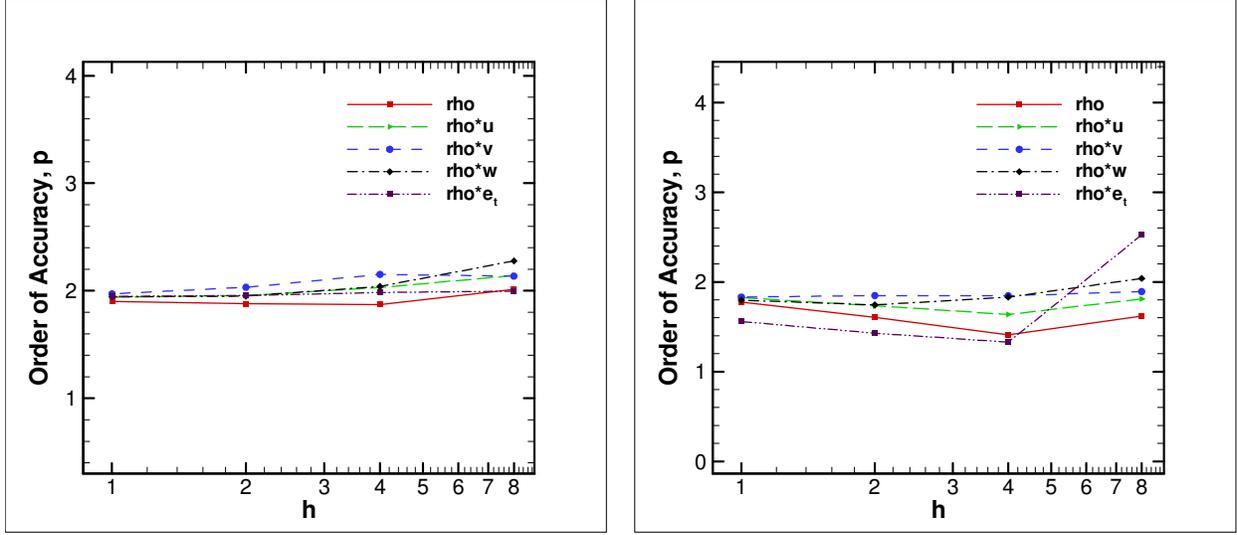
$$\begin{aligned}
F(x, y, z) &= y = 0 \\
F(x, y, z) &= 2y - x\sin(5\pi/180) - z\sin(6\pi/180) - 0.06\sin(2\pi x) \\
&\quad - 0.05\sin(2\pi z) = 0
\end{aligned} \tag{4.23}$$

The constants used in the 2D and 3D Manufactured Solutions are presented in Appendix C in Table C1 and Table C2, respectively. By selecting the Manufactured Solution as shown above, all the velocity components become zero on the  $F = 0$  boundary and the normal derivatives of density and pressure become zero at  $F = 0$  boundary which makes the normal derivative of temperature also zero at that boundary satisfying both the no-slip condition and the adiabatic condition. The temperature contours in a 3D domain when the bottom boundary is defined as an adiabatic no-slip wall are shown in Figure 20.



**Figure 20: Temperature contours when the bottom boundary is defined as a no-slip adiabatic wall**

The no-slip wall is tested as an adiabatic boundary on the 2D hybrid mesh and 3D hybrid mesh with curved boundaries and the observed order of accuracy calculated from the numerical solutions approached two with mesh refinement for the meshes considered. The results are shown here for the 3D hybrid mesh with curved boundaries and the order of accuracy results using  $L_1$  and  $L_\infty$  norms of the discretization error for the adiabatic no-slip wall boundary on a 3D hybrid mesh are shown in Figure 21.



a)

b)

**Figure 21: Order of accuracy results for adiabatic no-slip wall boundary on a 3D hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.4.2.2 No-slip Isothermal Wall

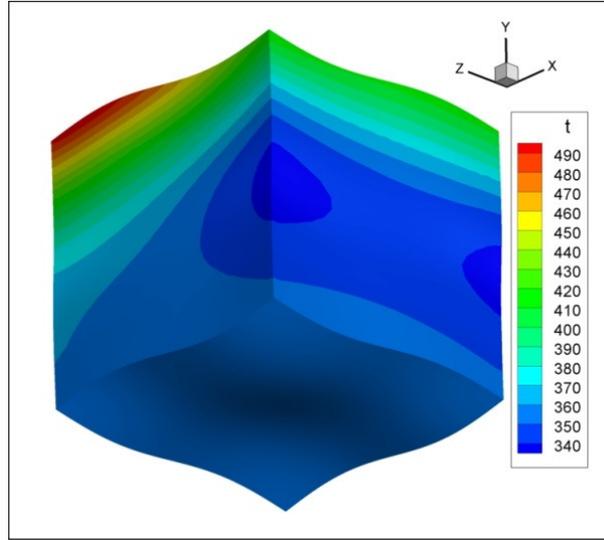
By testing the no-slip wall as an isothermal boundary, a Dirichlet boundary condition for temperature ( $T = constant$ ) is verified along with the no-slip condition ( $\vec{V} = 0$ ) on a particular boundary. The no-slip wall is tested as an isothermal boundary on meshes with both straight and curved boundaries in 2D and 3D. The 2D Manufactured Solution used for the verification of no-slip wall as an isothermal boundary is of the form

$$\begin{aligned}
 \rho &= \frac{p_0 + (F(x, y))^2 (\phi_p(x, y))}{Rt (T_0 + (F(x, y))(\phi_T(x, y)))} \\
 u &= (F(x, y))(u_0 + \phi_u(x, y)) \\
 v &= (F(x, y))(v_0 + \phi_v(x, y)) \\
 p &= p_0 + (F(x, y))^2 (\phi_p(x, y))
 \end{aligned} \tag{4.24}$$

The 3D Manufactured Solution used for the verification of no-slip wall as an adiabatic boundary is of the form

$$\begin{aligned}
\rho &= \frac{p_0 + (F(x, y, z))^2 (\phi_p(x, y, z))}{Rt (T_0 + (F(x, y, z))(\phi_T(x, y, z)))} \\
u &= (F(x, y, z))(u_0 + \phi_u(x, y, z)) \\
v &= (F(x, y, z))(v_0 + \phi_v(x, y, z)) \\
w &= (F(x, y, z))(w_0 + \phi_w(x, y, z)) \\
p &= p_0 + (F(x, y, z))^2 (\phi_p(x, y, z))
\end{aligned} \tag{4.25}$$

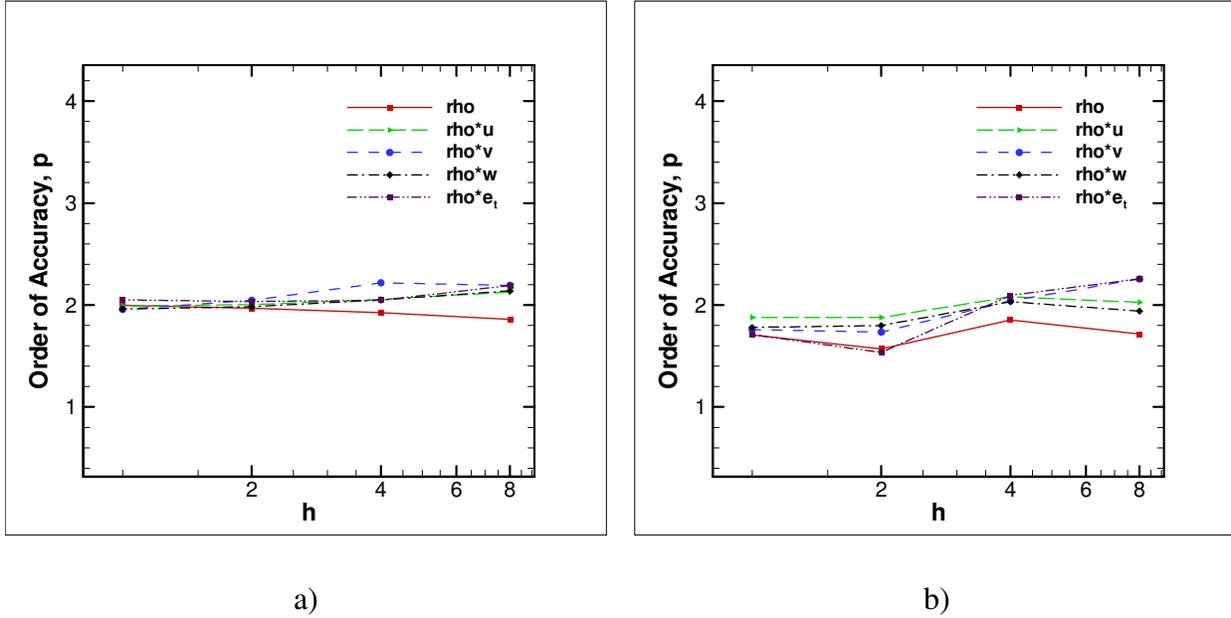
The constants used in the 2D and 3D Manufactured Solutions are presented in Appendix C in Table C1 and Table C2, respectively. By selecting the Manufactured Solution as shown above, all the velocity components become zero on the  $F = 0$  boundary and the temperature becomes constant at that boundary satisfying both the no-slip condition and the isothermal condition. Here, the Manufactured Solution for temperature is not used, but a constant temperature boundary needs to be satisfied. Hence, the Manufactured Solution for density is selected in terms of pressure and temperature which satisfies the needed conditions for isothermal boundary. The temperature contours in a 3D domain when the bottom boundary is defined as an isothermal no-slip wall are shown in Figure 22.



**Figure 22: Temperature contours when the bottom boundary is defined as a no-slip isothermal wall**

The no-slip wall is tested as an isothermal boundary on the 2D hybrid mesh and 3D hybrid mesh with curved boundaries and the observed order of accuracy calculated from the

numerical solutions approached two with mesh refinement for the meshes considered. The results are shown here for the 3D hybrid mesh with curved boundaries and the order of accuracy results using  $L_1$  and  $L_\infty$  norms of the discretization error for the isothermal no-slip wall boundary on a 3D hybrid mesh are shown in Figure 23.



**Figure 23: Order of accuracy results for isothermal no-slip wall boundary on a 3D hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.4.3 Slip Wall

Testing the slip wall is verifying the slip condition  $V_n = 0$  on a particular boundary where  $V_n$  is the velocity component normal to the surface. Also, on a slip wall boundary, the viscous terms need to be zero. This boundary condition is defined with impermeable or reflecting options available in the Loci-CHEM code. The slip wall boundary condition is tested with both the Euler and the Navier-Stokes equations. The 2D Manufactured Solution used for the verification of slip wall boundary is of the form

$$\begin{aligned}
 \rho &= \rho_0 + (F(x, y))^2 (\phi_\rho(x, y)) \\
 u &= u_0 + (F(x, y))^2 (\phi_u(x, y)) \\
 v &= (F(x, y))^2 (v_0 + \phi_v(x, y))
 \end{aligned} \tag{4.26}$$

$$p = p_0 + (F(x, y))^2 (\phi_p(x, y))$$

The 3D Manufactured Solution used for the verification of slip wall boundary is of the form

$$\begin{aligned} \rho &= \rho_0 + (F(x, y, z))^2 (\phi_\rho(x, y, z)) \\ u &= u_0 + (F(x, y, z))^2 (\phi_u(x, y, z)) \\ v &= (F(x, y, z))^2 (v_0 + \phi_v(x, y, z)) \\ w &= w_0 + (F(x, y, z))^2 (\phi_w(x, y, z)) \\ p &= p_0 + (F(x, y, z))^2 (\phi_p(x, y, z)) \end{aligned} \quad (4.27)$$

The constants used in the 2D and 3D Manufactured Solutions are presented in Appendix C in Table C1 and Table C2, respectively. The above Manufactured Solutions for slip wall boundary condition verification will work only for testing on straight boundaries since the normal velocity component of zero cannot be achieved for a curved boundary with this Manufactured Solution. These Manufactured Solutions are used while testing slip wall along with the Navier-Stokes equations. For the curved boundaries, selecting the Manufactured Solutions while testing slip wall along with the Navier-Stokes equations is challenging and testing of slip wall with the Navier-Stokes equations on curved boundaries is not performed in this work. Hence, the slip wall with the Navier-Stokes equations is tested on 2D rectangular mesh with straight boundary and 3D hybrid mesh with straight boundary.

In the case of verification of slip wall with the Euler equations, selecting the Manufactured Solution for testing on curved boundaries is relatively easier since there is no need to deal with the viscous terms on the curved boundary. The velocity component normal to the slip wall boundary is zero and can be derived using the expression

$$\begin{aligned} \nabla F \cdot V &= 0 \\ \text{In 2D, } \quad \frac{\partial F}{\partial x} u + \frac{\partial F}{\partial y} v &= 0 \\ \text{In 3D, } \quad \frac{\partial F}{\partial x} u + \frac{\partial F}{\partial y} v + \frac{\partial F}{\partial z} w &= 0 \end{aligned} \quad (4.28)$$

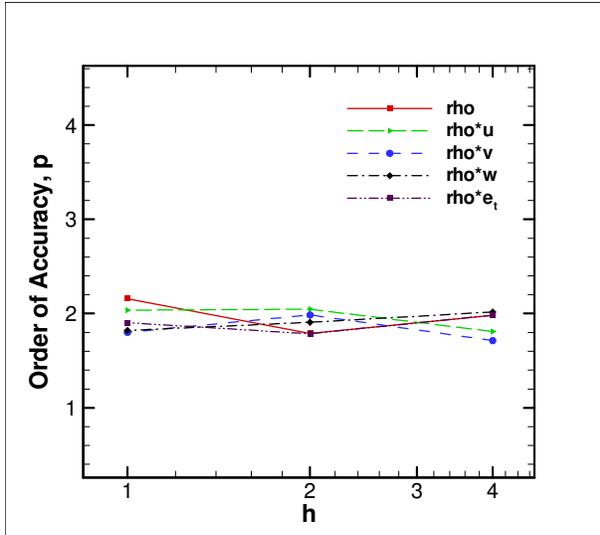
The 2D Manufactured Solution for verification of slip wall when using curved boundaries and Euler equations is of the form

$$\begin{aligned}
\rho &= \rho_0 + (F(x, y))^2 (\phi_\rho(x, y)) \\
u &= u_0 + \phi_u(x, y) \\
v &= -\frac{\partial(F(x, y))}{\partial x} (u_0 + \phi_u(x, y)) \\
p &= p_0 + (F(x, y))^2 (\phi_p(x, y))
\end{aligned} \tag{4.29}$$

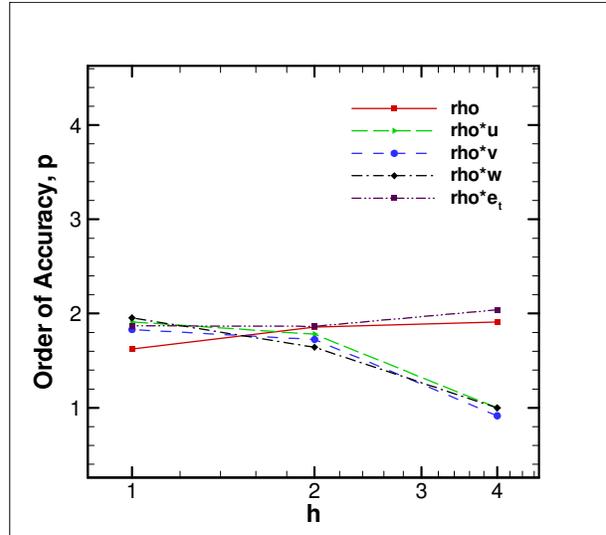
The 3D Manufactured Solution used for the verification of slip wall when using curved boundaries and Euler equations is of the form

$$\begin{aligned}
\rho &= \rho_0 + (F(x, y, z))^2 (\phi_\rho(x, y, z)) \\
u &= u_0 + \phi_u(x, y, z) \\
v &= -\frac{\partial(F(x, y, z))}{\partial x} (u_0 + \phi_u(x, y, z)) \\
&\quad - \frac{\partial(F(x, y, z))}{\partial z} (w_0 + \phi_w(x, y, z)) \\
w &= w_0 + \phi_w(x, y, z) \\
p &= p_0 + (F(x, y, z))^2 (\phi_p(x, y, z))
\end{aligned} \tag{4.30}$$

The constants used in the 2D and 3D Manufactured Solutions are presented in Appendix C in Table C1 and Table C2, respectively. The slip wall with the Euler equations is tested on 2D hybrid mesh and 3D skewed hybrid mesh with curved boundaries also and the observed order of accuracy calculated from the numerical solutions approached two with mesh refinement for the meshes considered. The results are shown here for the 3D skewed hybrid mesh with curved boundaries and the order of accuracy results using  $L_1$  and  $L_\infty$  norms of the discretization error for the slip wall boundary with the Euler equations on a 3D skewed hybrid mesh are shown in Figure 24. The slip wall with the Navier-Stokes equations is tested on 2D rectangular mesh and 3D hybrid mesh with straight boundaries also and the observed order of accuracy calculated from the numerical solutions approached two with mesh refinement for the meshes considered. The results shown here are for the 3D hybrid mesh with straight boundaries and the order of accuracy results using  $L_1$  and  $L_\infty$  norms of the discretization error for the slip wall boundary with the Navier-Stokes are shown in Figure 25.

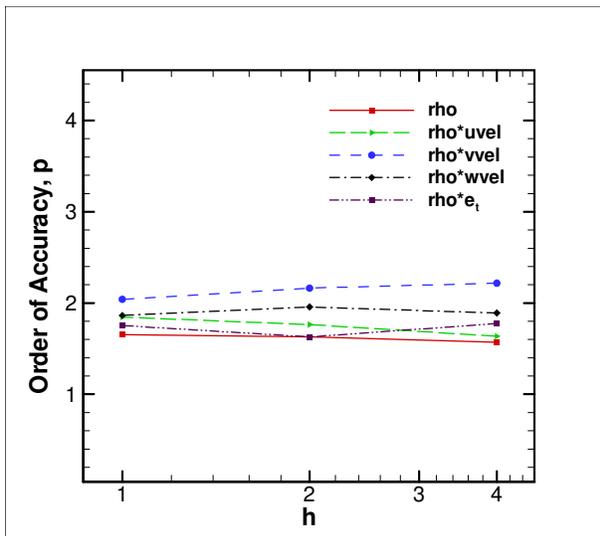


a)

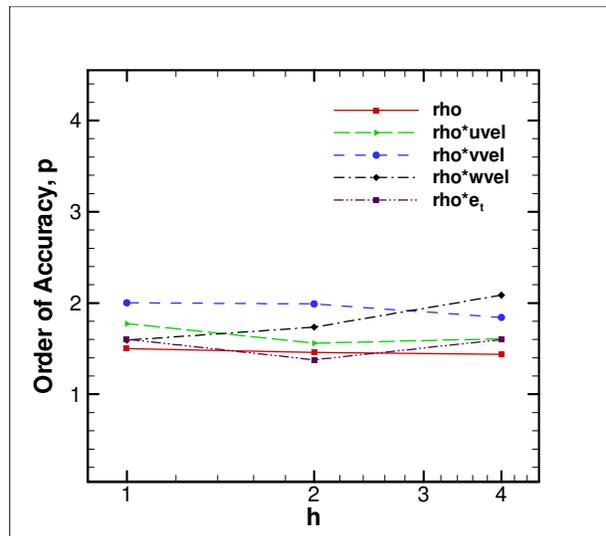


b)

**Figure 24: Order of accuracy results for slip wall boundary with the Euler equations on a 3D skewed hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**



a)



b)

**Figure 25: Order of accuracy results for slip wall boundary with the Navier-Stokes equations on a 3D hybrid mesh with straight boundaries using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.4.4 Isentropic Inflow

This boundary condition is an inflow boundary that requires the total temperature and total pressure to be constant on the inflow boundary. This boundary condition is applicable to only subsonic inflow conditions in the code. The isentropic inflow boundary is tested on 2D rectangular mesh and 3D hybrid mesh with straight boundaries. The governing equations used while testing isentropic boundary condition are the Euler equations. The 2D Manufactured Solution used for the verification of isentropic inflow boundary condition is of the form

