

8-1-2016

The Validation and Verification of LES Modeling Using KIVA

David Fyda

University of Nevada, Las Vegas, fydad@unlv.nevada.edu

Follow this and additional works at: <https://digitalscholarship.unlv.edu/thesesdissertations>

 Part of the [Aerospace Engineering Commons](#), and the [Mechanical Engineering Commons](#)

Repository Citation

Fyda, David, "The Validation and Verification of LES Modeling Using KIVA" (2016). *UNLV Theses, Dissertations, Professional Papers, and Capstones*. 2778.

<https://digitalscholarship.unlv.edu/thesesdissertations/2778>

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Theses, Dissertations, Professional Papers, and Capstones by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

THE VALIDATION AND VERIFICATION OF LES MODELING USING KIVA

By

David Fyda

Bachelor of Science in Mechanical Engineering

University of Nevada Las Vegas

2012

A thesis submitted in partial fulfillment

of the requirements for the

Master of Science in Aerospace Engineering

Department of Mechanical Engineering

Howard R. Hughes College of Engineering

The Graduate College

University of Nevada, Las Vegas

August 2016



Thesis Approval

The Graduate College
The University of Nevada, Las Vegas

July 15, 2016

This thesis prepared by

David Fyda

entitled

The Validation and Verification of Les Modeling Using Kiva

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Aerospace Engineering
Department of Mechanical Engineering

Darrell Pepper, Ph.D.
Examination Committee Chair

Kathryn Hausbeck Korgan, Ph.D.
Graduate College Interim Dean

Alexander Barzilov, Ph.D.
Examination Committee Member

William Culbreth, Ph.D.
Examination Committee Member

Laxmi Gewali, Ph.D.
Graduate College Faculty Representative

Abstract

In recent years, the use of large-eddy simulation (LES) has grown into new research methods. LES is preferred when compared to Reynold's Averaged Navier Stokes (RANS), which separates velocity components into steady and fluctuating components. While RANS is relatively easy to implement, it does not fully resolve the range of turbulence eddies and has limitations that make it inaccurate in many practical circumstances. Direct numerical simulation, DNS, fully resolves all turbulent eddies to the smallest grid scale, but requires an extremely fine grid. This makes it computationally impractical to use as the computational power required to solve even the simplest case is severely high. In LES, the large eddies are resolved while the small eddies are modelled. This can be advantageous as we lower the computational resources required for solving the flow while still maintaining accuracy.

The use of RANS as well as DNS make them highly difficult in solving issues involving combustion processes. LES models tend to be simpler and require fewer adjustments when applied to a wide range of flows. In addition to turbulence, LES can also handle species transport and chemical reactions typically found in engine configurations, and makes it a very suitable choice.

LES has many different approaches, including the popular Smagorinsky model. The main problem with the elementary Smargoinisky approach is that a model parameter, C_d , is assumed constant over the entire region, which is generally not true for turbulent flows. Making the model dynamic to localize the parameter, we still have problems of the parameter varying too much (as much as ten times the mean) and the eddy viscosity becoming negative, resulting in instability. The Vreman LES model, as employed in this work, performs as well if not better than the dynamic Smagorinsky model, and does not require ad hoc procedures, including issues involving clipping.

In this study, the LES Vreman model with a finite element method with h-adaptation technique is verified and validated using several benchmark cases. The end result being part of an on-going effort to enhance combustion predictability and increase efficiencies within engines.

Table of Contents

Abstract.....	iii
Table of Contents.....	v
List of Figures.....	vii
Chapter 1: Introduction.....	1
Chapter 2: Literature Survey.....	3
2.1 Numerical Methods.....	3
2.1.1 Finite Difference Method (FDM).....	3
2.1.2 Finite Volume Method (FVM).....	4
2.2 The Method of Weighted Residuals (MWR).....	5
2.2.1 Finite Element Method (FEM).....	7
2.2.2 H-P Adaptation.....	7
2.2.3 Galerkin's Method.....	10
2.2.4 Isoparametric Elements.....	11
2.3 Projection Algorithm.....	14
2.3.1 WEAK Statements.....	18
2.3.2 Weighted Residual Statement of Velocity and Thermal Energy.....	20
2.4 Solution Process.....	20
2.4.1 Time Advancement.....	21
2.4.2 Mass Lumping.....	22
2.4.3 Petrov-Galerkin.....	23
2.4.4 Stability and Time Dependent Solution.....	24
2.5 Boundary Conditions.....	25

2.5.1 Boundary Conditions for Velocity	26
2.5.2 Boundary Conditions for Pressure and Velocity Corrector	27
2.5.3 Euler-Lagrange Multiplier	28
2.5.4 Boundary Conditions for Thermal Transport	28
2.6 Turbulence Modeling.....	29
2.6.1 Reynolds-Averaged Navier-Stokes (RANS)	29
2.6.2 Direct Numerical Simulation (DNS)	30
2.6.3 Large Eddy Simulation (LES)	30
Chapter 3: LES Modeling Approach	32
3.1 Filtering Definition and Properties	30
3.2 Favre-Filtered Navier-Stokes Equations.....	34
3.2.1 Leonard’s Decomposition.....	36
3.3 Sub-grid Scale Viscosity.....	37
3.3.1 Smargorinsky Model.....	37
3.3.2 Vreman Model	39
Chapter 4: Results and Discussion.....	41
4.1 Geometry.....	41
4.2 Mesh and Velocity Contours	42
4.3 Comparison of U versus Y and X versus V	44
4.4 Comparison to COMSOL Simulation.....	46
Chapter 5: Conclusion.....	50
References.....	51
Curriculum Vitae	54

List of Figures

Figure 2.1 Example of h-adaptation.....	8
Figure 2.2 H-adaptive method for unstructured finite element grids	9
Figure 2.3 Mapping from physical to computational domain.....	12
Figure 2.4 Decomposition of V into U and P	16
Figure 3.1 Prediction of RANS, LES, and DNS.....	33
Figure 3.2 Kolmogorov scale of eddy energy.....	39
Figure 4.1 Mesh and geometry of cavity lid.....	41
Figure 4.2 Mesh and velocity contour at different Reynold's numbers using LES.....	42
Figure 4.3 Mesh and velocity contour at different Reynolds using RANS simulation with adaptation ...	43
Figure 4.4 U versus Y at different Reynolds using both LES and RANS	44
Figure 4.5 X versus V at different Reynolds using both LES and RANS	45
Figure 4.6 Mesh of COMSOL simulation	46
Figure 4.7 Velocity profile of COMSOL simulation for $Re=3,200$	47
Figure 4.8 Pressure contour of COMSOL simulation for $Re=3,200$	47
Figure 4.9 U versus Y at different Reynolds using COMSOL.....	48
Figure 4.10 X versus V at different Reynolds using COMSOL	49

Chapter 1 Introduction

The world is evolving to a more clean, greener and energy efficient place. Bills and laws such as the Clean Air Act passed in the 1990s required automotive makers to improve engine design as well as efficiency to reduce pollutant emissions and decrease fuel costs. Even today, the idea of improving efficiency and design has not changed. Automotive makers look to renewable fuel sources, as well as liquid forms of energy, such as hydrogen or electricity to replace the cost and reliance on oil. In the past, improving efficiency was difficult because of the complexity of simulating a combustion engine process and the lack of computer resources. Internal combustion engines are extremely intricate systems that include turbulent fluid flow, flame propagation, heat transfer, moving parts, and chemical reactions. Modeling such a system is a difficult task that requires a high performance computer to produce the calculation. In addition, creating the models to accurately simulate the physical process is a complicated process for internal combustion engines.

Kiva [38] is a powerful multi-dimensional internal combustion engine code capable of solving many different systems that occur during engine processes such as turbulent fluid flow, heat transfer, moving meshes, and reactions. One of the difficulties of modeling turbulent flow is the accuracy required near the walls. Normally wall functions are used to approximate wall velocities in the viscous sublayer. While they do offer a good approximation for this layer, wall functions diverge rapidly during the buffer layer between the viscous sublayer and the transition region. In addition, wall functions are generally only good with $y^+ < 5$ wall units and error generates rapidly past that. The y^+ represents the distance to the wall made dimensionless in regards to shear velocity and kinematic velocity.

A popular approach is the Smargorinsky model which is particularly effective at resolving the turbulence scales. The downfall to Smargorinsky is that it models the eddy viscosity using a constant scale throughout the entire domain which is not accurate. In addition, the scaling can lead to fluctuations in the constant value and can even become negative resulting in instability. Lastly, a filter or some sort of clipping is required of the small scale eddies to produce good results. These small eddies are needed as they play an important role in accurately simulating reacting flow, multi-phase flow, and flow near the wall boundary. This issue makes the Smargorinsky model unappealing for use in combustion engines.

Vreman [15], more recently developed a model that resolves the turbulence to the scale of the mesh by modelling the sub-grid scale eddies. These turbulence eddies are not lost, unlike Smargorinsky. The model uses first order velocity derivatives and does not involve any sort of filtering or clipping associated with Smargorinsky. The Vreman model is also capable of handling transitional flows whereas the Smargorinsky model cannot. In addition, the Vreman modeled has been found to be as accurate, if not more, as the Smargorinsky model and as good as the dynamic Smargorinsky model. This makes the model more suitable for combustion processes.

Chapter 2 Literature Survey

This chapter presents an outline of research related to the numerous experimental and numerical investigations concerning turbulent LES modeling.

2.1 Numerical Methods

Many engineering problems often have analytical solutions associated with them that can be described by linear and homogenous differential equations. However, there are many other problems that have no analytical solution associated with them. These equations are nonlinear and nonhomogenous. Numerical methods are used to model differential equations that have no analytical solutions, whether linear or nonlinear, homogenous or nonhomogenous, or 1st and 2nd order. The three main methods used to model such equations are: finite difference method (FDM), finite volume method (FVM) and finite element method (FEM).

2.1.1 Finite Difference Method (FDM)

The finite difference method (FDM) is based on a discretization of the differential forms of the conservation equations based on truncation of Taylor series. FDM is the oldest technique of the three methods. The drawback of the FDM is due to its lack of flexibility on the geometry; it can only be effectively used with orthogonal meshes. Its implementation is simple, so numerical programs are easily developed. There are three main schemes to solve a differential equation using FDM. The first is forward difference where the first and second order derivatives are approximated respectively by:

$$\frac{df}{dx} = \frac{f_{i+1} - f_i}{\Delta x} + O(\Delta x) \text{ and } \frac{d^2 f}{dx^2} = \frac{f_{i+2} - 2f_{i+1} + f_i}{\Delta x^2} + O(\Delta x) \quad (2.1)$$

The rate of change is then approximated between the current time step and the next time step.

The backward difference is similar to the forward difference scheme:

$$\frac{df}{dx} = \frac{f_i - f_{i-1}}{\Delta x} + O(\Delta x) \text{ and } \frac{d^2 f}{dx^2} = \frac{f_i - 2f_{i-1} + f_{i-2}}{\Delta x^2} + O(\Delta x) \quad (2.2)$$

Lastly, the central difference which incorporates both approaches and produces second order accuracy:

$$\frac{df}{dx} = \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x^2) \text{ and } \frac{d^2 f}{dx^2} = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + O(\Delta x^2) \quad (2.3)$$

The main problem with the FDM is that the error of the numerical solution increases with the number of steps; this error is referred to as accumulative error. Additionally, step size has strong effects on the accuracy of the model. Numerical stability is not always guaranteed and because of this FDM is typically used to approximate simple configurations involving partial differential equations (PDEs). [1]

2.1.2 Finite Volume Method (FVM)

The finite volume method (FVM) is a method for representing and evaluating partial differential equations in the form of algebraic equations [2]. Similar to the FDM, values are calculated at discrete places on a domain. The term finite volume refers to the small volume surrounding each node point on a domain. In the FVM, volume integrals in a partial differential equation that contain a divergence term are converted to surface integrals, using the divergence theorem, that is

$$\iiint_V (\nabla \cdot F) dV = \oiint_S (F \cdot n) dS \quad (2.4)$$

These terms are then evaluated as fluxes at the surfaces of each element. FVM is

naturally conservative as the flux entering a given volume is the same as that leaving the adjacent volume. Another advantage of the FVM is its flexibility as it can be used with both structured and unstructured meshes, and it is not limited in regards to geometry as FDM. FVM is widely used in commercial applications and in computational fluid dynamics (CFD) due to its simplistic yet accurate approach.

$$\frac{df}{dx} = \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} \quad (2.5)$$

2.2 The Method of Weighted Residuals (MWR)

The governing equations can be formulated into weak integral statements that are true over any domain size by approximating the dependent variables with an orthogonal basis set. These approximations are substituted for the dependent variables and the derivatives are performed on the polynomial approximation. The Method of Weighted Residuals (MWR) can be used to obtain the weak statements for the governing equations which require the use of Green's theorem. Green's theorem allows for the reduction of a second order equation integrated over a volume or area to a first order equation integrated over an area or line, respectively. This allows us to produce the weak integral statement of the governing equations. The value of using the MWR is that it is a simplistic method for the creation of the weak statement.

Next, a residual statement is introduced which states that the weighted integral of the weak statement over an element be equal to zero which allows the MWR to be applied to any domain size. The ability to solve the weighted residual statements in an accurate manner is coupled to the elements which discretizes the domain. The finite element method is employed to achieve the solution over discrete portions of the domain. These discrete domains are assembled

in linear fashion to solve the equations over the entire domain. The more refined the grid the better the accuracy of the solution.

Since the problem domain is a discrete system, the finite element method seeks to minimize the residual, R , over the entire domain. For example, if the residual equation [4] is determined by:

$$R(T, x_i) \equiv -k\nabla^2 T - Q \quad (2.6)$$

where Q is the source term, k is the thermal conductivity, and T the temperature approximation and T is given by:

$$T(x_i) = \sum_{i=1}^n T_i N_i \quad (2.7)$$

This is a polynomial expansion of order n . The term N_i is the weight, and T_i is the nodal temperatures.