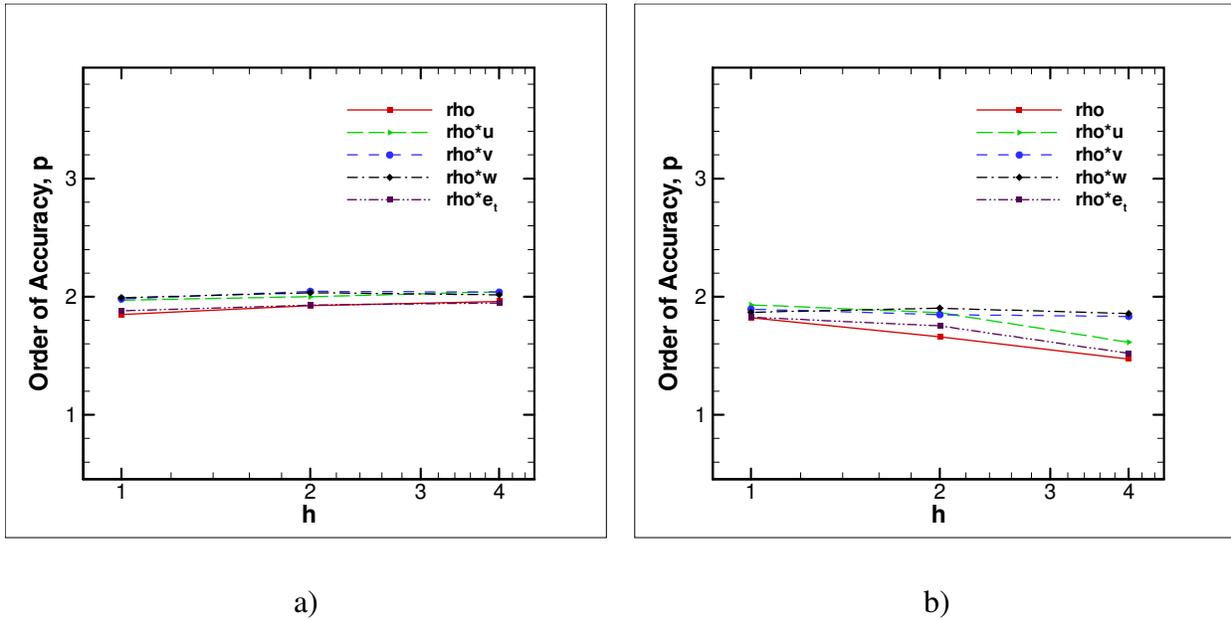
$$\begin{aligned}
 \rho &= \rho_0 + (F(x, y))^2 (\phi_\rho(x, y)) \\
 u &= (F(x, y))^2 (u_0 + \phi_u(x, y)) \\
 v &= v_0 + (F(x, y))^2 (\phi_v(x, y)) \\
 p &= p_0 + (F(x, y))^2 (\phi_p(x, y))
 \end{aligned} \tag{4.31}$$

The 3D Manufactured Solution used for the verification of isentropic inflow boundary condition is of the form

$$\begin{aligned}
 \rho &= \rho_0 + (F(x, y, z))^2 (\phi_\rho(x, y, z)) \\
 u &= (F(x, y, z))^2 (u_0 + \phi_u(x, y, z)) \\
 v &= v_0 + (F(x, y, z))^2 (\phi_v(x, y, z)) \\
 w &= (F(x, y, z))^2 (w_0 + \phi_w(x, y, z)) \\
 p &= p_0 + (F(x, y, z))^2 (\phi_p(x, y, z))
 \end{aligned} \tag{4.32}$$

The constants used in the 2D and 3D Manufactured Solutions are presented in Appendix C in Table C1 and Table C2, respectively. The above Manufactured Solutions for isentropic inflow boundary condition verification are developed for testing on straight boundaries. For curved boundaries, the Manufactured Solutions need to be selected very carefully and verification of isentropic inflow on a curved boundary is not performed in this work. By selecting the Manufactured Solution as above, all the variables are constant on the  $F = 0$  inflow boundary and also the derivatives of the variables are zero normal to the boundary. The u-velocity and w-velocity are zero on the  $F = 0$  boundary and with only v-velocity defined on that boundary, the velocity is normal to the inflow boundary. With density, pressure, and the velocity components constant on the boundary, the stagnation pressure and stagnation temperature are

also constant which satisfies the isentropic boundary condition on the inflow boundary. The isentropic inflow boundary condition is successfully verified on a 2D rectangular mesh with straight boundaries and 3D hybrid mesh with straight boundaries and the observed order of accuracy calculated from the numerical solutions approached two with mesh refinement for the meshes considered. The results are shown here for the 3D hybrid mesh with straight boundaries and the order of accuracy results using  $L_1$  and  $L_\infty$  norms of the discretization error for the isentropic inflow boundary on a 3D hybrid mesh are shown in Figure 26.



**Figure 26: Order of accuracy results for isentropic inflow boundary on a 3D hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.4.5 Outflow

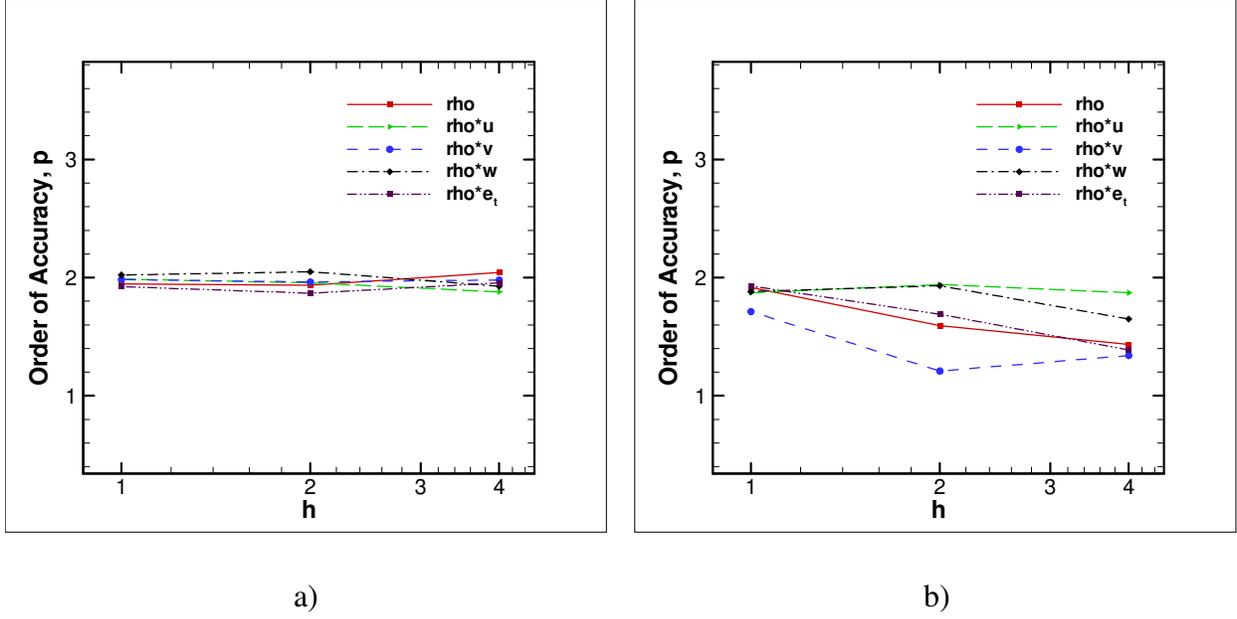
This boundary condition is a characteristic based outflow condition. The outflow boundary condition is tested as a subsonic boundary. For this outflow boundary condition, a static pressure is imposed at the boundary. The outflow boundary is tested on 2D rectangular mesh with straight boundary and 3D hybrid mesh with straight boundary. The governing equations used while testing outflow boundary condition are the Euler equations. The 2D Manufactured Solution used for the verification of the outflow boundary condition as a subsonic outflow is of the form

$$\begin{aligned}
\rho &= \rho_0 + (F(x, y))^2 (\phi_\rho(x, y)) \\
u &= (F(x, y))^2 (-u_0 + \phi_u(x, y)) \\
v &= -v_0 + (F(x, y))^2 (\phi_v(x, y)) \\
p &= p_0 + (F(x, y))^2 (\phi_p(x, y))
\end{aligned} \tag{4.33}$$

The 3D Manufactured Solution used for the verification of outflow boundary condition as a subsonic outflow is of the form

$$\begin{aligned}
\rho &= \rho_0 + (F(x, y, z))^2 (\phi_\rho(x, y, z)) \\
u &= (F(x, y, z))^2 (-u_0 + \phi_u(x, y, z)) \\
v &= -v_0 + (F(x, y, z))^2 (\phi_v(x, y, z)) \\
w &= (F(x, y, z))^2 (-w_0 + \phi_w(x, y, z)) \\
p &= p_0 + (F(x, y, z))^2 (\phi_p(x, y, z))
\end{aligned} \tag{4.34}$$

The constants used in the 2D and 3D Manufactured Solutions are presented in Appendix C in Table C1 and Table C2, respectively. The above Manufactured Solutions for outflow boundary condition verification work for testing on straight boundaries. For the curved boundaries, the Manufactured Solutions need to be selected very carefully and verification of outflow on a curved boundary is not performed here. By selecting the Manufactured Solution as above, the u- and w-velocities are zero on the  $y = 0$  boundary and only a negative v-velocity is defined on this boundary which ensures that the flow is going out and is normal to that boundary. The outflow boundary condition is tested as a subsonic outflow on a 2D rectangular mesh with straight boundaries and 3D hybrid mesh with straight boundaries and the observed order of accuracy calculated from the numerical solutions approached two with mesh refinement for the meshes considered. The results are shown for the 3D hybrid mesh with straight boundaries and the order of accuracy results using  $L_1$  and  $L_\infty$  norms of the discretization error for the outflow boundary tested as a subsonic boundary on a 3D hybrid mesh are shown in Figure 27.



**Figure 27: Order of accuracy results for subsonic outflow boundary on a 3D hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.4.6 Extrapolation

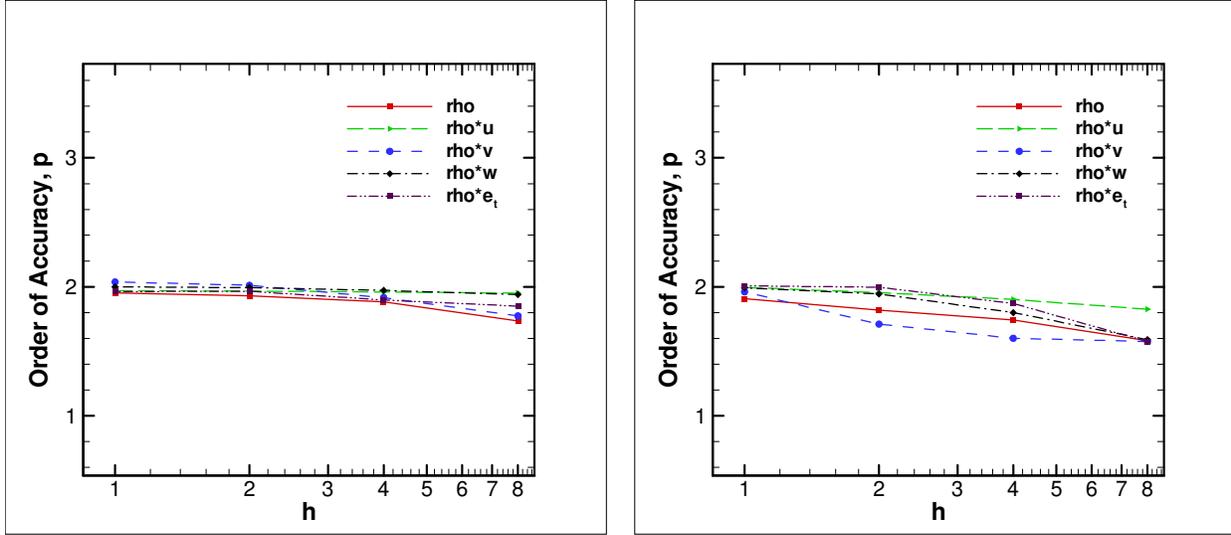
The extrapolation-based boundary condition is useful for supersonic outflow conditions. The outflow boundary condition is tested with supersonic flow conditions. The extrapolation boundary condition is tested on a 2D rectangular mesh and a 3D hybrid mesh with straight boundaries. The governing equations used while testing outflow boundary condition are the Euler equations. The 2D Manufactured Solution used for the verification of the extrapolation boundary condition with supersonic flow is of the form

$$\begin{aligned}
 \rho &= 1 + (F(x, y))^2 (\phi_\rho(x, y)) \\
 u &= (F(x, y))^2 (-700 + \phi_u(x, y)) \\
 v &= -900 + (F(x, y))^2 (\phi_v(x, y)) \\
 p &= 100000 + (F(x, y))^2 (\phi_p(x, y))
 \end{aligned} \tag{4.35}$$

The 3D Manufactured Solution used for the verification of extrapolation boundary condition with supersonic flow is of the form

$$\begin{aligned}
\rho &= 1 + (F(x, y, z))^2 (\phi_\rho(x, y, z)) \\
u &= (F(x, y, z))^2 (-700 + \phi_u(x, y, z)) \\
v &= -900 + (F(x, y, z))^2 (\phi_v(x, y, z)) \\
w &= (F(x, y, z))^2 (-800 + \phi_w(x, y, z)) \\
p &= 100000 + (F(x, y, z))^2 (\phi_p(x, y, z))
\end{aligned} \tag{4.36}$$

The constants used in the 2D and 3D Manufactured Solutions are presented in Appendix C in Table C1 and Table C2, respectively. The above Manufactured Solutions for extrapolation boundary condition verification are only for testing on straight boundaries. For the curved boundaries, the Manufactured Solutions need to be handled carefully and verification of extrapolation boundary condition on a curved boundary is not performed here. By selecting the Manufactured Solution as above, the  $u$ - and  $w$ -velocities are zero on the  $y = 0$  boundary and only a negative  $v$ -velocity is defined on the  $y = 0$  boundary which ensures that the flow is going and the flow is normal to that boundary. The Manufactured Solution also ensures that the derivatives of all the variables at the boundary are zero. The extrapolation boundary condition is tested on a 2D rectangular mesh and a 3D hybrid mesh with straight boundaries and the observed order of accuracy calculated from the numerical solutions approached two with mesh refinement for the meshes considered. The results are shown for the 3D hybrid mesh with straight boundaries and the order of accuracy results using  $L_1$  and  $L_\infty$  norms of the discretization error for the extrapolation boundary with supersonic flow on a 3D hybrid mesh are shown in Figure 28.



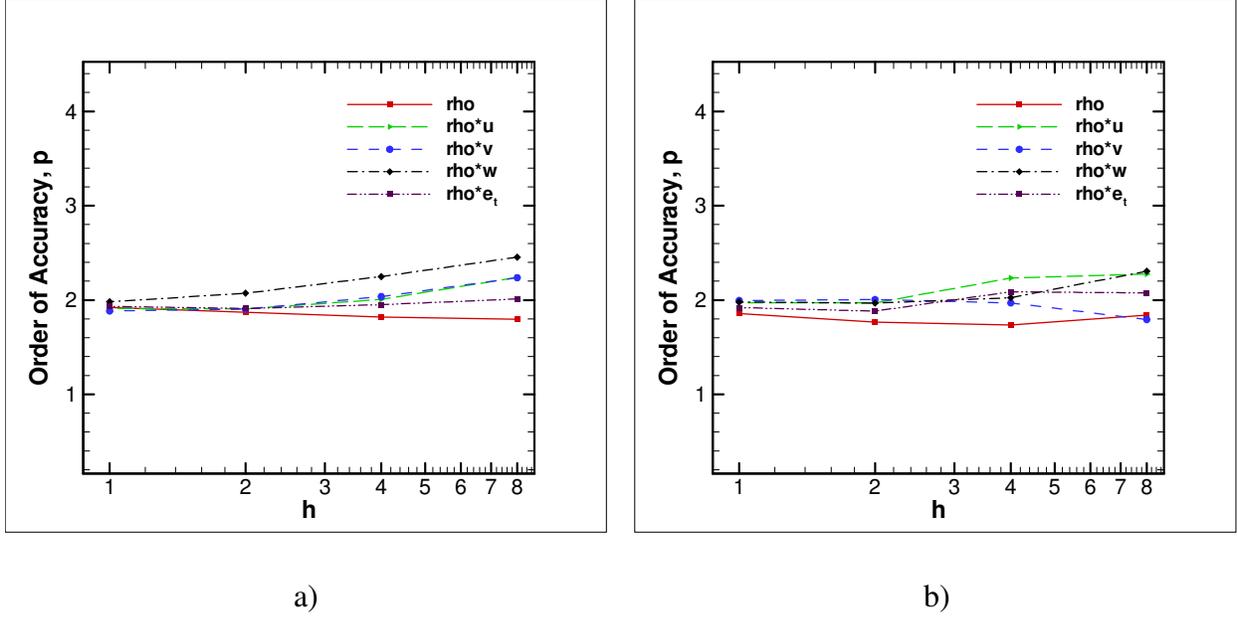
a)

b)

**Figure 28: Order of accuracy results for extrapolate boundary on a 3D hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.4.7 Farfield

The farfield boundary condition is an inflow-outflow characteristic based boundary condition. It is mostly suitable for a farfield condition used in external flows. It is suitable for both subsonic and supersonic flow conditions. Here, the farfield boundary condition is tested with subsonic flow conditions on 2D hybrid and 3D skewed hybrid meshes with curved boundaries. While testing the farfield boundary condition, the Navier-Stokes equations are used as the governing equations. The same Manufactured Solution used for testing the baseline governing equations, i.e., the Euler and the Navier-Stokes equations are used for testing the farfield boundary condition. The farfield boundary condition is successfully verified on the 2D and 3D hybrid meshes and the observed order of accuracy calculated from the numerical solutions approached two with mesh refinement for the meshes considered. The results are shown for the 3D skewed hybrid mesh and the order of accuracy results using  $L_1$  and  $L_\infty$  norms of the discretization error for the farfield boundary condition on a 3D skewed hybrid mesh are shown in Figure 29.



**Figure 29: Order of accuracy results for farfield boundary on a 3D skewed hybrid mesh using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.5 Verification of Turbulence Models

Verification of turbulence models provides additional challenges for MMS for different reasons.<sup>29</sup> One of the reasons is that the turbulence models often employ min or max functions to switch from one behavior to another, thus causing the source terms to no longer be continuously differentiable. The turbulence models tested in the finite volume code are the basic  $k-\omega$  turbulence model and the  $k-\varepsilon$  turbulence model which are part of the baseline version of the Menter's  $k-\omega$  model<sup>27</sup> and the Menter's Shear Stress Transport  $k-\omega$  model.<sup>27</sup> In both models, the  $k-\omega$  turbulence model gets activated in the boundary layer region and the  $k-\varepsilon$  turbulence model gets activated away from the wall boundaries in the free shear layers. The general form of the turbulent kinetic energy equation and the turbulent frequency equation for compressible flow are:

$$\begin{aligned}
 & \frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u k)}{\partial x} + \frac{\partial(\rho v k)}{\partial y} + \frac{\partial(\rho w k)}{\partial z} - P + \beta^* \rho \omega k \\
 & - \frac{\partial}{\partial x} \left[ (\mu + \sigma_k \mu_T) \frac{\partial k}{\partial x} \right] - \frac{\partial}{\partial y} \left[ (\mu + \sigma_k \mu_T) \frac{\partial k}{\partial y} \right] \\
 & - \frac{\partial}{\partial z} \left[ (\mu + \sigma_k \mu_T) \frac{\partial k}{\partial z} \right] = 0
 \end{aligned} \tag{4.37}$$

$$\begin{aligned}
& \frac{\partial(\rho\omega)}{\partial t} + \frac{\partial(\rho u\omega)}{\partial x} + \frac{\partial(\rho v\omega)}{\partial y} + \frac{\partial(\rho w\omega)}{\partial z} - \frac{\gamma\rho}{\mu_T}P + \beta^*\rho\omega^2 \\
& - \frac{\partial}{\partial x} \left[ (\mu + \sigma_k\mu_T) \frac{\partial\omega}{\partial x} \right] - \frac{\partial}{\partial y} \left[ (\mu + \sigma_k\mu_T) \frac{\partial\omega}{\partial y} \right] \\
& - \frac{\partial}{\partial z} \left[ (\mu + \sigma_k\mu_T) \frac{\partial\omega}{\partial z} \right] \\
& - 2(1-F)\rho\sigma_{\omega 2} \frac{1}{\omega} \left[ \frac{\partial k}{\partial x} \frac{\partial\omega}{\partial x} + \frac{\partial k}{\partial y} \frac{\partial\omega}{\partial y} + \frac{\partial k}{\partial z} \frac{\partial\omega}{\partial z} \right] = 0
\end{aligned} \tag{4.38}$$

The turbulence production terms is given by

$$\begin{aligned}
P = & \tau'_{xx} \frac{\partial u}{\partial x} + \tau'_{yy} \frac{\partial v}{\partial y} + \tau'_{zz} \frac{\partial w}{\partial z} + \tau'_{xy} \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) + \tau'_{yz} \left( \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) \\
& + \tau'_{zx} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)
\end{aligned} \tag{4.39}$$

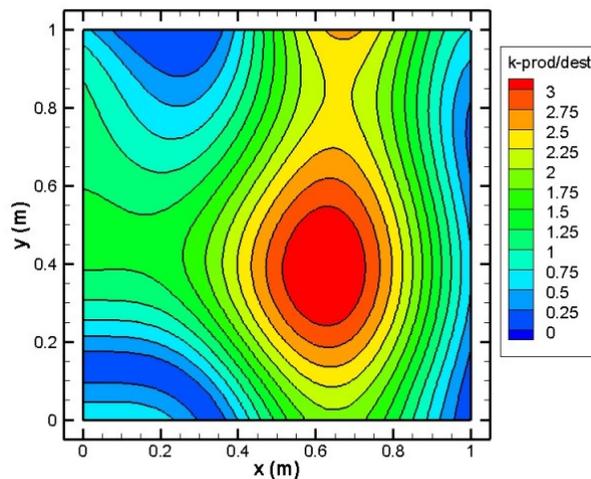
where the full compressible stress tensor is used:

$$\begin{aligned}
\tau'_{xx} &= \frac{2}{3}\mu_T \left( 2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} - \rho k \right), \\
\tau'_{yy} &= \frac{2}{3}\mu_T \left( 2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} - \rho k \right), \\
\tau'_{zz} &= \frac{2}{3}\mu_T \left( 2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \rho k \right) \\
\tau'_{xy} &= \mu_T \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \quad \tau'_{yz} = \mu_T \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right), \quad \tau'_{xz} = \mu_T \left( \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right)
\end{aligned} \tag{4.40}$$

The blending function  $F$  in the turbulent dissipation rate equation can be used to activate/deactivate the cross diffusion term which shifts the turbulence equations between the  $k$ - $\omega$  turbulence model and the  $k$ - $\varepsilon$  turbulence model. By setting the blending function  $F$  to zero, the cross diffusion term in the turbulent dissipation rate equation is activated and the  $k$ - $\varepsilon$  turbulence model is tested. By setting the blending function  $F$  to unity, the cross diffusion term in the turbulent dissipation rate equation is deactivated and the  $k$ - $\omega$  turbulence model is tested.