The finite element method seeks to minimize this Residual over a domain [5]. Requiring the residual to be zero on average is accomplished by multiplying the residual equation by the appropriate weighting function W_i and integrating over the entire domain.

$$\int_{\Omega} W_i R(T, x_i) d\Omega = 0 \quad (2.8)$$

The inner product when equated to zero is an orthogonal projection of the residual since it seeks to find values for T to make the statement true. When applied over a domain, which is discretized into finite elements, the resulting set of algebraic equations can be solved for the unknown values, in our case, T at the nodes.

2.2.1 Finite Element Method (FEM)

In mathematics, the finite element method (FEM) is a numerical technique for finding approximate solutions to boundary value problems for partial differential equations. It is based on the principle of variational calculus to minimize the error function and produce a stable solution. FEM uses many simple element equations over many small subdomains, i.e. finite elements, to approximate a more complex equation over a larger domain. The main advantages [3] of using FEM are:

- Accurate representation of complex geometry
- Easy representation of the total solution
- Capture of local effects
- Inclusion of dissimilar material properties

A FEM solution involves dividing the domain of the problem into a collection of subdomains, where each subdomain is represented by a set of element equations. We then assemble all sets of the element equations into a global system of equations. This global system of equations can then be solved using several different techniques.

2.2.2 H-P Adaptation

There are two main types of adaptation available: mesh refinement and equation refinement, h and p-adaptation, respectively. H-adaptation is the process of taking a cell and creating a new cell within it via the block method. This maintains any curvature of the cell and the original element is not affected. H-adaptation eliminates the need of having a small mesh over the entire domain, instead locally refining where required, usually in areas of high gradient changes. The use of h-adaptation yields accurate solutions and exponential convergence rates at the cost of

computational power. This is particularly important in the case of singularities such as those involved in the cavity-driven lid. Singularities are best handled with adaptation occurring early in the solution process.

Determining areas of mesh refinement is commonly done using a posteriori error estimate and a least-squares method to “smooth” the values at the nodal points. This requires a solution set of a linear matrix [35]. This process is done by using smoothed values and equal order interpolation used by the finite element method [36]. Using a local least squares method, we can minimize the number of nodes which need to be refined using a maximum tolerance error. The current error estimator is based on work by Oden and Ainsworth [34] which solves for these errors and expresses them as an “energy” or “ L_2 ” norm [34]. The mesh adaptation process includes division and recovery of elements as well as removal of holes in the grid after dividing and recovering. This process is shown below in the flow chart in Figure 2.2.

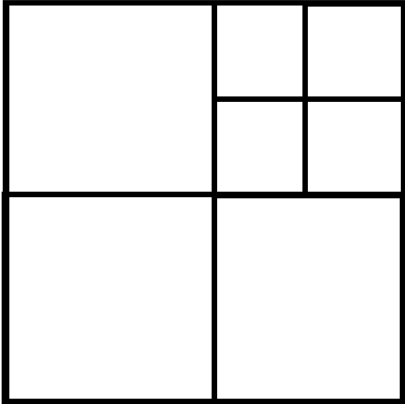


Figure 2.1 Example of h-adaptation

Another type of adaptation is P-adaptation, which uses a higher degree polynomial for the basis function. The complexity of higher order degrees beyond quadratic becomes difficult to manage for the basis function. P-adaptation also offers exponential convergence at the cost of computational power.

h-ADAPTATION

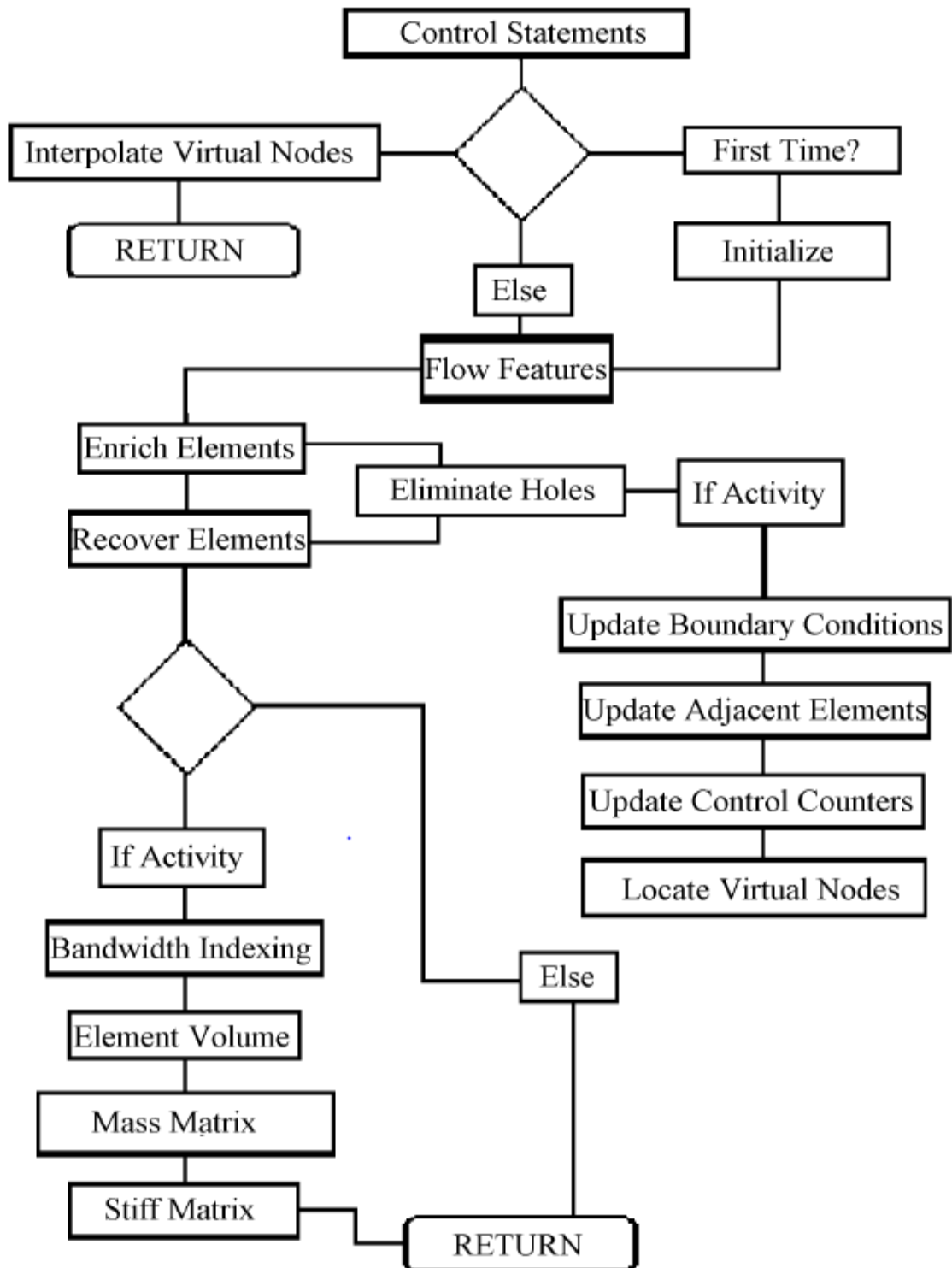


Figure 2.2 h-adaptive method for unstructured finite element grids

2.2.3 Galerkin's Method

In choosing $W_i = N_i$, the method of weighted residuals becomes Galerkin's method.

Although other methods exist, we will be focusing exclusively on Galerkin's method. Although there are other weights, this choice of weight is popular and allows for the formulation of one set of basis functions. Integration is done over discrete elements in the domain where the accumulation of these integral equations over individual elements yields a system of integral equations describing the entire domain. I

By using Green's Theorem, we obtain the weak formulation of the integral equations.

Continuing from 2.7, the weak formulation for the diffusion of heat is:

$$\int_{\Omega} \frac{\partial W_i}{\partial x_i} k \frac{\partial T}{\partial x_i} d\Omega - \int_{\Omega} W_i Q d\Omega + \int_{\Gamma} W_i \left(-k \frac{\partial T}{\partial n} \right) d\Gamma = 0 \quad (2.10)$$

Particularly beneficial is the way the integral expression automatically incorporates the surface fluxes which describes the flux of energy moving across the domain boundaries.

$$q = \int_{\Gamma} W_i \left(-k \frac{\partial T}{\partial n} \right) d\Gamma \quad (2.11)$$

Determining the exactness of the approximated solution is fundamental to all approximation procedures. A measure for this bilinear equation presented above can be defined by the norm:

$$\|T - T_h\|_{H^0} \leq ch^p \quad (2.12)$$

where c is a constant dependent on the domain, h is the size of the element, and p is the order of the approximating polynomial. The difficulty with determining the error with measure as currently described is that the exact or true solution T is not usually known. As first suggested by Zienkiewicz and Zhu [6], an approximation to the exact solution can be made if we know that

the solution lies within the bounds of the approximated projection or solution. A data smoothing process, such as the least squares method, will bring the approximate solution close to the actual solution. This method is applied to the gradient of velocity since the derivatives of the finite element using linear approximating functions are discontinuous at the nodes [7]. Therefore, 2.12 may be used to evaluate the error.

2.2.4 Isoparametric Elements

It remains to choose an appropriate weight function and choosing the eigenvalues would accomplish the task. The 1st order Lagrangian interpolating polynomial is used as the eigenfunction. Choosing a normalized transformation for the interpolating function and an approximate weight produces Galerkin's method with isoparametric elements and a normalized computational space. The linear system of equations is assembled from each element in the geometric domain, which is transformed to an element of normalized length in the computational domain.

This mapping or transformation allows for the development of generic interpolating functions known as shape functions. These basis or shape functions for isoparametric tri-linear hexahedral elements are listed below [8].

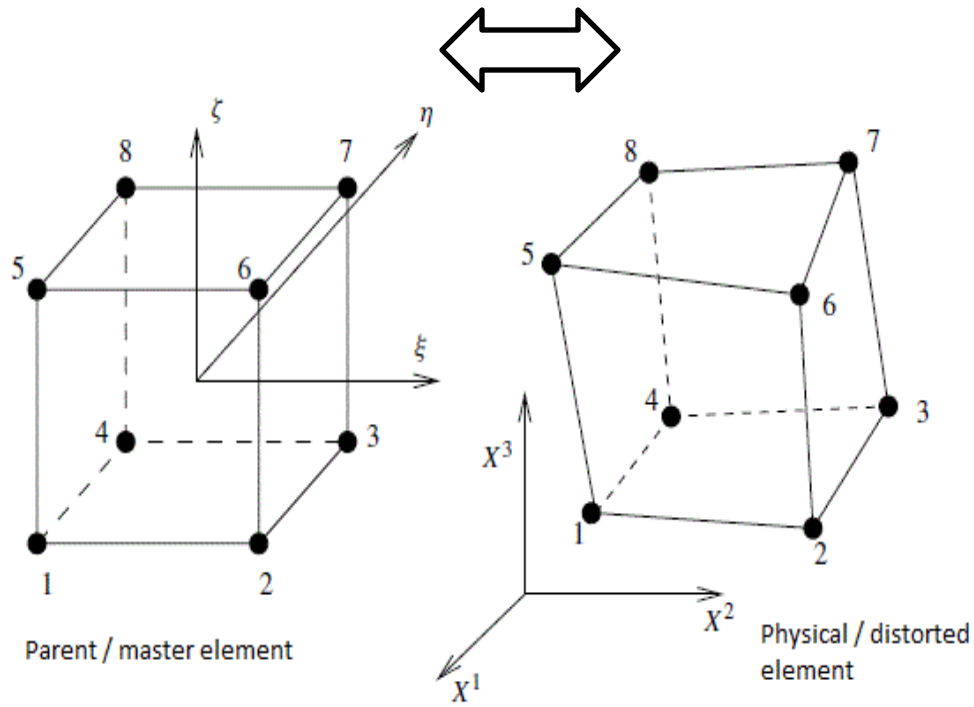


Figure 2.3 Mapping from physical to computational domain

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \\ N_8 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} (1-\xi)(1-\eta)(1-\zeta) \\ (1+\xi)(1-\eta)(1-\zeta) \\ (1+\xi)(1+\eta)(1-\zeta) \\ (1-\xi)(1+\eta)(1-\zeta) \\ (1-\xi)(1-\eta)(1+\zeta) \\ (1+\xi)(1-\eta)(1+\zeta) \\ (1+\xi)(1+\eta)(1+\zeta) \\ (1-\xi)(1+\eta)(1+\zeta) \end{bmatrix} \quad (2.13)$$

Derivatives of the shape functions are obtained from the chain rule

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x_j} \frac{\partial x_j}{\partial \xi} \delta_{ij} \quad (2.14)$$

Derivatives of a global quantity in this computational domain are easily obtained from the shape functions and the Jacobian [9]

$$\begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{bmatrix} = J \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \quad (2.15)$$

where the Jacobian is defined as:

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \quad (2.16)$$

The relationship between global and computational domain derivatives of a function is given by:

$$\begin{bmatrix} \frac{\partial N_j}{\partial \xi} \\ \frac{\partial N_j}{\partial \eta} \\ \frac{\partial N_j}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_j}{\partial x} \\ \frac{\partial N_j}{\partial y} \\ \frac{\partial N_j}{\partial z} \end{bmatrix} \quad (2.17)$$

This transformation is nonsingular, meaning it is one to one onto the domain of the natural coordinate system. For each element, the derivatives of variables (i.e. trial functions) in the global domain can be defined as:

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix} = J^{-1} \begin{bmatrix} \sum \frac{\partial N_j}{\partial \xi} \\ \sum \frac{\partial N_j}{\partial \eta} \\ \sum \frac{\partial N_j}{\partial \zeta} \end{bmatrix} \quad (2.18)$$

where the inverse of the Jacobian J^{-1} is given by:

$$J^{-1} = \frac{1}{|J|} \begin{bmatrix} \frac{\partial x}{\partial \xi} & -\frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & -\frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & -\frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \quad (2.19)$$

Integration takes place over sampling points within the domain. Since we have made a transformation to a normalized computational domain, the implementation of Gauss-Legendre quadrature for the numerical integration can be applied directly to the integral equations without a change of limits. The integration in three dimensions is shown as follows:

$$\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 F(\xi, \eta, \zeta) |J| d\xi d\eta d\zeta = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n W_i W_j W_k F(\xi, \eta, \zeta) J(\xi, \eta, \zeta) \quad (2.20)$$

where the Gauss weights of integration W_i are evaluated at the Gauss points ξ, η, ζ . For two

point quadrature, the weights are 1.0 at points $\pm \frac{1}{\sqrt{3}}$ for each direction. Therefore, in three

dimensions, there are eight weighting points. Higher integration accuracy can be obtained with more points of integration. The integration for one point quadrature occurs at the centroid of the element in the computational domain with a weight of 2.0.