The constants used in the 2D and 3D Manufactured Solutions for the verification of turbulence equations are presented in Appendix C in Table C1 and Table C2, respectively. In the process of selecting the Manufactured Solution for code verification purposes, it is required that different terms in the governing equations are roughly the same order of magnitude such that the contribution from each term in the governing equation is of same order of magnitude. This prevents the larger magnitude terms from masking errors in other terms of smaller magnitude.

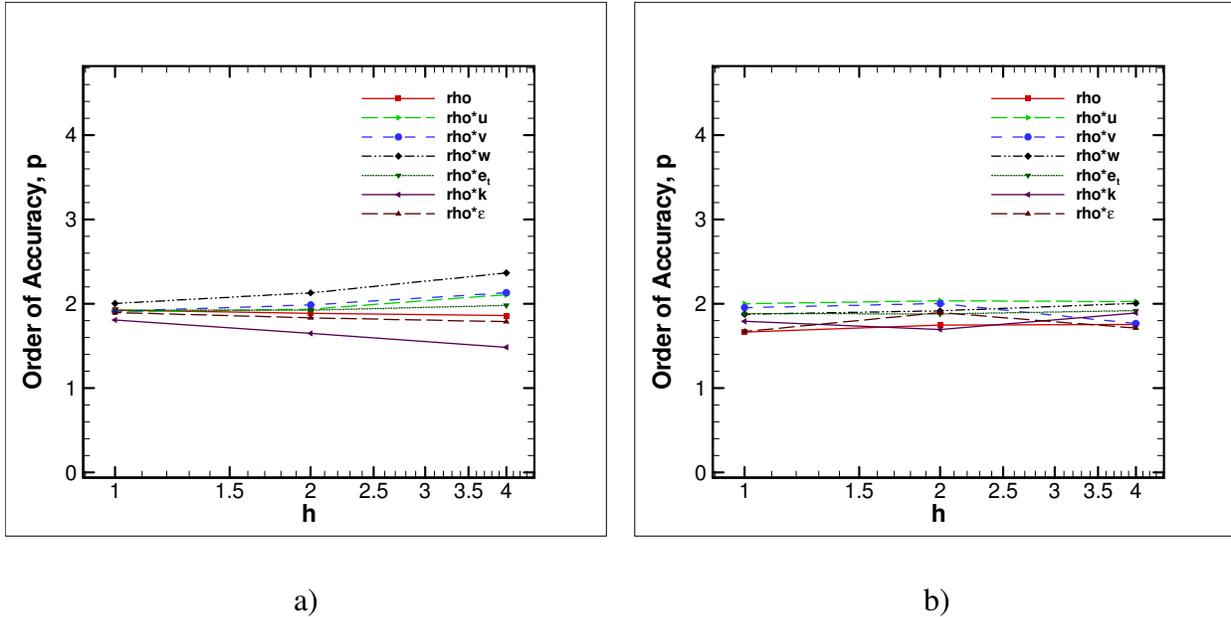
During the verification of turbulence models, the Manufactured Solutions are generated such that all terms in the turbulence models are roughly same order of magnitude over the domain considered for verification. For example, when the source terms are generated for the turbulent kinetic energy equation, the ratios of the convection, diffusion, and production terms to the destruction terms are calculated to check that all the terms in the turbulent kinetic energy equation are of similar orders of magnitude. As an example, the ratio of production term to the destruction terms in the turbulent kinetic energy equation in a 2D rectangular domain is shown in Figure 30. From the figure, the ratio of the production term to the destruction term is maintained such that the production term and the destruction term are of similar orders of magnitude in the complete domain. Similar source term ratios for the turbulent dissipation rate equation are calculated to check all the terms are of similar orders of magnitude.<sup>29</sup>



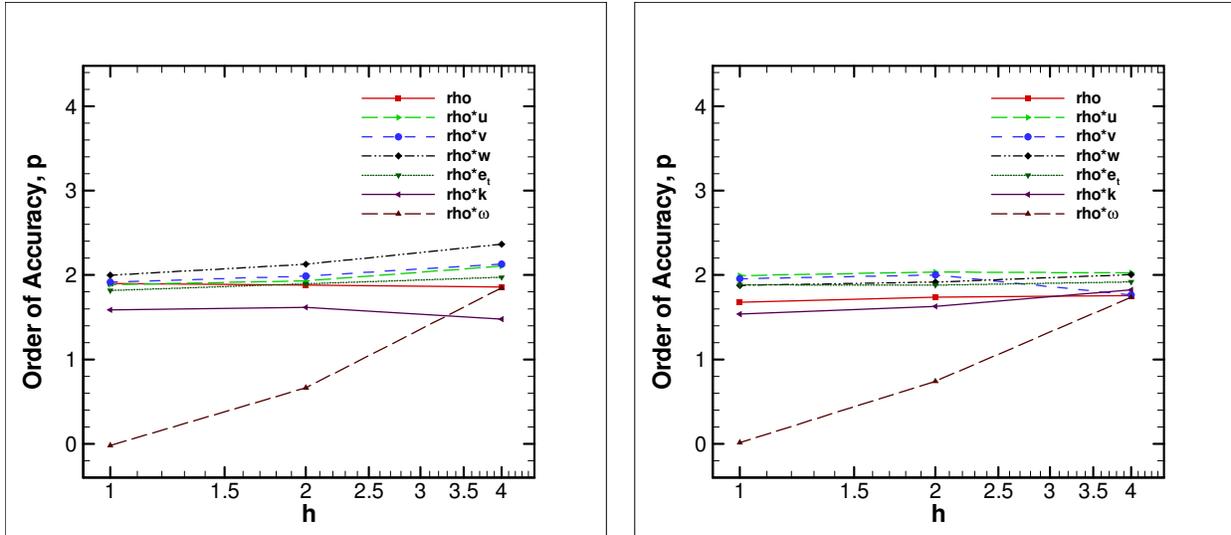
**Figure 30: Ratio of production term to the destruction term in turbulent kinetic energy equation in a 2D rectangular domain**

The  $k-\omega$  turbulence model is tested on the 3D skewed hybrid mesh and the observed order of accuracy approaches two with mesh refinement. The observed order of accuracy result for the  $k-\omega$  turbulence model on the 3D skewed hybrid mesh is shown in Figure 31. During the testing of  $k-\epsilon$  turbulence model, a problem with the turbulent dissipation rate equation was observed and the discretization error for that equation did not decrease at expected rate. The observed order of accuracy of the  $\rho\omega$  discretization error norms dropped to zero with mesh

refinement, but all the other conserved variable discretization norms approached two with mesh refinement. The behavior of the observed order of accuracy with mesh refinement for the k- $\epsilon$  turbulence model on the 3D skewed hybrid mesh is shown in Figure 32.



**Figure 31: Order of accuracy results on the 3D skewed hybrid mesh for k- $\omega$  turbulence model (F=1) using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

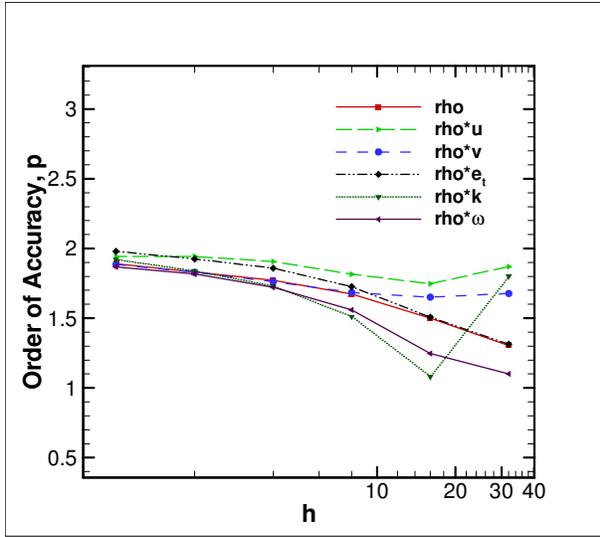


a)

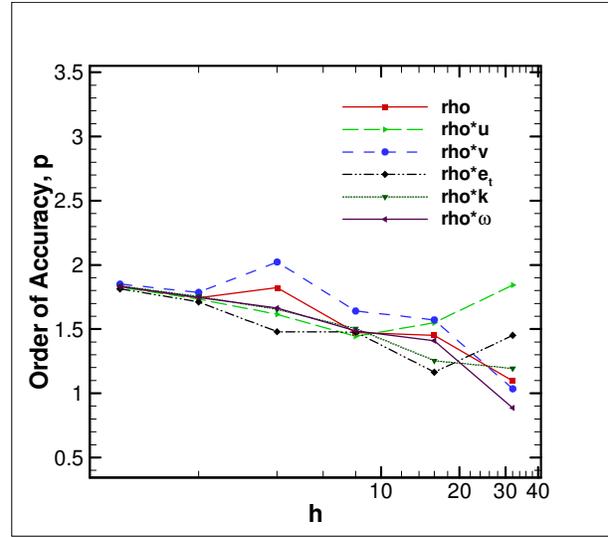
b)

**Figure 32: Order of accuracy results on the 3D skewed hybrid mesh for k- $\epsilon$  turbulence model ( $F=0$ ) using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

To explore the reason for failure of the verification test for the k- $\epsilon$  turbulence model on the 3D skewed hybrid mesh, it is tested on simpler meshes. Initially the k- $\epsilon$  turbulence model is tested on the 2D hybrid mesh and it is observed that the verification is successful with all the norms of the discretization errors approaching two with mesh refinement. The order of accuracy result for the k- $\epsilon$  turbulence model on the 2D hybrid mesh is shown in Figure 33. The above test is also done on a highly skewed 3D curvilinear mesh with hexahedral cells and the k- $\epsilon$  turbulence model is successfully verified with all the norms of the discretization error approaching two with mesh refinement. The order of accuracy result for the k- $\epsilon$  turbulence model in the 3D highly skewed curvilinear mesh is shown in Figure 34.

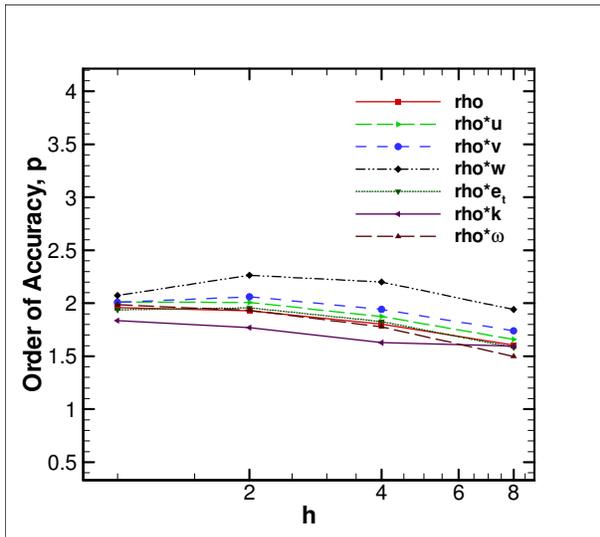


a)

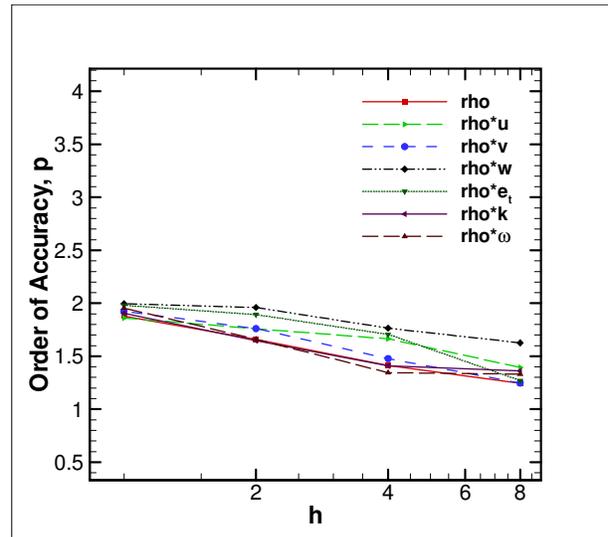


b)

**Figure 33: Order of accuracy results on the 2D hybrid mesh for  $k-\epsilon$  turbulence model ( $F=0$ ) using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**



a)

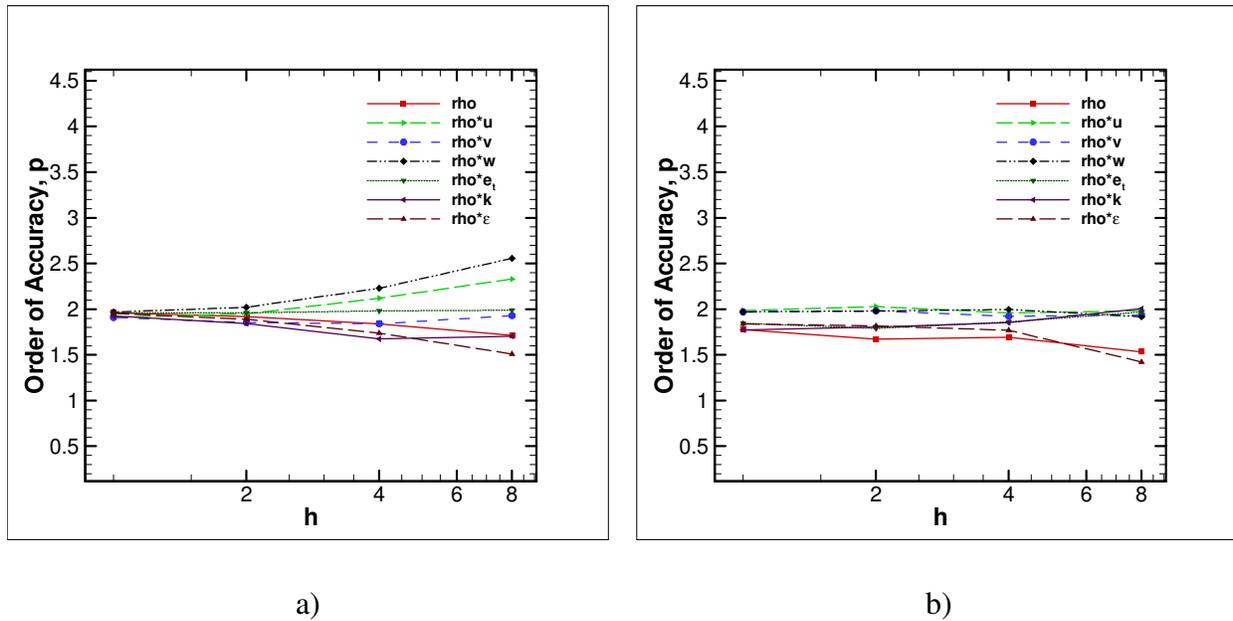


b)

**Figure 34: Order of accuracy results on the 3D highly skewed curvilinear (i.e., structured) mesh with hexahedral cells for  $k-\epsilon$  turbulence model ( $F=0$ ) using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

In addition, the  $k-\epsilon$  turbulence model is tested on a 3D unstructured mesh with tetrahedral cells and it is successfully verified with all the norms of the discretization error approaching two

with mesh refinement. The order of accuracy result for the  $k-\varepsilon$  turbulence model on 3D unstructured mesh with tetrahedral cells is shown in Figure 35. By testing on different meshes, it can be concluded that there is an issue in the discrete formulation of some of the terms in the turbulent dissipation rate equation as it works correctly only on 2D mesh topologies, 3D structured mesh topologies, and 3D unstructured mesh with tetrahedral cells but it fails on 3D unstructured mesh topologies with skewed cells. From the above analysis, it is determined that the issue is isolated to the cross-diffusion term in the turbulent dissipation rate equation on 3D unstructured mesh with skewed tetrahedral cells.



**Figure 35: Order of accuracy results on the 3D unstructured mesh with tetrahedral cells for  $k-\varepsilon$  turbulence model ( $F=0$ ) using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

#### 4.6 Verification of Time Accuracy for Unsteady Flows

It is more difficult to apply the verification procedure using the order of accuracy test to problems that involve both spatial and temporal discretization, especially when the spatial order is different from the temporal order. A combined spatial and temporal order verification method was developed by Kamm et al.<sup>64</sup> In their approach, they use the Newton-type iterative procedure to solve a coupled, non-linear set of algebraic equations to calculate the coefficients and observed order of accuracies for the spatial and temporal terms in the discretization error

expansion. In this present work, a simpler approach for spatial and temporal order verification is proposed.