2.3 Projection Algorithm

There exists countless methods to find the solution to these nonlinear equations. A method to find the velocity, pressure, and vorticity formulation employing Newton's method to linearize the momentum equation [16], and using the least squares FEM with a conjugate

gradient technique has been demonstrated to work well [17]. Without some form of projection in the FEM, mixed methods are required. That is, different approximations or finite dimensional spaces are necessary for the velocity (Lebesgue or L_2) and pressure (Sobolev or H) are needed to satisfy the Div-Stability condition [18]. These conditions are also known as the LBB conditions. With the use of a projection method, where the pressure is being estimated from the flow, as well as in SIMPLE methods [18 & 19], the LBB conditions are satisfied.

A self-adjoint projection scheme was developed by various researchers including Gresho and Chan [20], Lohner [19] and Ramaswamy [21] which provides a solution for nonlinear problems. This semi-implicit scheme has an advantage over iterative methods that may not have good convergence rates, i.e. supplied with a reasonable first guess.

The projection method for the solution of the Navier-Stokes equations is a self-adjoint system created by decomposing the momentum into gradient driven or curl-free portions and divergence-free portions. A divergence free velocity field is maintained by the projection of the predicted velocity onto the divergence free space. An Euler-Lagrangian variational seeks to minimize the function [22]

$$E(V, \lambda) = \frac{1}{2dt} \int_{\Omega} \left([V - V^*]^2 + [\lambda][C]\{V\} \right) d\Omega \quad (2.21)$$

In the incompressible case, pressure is recovered from the divergence of the momentum equation using some initial velocity or recently calculated velocity. The velocities are then updated from the pressure, which enforces continuity. This splitting method is discussed below as an Euler-Lagrange Variational projection into divergence free space.

The projection-step algorithm used in this computer model is based on the method initially developed by Chorin [23]. Using Helmholtz-Hodge decomposition theorem, which states that any vector field in the domain Ω can be uniquely decomposed as:

$$V = U + \nabla P \quad (2.22)$$

where U is the divergence free velocity vector. $\nabla \cdot U = 0$ in Ω and $U \cdot n = 0$ on the boundary Γ .

This portion of the decomposition is a projection onto a divergence free field [24 & 25].

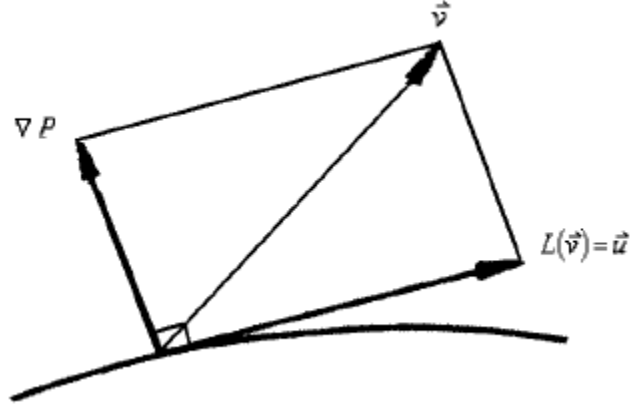


Figure 2.4 Decomposition of V into U and P

The projection is shown in Fig. 4 for the velocity field, V . Notice the gradient portion has zero curl under the decomposition since the vector identity $\nabla \times \nabla P = 0$. The curl of a vector field that is a function of the gradient of a scalar is irrotational, i.e. curl free.

Under the projection, we seek the proper P such that

$$\nabla P = V + U \quad (2.23)$$

Taking the divergence of each side, we get

$$\nabla \cdot \nabla P = \nabla^2 P = \nabla \cdot (V + U) = \nabla \cdot (V) + \nabla \cdot (U) = \nabla \cdot (V) \quad (2.24)$$

The linear orthogonal projection operator L , applied to the incompressible Navier-Stokes vector fields yields

$$L\left(\frac{\partial U}{\partial t} + \nabla P\right) = L\left(- (U \cdot \nabla) \rho U + \mu \nabla^2 U\right) \quad (2.25)$$

And since L is a linear operator and $L(\nabla P) = 0$ as shown previously, pressure is removed from the equation set. The projection under L is given as:

$$\frac{\partial U}{\partial t} = -(U \cdot \nabla) \rho U + \mu \nabla^2 U \quad (2.26)$$

where U is the divergence free averaged time velocity. During the time advancement of the averaged field a projection onto the divergence free space is performed, maintaining a divergence free velocity field. This is accomplished with the proper choice of P , and splitting the velocity into the divergence-free field and the perturbed field or predictor.

Splitting the velocity into two averaged components, V^* and V , the momentum equations under the linear orthogonal projection operator L becomes:

$$\rho \frac{V_{n+1}^* - V_n}{dt} + \rho V_n \cdot \nabla V_n = \mu \nabla^2 V_n \quad (2.27)$$

where the velocity components of V are either from the initial guess or from the previously calculated time step. This is the divergence free velocity attained through the proper choice of $\text{grad}(P)$.

Given the approximate velocity just advanced from the previous explicit marching, the goal is to find some velocity V that satisfies continuity. We seek the projection of V^* , a perturbed velocity, onto the divergence free space to complete the calculation of the velocities subject to incompressibility. Under the decomposition of the vector field $L(V^*)$, we make the projection

$$V^* = V + dt \nabla P \text{ where } \nabla \cdot V = 0 \quad (2.28)$$

Taking the gradient of both sides, a Poisson equation for P is obtained in the form

$$\nabla^2 P = \frac{-\nabla \cdot V^*}{dt} \quad (2.9)$$

In discretized finite element representation, we have

$$\frac{M(V - V^*)}{dt} + \nabla P = 0 \quad (2.30)$$

where M is the mass matrix. The equations above are essentially the Euler-Lagrange equation

$$\frac{M}{dt}(V - V^*) + CP = 0 \quad (2.31)$$

where C is the gradient operator. The equation is subject to the constraint of continuity.

$$C^T V = 0 \quad (2.32)$$

The system is solved sequentially by creating a diagonal form of the mass matrix (a lumped matrix), multiplying by its inverse, and then by taking the gradients of both sides and enforcing continuity as shown below:

$$\begin{aligned} C^T M^{-1} CP &= C^T V^* \\ V &= V^* - dt M^{-1} CP \end{aligned} \quad (2.33)$$

2.3.1 Weak Statements

The weak statement for the projection is

$$\int_{\Omega} \left[\frac{\partial N_j}{\partial x_j} \right] \{N_k\} \sum_{l=1}^n [N_l] \left\{ \frac{\partial N_l}{\partial x_i} \right\} \{P_i\} d\Omega = \int_{\Omega} \left[\frac{\partial N_j}{\partial x_j} \right] \{V_i^*\} \frac{d\Omega}{dt} \quad (2.34)$$

where the summation creates the diagonalized mass matrix. Solving for the averaged V from the weighted residual statement produces the divergence free velocity. $[N_j]$ is the shape function notation for the element, $[]$ represents a row vector (or matrix), $\{ \}$ represents the transpose of a column, and Ω indicates the domain.

$$\{V_i\} = \{V_i^*\} - dt \left[\{N_k\} \sum_{l=1}^n [N_l] d\Omega \right]^{-1} \left[\int_{\Omega} \left\{ \frac{\partial N_i}{\partial x} \right\} [N_j] \{P\}_i d\Omega \right] \quad (2.35)$$

A time explicit advancement of velocity is made using the weakened momentum equations and an assumed initial pressure at time $n=0$. The projection onto a divergent free field is made to ensure mass continuity. Pressure can be determined if desired. Then the whole process is repeated as time progresses. After the inverse matrix is established for the solution of the pressure, the most time consuming part of the process is the solution of the Euler-Lagrange equation enforcing mass continuity.

To apply the finite element method to the solution of the governing equations the weak statements of the equations are found and then coded. The energy and mass transport equations are included below. The method of weighted residuals is applied to the weak statements for the governing equations. The dependent variables are replaced with their trial functions.

$$Z_i = \sum_{i=1}^n \phi_n z_i^n(t) = [N_j] \{Z_i\} \quad (2.36)$$

where Z_i are the dependent variables, $[N_j]$ are the basis or shape function notation for the element, $[]$ is a row vector (or matrix), $\{ \}$ is the transpose of a column, Ω indicates the domain, and Γ is the surface boundary.

Substituting the dependent variables trial functions into the governing equations and the 2nd order terms described above, produces the following set of integrated ODE's.

2.3.2 Weighted Residual Statement of Velocity under Decomposition:

$$\begin{aligned}
& \left(\rho \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \right) \{\dot{V}_i\} + \rho \left(\int_{\Omega} \{N_i\} (N_k V_k) \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{V_i\} + \int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \{P_i\} d\Omega \\
& + \int_{\Omega} \{N_i\} \frac{2}{3} \left[\frac{\partial N_j}{\partial i} \right] d\Omega \{k_i\} - \int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \{\mu_i\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \{V_i\} + \int_{\Omega} [\mu + \mu_t] \left\{ \frac{\partial N_j}{\partial x_i} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \{V_i\} \quad (2.37) \\
& + \rho \int_{\Omega} f(x_i) \{N_i\} d\Omega - \int_{\Gamma} \{N_i\} [\mu + \mu_t] \left[\frac{\partial N_j}{\partial x_i} \right] \{V_i\} \widehat{n_i} d\Gamma = 0
\end{aligned}$$

where $f(x_i)$ is the body force per unit mass, typically gravity. As mentioned previously, for slightly compressible fluids, those that are subject to the Boussinesq approximation for density changes as a function of temperature, this body force is the difference in gravity forces and buoyant forces $(\rho_0 - \rho) g_i$. Similarly, we can derive the weighted residual statement of thermal energy:

$$\begin{aligned}
& \left(\rho C_p \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \right) \{\dot{T}_i\} + \left(\rho C_p \int_{\Omega} \{N_i\} (N_k V_k) \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{T_i\} \\
& - \int_{\Omega} \{N_i\} \left(\left[\frac{\partial N_j}{\partial x_i} \right] \left\{ \frac{\mu_t}{\text{Pr}} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] \right) d\Omega \{T_i\} + \left(\int_{\Omega} \left[\kappa + \frac{\mu_t}{\text{Pr}} \right] \left\{ \frac{\partial N_j}{\partial x_i} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) \{T_i\} \quad (2.39) \\
& - \left(\int_{\Omega} \{N_i\} \{Q_i\} d\Omega \right) - \left(\int_{\Omega} \{N_i\} \{q_i\} \Gamma d\Gamma \right) = 0
\end{aligned}$$

2.4 Solution Process

By integrating over each element and combining the contributions from each element to the nodes in common to those elements, a matrix equation is formed that will be solved for these nodal values. It is important to note that when integrating over each element, the contributions of the surface flux cancel everywhere except at the boundaries. This is an important distinction

between the finite element method and the finite volume method. It also leads some to the idea that the finite element method is not locally conservative. On the contrary, it is precisely conservative whereas the finite volume method is not because of the truncation error associated with evaluating the surface fluxes everywhere within the domain. The matrix equations for the explicit time advancement of momentum and heat can be written as:

$$\begin{aligned} [M]\{\dot{V}\} + [A(V)]\{V\} - C\{P\} + [K_v]\{V\} &= \{F_v\} \\ [M_T]\{\dot{T}\} + [A(V)]\{T\} + [K_T]\{T\} &= \{F_T\} \end{aligned} \quad (2.39)$$

The individual matrices for these equations are defined as:

$$\begin{aligned} M &= \rho \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \\ M_T &= \rho C_p \int_{\Omega} \{N_k\} \sum_{l=1}^n [N_l] d\Omega \\ A(V) &= \rho \int_{\Omega} N_i (N_k V_k) \frac{\partial N_i}{\partial x_i} d\Omega \\ K_v &= - \left(\int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \mu_t \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega \right) + \int_{\Omega} \left([\mu + \mu_t] \frac{\partial N_j}{\partial x_i} \frac{\partial N_j}{\partial x_i} \right) d\Omega \\ C &= \int_{\Omega} \frac{\partial N_j}{\partial x_i} N_i d\Omega \\ F_v &= \int_{\Omega} N_j f(x_i) d\Omega + \int_{\Gamma} [\mu + \mu_t] N_j \hat{n}_j \frac{\partial V_j}{\partial x_j} d\Gamma \\ K_T &= \int_{\Omega} \{N_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \left\{ \frac{\mu_t}{\text{Pr}} \right\} \left[\frac{\partial N_j}{\partial x_i} \right] d\Omega + \int_{\Omega} \left(\left[\kappa + \frac{\mu_t}{\text{Pr}} \right] \frac{\partial N_i}{\partial x_j} \frac{\partial N_i}{\partial x_j} \right) d\Omega \\ F_T &= \left(\rho \int_{\Omega} \{N_i\} \{Q_i\} d\Omega \right) + \left(\int_{\Gamma} \{N_i\} [N_j] \{q_i\} d\Gamma \right) \end{aligned}$$

2.4.1 Time Advancement

The initial guess of velocity is time marched explicitly by:

$$\{V_i^{n+1}\} = \{V_i^n\} + \Delta t [M^{-1}] \left[\{F_v\} - [K_v] \{V_i^n\} - [A(V)] \{V_i^n\} + [C] \{P_i^n\} \right] \quad (2.40)$$

This explicit marching also applies equally to the scalar quantities of temperature as well. Before marching these quantities forward in time, the velocities need to be projected onto the divergence free field. The velocities are updated from the components of P

$$V^{n+1} = V^* + dtM^{-1}CP \quad (2.41)$$

The pressure is calculated from either the discretized Poisson equation or is extracted directly from the projection algorithm by dividing by λ with dt. This pressure is associated with the projection, the time advanced divergent velocity. To calculate the pressure experienced in the momentum equations, the gradient is taken of the divergent free Navier-Stokes equations, resulting in the Poisson equation.