Neglecting the higher order terms, the discretization error for a scheme with spatial and temporal terms, can be written as

$$\epsilon_{h_x}^{h_t} = g_x h_x^{\hat{p}} + g_t h_t^{\hat{q}} \quad (4.41)$$

where  $\hat{p}$  and  $\hat{q}$  are the observed orders of accuracy in space and time, respectively,  $g_x$  and  $g_t$  are the coefficients of spatial and time terms, respectively, and  $h_x$  and  $h_t$  are normalized spatial and temporal discretizations respectively. Similarly, with different set of coefficients, the norm of the discretization error can be found as

$$\|\epsilon_{h_x}^{h_t}\| = g_x h_x^{\hat{p}} + g_t h_t^{\hat{q}} \quad (4.42)$$

Initially, a spatial mesh refinement study is performed with a fixed time step to calculate  $\hat{p}$  and  $g_x$  using three mesh levels which makes the discretization error equation

$$\|\epsilon_{h_x}^{h_t}\| = g_x h_x^{\hat{p}} + \phi \quad (4.43)$$

where  $\phi = g_t h_t^{\hat{q}}$  is the fixed temporal error term. Using three mesh solutions, refined by the factor  $r_x$ , coarse ( $r_x^2 h_x$ ), medium ( $r_x h_x$ ), and fine ( $h_x$ ), the observed order of accuracy  $\hat{p}$  can be calculated<sup>7</sup> as

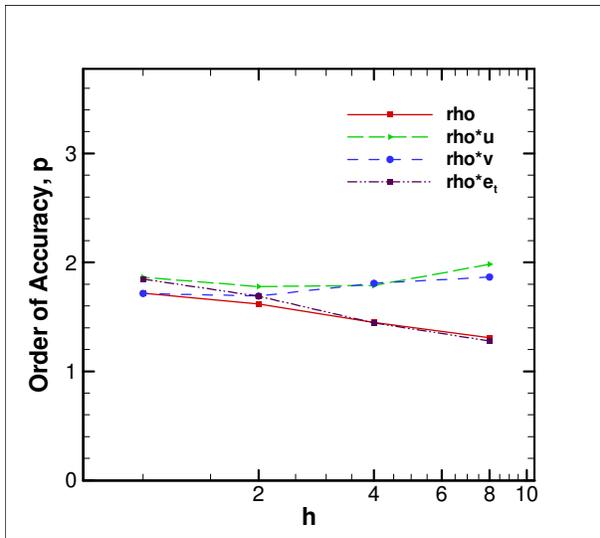
$$\begin{aligned} \frac{g_x (r_x^2 h_x)^{\hat{p}} - g_x (r_x h_x)^{\hat{p}}}{g_x (r_x h_x)^{\hat{p}} - g_x (h_x)^{\hat{p}}} &= \frac{(\|\epsilon_{r_x^2 h_x}^{h_t}\| - \phi) - (\|\epsilon_{r_x h_x}^{h_t}\| - \phi)}{(\|\epsilon_{r_x h_x}^{h_t}\| - \phi) - (\|\epsilon_{h_x}^{h_t}\| - \phi)} \\ \frac{g_x (r_x h_x)^{\hat{p}} [r_x^{\hat{p}} - 1]}{g_x (h_x)^{\hat{p}} [r_x^{\hat{p}} - 1]} &= \frac{\|\epsilon_{r_x^2 h_x}^{h_t}\| - \|\epsilon_{r_x h_x}^{h_t}\|}{\|\epsilon_{r_x h_x}^{h_t}\| - \|\epsilon_{h_x}^{h_t}\|} \\ \hat{p} &= \frac{\ln\left(\frac{\|\epsilon_{r_x^2 h_x}^{h_t}\| - \|\epsilon_{r_x h_x}^{h_t}\|}{\|\epsilon_{r_x h_x}^{h_t}\| - \|\epsilon_{h_x}^{h_t}\|}\right)}{\ln(r_x)} \end{aligned} \quad (4.44)$$

where  $r_x$  is the spatial refinement factor between two mesh levels and the coefficient of the spatial term  $g_x$  can be calculated as

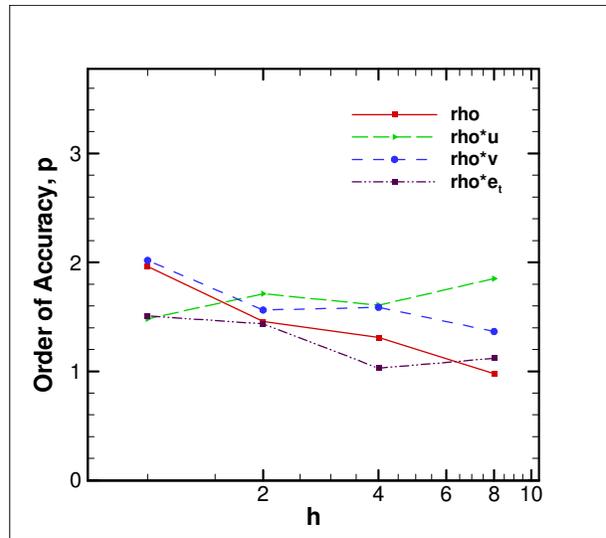
$$g_x = \frac{\left\| \epsilon_{r_x h_x}^{h_t} \right\| - \left\| \epsilon_{h_x}^{h_t} \right\|}{h_x^{\hat{p}} (r_x^{\hat{p}} - 1)} \quad (4.45)$$

Similarly, a temporal refinement study is performed on a fixed mesh to calculate  $\hat{q}$  and  $g_t$  using three temporal discretizations, coarse ( $r_t^2 h_t$ ), medium ( $r_t h_t$ ), and fine ( $h_t$ ). With all the coefficients calculated, the spatial step size and the temporal step size can be chosen such that the spatial discretization error term has the same order of magnitude as the temporal discretization error term. It is required that the spatial and temporal discretization error terms have the same order of magnitude such that both the terms have the same effect on the discretization error of the scheme. With one error term much smaller than the other error term, it makes it difficult to verify the order of accuracy of the smaller error term. If the temporal discretization error term is so small when compared to the spatial discretization error term, then mistakes in the temporal discretization will not be seen on very fine meshes.

Once these two terms are approximately the same order of magnitude, combined spatial and temporal order verification is conducted by choosing the temporal refinement factor such that the temporal error term drops by the same order of magnitude as the spatial error term with refinement, i.e.,  $r_t = r_x^{\hat{p}/\hat{q}}$ . Here  $r_t$  is the temporal refinement factor,  $r_x$  is the spatial refinement factor,  $\hat{p}$  is the spatial order and  $\hat{q}$  is the temporal order. In our case, the formal order is 2 in both space and time, i.e.,  $\hat{p} = \hat{q} = 2$ . Using this procedure, the unsteady time term is verified on the 2D hybrid mesh and the 3D hybrid mesh for Navier-Stokes equations. The observed order of accuracy on both the meshes approached two with mesh refinement. The order of accuracy results using  $L_2$  discretization error for the time accuracy of the unsteady flows on 2D hybrid mesh and 3D hybrid mesh are shown in Figure 36 and Figure 37, respectively.

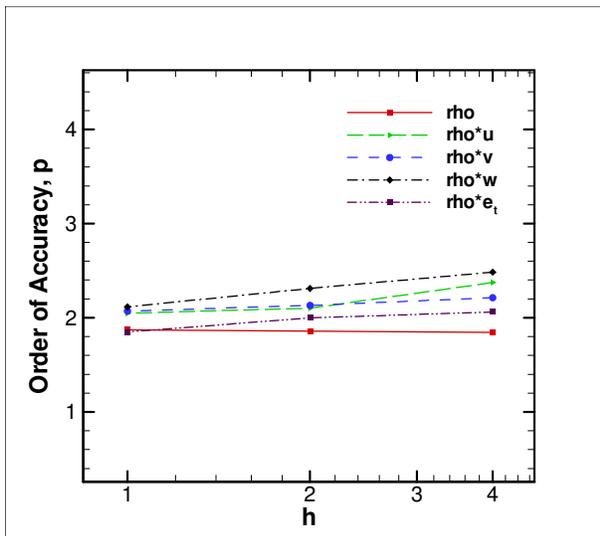


a)

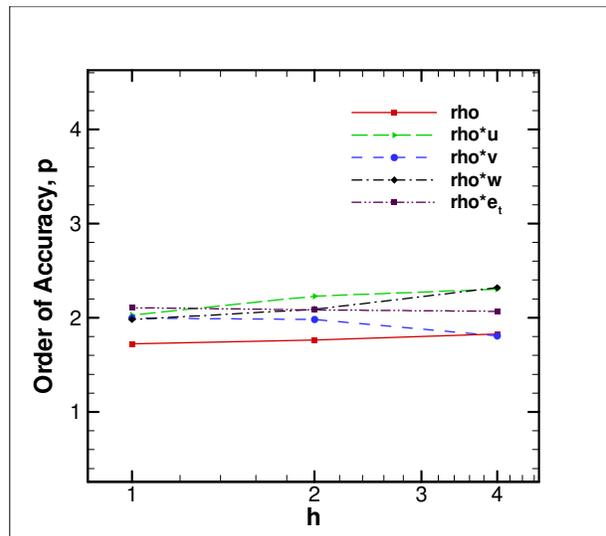


b)

**Figure 36: Order of accuracy results for time accuracy of the unsteady flows on the 2D hybrid grid using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**



a)



b)

**Figure 37: Order of accuracy results for time accuracy of the unsteady flows on the 3D hybrid grid using a)  $L_1$  norm of the discretization error and b)  $L_\infty$  norm of the discretization error**

## **4.7 Issues Uncovered During Code Verification**

During the verification of the finite volume code, several coding mistakes and algorithm inconsistencies were uncovered. Some code options failed the order of accuracy test for the first time and the verification studies helped in learning more about the code or the algorithm and ultimately removing some of the mistakes in the code or algorithm. Because the verification procedure is so sensitive to minor issues like the mesh topology or mesh quality, they can often uncover sensitivities to these issues. Issues uncovered during the code verification are documented here.

### **4.7.1 Coding Mistakes/Algorithm Inconsistencies**

A coding mistake was found and corrected in the formulation of the diffusion operator while testing the Navier-Stokes equations in the finite volume code on the 2D skewed curvilinear mesh. Initially, the discretization error did not decrease with mesh refinement and a modification was done to the diffusion operator by Luke.<sup>39</sup> The new diffusion operator rectified the problem and an observed order of accuracy of two was attained with mesh refinement.

A coding mistake was found and corrected in the formulation of the diffusion operator while testing the Navier-Stokes equations on 2D unstructured mesh with triangular cells. Initially, the diffusion operator was found to be only first order accurate on 2D unstructured meshes and after modifying the diffusion operator formulation for unstructured grids, the diffusion operator tested was found to be second order accurate.

Testing the  $k-\epsilon$  turbulence model on different mesh topologies, it was found that there was an issue in the discrete formulation of the cross-diffusion term in the turbulent dissipation rate equation. The present formulation works fine on 2D mesh topologies, 3D structured mesh topologies, and 3D unstructured mesh with tetrahedral cells but failed on 3D unstructured mesh topologies with skewed cells.

### **4.7.2 Grid Sensitivities**

Systematic mesh refinement was found to be important for the verification process. Lack of systematic mesh refinement on unstructured meshes led to the failure of code verification test. As an example, initially 3D unstructured meshes containing tetrahedral cells were generated using GRIDGEN. Using this mesh generation tool, a domain was filled with tetrahedral cells

automatically. Generating different mesh levels in GRIDGEN, the meshes were not systematically refined. When Navier-Stokes equations were tested using these meshes generated in GRIDGEN, the code verification test failed where the observed order of accuracy did not match the formal order. Later, the 3D unstructured mesh with tetrahedral cells was generated using the developed mesh generation code and meshes from this code were systematically refined between the mesh levels. Using these meshes, the Navier-Stokes equations were successfully verified to second order accurate, hence explaining the importance of systematic mesh refinement for code verification.

During the verification of the no-slip wall boundary conditions it was found that the mesh should be normal to the wall and the order of accuracy test failed if the mesh is not normal to the wall. To explain this, initially while testing the adiabatic no-slip wall boundary condition in 2D, the meshes which were used for verification did not have mesh normal to the boundary. While testing the adiabatic no-slip wall boundary on these meshes, the calculated observed order of accuracy did not match the formal order and error was found to be near the boundary. Later, when the meshes are generated again such that the mesh is normal to the boundary, the adiabatic no-slip wall boundary was successfully verified.

The discrete formulation of the governing equations was found to be sensitive to highly skewed tetrahedral and prismatic cells. The baseline governing equations were successfully verified on the highly skewed 2D hybrid mesh and highly skewed 3D curvilinear mesh with hexahedral cells, but failed the verification test on a highly skewed 3D hybrid mesh.

## 5. Finite Volume Diffusion Operators

For well over a decade, there has been significant work on the development of algorithms for the compressible Navier-Stokes equations on unstructured grids. This work is motivated by the fact that unstructured meshes allow more automation of the mesh generation process and thus less of the analyst's time relative to structured meshes. One of the challenging tasks during the development of these algorithms is the formulation of consistent and accurate diffusion operators. A robust approach is desired for the treatment of diffusion operators such that they are at least second order accurate on various structured and unstructured cell topologies. In addition, the accuracy of diffusion operators also depends on mesh quality.

A survey of diffusion operators for compressible CFD solvers was conducted to understand different formulation procedures for diffusion fluxes. A patch-wise version of the Method of Manufactured Solutions was used to test the accuracy of some of the selected diffusion operators. These diffusion operators were tested and compared on different 2D mesh topologies to study the effect of mesh quality, mesh resolution, and the solution behavior. For 2D structured meshes, mesh quality includes stretching, aspect ratio, skewness, and curvature, but for 2D unstructured meshes, these mesh quality attributes cannot be isolated. As an example, a triangular mesh with curvature, aspect ratio or stretching will also have skewness. Quantities examined include the numerical approximation errors and order of accuracy associated with face gradient reconstruction. In the present work, the testing of diffusion operators is limited to cell-centered finite volume methods in 2D which are formally second order accurate.

### 5.1 Robustness and Accuracy

In the development of a numerical diffusion operator, one generally desires the creation of an accurate and robust operator. The consistent and accurate treatment of diffusion fluxes are particularly challenging for finite volume solvers. Generally, it is required for the discrete operator to share important properties with the original continuous operator. For diffusion operators there are three properties that we would like to satisfy and these are as follows:<sup>39</sup>

1. The operator should be numerically conservative. If the operator is defined as a sum of fluxes over all of the faces of a cell, then the operator is conservative if the shared faces utilize the same flux.

2. The operator should not generate new extrema. That is, the maximum and minimum solution values for diffusion processes should occur at the boundaries.
3. The operator should be linearity preserving. That is, the operator should evaluate identically to zero when given a linear function.

Satisfaction of the first property of conservation when using a finite-volume scheme is easily achieved. If the operator is defined as a sum of fluxes over all of the faces of a cell, and the shared faces utilize the same flux, then conservation is automatically achieved. The challenge then turns to the formulation of the cell interface fluxes. The second property can be evaluated simply for linear schemes. For linear schemes the operator which is formed from the sum of interface fluxes is written as a weighted sum of neighboring cell values. If all of the weights in this sum are positive then the operator will not produce new extrema as each cell value is an average of neighboring values and thus will be bounded by them. For this argument to hold, the weights of immediately adjacent cells must be non-zero, otherwise a degenerated ‘rotated’ Laplacian could result. An operator that does not satisfy this property could result in solutions where there is unbounded growth in the solution. As a result, the second property is required to produce a robust scheme. It has been shown for finite-element schemes that a positive stencil will be generated for Delaunay simplex meshes. For mixed element meshes, such as those found in typical unstructured viscous meshes, no such guarantee has been shown. For finite-volume meshes, the most typical strategy for formulating the viscous fluxes is to employ a directional derivative technique whereby the derivative in the direction of the vector connecting cell centroids is computed using the cell values on either side of the face, while gradients in other directions are computed through an average of the cell centered gradients. This averaging technique avoids the ‘rotated’ Laplacian stencil that will result if a simple average of cell centered gradients were used. Strang et al.<sup>62</sup> showed that for VGRID generated meshes up to the thirty percent of mesh cells had negative coefficients in the diffusion operator stencil. Haselbacher et al.<sup>42</sup> showed that for a linear scheme, one could not in general satisfy all of the three conditions listed above; therefore one must either relax these requirements or resort to some form of non-linear limiting procedure to create a robust diffusion operator within the context of a finite volume scheme. When the stencil positivity property is violated, it usually doesn’t result in failure of the scheme; however the robustness of such a scheme is not satisfactory for production use. Finally, linearity preserving is generally required to achieve a

second order reconstruction. At the very least, failure to satisfy the linearity preservation property guarantees that the solution will be highly sensitive to the local quality and distribution of mesh elements.

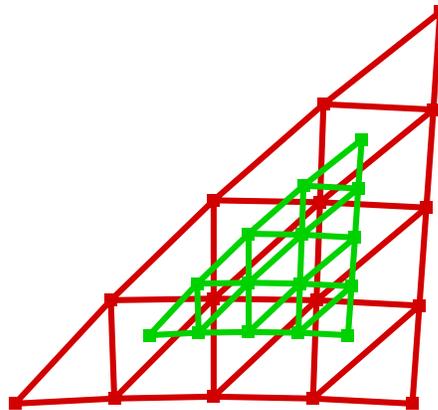
## 5.2 Testing Framework

The accuracy of a diffusion operator in the Navier-Stokes equations can be tested using approaches that are similar to those from the Method of Manufactured Solutions<sup>9</sup> often employed for code verification. During this procedure, a smooth analytic solution is selected to assess the diffusion operator. For the diffusion operator to be verified, the observed order of accuracy is required to match the formal order of accuracy.

An analytic solution is used to find the exact solution which is used for the calculation of the error for the numerical diffusion operator. The difference in the procedure applied here when compared to the MMS is that the analytic solution is not applied to any governing equation to generate a source term. During the formulation of the diffusion operator, the common procedure is to calculate the gradients on the cell faces and these solution gradients on the cell faces are tested for the accuracy of the diffusion operator. For testing the diffusion operators in finite volume CFD codes, initially, the numerical solution values at the cell centers are obtained from the analytic solution selected for testing and the solution values at the cell centers are used to calculate the numerical solution gradients at the face centers. The values of the solution gradients may be different depending on the type of formulation used for the calculation of the gradients. The analytic solution can also be used to calculate the exact solution gradients at the face centers. Then the error in the numerical formulation of the gradient can be calculated as the difference between the numerical solution gradients and the exact solution gradients at the face centers. The procedure is repeated on multiple mesh levels to get the error from each mesh level which are used to calculate the observed order of accuracy. If the observed order of accuracy calculated from multiple mesh levels matches the formal order, the diffusion operator can be considered verified. Since the mesh quality can affect the accuracy of the diffusion operator, it can be altered during the testing process to build an accurate and consistent diffusion operator.

In the process of calculating the observed order of accuracy, different mesh levels are required to obtain the discretization error on different systematically refined meshes. By doing a traditional way of refinement as already discussed in the order of accuracy verification using

MMS, the computations become expensive particularly for 3D problems where each level of refinement is 8 times larger. So, particularly for testing the diffusion operators a different approach is used in terms of mesh refinement. This alternate method used is similar to the downscaling approach to order verification<sup>17,18</sup> which employs a single mesh which is scaled down about a single point in the domain. In the downscaling approach, several parts of the mesh in a domain are selected and they are repeatedly shrunk towards a focal point. In our approach, only the stencil or the mesh to be scaled is only considered for testing the numerical formulation instead of the complete domain. An example of shrinking a mesh over a focal point for a 2D unstructured mesh is shown in Figure 38. In the figure, the mesh is shrunk over the cell centroid of the central triangle which is the focal point and the whole mesh is scaled by a factor of two. By shrinking the mesh in this fashion, a systematic mesh refinement is achieved which means that the refinement is uniform and the mesh quality remains same with refinement. Similarly, the mesh can be further scaled down several times to obtain multiple mesh levels for testing the formulation of a diffusion operator.



**Figure 38: Shrinking of a mesh over a focal point for a 2D unstructured mesh**

### **5.3 Cell-Centered Finite Volume Diffusion Operators**

After a brief literature survey, a few commonly used cell centered finite volume diffusion operator formulations in the literature are selected and tested for their accuracy on different mesh topologies with different cell quality aspects. The different numerical formulations tested include the node-averaged method, unweighted least squares, weighted least squares, the numerical

formulation used by the Loci-CHEM CFD code, and the Face Approximate Mapped Least Squares (FAMLSQ) method<sup>44</sup> which uses a compact stencil. These methods are briefly discussed below.

### 5.3.1 Node Averaging Method

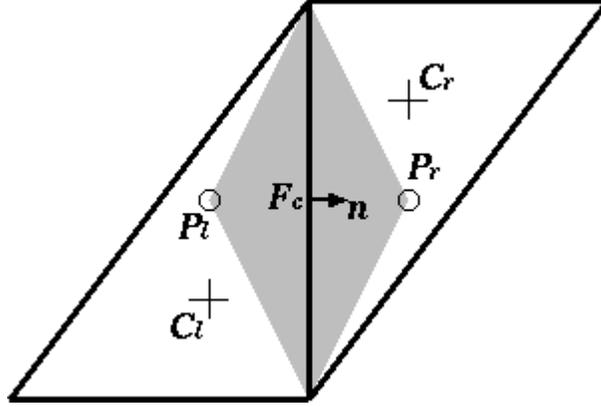
For cell centered formulations, a common method for calculating the solution gradients at the face centers is the node averaging method.<sup>44</sup> In this method, the solution values are reconstructed at the nodes from the surrounding cell centers. The solution reconstruction is proposed in [65, 66] and used in [67] as an averaging procedure that is based on a constrained optimization to satisfy the Laplacian properties. For the calculation of gradients on the face center, the derivative tangent to the face is computed as the divided difference between the solution values reconstructed at the nodes and the derivative normal to the face is computed using the solution values at the centroids of the cells sharing the face. The gradient is resolved from the derivative along the face and the derivative along the line connecting the cell centers across the face.

### 5.3.2 Unweighted and Weighted Least Squares

In the weighted least squares method,<sup>44</sup> the contributions from the adjacent cells to the minimized functional are weighted with weights inversely proportional to the distance from the central point, i.e., the face center when calculating the gradients on the face. In the unweighted least squares method,<sup>44</sup> all contributions from the adjacent cells are equally weighted.

### 5.3.3 Loci-CHEM Diffusion Operator

In order to more reliably obtain second order gradients at the face, an alternative method for computing these gradients is introduced in the Loci-CHEM code. In this method, a new control volume centered about the face is constructed. Initially, a least squares approach is used to calculate the solution gradients at the cell center.<sup>39</sup> Then, a second order function reconstruction is used to compute the nodal values of a control volume centered about a face. This is constructed by computing two points that are projected above and below the face centroid in the normal direction by a factor of  $0.5\vec{n} \cdot \vec{m}$  to form points  $P_l = F_c - 0.5(\vec{n} \cdot \vec{m})\vec{n}$ , and  $P_r = F_c + 0.5(\vec{n} \cdot \vec{m})\vec{n}$  as illustrated in Figure 39.  $\vec{n}$  is the face normal vector and  $\vec{m}$  is the vector that connects the left and right cell centroids.