Scalar transport for energy and species are performed as per the scalar transport equation.

$$\{T_i^{n+1}\} = \{T_i^n\} + \Delta t [M^{-1}] \left[\{F_{V_i}\} - [K_T] \{T_i^n\} - [A(V)] \{T_i^n\} \right] \quad (2.42)$$

Time step size should be a consideration of this explicit statement. The time scale of most engineering problems are governed by the faster time scales of turbulence and momentum transport.

2.4.2 Mass Lumping

Mass lumping is the combining of the time dependent terms in the mass matrix, row by row, into a diagonal matrix. This is done simply by adding the terms of each row. Mass lumping creates a new matrix that has its inverse as:

$$M_L^{-1} \equiv \frac{1}{M_L} \quad (2.43)$$

where

$$M_L = \int_{\Omega} N_i \sum_{j=1}^n N_j d\Omega$$

Mass lumping makes the time dependent equation an explicit equation [26 & 16]. Mass lumping speeds the transient solution significantly because multiplication by the inverted mass matrix is not required at each time step. The size of the time increment for explicit advancement is governed by stability requirements based on the Courant and Reynolds cell numbers. The Courant number is needed for the finite difference method to help calculate the appropriate time step for a given solution. If the time step is too big based on the Courant number, then the solution will diverge. The Reynolds cell number is used to make sure the discretization scheme behaves appropriately, especially at critical cell numbers where oscillations can occur.

2.4.3 Petrov Galerkin

A Petrov-Galerkin scheme is used to weight the advection terms

$$W_i = N_i + \frac{\alpha h_e}{2V} (V \cdot \nabla N_i) \quad (2.44)$$

where h_e is the element size, $\alpha = \coth \beta / 2 - 2 / \beta$, $\beta = \frac{h_e |V|}{2K_e}$ and K_e is an effective diffusion term in the direction of the local velocity vector [17, 27, 28 & 29]. This weighting introduces selective artificial diffusion into the numerical scheme that acts along the local streamline. This method is effective at removing numerical dispersion in very steep gradient areas, leaving between 1-2% noise in the solution. This dispersive error is associated with modeling advection, and is precisely measured prior to the time advancement and then removed during integration. It is important to note that this Petrov-Galerkin method is also useful as a shock capturing scheme, even in the absence of molecular viscosity, i.e. Peclet number.

For non-hydrostatic calculations, the pressure is obtained from the solution of a Poisson equation based on the discretized momentum equations

$$[K]\{p\} = C^T [M]^{-1} [\{F_v\} - ([K_v] + [A(V)])\{V\}] \quad (2.45)$$

Currently, Sparse Cholesky or Krylov solvers are used to solve the Poisson pressure equation. A time dependent form of the continuity equation is used to correct the velocities. A forward-in-time Euler scheme is employed to advance the discretized equations in time [16].

2.4.4 Stability and Time Dependent Solution

The explicit Euler time integration scheme has time advancement restrictions that are met by the requirements of the Courant and Reynolds cell numbers. The determining equations for a forward-in-time, centered-in-space finite difference scheme can be found using a Von Neumann stability analysis [30]. In fact, only the stability of the linear equations can be analyzed with this type of analysis. Linearizing a nonlinear equation can be performed and the stability analyzed, although it is only locally applicable [31]. The stability analysis produces guidelines to constrain the time increments.

Von Neumann stability analysis is based on Fourier mode analysis. Velocity can be expressed in its Fourier mode as:

$$u_j^n = U^n e^{ik\Delta x j} \quad (2.46)$$

where k is the wave number in the x -direction, $k\Delta x$ is the phase angle, $i = \sqrt{-1}$, and j are the discretized coordinate indices. These components are substituted into the discretization and reduced. An amplification factor “ G ” is introduced such that.

$$U^{n+1} = GU^n \quad (2.47)$$

Stability requires the absolute value of “G” be bounded for all values of $k\Delta x$. If it is assumed that the fluid motion is wave-like in nature, the discretization is made to represent the motion over a length Δx . The highest frequency in the interval that can be approximated is $2\Delta x$, i.e., it requires at least three points to approximately determine a sine wave between 0 and 2π .

Hindmarsh [32] determined the necessary and sufficient conditions for stability of the advection-diffusion equation. This analysis as applied to the explicit Euler forward scheme produces the time increment limits as:

$$\Delta t \leq \frac{1}{\sum_{j=1}^3 \frac{2K_j}{\Delta x_j^2}} \quad (2.48)$$

$$\Delta t \leq \frac{1}{\sum_{j=1}^3 \frac{u_j^2}{2K_j}}$$

where the term K_j refers to the j-th directional component of the diffusion matrix. Since the Galerkin method utilizing linear interpolating polynomials has a centered-in-space type architecture, these stability constants certainly give some idea as to what the time increments might be allowed.

Numerical experimentation with various types of problems has shown that the following stability conditions are usually satisfactory:

$$\Delta t \leq \frac{2}{\text{Re}} \frac{1}{\|U\|^2} \quad (2.49)$$

$$\Delta t \leq \frac{\text{Re}/2}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}$$

Construction of the Δx or h in three dimensions is performed by finding the average value for the coordinates of each face and then taking the difference between opposing faces. The entire grid is searched for the constraining values in order to optimize the time step.

2.5 Boundary Conditions

Evaluating the boundary integral for the second order bilinear equation as:

$$\int_{\Gamma} W_i \left(-k \frac{\partial \hat{T}}{\partial n} \right) d\Gamma \quad (2.50)$$

over the surface Γ , we break the surface integral into its x and y components:

$$\frac{\partial \hat{T}}{\partial n} = \hat{n} \cdot \nabla \hat{T} = n_x \frac{\partial \hat{T}}{\partial x} + n_y \frac{\partial \hat{T}}{\partial y} \quad (2.51)$$

Determining the value of the direction cosines, n_x and n_y are obtained from noticing that

$$n_x = \cos \theta = \frac{dy}{d\Gamma} \text{ and } n_y = \sin \theta = -\frac{dx}{d\Gamma} \quad (2.52)$$

Therefore, the equation for the surface integral in 2-D becomes

$$-\int_{\Gamma} kW_i \left(n_x \frac{\partial \hat{T}}{\partial x} + n_y \frac{\partial \hat{T}}{\partial y} \right) d\Gamma \quad (2.53)$$

2.5.1 Boundary Conditions for Velocity under Decomposition

Dirichlet boundary conditions for velocity are simple: either a no-slip condition for solid objects or a fixed velocity at inlets. Outlet boundaries can be made with the assumption of a zero gradient for velocity. The zero gradient assumption requires the computational domain to be

constructed to match this imposed boundary condition. This statement can be relaxed with the use of viscous boundary condition [22].

Another boundary condition exists when a weak statement is created. Weakening the second derivative viscous term results in

$$\int_{\Gamma} \{N_i\} \{\mu + \mu_i\} \left[\frac{\partial N_j}{\partial x_i} \right] \{V_i\} n_i d\Gamma \quad (2.54)$$

This equation is zero for walls where no-slip boundary conditions exist. It is also zero where an inlet or outlet velocity is normal to the boundary. Otherwise the components of these boundary integrals are evaluated and used to relax the requirements of zero gradients at an outflow velocity when combined with the calculated pressure at the outflow.

2.5.2 Boundary Conditions for Pressure and Velocity Corrector

Because the equation for pressure in incompressible flow is a Poisson equation, it is therefore elliptic. Boundary conditions must be imposed at all surfaces of the computational domain. The Neumann boundary condition is:

$$n \cdot \nabla P = \frac{\partial P}{\partial n} = -n \cdot \frac{\partial V}{\partial t} \quad (2.55)$$

This boundary condition, when combined with a Dirichlet condition at some reference point to eliminate the singularity in the equation set, is sufficient to determine the pressure up to an arbitrary constant.

The second half step is related to inviscid flow. The portion of the decomposition without curl relies on the determination of the proper scalar gradient to make the decomposition true. Therefore the proper boundary condition is related to the normal component of penetration through the boundary. This is defined as:

$$n \cdot V^{n+1} \Big|_{\Gamma} = n \cdot f^{n+1} \quad (2.56)$$

where f^{n+1} is either the prescribed boundary condition or is evaluated from the viscous terms at the boundary,

$$\int_{\Gamma} \mu [N_i] \left[\frac{\partial N_j}{\partial x_i} \right] \{V_i\} n_i d\Gamma \quad (2.57)$$

2.5.3 Euler-Lagrange Multiplier

If a Lagrangian multiplier is substituted for pressure, the boundary conditions for the projection equation are found in the same manner as the pressure Poisson equation. The boundary condition for this multiplier is derived from:

$$\nabla P = \frac{V - V^*}{dt} \quad (2.58)$$

And combined with the boundary conditions for pressure given earlier, the resulting traction equation is:

$$n \cdot \nabla \lambda = \frac{\partial \lambda}{\partial n} = n \cdot [V - V^*] \quad (2.59)$$

Continuity applies to $n \cdot V$ so the boundary condition for λ is:

$$\frac{\partial \lambda}{\partial n} = n \cdot V^* \text{ on } \lambda \quad (2.60)$$

2.5.4 Boundary Conditions for Thermal Transport

Thermal transport equations have either specified flux (Neumann) or fixed (Dirichlet) conditions. As noted earlier, a zero flux is automatically applied if no other boundary condition exists. The integral of thermal flux is calculated for the energy transport equation by:

$$\int_{\Gamma} \{N_i\} [N_j] \{q_i\} d\Gamma \quad (2.61)$$

The shape functions are now for 1-D line elements or 2-D surface elements depending on whether the problem is 2-D or 3-D, respectively.

2.6 Turbulence Modeling

The biggest challenge in solving turbulence is accuracy and resolution. Turbulence, by its very nature, is unpredictable and random and interacts on many different length scales which interact with each other. The three main methods are described below.

2.6.1 Reynolds-averaged Navier-Stokes (RANS)

The simplest method to use in solving turbulent flow is to employ the Reynolds-Averaged Navier Stokes (RANS) method. The three methods of averaging are time, spatial and ensemble, with the most appropriate being a time averaged because almost all engineering flow problems involve inhomogenous turbulence. The equations are averaged over time with a mean component along with a fluctuating component. When time averaged, the Reynolds stress tensor is created and six new variables are formed which have not been solved. We need a closure scheme to solve for these variables and the most common way is using a zero, one, or two equation model. The most popular being the two equation k- ϵ and the k- ω model. The k- ϵ model is simple to implement but has poor prediction for swirling or rotating flows and flows with strong separation. In addition, it is only suitable for fully turbulent flows. The k- ω predicts better under strong adverse pressure gradients and can be used in separation flow such as turbomachinery. The disadvantage is that it tends to predict separation early and too big as well

as requires a fine mesh resolution near the wall. In regards to turbulence, both have problems and limitations that make them inadequate to deal with combustion engine processes.

2.6.2 Direction Numerical Simulation (DNS)

The most accurate method for modeling turbulence is Direct Numerical Simulation (DNS). The governing equations are solved and all motions contained in the flow are resolved. The results for DNS simulations contain very detailed information about the flow which is used as a tool for understanding turbulence production, dissipation, energy dissipation, and other variables of interest. The size of the grid cannot exceed the Kolmogorov scale, $N \sim \frac{L}{\eta}$, where N is the number of points, L the characteristic length, and η the Kolmogorov scale. By letting

$\eta = \left(\frac{\nu^3}{\varepsilon} \right)^{1/4}$, where ν is the kinematic viscosity and ε is the dissipation per unit mass, which is

approximated by $\varepsilon \sim \frac{u^3}{L}$ where u is the characteristic velocity of the flow. The number of points

for three dimensions can be rewritten as:

$$N \sim \left(\frac{L}{\eta} \right)^3 \sim \left(\frac{uL}{\nu} \right)^{9/4} \sim \text{Re}^{9/4} \quad (2.62)$$

It can be seen that for high Reynolds numbers, DNS is impractical due to the amount of nodes. To overcome this problem, we turn to large eddy simulation (LES).

2.6.3 Large Eddy Simulation (LES)

Large eddy simulation (LES) separates the turbulence into large and small eddies. LES resolves the large eddies and the small eddies are modeled using a sub-grid scale model. LES is a

transient turbulence model that takes the best of RANS and DNS. Its main advantage is that it is highly adaptable and more accurate than RANS and its computational cost, while high, is less than that of DNS. The small eddies can be modeled using a form of the Smargorinsky method, which we will now go into more detail.

Chapter 3: LES Modelling Approach

3.1 Filtering Definition and Properties

In LES, the turbulent flow is separated into large and small scales based on a cut-off length. This cut-off length is assumed to be equal to the size of the computational grid as illustrated in Figure 3.1. The scales which have a characteristic size greater than the cut-off length are then called large or resolved scales. The others are known as small or sub-grid scales. The large turbulent eddies are resolved and the smaller eddies are modeled which results in less nodes.

Mathematically the scale separation is achieved by applying a scale high-pass filter to the exact solution. This is then represented by the convolution product in which the resolved component of an arbitrary variable $\bar{\Phi}(x_j, t)$ is defined by:

$$\bar{\Phi}(x_j, t) = \int_{\Omega} G(x_j - x'_j) \Phi(x'_j, t) dx'_j \quad (3.1)$$

Where Ω is the computational domain, G is the convolution kernel with a characteristic filter width Δ and $x_j = (x, y, z)$ are the axes of the Cartesian coordinate system.

There are several properties that the filter needs to demonstrate as the filtering operation is applied to the Navier-Stokes equations. They are:

$$\bar{a} = a \quad \text{i.e. conservation of constants} \quad (3.2)$$

$$\overline{\Psi + \Phi} = \bar{\Psi} + \bar{\Phi} \quad \text{i.e. linearity} \quad (3.3)$$

$$\overline{\frac{\partial \Phi}{\partial x_j}} = \frac{\partial \bar{\Phi}}{\partial x_j} \quad \text{i.e. commutation with derivation} \quad (3.4)$$

Applying a filter to the Navier-Stokes equations, a part of the exact solution spectrum is lost as indicated in Figure 3.1. It follows that in order to represent the effect of the sub-grid scales in the total energy spectrum; a model which normally takes the form of a statistical description is required.

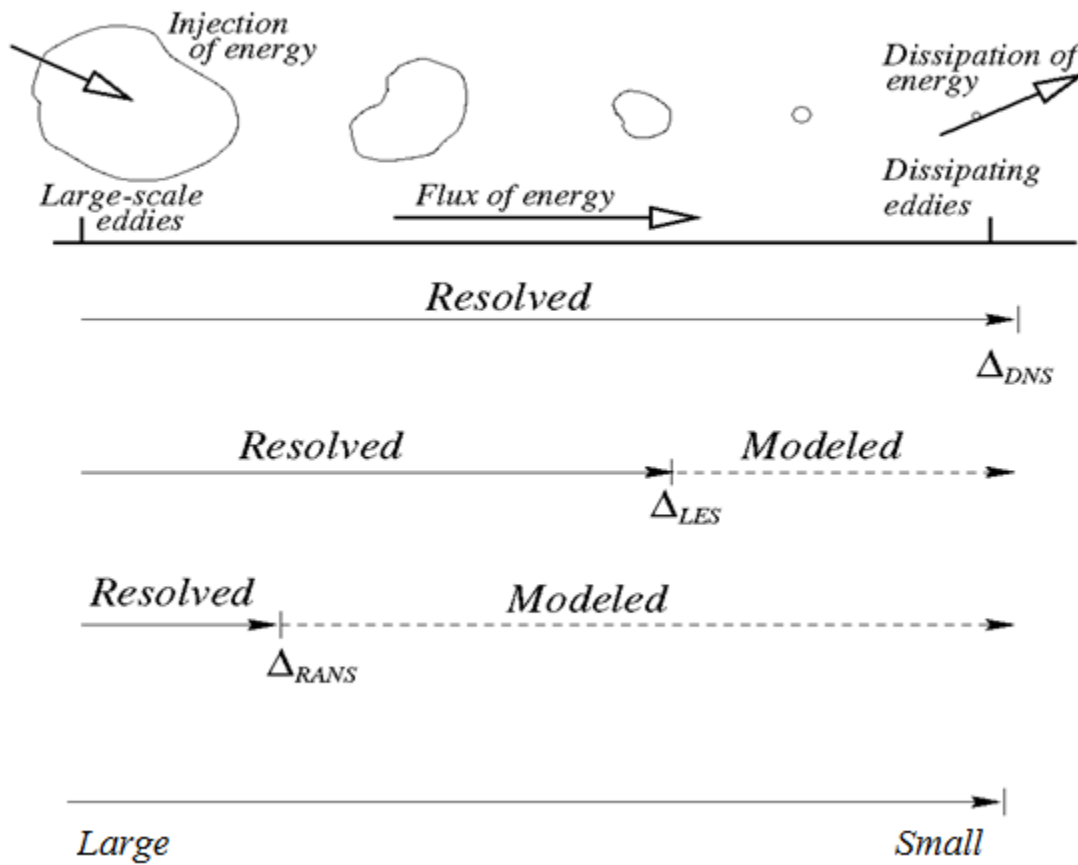


Figure 3.1 Prediction of RANS, LES, and DNS

This filtering technique involves a change of variable in which the filtered variables are weighted by density. Since the approach is similar to Favre averaging, it is commonly called Favre filtering. Mathematically, this is written as:

$$\tilde{\Phi}(x_j, t) = \frac{\overline{\rho\Phi(x_j, t)}}{\bar{\rho}} \quad (3.5)$$

$\tilde{\Phi}(x_j, t)$ is the Favre filtered quantity obtained from the grid filtered component $\bar{\Phi}(x_j, t)$ and ρ is the density. It should be noted that (\sim) is linear but does not commute with the derivative operator in space or time. We now rewrite the instantaneous variable $\Phi(x_j, t)$ as:

$$\Phi(x_j, t) = \bar{\Phi}(x_j, t) + \Phi''(x_j, t) \quad (3.6)$$

In which $\Phi''(x_j, t)$ represents the SGS component at a length smaller than the grid filter width Δ .

3.2 Favre-filtered Navier-Stokes Equations

The conservation laws of continuity, momentum, and energy can be expressed as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0 \quad (3.7)$$

$$\frac{\partial(\rho u_j)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_i} = -\frac{\partial p}{\partial x_j} + \frac{\partial \sigma_{ij}}{\partial x_i} + \rho g_i \quad (3.8)$$

$$\frac{\partial(\rho C_p T)}{\partial t} + \frac{\partial(\rho C_p u_i T)}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\mu C_p}{\text{Pr}} \frac{\partial T}{\partial x_i} \right) \quad (3.9)$$

where ρ is the density, u_j is the velocity vector, p is the pressure, T is the temperature, Pr is the Prandtl number, C_p is the specific heat at constant pressure. σ_{ij} is evaluated using Stoke's hypothesis:

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \quad (3.10)$$

In addition, μ is the dynamic viscosity, which can be written as a function of temperature using the Sutherland formula:

$$\mu = 18.27 \times 10^{-6} \frac{T + \lambda}{291.15 + \lambda} \left(\frac{T}{291.15} \right)^{3/2} \quad (3.11)$$

where λ varies depending on the ideal gas used; for air $\lambda = 120$.

To formulate the Navier Stokes equations which describe the evolution of the resolved scales, Favre-filtering is applied to the original equations. The Favre-filtered equations of continuity, momentum, and energy are obtained from equation (3.7), (3.8), and (3.9) are:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_j)}{\partial x_j} = 0 \quad (3.12)$$

$$\frac{\partial (\bar{\rho} \tilde{u}_j)}{\partial t} + \frac{\partial (\bar{\rho} u_i u_j)}{\partial x_i} = - \frac{\partial \bar{p}}{\partial x_j} + \frac{\partial \tilde{\sigma}_{ij}}{\partial x_i} + \bar{\rho} g_i \quad (3.13)$$

$$\frac{\partial (C_p \bar{\rho} \tilde{T})}{\partial t} + \frac{\partial (\bar{\rho} C_p u_i T)}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\mu C_p}{\text{Pr}} \frac{\partial \tilde{T}}{\partial x_i} \right) \quad (3.14)$$

We can see that there are two nonlinear terms, one each in the momentum and energy equation, $\bar{\rho} u_i u_j$ and $\bar{\rho} u_i T$ respectively. These nonlinear terms must be expressed in terms of the resolved scale variables and their fluctuating parts. We use Leonard's decomposition [10] to decompose these nonlinear terms.

3.2.1 Leonard's Decomposition

The nonlinear term is first rewritten as a function of the filtered velocity vector u_j and its fluctuating part u_j'' . Applying this to the first nonlinear term in the momentum equation

$$\bar{\rho} u_i u_j = \bar{\rho} (\tilde{u}_i + \tilde{u}_i'') (\tilde{u}_j + \tilde{u}_j'') = \bar{\rho} (\tilde{u}_i \tilde{u}_j + \tilde{u}_i u_j'' + \tilde{u}_j u_i'' + u_i'' u_j'') \quad (3.15)$$

All terms on the right hand side can be expressed in terms of the SGS tensor τ_{ij} as

$$\tau_{ij} = L_{ij} + C_{ij} + R_{ij} = \bar{\rho} (u_i u_j - \tilde{u}_i \tilde{u}_j) \quad (3.16)$$

where

$$\begin{aligned} L_{ij} &= \bar{\rho} (\tilde{u}_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j) \\ C_{ij} &= \bar{\rho} (\tilde{u}_i u_j'' + \tilde{u}_j u_i'') \\ R_{ij} &= \bar{\rho} u_i'' u_j'' \end{aligned} \quad (3.17)$$

where L_{ij} is the Leonard [10] tensor, C_{ij} is the cross-stress tensor, and R_{ij} is the Reynolds sub-grid tensor. L_{ij} represents the interactions between the large scales, C_{ij} represents the interactions between the large and small scales, and R_{ij} represents the interactions between the sub-grid scales. This decomposition technique allows the momentum equation (3.13) to be rewritten as:

$$\frac{\partial(\bar{\rho} \tilde{u}_j)}{\partial t} + \frac{\partial(\bar{\rho} \tilde{u}_i \tilde{u}_j)}{\partial x_i} = -\frac{\partial \bar{p}}{\partial x_j} + \frac{\partial \tilde{\sigma}_{ij}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x} + \bar{\rho} g_i \quad (3.18)$$

where $\tau_{ij} = \bar{\rho} (u_i u_j - \tilde{u}_i \tilde{u}_j)$

Using the same technique, the energy equation (3.14) can be expressed as:

$$\frac{\partial(C_p \bar{\rho} \tilde{T})}{\partial t} + \frac{\partial(\bar{\rho} C_p \tilde{u}_i \tilde{T})}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\mu C_p}{\text{Pr}} \frac{\partial \tilde{T}}{\partial x_i} \right) - \frac{\partial q_i}{\partial x_i} \quad (3.19)$$

where $q_i = \bar{\rho}(u_i T - \tilde{u}_i \tilde{T})$

The effects of the SGS terms are represented in τ_{ij} and q_i .

In order to ensure that the dynamics of the resolved scales remain accurate, the SGS terms have to be considered in the solutions of the governing equations. Since the small scales are more isotropic than the large scales, they can be modeled using an SGS model.

3.3 Sub-grid Scale Viscosity

The purpose of SGS models is to represent the energy loss due to the separation between large and small scales. These models do not attempt to produce the SGS stresses precisely; rather they take into account their effects of the resolved scale.

The Boussinesq approximation states the following:

$$\begin{aligned}\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} &= -2\mu_{sgs}\tilde{S}_{ij} \\ \tilde{S}_{ij} &= \frac{1}{2}\left(\frac{\partial\tilde{u}_i}{\partial x_j} + \frac{\partial\tilde{u}_j}{\partial x_i}\right)\end{aligned}\tag{3.20}$$

It should be noted that the isotropic part of the SGS stress tensor, τ_{kk} , is commonly neglected due to the incompressibility of the sub-grid scales [11]. We now have a way to model τ_{ij} but we have a new term that also needs to be modeled, μ_{sgs} .

3.3.1 Smagorinsky Model

A widely used SGS in LES modeling is the Smagorinsky model [33] where μ_{sgs} is expressed as:

$$\begin{aligned}\mu_{sgs} &= \bar{\rho} (C_{SM} \Delta)^2 |\tilde{S}| \\ |\tilde{S}| &= \sqrt{2\tilde{S}_{ij}\tilde{S}_{ij}}\end{aligned}\tag{3.21}$$

To obtain a proper value for C_{SM} it is assumed that:

- The cut-off wave number k_c lies within $k^{-5/3}$ Kolmogorov cascade in the energy spectrum
- The ensemble-averaged SGS dissipation ξ_{sgs} is identical to the dissipation of the spectrum ξ

This gives an approximate value for C_{SM} as:

$$C_{SM} = \frac{1}{\pi} \left(\frac{3C_k}{2} \right)^{-3/4}\tag{3.22}$$

where C_k is the Kolmogorov constant. The Smargorinsky method yields good results for homogenous turbulent flow when the cut-off is placed far enough into the inertial range of the spectrum [12-14]. In this case, $C_k \approx 1.4$ which leads to $C_{SM} \approx 0.17$. The Smagorinsky model is formulated on the assumption that the flow is turbulent, fully-developed, and isotropic. This is seldom the case. Figure 3.2 below shows the Kolmogorov scale.

Another approach which is used by many to obtain a natural adaption of the model for inhomogenous flows is to determine the model coefficient as a function of space and time. This eradicates the need of an ad hoc and a priori prescription of the model coefficient. It should be noted that a dynamic procedure does not change the form of the original model, but rather it aims to adapt the model to the local structure of the flow dynamics. Thus, any SGS model with a constant coefficient can be used as a model for a dynamic procedure.

In order to overcome the shortcomings of the Smargorinsky model, Vreman [15] proposed a different SGS model which guarantees vanishing SGS dissipation in regions where the flow is laminar.