**Figure 39: An illustration of the reconstruction of a centered control volume about face  $F_c$  as shown in the shaded region**

The cell gradients are then used in the respective cells to reconstruct the function values at the points  $P_l$  and  $P_r$  while at the function values at the face nodes are reconstructed using the average of the left and right cell function reconstructions. The gray area in Figure 39 illustrates this reconstructed diamond cell. The left and right points of the reconstruction along with the edges of the face form triangles that bound this volume. Green's theorem is then employed to compute a second order face gradient.

#### 5.3.4 FAMLSQ Method with Compact Stencil

A more general approximate mapping (FAMLSQ) method<sup>44</sup> that is based on the distance function, defined as the distance to the nearest boundary is explained here. The FAMLSQ method useful for curved boundaries, applies a least-square minimization in a locally constructed coordinate system.

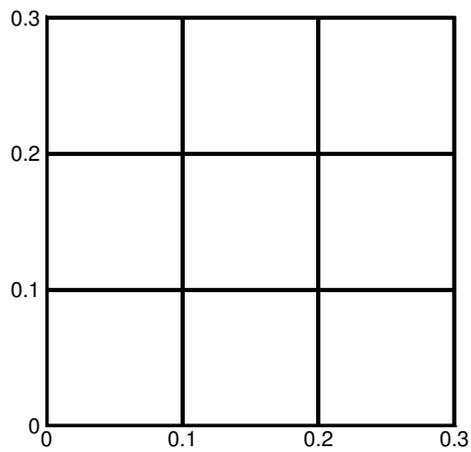
$$f^r = f_0 + \kappa \zeta' + \lambda \eta' \quad (5.1)$$

The local coordinates  $(\zeta', \eta')$  are constructed using the distance function, which provides information on the closest boundary point. First the distance function is reconstructed at the face center. The coordinate vectors at the face center are defined as a unit  $\eta'$ -directional vector pointing in the direction opposite to the closest boundary point and its orthonormal  $\zeta'$ -directional vector. The  $\eta'$ -coordinate is taken as the distance from the boundary and the  $\zeta'$ -coordinate is the projection of the vector connecting the stencil point with the face center onto the  $\zeta'$ -direction. The stencil selected for calculating the solution gradients at face centers is called the compact

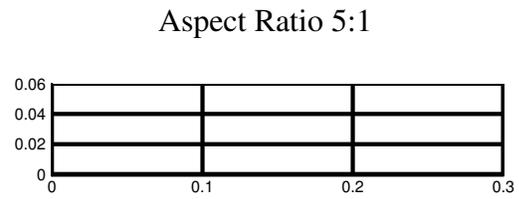
stencil.<sup>44</sup> The compact stencil typically involves two prime cells and two auxiliary cells; one for each prime cell. An auxiliary cell is chosen from the pool of the cells sharing the nodes of the face as the cell closest to the prime cell, but not its face neighbor. The compact stencil is important for discretizations on high-aspect-ratio grids to represent correctly the direction of the strong coupling.

#### **5.4 Mesh Topologies**

The different formulations of diffusion operators are tested on different mesh types which can isolate the mesh quality parameters including cell aspect ratio, cell stretching, cell skewness, and curvature. All the above types of cells are considered in the evaluation of the diffusion operators. In the process of evaluation, only a minimum grouping of cells required (i.e. the stencil) are considered for the calculation of the solution gradients on the face centers of a single cell. For 2D structured meshes, only 9 cells are used and the solution gradients are calculated on the face centers of the central cell. The 2D structured meshes with different cell quality aspects considered for testing the different formulations of diffusion operators are shown in Figure 40 and Figure 41. A uniform mesh is shown in Figure 40a, a mesh with aspect ratio cells is shown in Figure 40b, a stretched mesh is shown in Figure 40c, a mesh with skewed cells is shown in Figure 40d, and a curvilinear mesh is shown in Figure 41. For 2D unstructured meshes, the stencil contains 16 cells and only 13 cells are required for the calculation of solution gradients on the face centers of the central cell. The triangular cells in the three corners of the domain are not required for the calculation of solution gradients. The 2D unstructured meshes with different cell quality aspects considered for testing are shown in Figure 42. An isotropic triangular mesh is shown in Figure 42a, a triangular mesh with aspect ratio cells is shown in Figure 42b where the cells also have skewness associated with them, and a curvilinear triangular mesh is shown in Figure 42c.

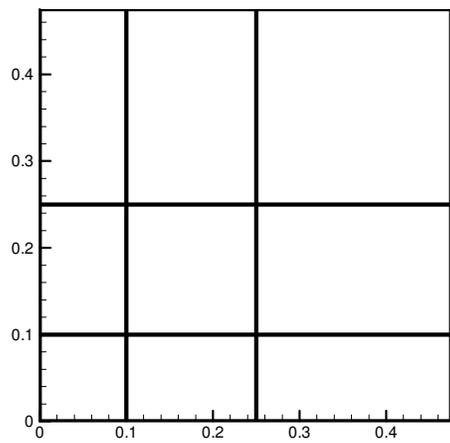


a)



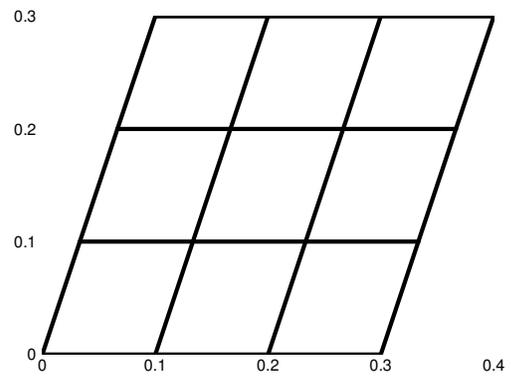
b)

Stretching Factor 1.5



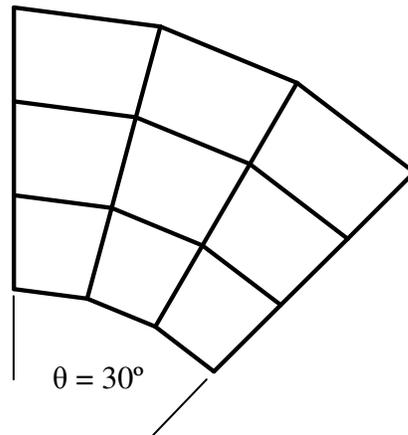
c)

Skew Angle 60°

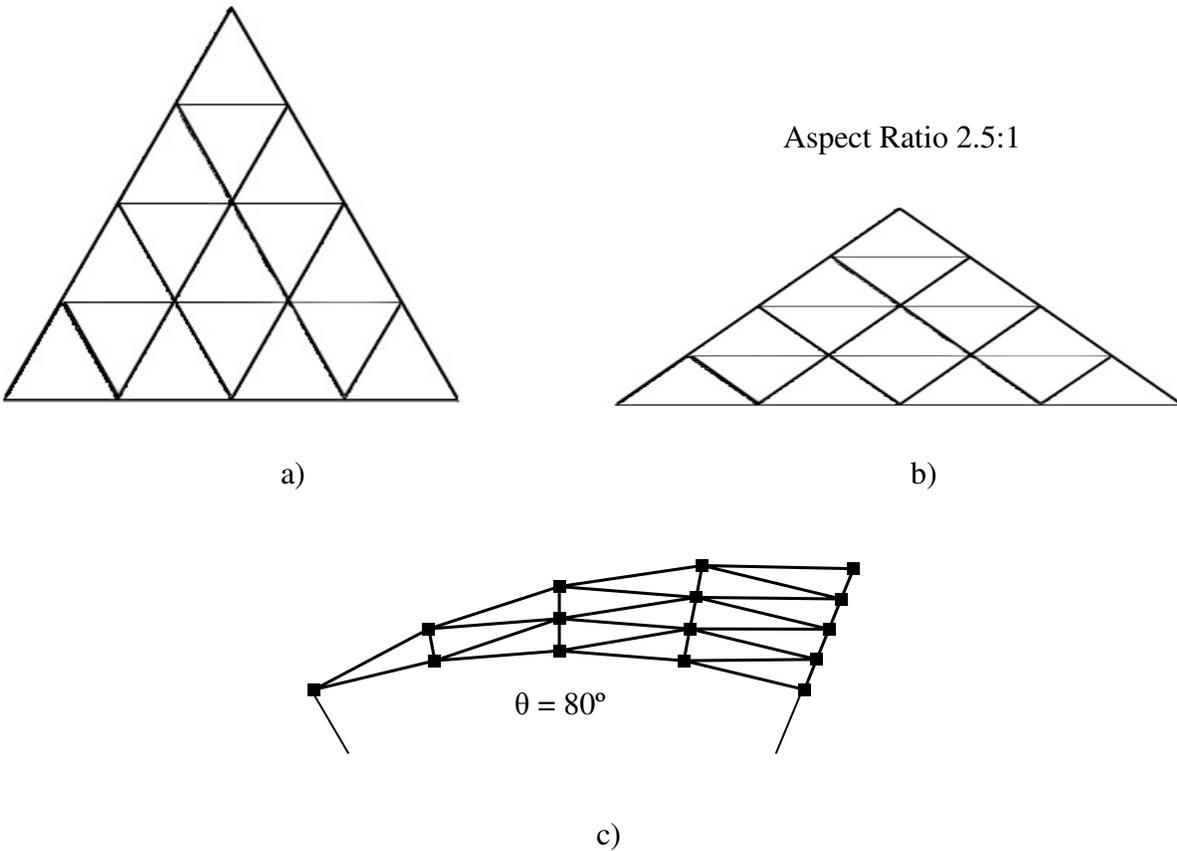


d)

**Figure 40: Structured mesh topologies in 2D: a) uniform mesh, b) mesh with aspect ratio cells, c) stretched mesh, and d) mesh with skewed cells**



**Figure 41: 2D structured mesh curvilinear mesh**



**Figure 42: Unstructured mesh topologies in 2D: a) mesh with isotropic cells, b) mesh with aspect ratio cells, and c) curvilinear mesh**

## 5.5 Effect of Mesh Quality

The evaluation criteria for diffusion operators are the calculation of the solution gradients on the face centers to compare them with the exact gradients for errors in the solution gradients and calculation of the observed order of accuracy with mesh refinement. The different numerical formulations considered are tested for accuracy on 2D structured meshes and 2D unstructured meshes which have some cell quality aspects. The analytic solution used while testing the different numerical formulations on 2D meshes is of the form given below:

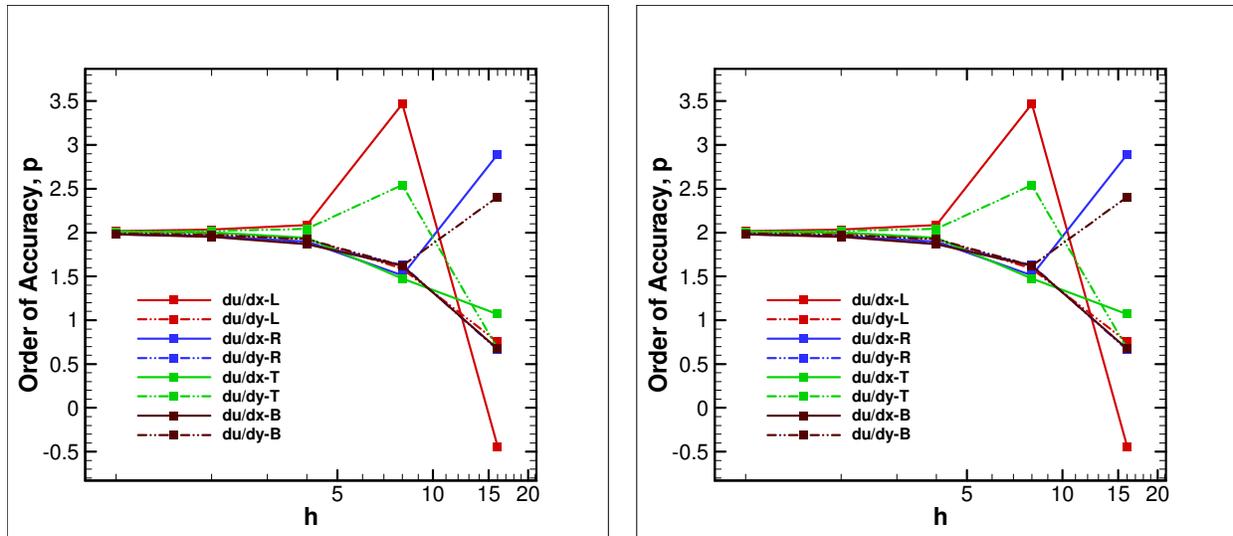
$$u(x, y) = u_0 + u_x \sin\left(\frac{2\pi(x - x_0)}{L}\right) + u_y \sin\left(\frac{2\pi(y - y_0)}{L}\right) + u_{xy} \sin\left(\frac{2\pi(x - x_0)(y - y_0)}{L^2}\right) \quad (5.2)$$

### 5.5.1 2D Structured Meshes

On 2D structured meshes, three numerical formulations of diffusion operators are tested. They include the node-averaged, unweighted least squares, and weighted least squares methods.

#### 5.5.1.1 Uniform Mesh

All three numerical formulations are tested on the 2D structured uniform mesh (Figure 40a). The solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on the four faces of the cell which is the central cell of the stencil considered. The mesh refinement is done by fixing the cell center of the central cell as the focal point and shrinking the remaining domain equally in all directions. On a uniform mesh all three numerical formulations methods produced similar errors. The solution gradients calculated at the face centers showed second order convergence with mesh refinement for all the numerical formulations tested on 2D structured uniform mesh. The observed order of accuracy results for the solution gradients at the face centers for the left, right, top, and bottom faces of the cell on a 2D structured uniform mesh using node averaging method and weighted least squares method are shown in Figure 43.



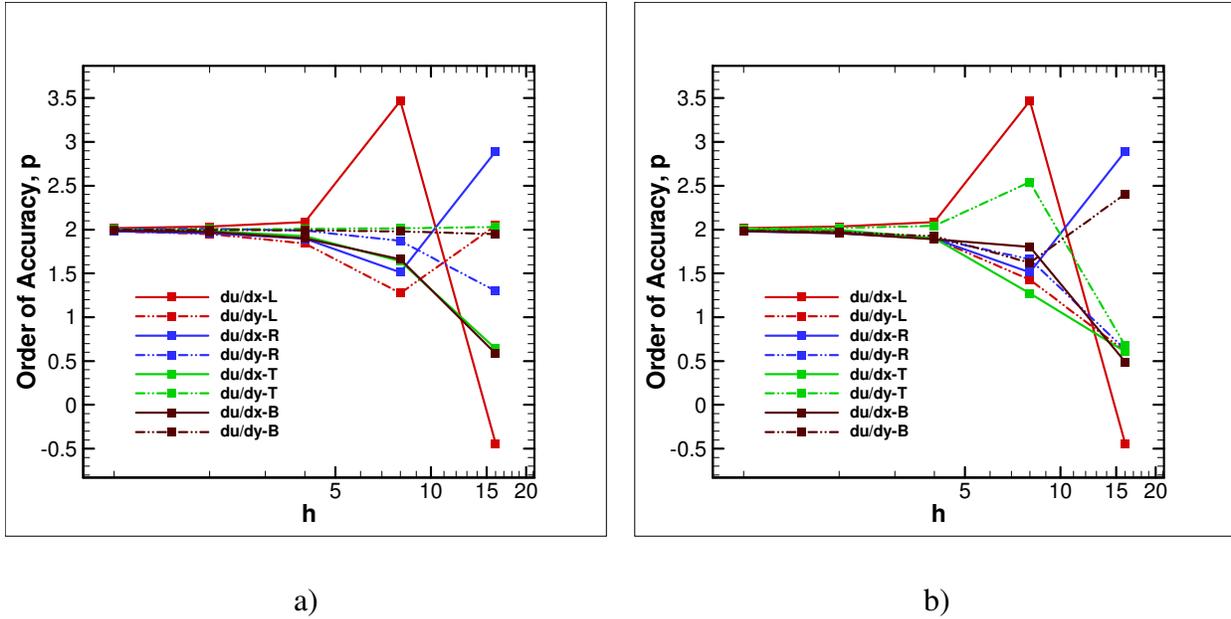
a)

b)

**Figure 43: Order of accuracy results for gradients at face center on a 2D structured uniform mesh using a) node averaging method and b) weighted least squares method**

### 5.5.1.2 Aspect Ratio

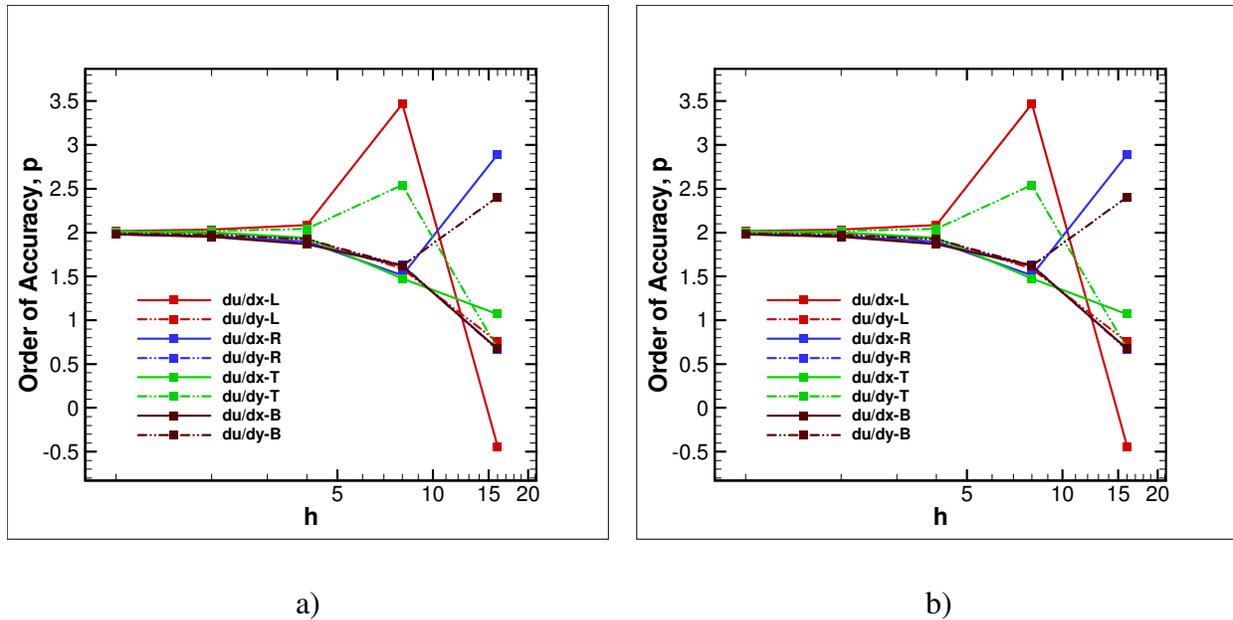
The effect of cell quality on the numerical formulations of the diffusion operators is tested. The numerical formulations of diffusion operators are tested on 2D structured mesh which has cells with a constant aspect ratio (Figure 40b). Meshes with aspect ratios of 5:1, 10:1 and 50:1 are used while testing the numerical formulations. The solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on the four faces of the cell which is the central cell of the stencil considered. The mesh refinement is done by fixing the cell center of the central cell as the focal point and shrinking the remaining domain equally in all directions. All three numerical formulations methods produced similar errors on the mesh with aspect ratio cells. The solution gradients calculated at the face centers showed second order convergence with mesh refinement for all the numerical formulations tested on the 2D structured mesh with aspect ratio cells even for cell with a high aspect ratio of 50:1. The change in aspect ratio has no effect on the accuracy and a second order behavior is observed for all aspect ratio cells considered. The observed order of accuracy results for the solution gradients at the face centers on a 2D structured mesh with aspect ratio cells of 50:1 using the node averaging method and the weighted least squares method are shown in Figure 44.



**Figure 44: Order of accuracy results for gradients at face center on a 2D structured mesh with aspect ratio using a) node averaging method and b) weighted least squares method**

### 5.5.1.3 Stretching

The numerical formulations of diffusion operators are tested on 2D stretched mesh with a constant stretching factor (Figure 40c). Stretched meshes with several stretching factors between 1 and 1.5 in both the coordinate directions are used while testing the numerical formulations. One important note here is that the cell centers of the stretched cells in the mesh are not considered as the geometric cell centers, but they are calculated depending on the stretching involved in the mesh. The solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on the four faces of the cell which is the central cell of the stencil considered. The mesh refinement is done by fixing the cell center of the central cell as the focal point and shrinking the remaining domain equally in all directions. All three numerical formulations methods produced similar errors on the 2D stretched mesh. The solution gradients calculated at the face centers showed second order convergence with mesh refinement for all the numerical formulations tested on stretched mesh with stretching factors up to 1.5. The change in stretching factor has no effect on the accuracy and a second order behavior is observed for all stretching factors considered. The observed order of accuracy results for the solution gradients at the face centers on a 2D stretched mesh with a stretching factor of 1.5 using node averaging method and weighted least squares method are shown in Figure 45.