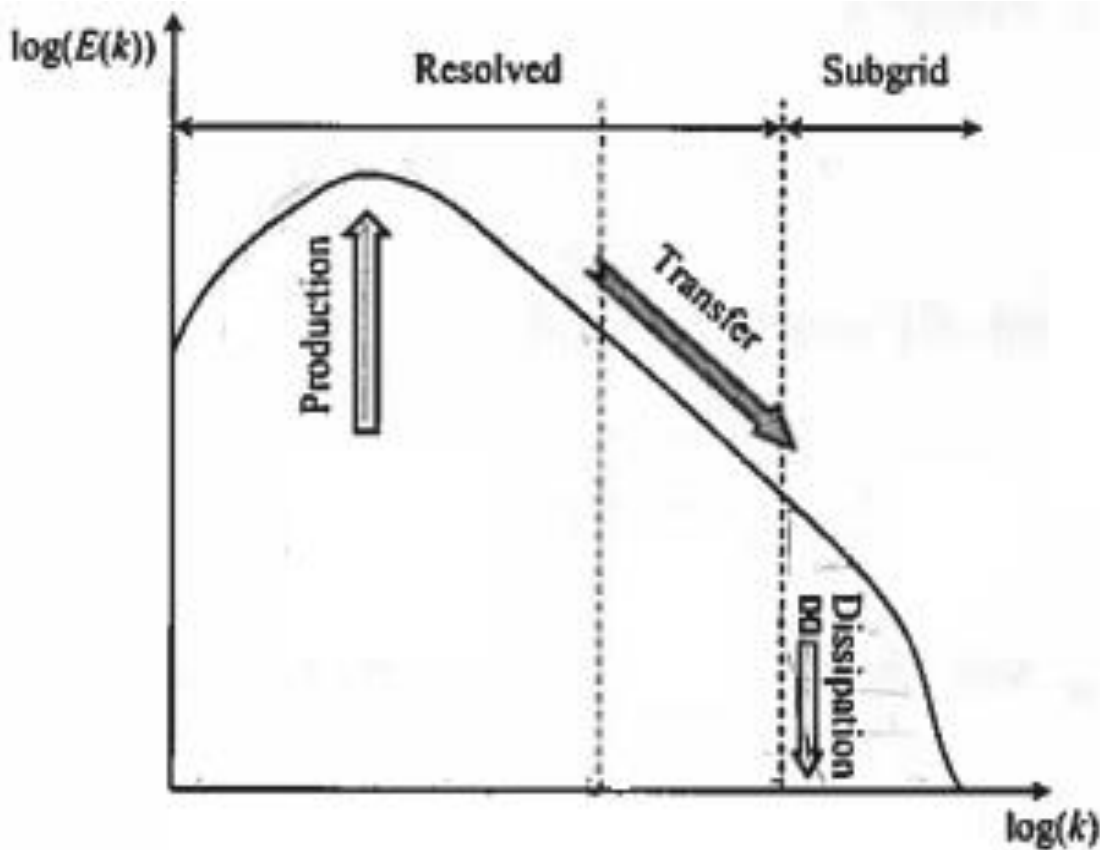


Figure 3.2 Kolmogorov Scale of Eddy Energy

3.3.2 Vreman Model

Unlike Smargorinsky, the Vreman [15] model does not involve any explicit filtering, averaging or clipping procedure to stabilize the numerical procedure. Furthermore, it is easy to compute since it does not need more than the local filter width and the first order derivatives of the filtered velocity field. The Vreman model is defined as:

$$\begin{aligned}
\mu_{sgs} &= \rho C_{VM} \sqrt{\frac{\beta_\beta^g}{\alpha_{ij} \alpha_{ij}}} \\
\alpha_{ij} &= \frac{\partial \tilde{u}_j}{\partial x_i} \\
\beta_{ij} &= \Delta_m^2 \alpha_{mi} \alpha_{mj} \\
\beta_\beta^g &= \beta_{11} \beta_{22} - \beta_{12}^2 + \beta_{11} \beta_{33} - \beta_{13}^2 + \beta_{22} \beta_{33} - \beta_{23}^2
\end{aligned} \tag{3.23}$$

The superscript g denotes a grid-filtered quantity and C_{VM} is related to C_{SM} by $CL\nu = 2.5C_{SM}^2$, giving $C_{VM} \approx 0.072$ for homogenous turbulence. The main feature that the Vreman model has over the Smagorinsky model is its ability to compute zero SGS viscosity in laminar flow. It should be noted that both the Smagorinsky and Vreman model use constant coefficients, but these coefficients can be localized when implementing a dynamic model. Dynamic models provide a more accurate local coefficient for flow features in that area as a constant coefficient over the entire flow is not realistic.

The Vreman model is self-dampening, hence laminar flow can be modeled with it including the laminar sublayer. Therefore no wall function is used to calculate the flow near the wall, making it a good choice for wall bounded flow. Other LES models require wall functions or hybrid RANS systems to simulate the wall flow.

Chapter 4 – Results and Discussion

We use the KIVA code to examine two different methods using a two equation $k-\omega$ RANS with wall functions and LES simulation to model the Navier-Stokes equations for both laminar and turbulent flow in a square cavity with moving velocity on top. KIVA uses a Predictor-Corrector split method along with Petrov-Galerkin weighting on the advection terms. The main solver uses a Krylov solver to solve the matrices. The cavity benchmark represents a difficult benchmark because of the singularity it contains in the corner. The first step is to verify that the solutions are indeed the right solution. Several cases at different Reynolds numbers are shown and compared to a $k-\omega$ RANS as well as an LES simulation. COMSOL, a popular FEM model, is also used to obtain a solution using a two step equation models of $k-\omega$ and $k-\epsilon$ RANS. All simulations are compared to Ghia [37] using the centerline geometric profile versus velocity.

4.1 Geometry

As shown below in Figure 5.1, we have defined the cavity with a height of 0.1 and a free stream velocity from 0.5017 to 1.5675, with the corresponding Reynolds from 3,200 to 10,000 respectively. All solutions used a dynamic viscosity of $1.846e-04$ to ensure proper Reynolds numbers.

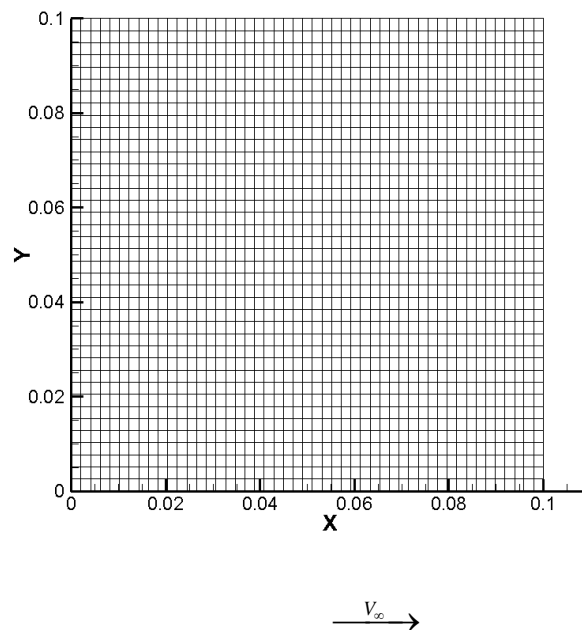


Figure 4.1 Mesh and geometry of cavity lid

4.2 Mesh and Velocity Contours

Using the KIVA code, two separate simulations were tested using LES and $k-\omega$ RANS with adaptation. Figure 2 and 3 below show the velocity profile of the cavity driven lid which will eventually be compared to Ghia [37]. Notice that for both, no singularity exists in the corner of the cavity which is typically found in most cases and notice that between Figure 2 and 3, the LES simulation has less overall damping than the RANS. This is especially evident at $Re = 3200$ and lower Reynolds numbers since at higher Reynolds numbers, the inertial forces are much larger than the viscous forces.

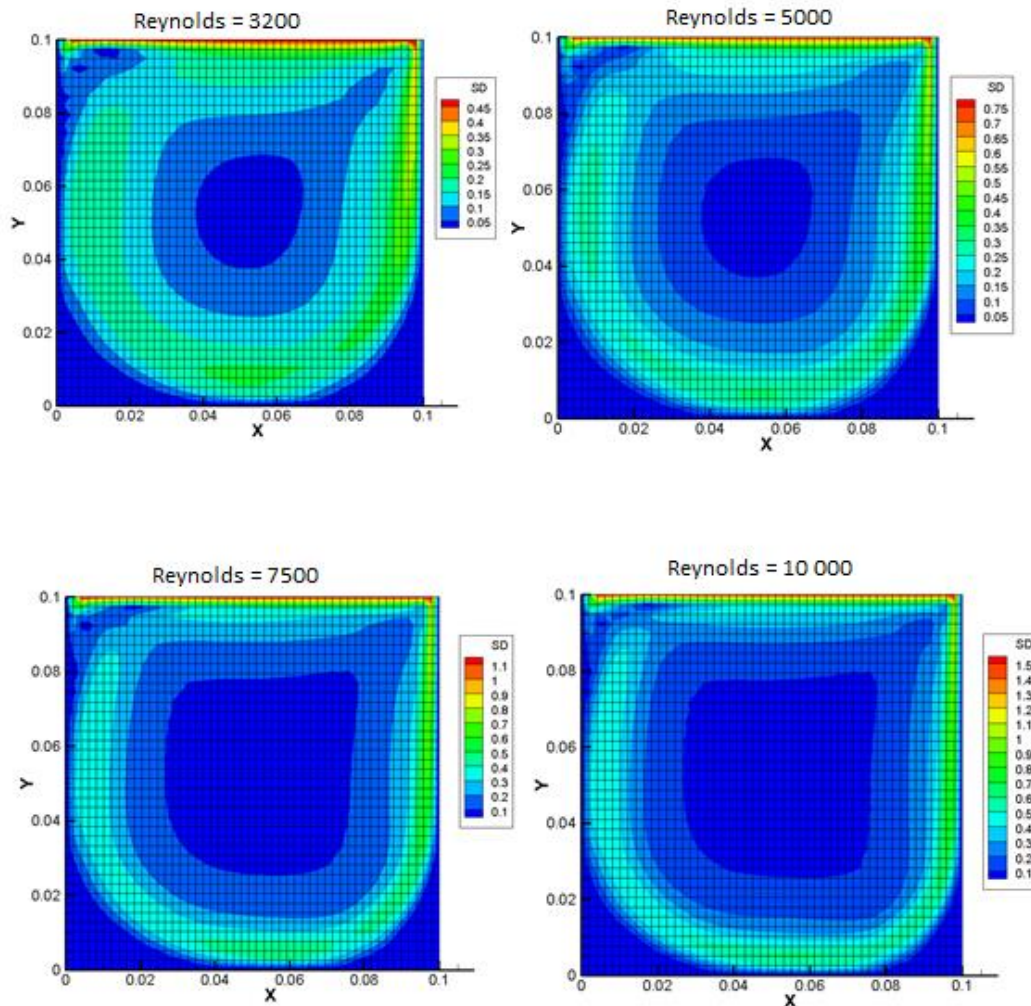


Figure 4.2 Mesh and velocity contour at different Reynold's numbers using LES

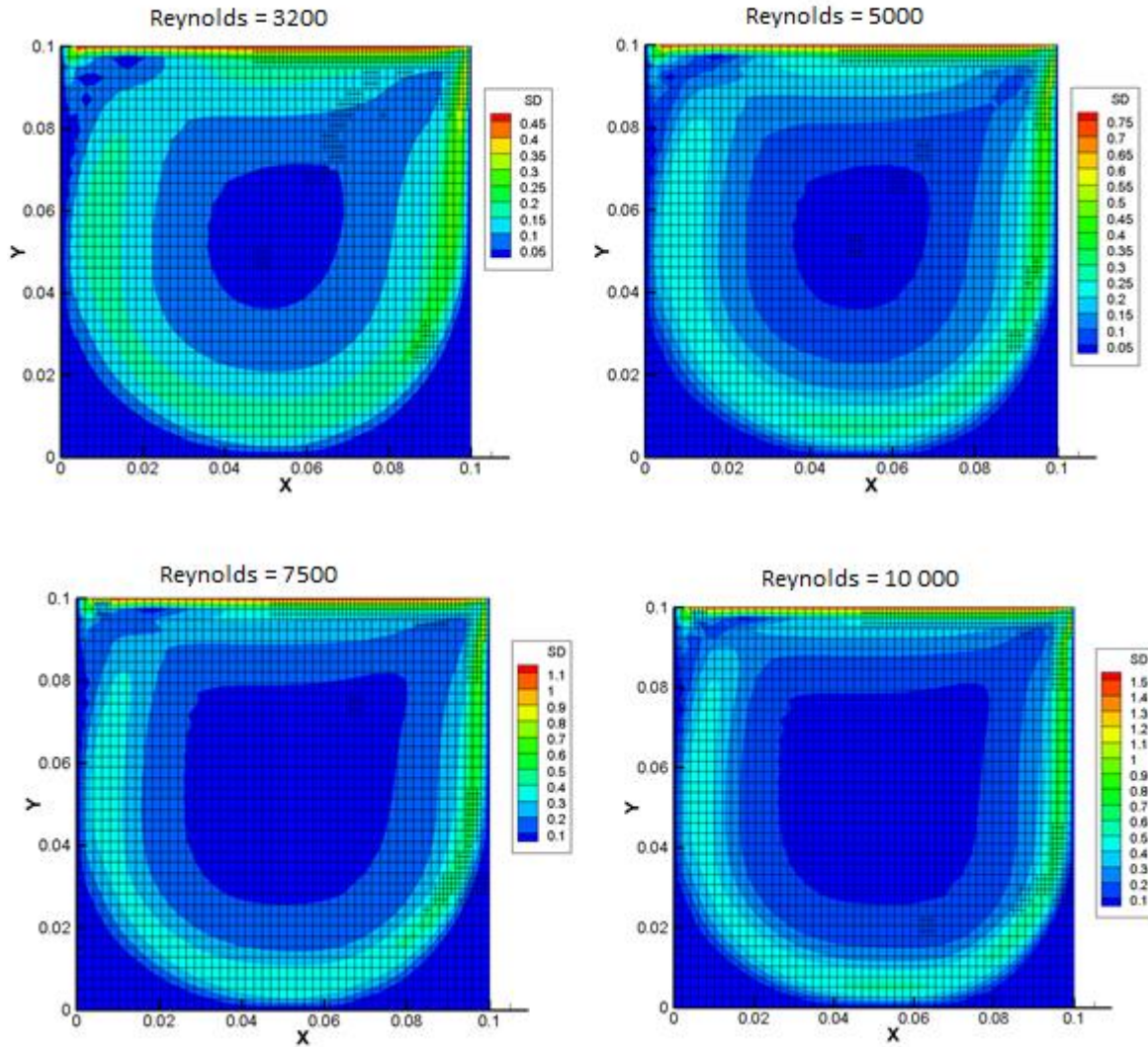


Figure 4.3 Mesh and velocity contour at different Reynolds using RANS simulation with adaptation

Examining Figures 2 and 3, our results are consistent with benchmark profiles of Ghia [37] at different Reynolds numbers. The cavity driven lid is a particularly interesting problem because of the singularity in the top right corner of the lid where the velocity is affected by the free stream velocity and the velocity of wall, which is zero. Secondly, we are also reaching higher Reynold's numbers which impose a potential problem in resolving the onset of turbulence.

4.3 Comparison of U versus Y and X versus V

While looking at velocity profiles is informative and can lead us to detect major problems if any, it does not tell us anything about the accuracy of the system. For that, we have to delve deeper into the profile. We compared the velocity profile to the geometric center of the cavity driven lid using Ghia [37]. Results are shown below in Figure 4 and 5. The red line shows the LES simulation and the blue line shows the RANS simulation.

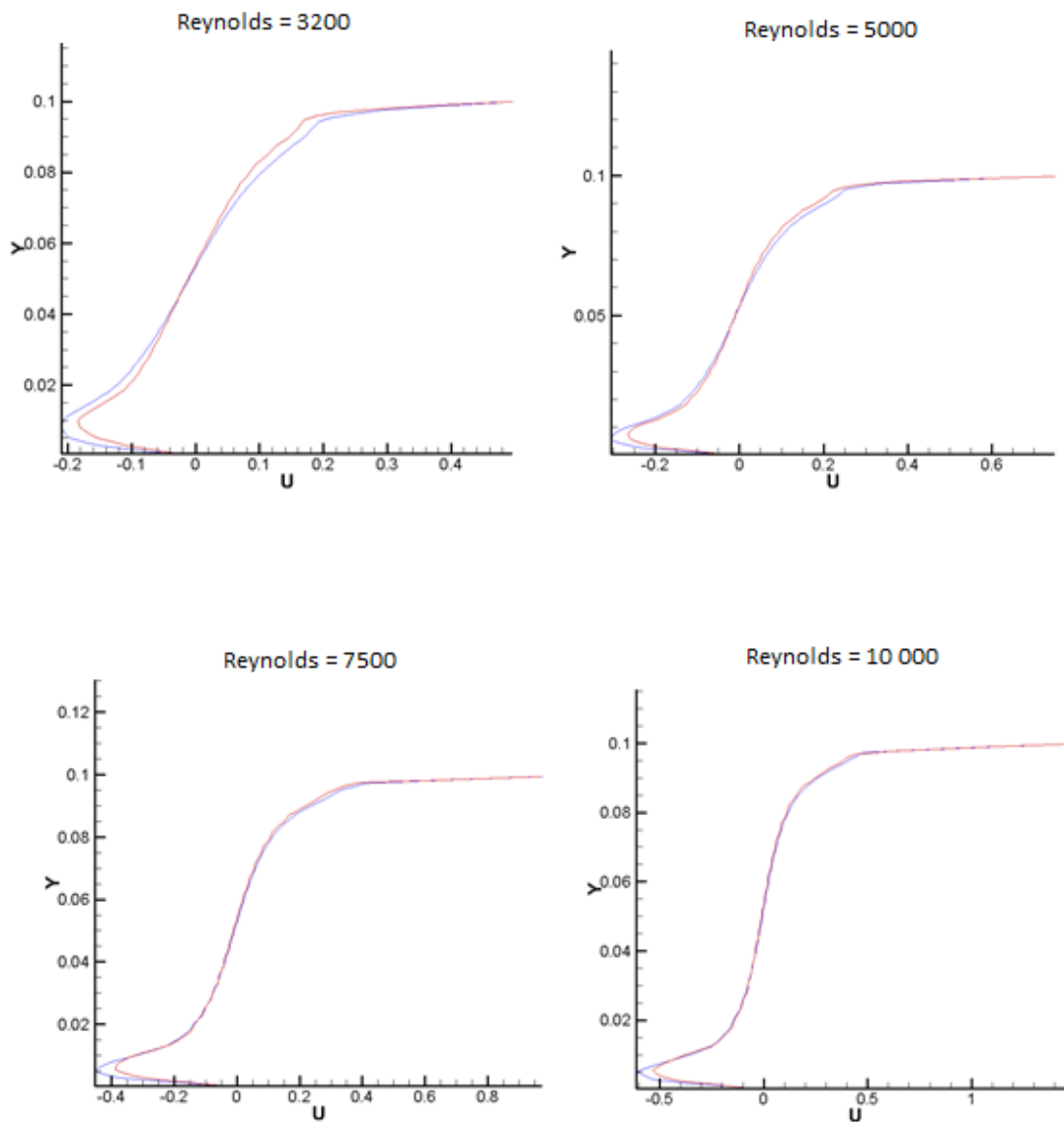


Figure 4.4 U versus Y at different Reynolds using both LES and RANS

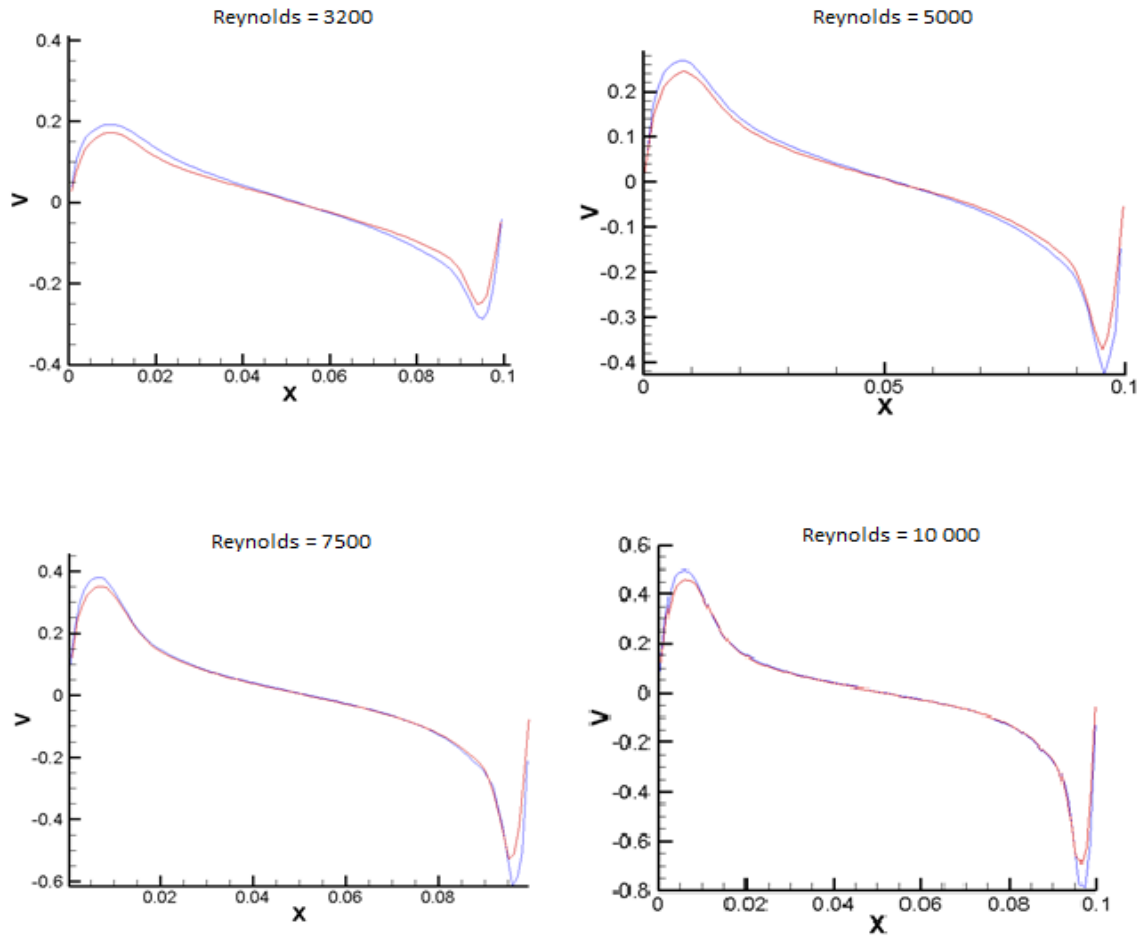


Figure 4.5 X versus V at different Reynolds using both LES and RANS

As shown in Figure 5, the velocity profiles of both X and Y versus the vertical and horizontal geometric means are a close match, indicating results for LES and RANS are consistent with each other. When compared to Ghia [37], we see that while the results are close, they do not match his results. The values of Ghia are higher than the values of the LES or RANS simulation. In addition, the curve has more linearity to it. The reason for this is because Ghia used a 2nd order central finite difference scheme for his simulation with a 1st order upwind scheme. Secondly, he also used a fully laminar solution method in a turbulent domain. Last, he had a smoothing factor to ensure stability in his procedure using a coupled strongly implicit method.

4.4 Comparison to COMSOL Simulation

Simulations used previously were also compared to a COMSOL simulation under the same conditions. Figure 6 below shows the mesh used comprised of approximately 66,000 grid cells using an automatic very fine grid generator. When compared against a fine grid of approximately 20,000 grid cells, no changes were detected in convergence rate. To be on the cautious side, the results were done using the very fine grid of approximately 66,000 grid cells. COMSOL uses a pressure-correction method to solve the Navier-Stokes equations. It uses nested loops with the inner and outer loop. It solves the momentum equation based on a provisional velocity and pressure taken from the previous loop and then corrects it by plugging it into the continuity equation. When dealing with incompressible flow, the divergence of the velocity must equal zero.

Figure 7 and Figure 8 show the velocity and pressure contour of the COMSOL simulation. Streamlines were added to show the proper representation of the flow due to the singularity of the top right corner of the simulation. That point exists because COMSOL is incapable of determining whether there is a moving boundary condition there or a no-slip boundary condition. The pressure contour was added in to better show this singularity.