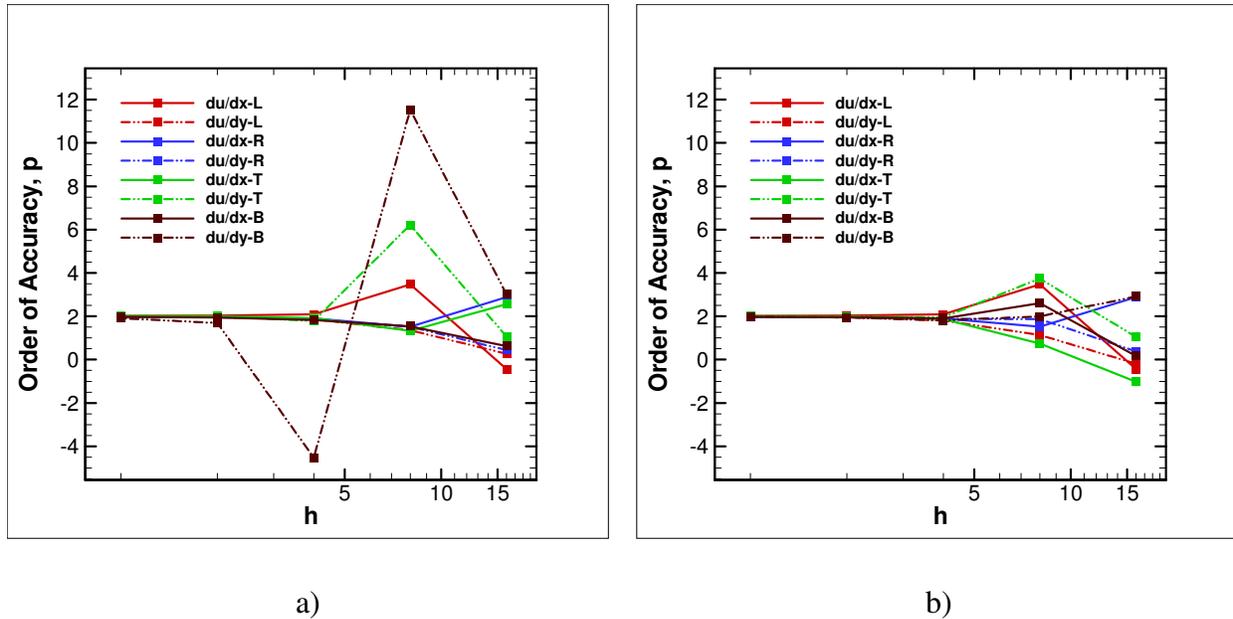


**Figure 45: Order of accuracy results for gradients at face center on a 2D structured mesh with stretched cells using a) node averaging method and b) weighted least squares method**

#### 5.5.1.4 Skewness

The numerical formulations of diffusion operators are tested on a 2D structured mesh with skewed cells (Figure 40d). Skewness in the structured mesh can be measured in many possible ways, but here skewness is measured as the minimum angle of the cell. The skewed meshes considered for testing have all the cells with same skewness. The 2D structured meshes with a minimum angle between two sides of a cell of  $60^\circ$ ,  $30^\circ$ ,  $15^\circ$ , and  $1^\circ$  are used while testing the numerical formulations. The solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on the four faces of the cell which is the central cell of the stencil considered. The mesh refinement is done by fixing the cell center of the central cell as the focal point and shrinking the remaining domain equally in all directions, thus maintaining the cell quality with mesh refinement. All three numerical formulations methods produced similar errors on the mesh with skewed cells. The solution gradients calculated at the face centers showed second order convergence with mesh refinement for all the numerical formulations tested on 2D structured mesh with skewed cells. The change in skewness of the mesh has no effect on the accuracy and a second order behavior is observed for all skewed cells considered. The observed order of accuracy results for the solution gradients at the face centers on a 2D structured mesh with highly skewed cells with a minimum

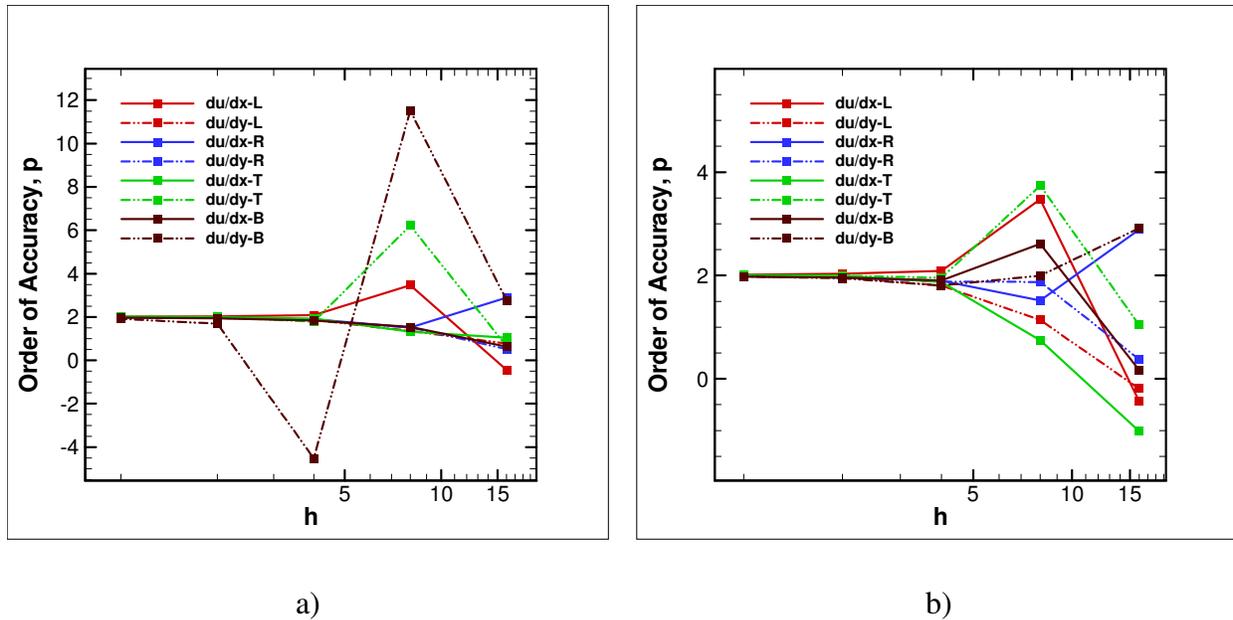
angle of  $1^\circ$  using node averaging method and weighted least squares method are shown in Figure 46.



**Figure 46: Order of accuracy results for gradients at face center on a 2D structured mesh with skewed cells using a) node averaging method and b) weighted least squares method**

### 5.5.1.5 Curvature

The effect of curvature as a mesh quality aspect is also examined for the different numerical formulations of diffusion operators. The mesh considered for testing the effect of curvature has an inner radius of 5 with different angles made by the stencil range from  $15^\circ$  to  $90^\circ$ . The solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on the four faces of the cell which is the central cell of the stencil considered. The mesh refinement is done by fixing the cell center of the central cell as the focal point and shrinking the remaining domain equally in all directions which maintains the quality of the mesh. The difference in the errors in the solution gradients is small for all the tested numerical formulations methods on the 2D structured curvilinear mesh. The solution gradients calculated at the face centers showed second order convergence with mesh refinement for all the numerical formulations tested on 2D structured curvilinear mesh (Figure 41). The change in the mesh curvature has no effect on the accuracy. The observed order of accuracy results for the solution gradients at the face centers on a 2D structured curvilinear mesh using node averaging method and weighted least squares method are shown in Figure 47.



**Figure 47: Order of accuracy results for gradients at face center on a 2D structured mesh with skewed cells using a) node averaging method and b) weighted least squares method**

As a conclusion, all the numerical formulations tested on 2D structured meshes are second order accurate on every type of mesh which has a particular mesh quality aspect associated to it. The effect of cell quality on the errors in the solution gradients calculated using any particular numerical method is negligible as the errors did not vary much between structured meshes with different cell quality. This approach of testing the numerical formulation can be extended to 3D without much effort in terms of mesh generation. Also, the shrinking aspect of mesh refinement for calculating the order of accuracy from the errors from different mesh levels can be applied easily in 3D saving computational resources. The accuracy of these numerical formulations is tested under more difficult situations when they are examined on 2D unstructured meshes.

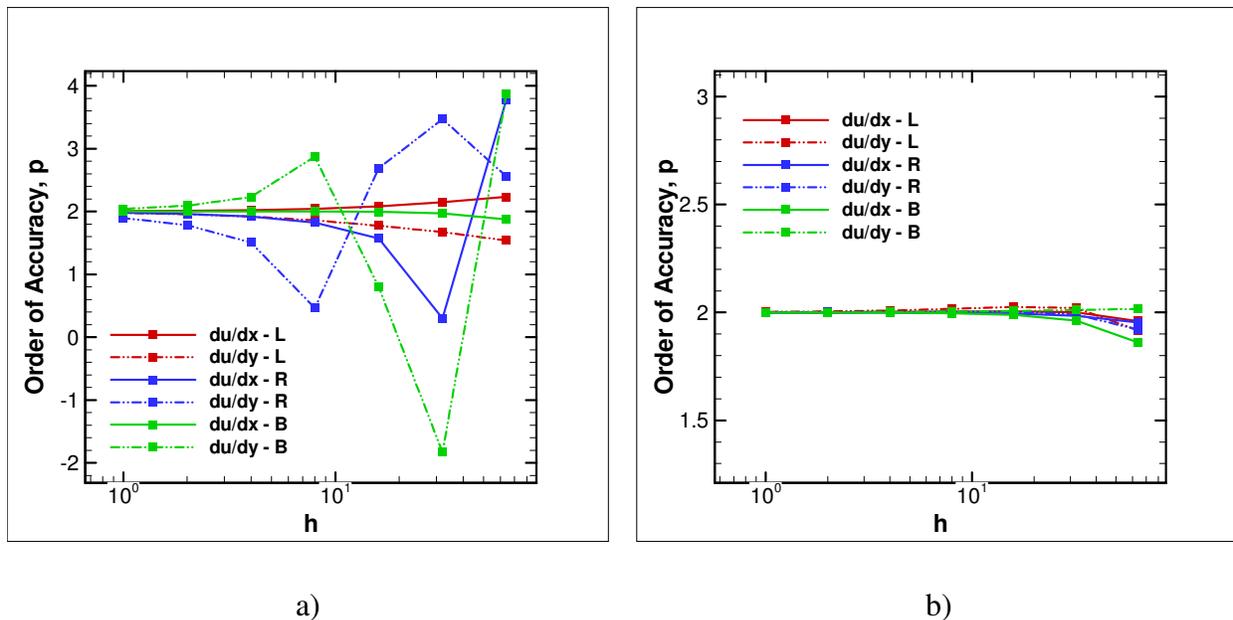
### 5.5.2 2D Unstructured Meshes

On 2D unstructured meshes, the four different numerical formulations of diffusion operators are tested. They include the node-averaged method, weighted least squares method, numerical formulation of diffusion operator used by the Loci-CHEM code, and the numerical formulation which uses a compact stencil and applied as a least squares procedure to calculate the solution gradients at the face centers which is called the Face Approximate Mapping Least

Squares (FAMLSQ).<sup>44</sup> The FAMLSQ method is tested only for unstructured meshes with curvature.

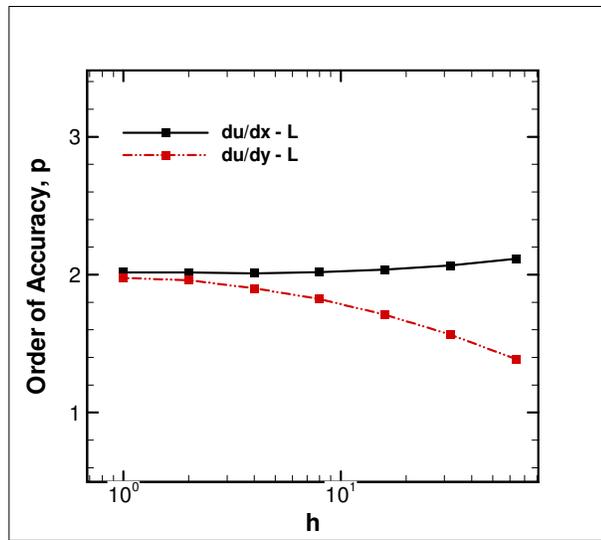
### 5.5.2.1 Mesh with Isotropic Cells

Apart from the FAMLSQ method, other numerical formulations are initially tested on the 2D unstructured mesh with isotropic cells (Figure 42a). The solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on the three cell faces of a triangle which is the central cell of the stencil considered. The mesh refinement is again achieved by fixing the cell centroid of the central cell as the focal point and shrinking the remaining domain equally in all directions. Three numerical formulations which include the node averaged method, the weighted least squares method, and the numerical formulation in the Loci-CHEM code are tested on a 2D unstructured mesh with equilateral triangles. The same analytic solution used for the 2D structured meshes is used here. The solution gradients calculated at the face centers showed second order convergence with mesh refinement for all the numerical formulations tested. The observed order of accuracy results for the solution gradients at the face centers for the three sides of the triangular cell using node averaging method and weighted least squares method are shown in Figure 48.

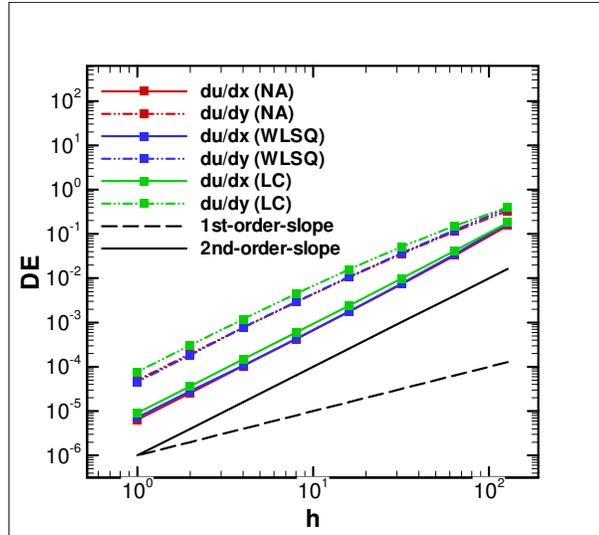


**Figure 48: Order of accuracy results for gradients at face center on a 2D unstructured mesh with isotropic triangular cells using a) node averaging method and b) weighted least squares method**

In the process of testing the diffusion operator in the Loci-CHEM code, the solution gradients are calculated at the face center of only one side of the triangular cell. The observed order of accuracy results for the solution gradients at one of the face centers of the triangular cell using numerical formulation of diffusion operator in Loci-CHEM code are shown in Figure 49. The error in the solution gradients calculated at the face center on a 2D unstructured mesh with isotropic cells using the three numerical methods (node averaging, weighted least squares and Loci-CHEM formulation) is compared and shown in Figure 50. The errors in the solution gradients are almost similar from the three methods and Loci-CHEM approach produced slightly higher errors.



**Figure 49: Order of accuracy results for gradients at face center on a 2D unstructured mesh with isotropic triangular cells using Loci-CHEM approach**

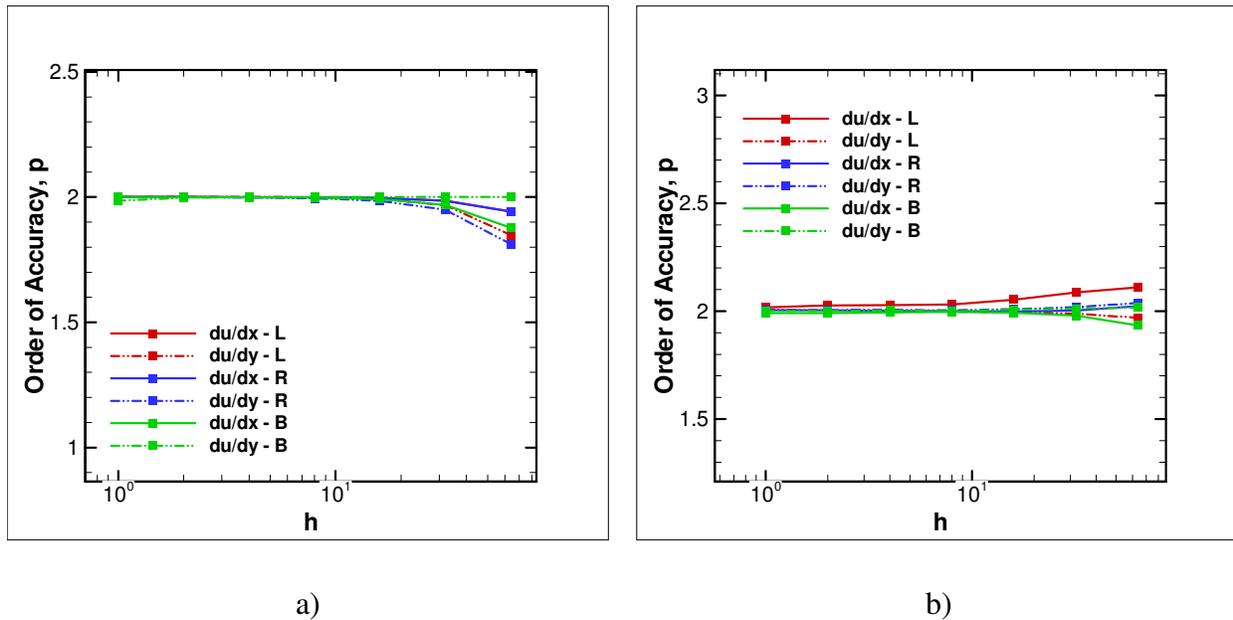


**Figure 50: Comparison of error in the gradients at the face center for the tested numerical methods on a 2D unstructured mesh with isotropic triangular cells**

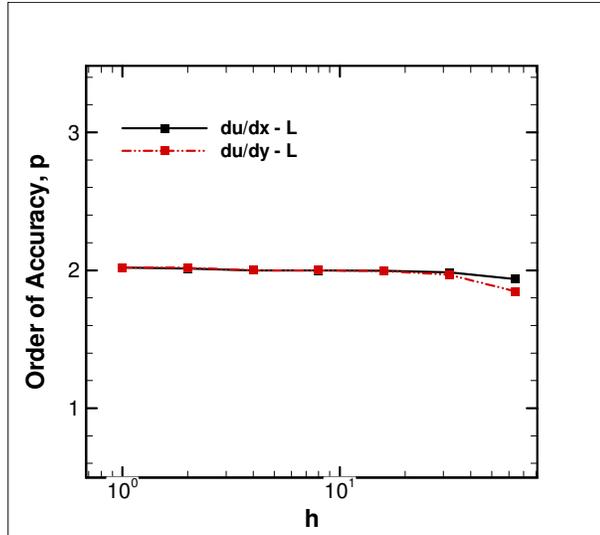
### 5.5.2.2 Skewness with Aspect Ratio

For 2D unstructured meshes, triangular cells with high aspect ratios will also be highly skewed. These two aspects of mesh quality cannot be tested separately when testing the numerical formulations on triangular cells. The aspect ratio of a triangular cell in this study is defined as the minimum value of the base to height ratio and the skewness is defined as the smallest angle made by any two sides in a triangle. The skewed meshes with a minimum angle of  $30^\circ$ ,  $10^\circ$ ,  $5^\circ$ ,  $1^\circ$ ,  $0.1^\circ$ , and  $0.01^\circ$  are used while testing the numerical formulations. The highest aspect ratio achieved using these meshes are approximately 7000:1. The numerical formulations of diffusion operators are tested on highly skewed triangular unstructured meshes. The skewed meshes considered for testing have all the cells with same skewness. The solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on the three faces of the triangular cell which is the central cell of the stencil considered. The mesh refinement is done by fixing the cell center of the central cell as the focal point and shrinking the remaining domain equally in all directions, thus maintaining the cell quality with mesh refinement. For the node averaging method and the weighted least squares method the solution gradients showed second order convergence with mesh refinement for both the numerical formulations tested on 2D unstructured triangular mesh with skewed and aspect ratio cells (Figure 42b). The change in skewness/aspect ratio of the mesh has no effect on the accuracy and a second order behavior is observed even for the highly skewed meshes. The

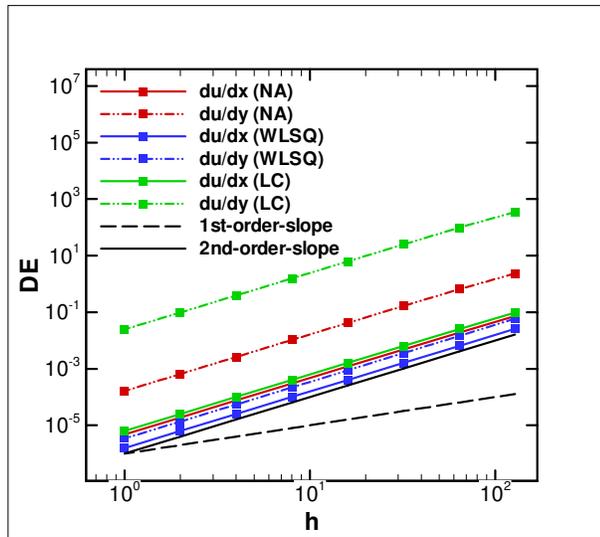
observed order of accuracy results for the solution gradients at the three face centers on a highly skewed triangular mesh with an aspect ratio of 70:1 and a minimum angle between sides of  $1^\circ$  using node averaging method and weighted least squares method are shown in Figure 51. While testing the diffusion operator in the Loci-CHEM code, the solution gradients are calculated at the face center of only one side of the triangular cell and the solution gradients showed second order convergence with mesh refinement for the Loci-CHEM formulation of the diffusion operator. The observed order of accuracy results for the solution gradients at one of the face centers of the triangular cell on a highly skewed mesh with an aspect ratio of 70:1 and a minimum angle between sides of  $1^\circ$  using numerical formulation of diffusion operator in Loci-CHEM code are shown in Figure 52. The error in the solution gradients calculated at the face center on the unstructured mesh with highly skewed and aspect ratio cells using the three numerical methods is compared and shown in Figure 53. Of the three numerical methods tested, Loci-CHEM approach produced higher errors and the weighted least squares approach produced least errors on unstructured mesh with highly skewed and aspect ratio cells.



**Figure 51: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and aspect ratio cells using a) node averaging method and b) weighted least squares method**



**Figure 52: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and aspect ratio cells using Loci-CHEM approach**



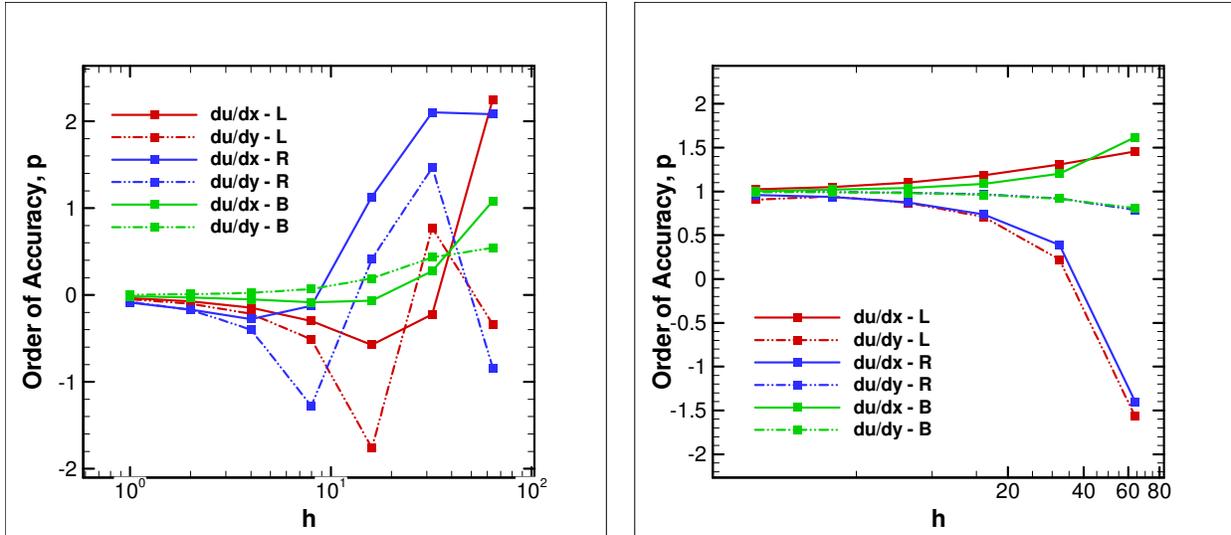
**Figure 53: Comparison of error in the gradients at the face center for the tested numerical methods on a 2D unstructured triangular mesh with skewed and aspect ratio cells**

### 5.5.2.3 Skewness with Curvature

Testing the effect of curvature for 2D unstructured meshes with triangular cells will also have skewness and possibly stretching in the radial direction associated with the 2D unstructured mesh. These mesh quality aspects are tested together while concentrating on the effect of curvature on the numerical formulations of the diffusion operators. The numerical formulations of diffusion operators are tested on meshes with high curvature which have an inner radius of 20

and the angle made by the stencil ranging from  $8^\circ$  to  $160^\circ$ . For the node averaging method and the weighted least squares methods, the solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on the three faces of the triangular cell which is the central cell of the stencil considered. The mesh refinement is done by fixing the cell center of the central cell as the focal point and shrinking the remaining domain equally in all directions, thus maintaining the cell quality with mesh refinement. For the node averaging method, it is observed that the solution gradient showed zeroth order accuracy with mesh refinement while testing on curved meshes. The errors in the calculation of the solution gradients do not decrease with mesh refinement. The inconsistency started showing up when the angle made by the stencil is above  $20^\circ$ . Certainly, the node average method is inconsistent while testing it on the 2D unstructured curvilinear mesh (Figure 42c). The observed order of accuracy results for the solution gradients at the three face centers using node averaging method on a 2D unstructured mesh with curvature (when angle made by the stencil is  $160^\circ$ ) is shown in Figure 54a.

For the weighted least squares method, it is observed that the solution gradient converged to first order accuracy with mesh refinement while testing on curved meshes. Again, the first order behavior started showing up when the angle made by the stencil is  $20^\circ$  and above. The weighted least squares method is considered to be first order accurate while testing it on the 2D unstructured curvilinear mesh. The observed order of accuracy results for the solution gradients at the three face centers weighted least squares method on a 2D unstructured mesh with curvature (when angle made by the stencil is  $160^\circ$ ) is shown in Figure 54b. So both the node averaging and weighted least squares methods fail to be second order accurate and an accuracy drop is observed while testing these methods on 2D unstructured meshes which curved boundaries.

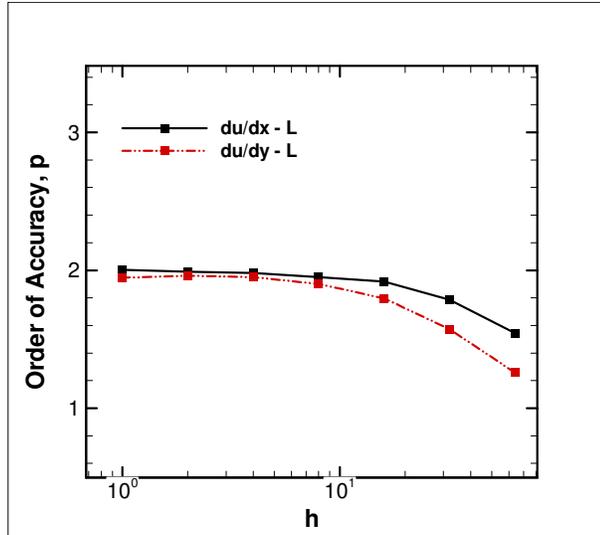


a)

b)

**Figure 54: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and curved cells using a) node averaging method and b) weighted least squares method**

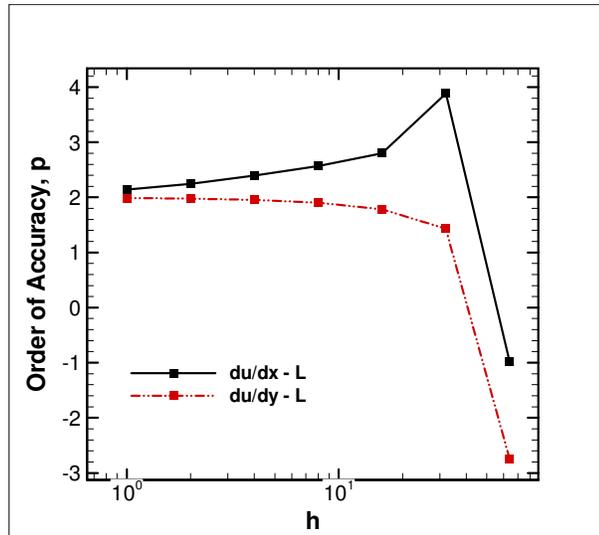
The next method tested on 2D unstructured curvilinear meshes is the Loci-CHEM formulation of diffusion operator. The solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on one of the three faces of the triangular cell which is the central cell of the stencil considered. The mesh refinement is done by fixing the face center at which the solution gradients are calculated as the focal point and shrinking the remaining domain equally in all directions. With the Loci-CHEM formulation tested on curved meshes, it is observed that the solution gradients calculated at the face center showed second order behavior with mesh refinement. The Loci-CHEM formulation is tested on a curved mesh where the angle made by the stencil is  $80^\circ$ . Also, the change in mesh curvature has no effect on the accuracy. The observed order of accuracy results for the solution gradients on the face center using Loci-CHEM formulation of diffusion operator on a 2D unstructured mesh with curvature is shown in Figure 55.



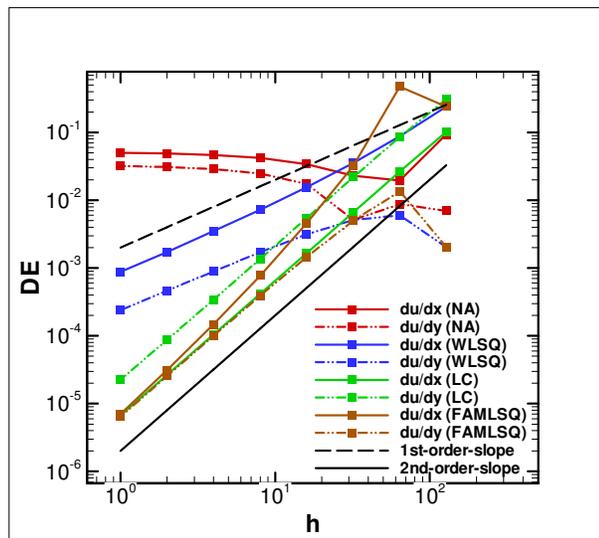
**Figure 55: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and curved cells using Loci-CHEM approach**

The other method tested on 2D unstructured curvilinear meshes is the Face Approximate Mapping Least Squares (FAMLSQ) method which uses a compact stencil to formulate the solution gradients at the face center. The testing is done in a slightly different way for this formulation where the mesh is shrunk such that the radial cell size and the angle made by the stencil are reduced by a factor of two for every refinement instead of refinement in the x and y coordinate directions. Systematic mesh refinement is satisfied even for this kind of refinement where the quality of the mesh improves instead of staying constant with every refinement. Using this formulation, the solution gradients  $\partial u/\partial x$  and  $\partial u/\partial y$  are calculated on one of the three faces of the triangular cell which is the central cell of the stencil considered. This formulation used for calculating the solution gradients at the face center showed second order accuracy with mesh refinement. The observed order of accuracy results for the solution gradients on the face center using the FAMLSQ formulation with a compact stencil on a 2D unstructured mesh with curvature is shown in Figure 56. The error in the solution gradients calculated at the face center on the unstructured mesh with highly skewed and curved cells using the four numerical methods tested is compared and shown in Figure 57. From the plot, the errors in the solution gradients using node averaging method do not reduce with mesh refinement and the errors in the solution gradients using weighted least squares method reduce at first order with mesh refinement. Loci-CHEM approach and FAMLSQ approach produced similar errors and the errors reduced at

second order with mesh refinement for both the approaches on 2D unstructured mesh with skewed and curved cells.



**Figure 56: Order of accuracy results for gradients at face center on a 2D unstructured triangular mesh with skewed and curved cells using FAML SQ approach**



**Figure 57: Comparison of error in the gradients at the face center for the tested numerical methods on a 2D unstructured triangular mesh with skewed and curved cells**

From the tests done on 2D unstructured meshes with different mesh qualities, it can be concluded that the node averaging method and the weighted least squares method are second order accurate on skewed unstructured mesh with aspect ratio cells, but their accuracy drops

when the formulations are tested on unstructured curvilinear meshes. The numerical formulation of diffusion operator in Loci-CHEM code and the FAMLSQ method using compact stencil are second order accurate on unstructured meshes with all mesh qualities and can be considered as consistent formulations for calculation of diffusion terms on 2D unstructured meshes from this analysis. This approach of testing the numerical formulation can be extended to 3D, since the shrinking aspect of mesh refinement for calculating the order of accuracy from different mesh levels can be applied easily in 3D saving significant computational power and time.

## 6. Conclusions

### 6.1 Summary of Results

A detailed code verification study of an unstructured finite volume code was performed. The finite volume code, Loci-CHEM, is used for reactive flow simulations of rocket engines. The MMS was used to generate exact solutions for complex governing equations. Systematic mesh refinement which is important for code verification was explained. Different options in the finite volume CFD code were verified which included the baseline steady-state governing equations, transport models, different boundary condition options, turbulence models, and time accuracy of unsteady flows. All the options were determined to be verified when the observed order of accuracy matched the formal order on the 2D hybrid and 3D hybrid mesh which contained all cell topologies (triangular, quadrilateral, hexahedral, tetrahedral, and prismatic cells). When the verification process failed on any one of these complex hybrid meshes, then simpler meshes were considered to isolate the problem. Coding mistakes, algorithm inconsistencies, and mesh quality sensitivities uncovered during code verification were presented. By testing the finite volume code on different meshes, the effect of cell quality and cell topology on the order of accuracy of the code was also assessed.

A survey of diffusion operators for compressible CFD solvers was conducted to understand the different formulation procedures for diffusion terms. Some of the diffusion operators extensively used in CFD solvers and some of the recently developed diffusion operators were evaluated by testing them on different structured and unstructured mesh topologies with different mesh quality aspects for accuracy. A patch-wise version of the Method of Manufactured Solutions for code verification was used for the evaluation of the diffusion operators. The numerical formulations were tested on 2D structured and 2D unstructured meshes with different cell quality aspects like skewness, aspect ratio, stretching and curvature. During the testing of the numerical formulations, the observed order of accuracy calculated from multiple mesh levels was compared with the formal order of accuracy. Multiple mesh levels were achieved by shrinking the mesh over a focal point rather than a conventional mesh refinement. All the numerical formulations tested on 2D structured meshes were accurate and consistent as they showed second order convergence on every type of mesh considered. All the numerical formulations tested on 2D unstructured meshes were accurate when tested on skewed

mesh with aspect ratio cells, but the node averaging method and weighted least squares method were not consistent as their accuracy dropped to less than second order while testing on 2D unstructured curved meshes. The diffusion operator formulation in the Loci-CHEM CFD code and the FAMLSQ approach using a compact stencil were found to be consistent and accurate diffusion operators while testing the formulations on 2D unstructured meshes as they achieved second order convergence on 2D unstructured meshes with all cell quality aspects.

## **6.2 Main Contributions of this Work**

There are some new contributions to the field of Computational Fluid Dynamics from this research work.

1. The requirements for generating 3D unstructured meshes with systematic mesh refinement were identified and FORTRAN codes were developed to generate multiple mesh levels for 3D unstructured meshes and 3D hybrid meshes for code verification purposes. To our knowledge, present commercial mesh generation software cannot generate different mesh levels of 3D unstructured meshes or 3D hybrid meshes which satisfy systematic mesh refinement.
2. The effect of mesh quality on numerical accuracy was studied by testing governing equations and different numerical formulations on different mesh topologies with varying mesh qualities.
3. A unique way of verifying the time accuracy of unsteady flows was identified. In the process of verifying the spatial and time terms in a governing equation simultaneously, a method for the selection of spatial and temporal step sizes such that the discretization error in both spatial and temporal terms are of similar magnitudes was developed.
4. The shrinking of meshes of over a focal point for generating multiple mesh levels was used for testing of different formulations without actually running the code. This procedure was applied successfully to decrease the computational effort in code verification.
5. Through the code verification work discussed herein, Loci-CHEM is currently the most verified compressible CFD code. We know of no other compressible CFD codes presently which are verified as extensively as Loci-CHEM.

### **6.3 Future Work and Recommendations**

Some future recommendations are mentioned here. One of them is generating Manufactured Solutions for verification of boundary conditions when the boundaries have curvature. It is required to come up with an intelligent way to design the Manufactured Solutions for some of the boundary condition options when applied to curved boundaries to satisfy constraints on normal derivatives. Another recommendation is the application of verification of different numerical formulations in 3D using the shrinking technique for generating multiple mesh levels which makes it easier in terms of computational resources for code verification purposes. In this case, the numerical formulations are tested without running the code.

## References

- <sup>1</sup> Hatton, L., "The T Experiments: Errors in Scientific Software," *IEEE Computational Science and Engineering*, Vol. 4, No. 2, 1997, pp. 27-38.
- <sup>2</sup> Abanto, J., Pelletier, D., Garon, A., Trepanier, J. and Reggio, M., "Verification of some Commercial CFD Codes on Atypical CFD Problems," AIAA-2005-682, 43<sup>rd</sup> AIAA Fluid Dynamics Conference and Exhibit, January 2005.
- <sup>3</sup> Kleb, B. and Wood, B., "CFD: A Castle in the Sand?," AIAA-2004-2627, 36<sup>th</sup> AIAA Fluid Dynamics Conference and Exhibit, June 2004.
- <sup>4</sup> Roache, P. J., "Verification of Codes and Calculations," *AIAA Journal*, Vol. 36, No. 5, May 1998, pp. 696-702.
- <sup>5</sup> Roache, P. J., *Verification and Validation in Computational Science and Engineering*, Albuquerque, NM, Hermosa Publishers, 1998.
- <sup>6</sup> Knupp, P. and Salari, K., *Verification of Computer Codes in Computational Science and Engineering*, Boca Raton, FL, Chapman & Hall/CRC, 2002.
- <sup>7</sup> Oberkampf, W. L. and Roy, C. J., *Verification and Validation of Scientific Computing*, Cambridge University Press, Cambridge, 2010.
- <sup>8</sup> Roache, P. J. and Steinberg, S., "Symbolic Manipulation and Computational Fluid Dynamics," *AIAA Journal*, Vol. 22, No. 10, 1984, pp. 1390-1394.
- <sup>9</sup> Roache, P. J., "Code Verification by the Method of Manufactured Solutions," *Journal of Fluids Engineering*, Vol. 124, No. 1, 2002, pp. 4-10.
- <sup>10</sup> Roy C. J., Nelson, C. C., Smith, T. M., and Ober, C. C., "Verification of Euler / Navier-Stokes Codes using the Method of Manufactured Solutions," *International Journal for Numerical Methods in Fluids*, Vol. 44, No. 6, 2004, pp. 599-620.
- <sup>11</sup> Roy, C. J., Smith, T. M., Ober, C. C., "Verification of a Compressible CFD Code Using the Method of Manufactured Solutions," 32<sup>nd</sup> AIAA Fluid Dynamics Conference and Exhibit, AIAA-2002-3110, St. Louis, Missouri, 24-26 June 2002.
- <sup>12</sup> Nelson, C. C. and Roy, C. J., "Verification of the Wind-US CFD Code Using the Method of Manufactured Solutions," 42<sup>nd</sup> AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2004.

- <sup>13</sup> Smith, T. M., Ober, C. C., Lorber, A. A., “SIERRA/Premo – A New General Purpose Compressible Flow Simulation Code, AIAA Paper 2002-3292, 2002.
- <sup>14</sup> Nelson, C. C. and Power, G. D., “CHSSI Project CFD-7: The NPARC Alliance Flow Simulation System,” AIAA Paper 2001-0594, 2001.
- <sup>15</sup> Hebert, S. and Luke, E., “Honey, I Shrunk the Grids! A New Approach to CFD Verification,” AIAA Paper 2005-0685, 43<sup>rd</sup> AIAA Aerospace Sciences Meeting and Exhibit, January, Reno, Nevada, 2005.
- <sup>16</sup> Luke, E. A., Tong, X.-L., Wu, J., and Cinnella, P., “CHEM 2: A Finite-Rate Viscous Chemistry Solver – The User Guide,” Tech. Rep. MSSU-COE-ERC-04-07, Mississippi State University, 2004.
- <sup>17</sup> Diskin, B., and Thomas, J. L., “Accuracy Analysis for Mixed-Element Finite Volume Discretization Schemes,” National Institute of Aerospace Technical Report TR 2007-8, Hampton, VA, 2007.
- <sup>18</sup> Thomas, J. L., Diskin, B., and Rumsey, C. L., “Towards Verification of Unstructured-Grid Solvers,” *AIAA Journal*, Vol. 46, No. 12, pp. 3070-3079, 2008.
- <sup>19</sup> Bond, R. B., Knupp, P. M., and Ober, C. C., “A Manufactured Solution for Verifying CFD Boundary Conditions, Part 2,” 43<sup>rd</sup> AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January, 2005
- <sup>20</sup> Bond, R. B., Ober, C. C., Knupp, P. M., and Bova, S. W., “Manufactured Solution for Computational Fluid Dynamics Boundary Condition Verification,” *AIAA Journal*, Vol. 45, No. 9, 2007, pp. 2224-2236.
- <sup>21</sup> Pelletier, D. and Roache, P. J., “CFD Code Verification and the Method of the Manufactured Solutions,” 10<sup>th</sup> Annual Conference of the CFD Society of Canada, Windsor, Ontario, Canada, June 2002.
- <sup>22</sup> Pelletier, D., Turgeon, E., and Tremblay, D., “Verification and Validation of Impinging Round Jet Simulations using an Adaptive FEM,” *International Journal for Numerical Methods in Fluids*, Vol. 44, 2004, pp. 737-763.
- <sup>23</sup> Eca, L. and Hoekstra, M., “An Introduction to CFD Code Verification Including Eddy-Viscosity Models,” European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006, Wesseling, P., Onate, E., and Periaux, J. (eds.), 2006.

- <sup>24</sup> Eca, L. and Hoekstra, M., “Verification of Turbulence Models with a Manufactured Solution,” European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006, Wesseling, P., Onate, E., and Periaux, J. (eds.), 2006.
- <sup>25</sup> Eca, L., Hoekstra, M., Hay, A., and Pelletier, D., “On the Construction of Manufactured Solutions for One and Two-Equation Eddy-Viscosity Models,” *International Journal for Numerical Methods in Fluids*, Vol. 54, No. 2, May 2007, pp. 119-154.
- <sup>26</sup> Spalart, P. R., and Allmaras, S. R., “A One-Equation Turbulence Model for Aerodynamic Flows,” AIAA Paper 92-0439, 1992.
- <sup>27</sup> Menter, F. R., “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications,” *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1598-1605.
- <sup>28</sup> Kok, J. C., “Resolving the Dependence on Free-Stream Values for the  $k-\omega$  Turbulence Model,” NLR-TP-00205, 1999.
- <sup>29</sup> Roy, C. J., Tendeau, E., Veluri, S. P., Rifki, R., Luke E. A., and Hebert, S., “Verification of RANS Turbulence Models in Loci-CHEM using Method of Manufactured Solutions,” AIAA-2007-4203, 18<sup>th</sup> AIAA Computational Fluid Dynamics Conference, June 2007.
- <sup>30</sup> Salas, M. D., “Some Observations on Grid Convergence,” *Computers and Fluids*, 35, March 2006, pp. 688-692.
- <sup>31</sup> Etienne, S., Garon, A., Pelletier, D., “Code verification for unsteady flow simulations with high order time-stepping schemes,” 47<sup>th</sup> AIAA Aerospace Sciences Meeting, AIAA 2009-169, Orlando, Florida, January, 2009.
- <sup>32</sup> Tremblay, D., Etienne, S., and Pelletier, D., “Code Verification and the Method of Manufactured Solutions for Fluid-Structure Interaction Problems,” 36<sup>th</sup> AIAA Fluid Dynamics Conference and Exhibit, AIAA-2006-3218, San Francisco, California, 5-8 June, 2006.
- <sup>33</sup> Knight, D., “A Fully Implicit Navier-Stokes Algorithm Using an Unstructured Grid and Flux Difference Splitting,” AIAA-93-0875, 31<sup>st</sup> Aerospace Sciences Meeting and Exhibit, January 1993.
- <sup>34</sup> Ollivier-Gooch, C. F. and Van Altena, M., “A High-Order-Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation,” *Journal of Computational Physics*, 181, 2002, 729-752.

- <sup>35</sup> Vaassen, J. M., Vigneron, D., Essers, J. A., “An implicit high order finite volume scheme for the solution of the 3D Navier-Stokes equations with the new discretization of diffusive terms,” *Journal of Computational and Applied Mathematics*, 2008.
- <sup>36</sup> Coirier, W. J., An adaptively-refined, Cartesian, cell based scheme for the Euler and Navier-Stokes equations, Ph.D. Thesis, University of Michigan, 1994.
- <sup>37</sup> Grismer, M. J., Strang, W. Z., Tomoro, R. F., and Witzeman, F. C., “Cobalt: A parallel, implicit, unstructured Euler/Navier-Stokes solver,” *Advances in Engineering Software*, Vol. 29, No. 3-6, pp. 365-373, 1998.
- <sup>38</sup> Veluri, S. P., Roy, C. J., Hebert, S., and Luke, E. A., “Verification of Loci-CHEM CFD Code using the Method of Manufactured Solutions,” AIAA-2008-661, 46<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit, January 2008.
- <sup>39</sup> Luke, E., “On Robust and Accurate Arbitrary Polytope CFD Solvers (Invited),” AIAA-2007-3956, 18<sup>th</sup> AIAA Computational Fluid Dynamics Conference, Miami, Florida, June 2007.
- <sup>40</sup> Mavriplis, D. J., “Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes,” 16<sup>th</sup> AIAA Computational Fluid Dynamics Conference, AIAA-2003-3986, Orlando, Florida, 23-26 June, 2003.
- <sup>41</sup> Delanaye, M., and Essers, J. A., “Quadratic-Reconstruction Finite Volume Scheme for Compressible Flows on Unstructured Adaptive Grids,” *AIAA Journal*, Vol. 35, No. 4, April 1997.
- <sup>42</sup> Haselbacher, A., McGuirk, J. J., and Page, G. J., “Finite Volume Discretization Aspects for Viscous Flows on Mixed Unstructured Grids,” *AIAA Journal*, Vol. 37, No. 2, February 1999.
- <sup>43</sup> Haselbacher, A., and Blazek, J., “Accurate and Efficient Discretization of Navier-Stokes Equations on Mixed Grids,” *AIAA Journal*, Vol. 38, No. 11, November 2000.
- <sup>44</sup> Diskin, B., Thomas, J. L., Nielsen, E. J., Nishikawa, H., White, J. A., “Comparison of node-centered and cell-centered unstructured finite-volume discretizations. Part I: viscous fluxes,” 47<sup>th</sup> AIAA Aerospace Sciences Meeting, AIAA 2009-597, Orlando, Florida, January, 2009.

- <sup>45</sup> Barth, T. J., “Numerical aspects of computing high Reynolds number flow on unstructured meshes, AIAA Paper 91-0721, 29<sup>th</sup> AIAA Aerospace Sciences Meeting, Reno, NV, January 1991.
- <sup>46</sup> Holmes, D. G., and Connell, S. D., “Solution of the 2D Navier-Stokes equations on unstructured adaptive grids, AIAA Paper 89-1392, 9<sup>th</sup> AIAA CFD conference, Washington, DC, June 1989.
- <sup>47</sup> Sun, Y., Wang, Z. J., Liu, Y., “Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow,” *Journal of Computational Physics*, 215, 2006, pp. 41-58.
- <sup>48</sup> Kannan, R., Sun, Y., Wang, Z. J., “A study of Viscous Flux Formulations for an Implicit P-Multigrid Spectral Volume Navier-Stokes Solver,” 46<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit, January 2008, AIAA-2008-783.
- <sup>49</sup> Bassi, S., Rebay, S., “A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations,” *Journal of Computational Physics*, 131, 1997, pp. 267-279.
- <sup>50</sup> Fidkowski, K. J., Darmofal, D. L., “A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations,” *Journal of Computational Physics*, 225, 2007, pp. 1653-1672.
- <sup>51</sup> DuFort, E. C. and Frankel, S. P., “Stability Conditions in the Numerical Treatment of Parabolic Differential Equations”, *Mathematical Tables and Other Aids to Computation*, Vol. 7, 1953, pp. 135-152.
- <sup>52</sup> Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Taylor and Francis, second edition, 2004.
- <sup>53</sup> Roy, C. J., “Grid Convergence Error Analysis for Mixed-Order Numerical Schemes,” *AIAA Journal*, Vol. 41, No. 4, April 2003, pp. 595-604.
- <sup>54</sup> Carpenter, M. H., Atkins, H. L., and Singh, D. J. (1994), “Characteristic and Finite-Wave Shock-Fitting Boundary Conditions for Chebyshev Methods,” *Transition, Turbulence, and Combustion*, edited by M. Y. Hussaini, T. B. Gatski, and T. L. Jackson, Vol. 2, Kluwer Academic, Norwell, MA, pp. 301-312.
- <sup>55</sup> Lyubimov, A. N., and Rusanov, V. V. (1973), “Gas Flows Past Blunt Bodies, Part II: Tables of the Gasdynamic Functions,” NASA TT F-715.

- <sup>56</sup> Trucano, T. G., Pilch, M. M., Oberkampf, W. L., “On the Role of Code Comparisons in Verification and Validation,” SAND 2003-2752, Sandia National Laboratories, Albuquerque, New Mexico, 2003.
- <sup>57</sup> Pointwise Inc., GRIDGEN v.15, User Manual.
- <sup>58</sup> Luke, E. A., and Cinnella, P., “Numerical Simulations of Mixtures of Fluids Using Upwind Algorithms,” *Computers and Fluids*, Volume 36, December 2007, pp. 1547-1566.
- <sup>59</sup> Zang, Y., and Luke, E., “Concurrent Composition Using Loci,” *Computing in Science and Engineering*, Volume 11, Issue 3 (May 2009), pp. 27-35.
- <sup>60</sup> Luke, E., and George, T., “Loci: A Rule-Based Framework for Parallel Multidisciplinary Simulation Synthesis,” *Journal of Functional Programming*, Volume 15, Issue 03, 2005, pp. 477-502, Cambridge University Press.
- <sup>61</sup> Wilcox, D. C., *Turbulence Modeling for CFD*, 2<sup>nd</sup> ed., DCW Industries, La Canada, CA, 1998.
- <sup>62</sup> Strang, W., Tomaro, R., and Grismer, M., “The defining Methods of Cobalt60: A parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver,” AIAA paper 1999-0786, 1999.
- <sup>63</sup> Luke, E., personal communication, April 2010.
- <sup>64</sup> Kamm, J. R., Rider, W. J., and Brock, J. S., “Combined Space and Time Convergence Analyses of a Compressible Flow Algorithm,” AIAA-2003-4241, 16<sup>th</sup> AIAA Computational Fluid Dynamics Conference, June 2003.
- <sup>65</sup> Holmes, D. G., and Connell, S. D., “Solution of Navier-Stokes equations on unstructured adaptive grids,” AIAA-1989-1392, 9<sup>th</sup> AIAA CFD Conference, Washington, DC, June 1989.
- <sup>66</sup> Rausch, R. D., Batina, J. T., and Yang, H. T., “Spatial adaptation procedures on unstructured meshes for accurate unsteady aerodynamic flow computation,” AIAA-1989-1392, 9<sup>th</sup> AIAA CFD Conference, Washington, DC, June 1989.
- <sup>67</sup> Frink, N. T., “Assessment of an unstructured-grid method for predicting 3-D turbulent viscous flows,” AIAA-1996-0292, 34<sup>th</sup> AIAA Aerospace Science Meeting and Exhibit, Reno, NV, January, 1996.

- <sup>68</sup> Veluri, S. P., Roy, C. J., and Luke, E. A., “Comprehensive Code Verification for an Unstructured Finite Volume CFD Code,” AIAA-2010-127, 48<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit, Orlando, Florida, January 2010.
- <sup>69</sup> Oberkampf, W. L., Blottner, F. G., Aeschliman, D. P., “Methodology for Computational Fluid Dynamics Code Verification/Validation,” 26<sup>th</sup> AIAA Fluid Dynamics Conference, June 19-22, San Diego, CA, 1995.
- <sup>70</sup> Batdorf, M., Freitag, L. A., Ollivier-Gooch, C., “Computational Study of the Effect of Unstructured Mesh Quality on Solution Efficiency,” American Institute of Aeronautics and Astronautics, 1997.
- <sup>71</sup> Turkel, E., Tel-Aviv, Yaniv, S., Landau, U., “Accuracy of Schemes for the Euler Equations with Non-Uniform meshes,” 24<sup>th</sup> AIAA Aerospace Sciences Meeting, Reno, NV, January, 1986.
- <sup>72</sup> Smith, T. M., Barone, M. F., Bond, R. B., Lorber, A. A., and Baur, D. G., “Comparison of Reconstruction Techniques for Unstructured Mesh Vertex Centered Finite Volume Schemes,” 18<sup>th</sup> Computational Fluid Dynamics Conference, AIAA-2007-3958, Miami, Florida, 25-28 June, 2007.
- <sup>73</sup> Gu, X., Schock, H. J., Shih, T. I-P., Hernandez, E. C., Chu, D., Keller, P. S., and Sun, R. L., “Grid Quality Measures for Structured and Unstructured Meshes,” 39<sup>th</sup> AIAA Aerospace Sciences Meeting, AIAA-2001-0652, Reno, Nevada, 8-11 January, 2001.
- <sup>74</sup> Spiegel, S. C., Stefanski, D. L., Luo, H., and Edwards, R. E., “A Cell-centered Finite Volume Method for Chemically Reacting Flows on Hybrid Grids,” 48<sup>th</sup> AIAA Aerospace Sciences Meeting, AIAA-2010-1083, Orlando, Florida, 4-7 January, 2010.
- <sup>75</sup> Edwards, J. R., “A Low-Diffusion Flux-Splitting Scheme for Navier-Stokes Calculations,” *Computers and Fluids*, Vol. 26, No. 6, pp. 635-659, 1997.
- <sup>76</sup> Edwards, J. R., and Liou, M. S., “Low-Diffusion Flux-Splitting Methods for Flows at All Speeds,” *AIAA Journal*, Vol. 36, pp. 1610-1618, 1998.

# Appendix A

A summary of several options verified in the finite volume Loci-CHEM CFD code is shown in Figure A1.

	2D Structured			2D Unstructured		2D Hybrid	3D Structured		3D Unstructured		3D Hybrid	3D Hybrid
	Rectangular Cartesian	Stretched Cartesian	Curvilinear	Unidirectional	Alternate Diagonal		Isotropic	Curvilinear	All	Isotropic	Skewed	
Euler (supersonic)	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Navier-Stokes	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Navier-Stokes (Highly Skewed Mesh)	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Thermally Perfect Thermodynamic Model	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Sutherland	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Wilson k-omega 1988 Model	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
k-epsilon Model	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Dirichlet	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Supersonic Inflow	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Farfield	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
No-Slip Adiabatic Wall	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
No-Slip Isothermal Wall	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Slip Wall (Euler)	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Slip Wall (Navier-Stokes)	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Isentropic Inflow	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Outflow	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Extrapolate	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Unsteady Time Marching	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green

Green	Verified 2 <sup>nd</sup> Order
Grey	(Assumed Verified) -
Red	Non-Convergence

† Cases that were not actually run, but that are assumed to be verified because a more complex element type was verified

Figure A1: Verification of different options in the finite volume Loci-CHEM CFD code

## Appendix B

After the Navier-Stokes equations were verified to be second order accurate on a 3D skewed hybrid mesh but showed a reduced accuracy on the 3D highly skewed hybrid mesh, the mesh quality was quantified with some parameters. The cell quality aspects were quantified and tabulated for the 65×65×65 meshes. These numbers which quantify the cell quality did not change much for different mesh levels and improved with mesh refinement for a particular mesh type which satisfied systematic mesh refinement. The cell quality is quantified for the 3D skewed hybrid mesh and the 3D highly skewed hybrid mesh are shown in Table B1.

**Table B1. Quantification of cell quality**

	3D skewed hybrid mesh	3D highly skewed hybrid mesh
Mesh Size	65×65×65	65×65×65
Maximum Cell Angle	37.4523	58.1545
Maximum Twist	0.0100871	0.00480552
Maximum Shear Twist	0.000306531	0.00100128

In the above table, maximum cell angle is defined as the maximum cell to face angle and it is the angle between the face normal and the cell centroids. This angle provides an indication of mesh isotropy. Lower the maximum cell angle better is the mesh quality. For non-triangular faces, the faces can be non-planar i.e. twisted. The twist metric is the measure of the non-planar component of the face geometry. Twist is the deviation of the face in the direction away from the plane of the face and shear twist is the deviation of the face in the plane of the face itself. A value of 0.1 indicates that the face geometry deviates from the planar description by 10 percent.

## Appendix C

For boundary conditions and turbulence models, the constants and the trigonometric functions used in the Manufactured Solutions in 2D and 3D are in Table C1 and Table C2, respectively.

**Table C1. Constants for 2D Manufactured Solutions**

Equation, $\phi$	$\phi_0$	$\phi_x$	$\phi_y$	$\phi_{xy}$
$\rho$ (kg/m <sup>3</sup> )	1	0.15	-0.1	0.08
u (m/s)	70	7	-8	5.5
v (m/s)	90	-5	10	-11
p (N/m <sup>2</sup> )	$1 \times 10^5$	$0.2 \times 10^5$	$0.175 \times 10^5$	$-0.25 \times 10^5$
T (K)	350	10	-30	15
k (m <sup>2</sup> /s <sup>2</sup> )	780	160	-120	80
$\omega$ (1/s)	150	-30	22.5	40
Equation, $\phi$		$a_{\phi_x}$	$a_{\phi_y}$	$a_{\phi_{xy}}$
$\rho$ (kg/m <sup>3</sup> )		0.75	1	1.25
u (m/s)		1.5	1.5	0.6
v (m/s)		1.5	1	0.9
p (N/m <sup>2</sup> )		1	1.25	0.75
T (K)		0.75	1.25	1.25
k (m <sup>2</sup> /s <sup>2</sup> )		0.65	0.7	0.8
$\omega$ (1/s)		0.75	0.875	0.6
Equation, $\phi$		$f_s(x)$	$f_s(y)$	$f_s(xy)$
$\rho$ (kg/m <sup>3</sup> )		cos	sin	cos
u (m/s)		sin	cos	cos
v (m/s)		sin	cos	cos
p (N/m <sup>2</sup> )		cos	sin	sin
T (K)		cos	sin	cos
k (m <sup>2</sup> /s <sup>2</sup> )		cos	sin	cos
$\omega$ (1/s)		cos	sin	cos

**Table C2. Constants for 3D Manufactured Solutions**

Equation, $\phi$	$\phi_0$	$\phi_x$	$\phi_y$	$\phi_z$	$\phi_{xy}$	$\phi_{yz}$	$\phi_{zx}$
$\rho$ (kg/m <sup>3</sup> )	1	0.15	-0.1	0.1	0.08	0.05	0.12
u (m/s)	70	7	-15	-10	7	4	-4
v (m/s)	90	-5	10	5	-11	-5	5
w (m/s)	80	-10	10	12	-12	11	5
p (N/m <sup>2</sup> )	$1 \times 10^5$	$0.2 \times 10^5$	$0.5 \times 10^5$	$0.2 \times 10^5$	$-0.25 \times 10^5$	$-0.1 \times 10^5$	$0.1 \times 10^5$
T (K)	350	10	-30	20	10	-12	15
k (m <sup>2</sup> /s <sup>2</sup> )	780	160	-120	80	80	60	-70
$\omega$ (1/s)	150	-30	22.5	20	40	-15	25
Equation, $\phi$		$a_{\phi x}$	$a_{\phi y}$	$a_{\phi z}$	$a_{\phi xy}$	$a_{\phi yz}$	$a_{\phi zx}$
$\rho$ (kg/m <sup>3</sup> )		0.75	0.45	0.8	0.65	0.75	0.5
u (m/s)		0.5	0.85	0.4	0.6	0.8	0.9
v (m/s)		0.8	0.8	0.5	0.9	0.4	0.6
w (m/s)		0.85	0.9	0.5	0.4	0.8	0.75
p (N/m <sup>2</sup> )		0.4	0.45	0.85	0.75	0.7	0.8
T (K)		0.75	1.25	0.5	0.65	0.75	0.5
k (m <sup>2</sup> /s <sup>2</sup> )		0.65	0.7	0.8	0.8	0.85	0.6
$\omega$ (1/s)		0.75	0.875	0.65	0.6	0.75	0.8
Equation, $\phi$		$f_s(x)$	$f_s(y)$	$f_s(z)$	$f_s(xy)$	$f_s(yz)$	$f_s(zx)$
$\rho$ (kg/m <sup>3</sup> )		cos	sin	sin	cos	sin	cos
u (m/s)		sin	cos	cos	cos	sin	Cos
v (m/s)		sin	cos	cos	cos	sin	cos
w (m/s)		cos	sin	cos	sin	sin	cos
p (N/m <sup>2</sup> )		cos	cos	sin	cos	sin	cos
T (K)		cos	sin	sin	cos	sin	cos
k (m <sup>2</sup> /s <sup>2</sup> )		cos	cos	sin	cos	cos	sin
$\omega$ (1/s)		cos	cos	sin	cos	cos	sin