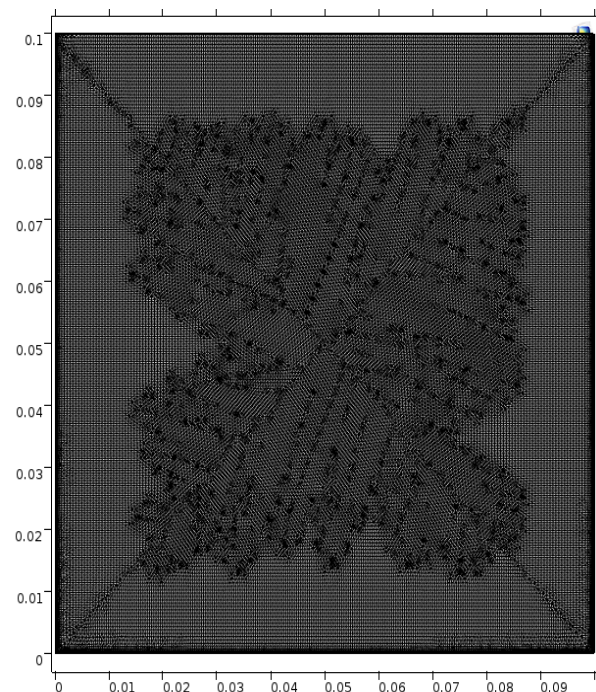


Figure 4.6 Mesh of COMSOL simulation

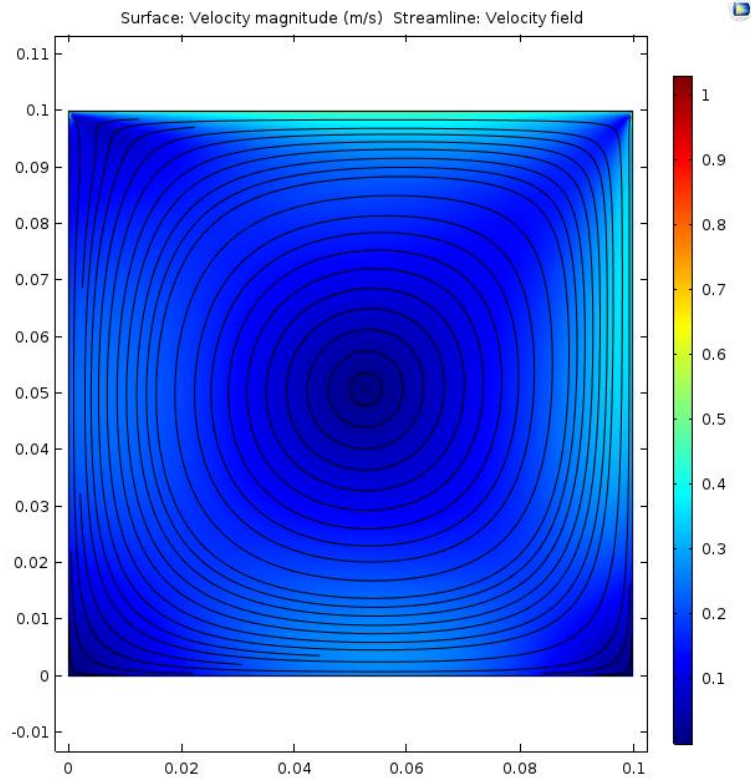


Figure 4.7 Velocity profile of COMSOL simulation for $Re=3,200$

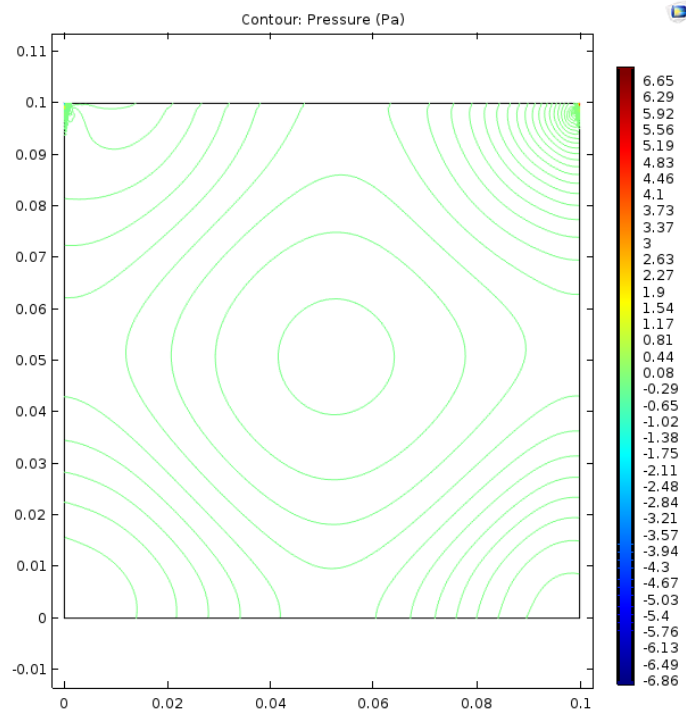


Figure 4.8 Pressure contour of COMSOL simulation for $Re=3,200$

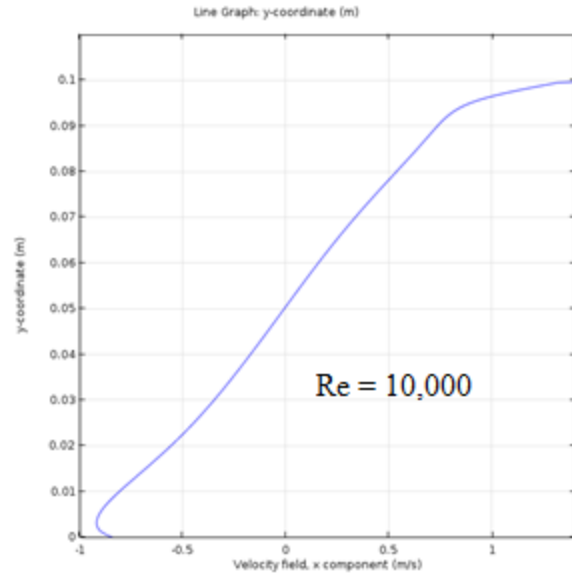
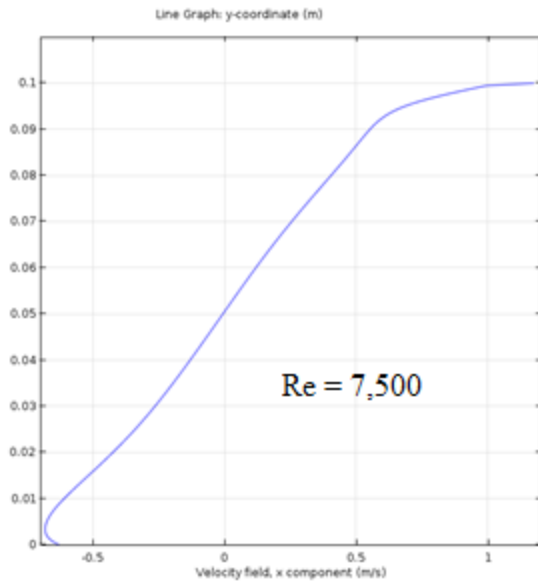
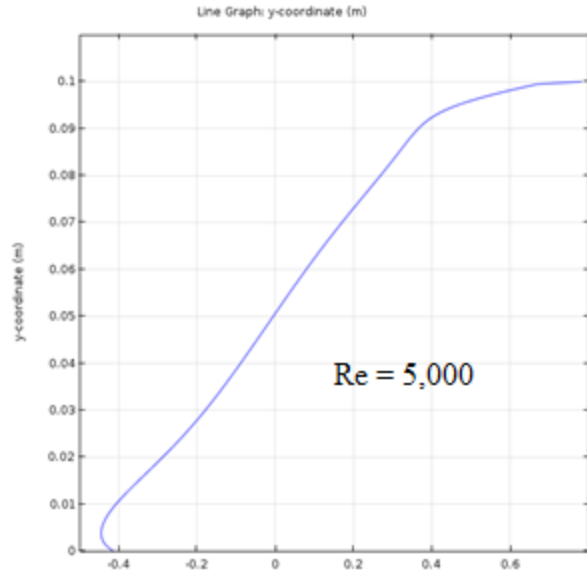
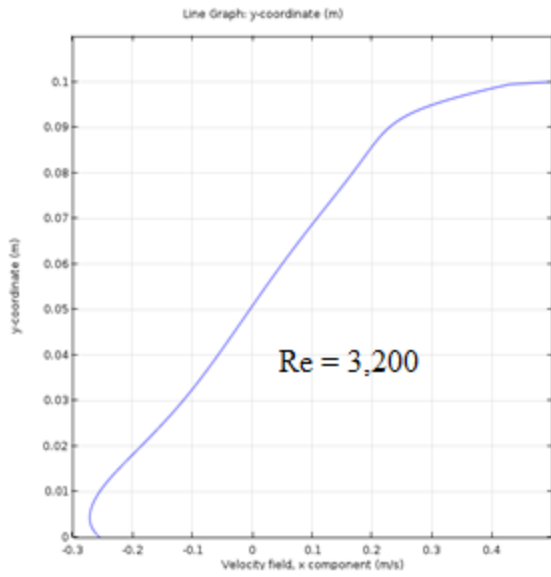


Figure 4.9 U versus Y at different Reynolds using COMSOL

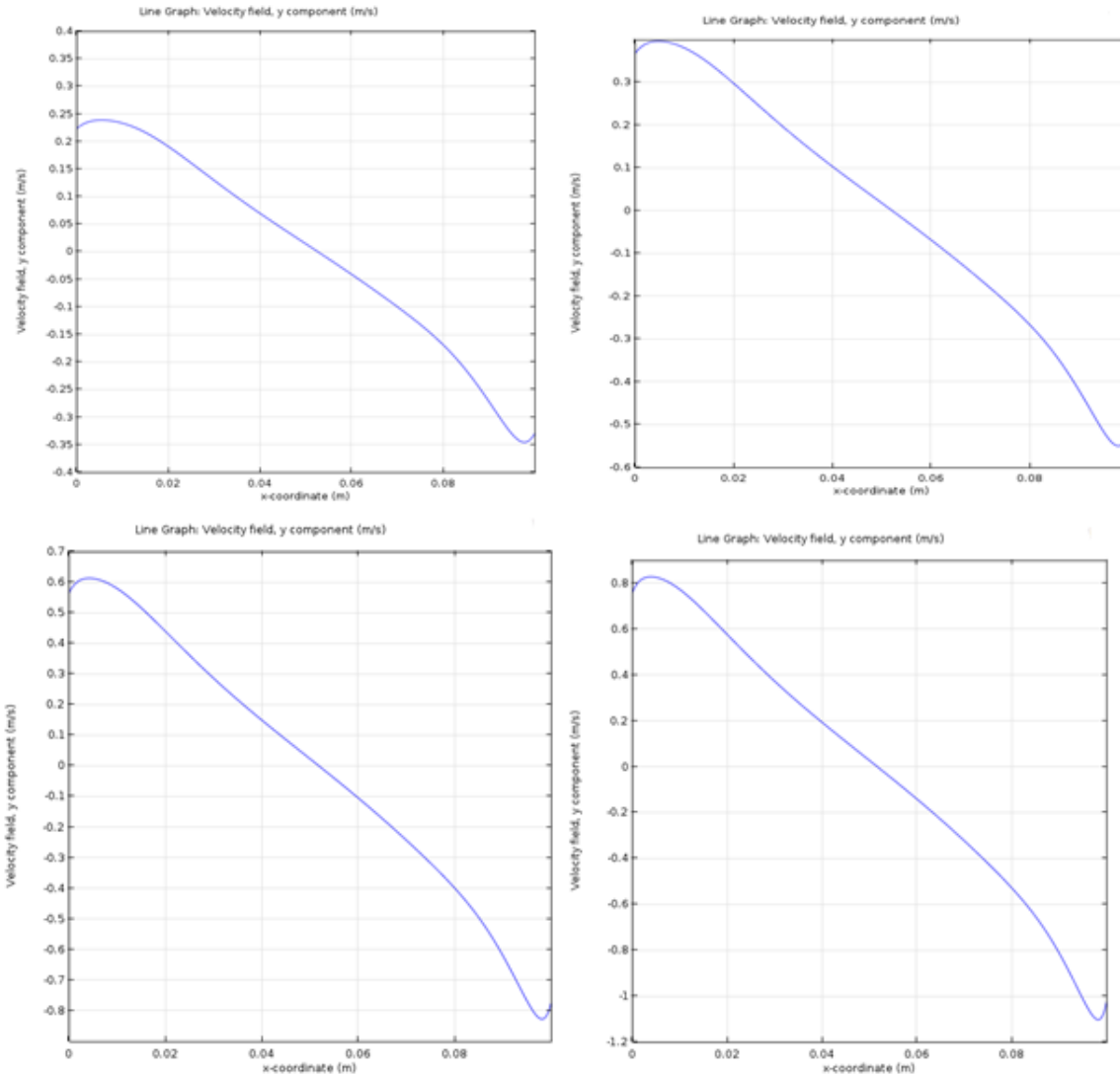


Figure 4.10 X versus V at different Reynolds using COMSOL

Figures 9 and 10 show the COMSOL results for the cavity driven lid comparing U versus Y and X versus V at the geometric center profile. We see that in COMSOL, much like in KIVA, the solutions are not as laminar as in Ghia's [37] results. In addition COMSOL has higher values at the boundaries. This is because COMSOL uses a law of the wall function as well as RANS modeling.

Chapter 5 Conclusion

Comparisons of KIVA, COMSOL, and Ghia [37] have been performed using RANS and LES for a cavity driven flow with Reynolds ranging from 3,200 to 10,000. The turbulence models used were $k-\omega$ and $k-\epsilon$ for RANS and Vreman for LES. One of the advantages of using an LES model over RANS is that RANS simulates the vortex over an average time result whereas the LES can simulate different sizes and scales. In addition, LES can simulate transient flow while RANS cannot, thus predicting turbulence with a much higher accuracy. Additionally, RANS is incapable of modeling close to the boundary layer of the wall.

When comparing the geometric center profiles of all cases, Ghia [37] only uses a laminar flow test case. That is to be expected as they did not have the computational power required to run any high turbulence simulation. COMSOL while more accurate, was still inaccurate near the wall. The KIVA code employed both RANS with wall bounded functions and LES. LES did a better job of modeling flows near the wall as well as more realistically modelling the vortices. Overall, all results were in good agreement with each other.

The next step is to utilize the LES model in a turbulent flow using a dynamic Vreman model for compressible flows at high Mach numbers. Initial tests appear promising but more work needs to be done to improve the accuracy of the model.

Additional work also needs to be done to parallelize for high performance computing. Since problems such as turbulent fluid flow are time intensive, reducing the amount of time using multiple processors would be extremely beneficial. Other work also includes optimizing p-adaptation in KIVA to enhance convergence rate.

References

- [1] Oliver Rübenkönig, The Finite Difference Method (FDM) - An introduction, (2006) Albert Ludwigs University of Freiburg.
- [2] LeVeque, Randall (2002), Finite Volume Methods for Hyperbolic Problems, Cambridge University Press.
- [3] Reddy, J.N. (2005). An Introduction to the Finite Element Method (Third ed.). McGraw-Hill.
- [4] Carrington, D., Wang, X., and Pepper, D. A Fractional Step hp-adaptive Finite Element Method for Turbulent Reactive Flow. 2006.
- [5] Zienkiewicz, O. C. and Morgan K. Finite Elements and Approximations, John Wiley & Sons, 1993.
- [6] Zienkiewicz O.C., Zhu, J. Z., Gong, N. G., “Effective and Practical h-p Version Adaptive Analysis Procedures for the Finite element Method,” International Journal for Numerical Methods in Engineering, Vol. 28, John Wiley & Sons, ltd., pp. 879-891.
- [7] Szabo, B., and Babuska, I., Finite Element Analysis, John Wiley & sons, inc., N.Y., NY, 1991.
- [8] Pepper, D. W. and Heinrich, J. C., The Finite Element Method, Basic Concepts and Application, Taylor and Francis Publishing, Hemisphere Publishing Co., 1992.
- [9] Beer, G. and Watson, J. O. Introduction to Finite and Boundary Element Methods for Engineers, John Wiley & Sons, Ltd, 1992.
- [10] A. Leonard, Energy cascade in large-eddy simulations of turbulent fluid flow, Advances in Geophysics, vol. 18, pp. 237-248, 1974.
- [11] G. Erlebacher, M. Y. Hussaini, V. G. Speziale and T. A. Zang, Towards the large-eddy simulations of compressible turbulent flows, Journal of Fluid Mechanics, vol. 238, pp. 155-185, 1992.
- [12] R. A. Clark, J. H. Ferziger, and W. C. Reynolds, Evaluation of subgrid-scale models using an accurately simulated turbulent flow, Journal of Fluid Mechanics, vol.91, pp 1-16, 1997.
- [13] D. K. Lilly, On the application of the eddy viscosity concept in the inertial subrange of turbulence, NCAR Report No. 123, 1966.
- [14] D. K. Lilly, The representation fo small-scale turbulence in numerical simulation experiments. Proceedings of the IBM Scientific Computing Symposium and Environmental Science, 1967.

- [15] A. W. Vreman, An eddy-viscosity subgrid scale model for turbulent shear flow: Algebraic theory and applications, *Physics of Fluids*, vol. 16, pp 3670-3681, 2004.
- [16] Carrington, D.B., and Pepper D. W., Convective Heat Transfer Downstream of a 3-D Backward-Facing Step, *Numerical Heat Transfer, Part A*, Vol. 41, pp. 555-578, 2002.
- [17] Kelly, D. W., Nakazawa O. C., Heinrich J. C., A note on Upwinding and Anisotropic Balancing Dissipation in the finite element approximations to convective diffusion problems, *International Journal for Numerical Methods in Engineering*, vol. 15, John Wiley & Sons, Ltd., pp. 1705-1711, 1980.
- [18] Chorin, A. J., "Numerical Solution of the Navier-Stokes Equations, *Mathematics of Computation*, Vol. 22, American Mathematical Society, Providence, RI, pp. 745-762, 1968.
- [19] Lohner, R., A Fast Finite Element Solver for Incompressible Flows, AIAA-90-0398, 28th Aerospace Sciences Meeting, 1990.
- [20] Gresho, P. M., Chan, S. T., On the Theory of Semi-Implicit Projection Methods for Viscous Incompressible Flow and its Implementation via a Finite Element Method that also introduces a Nearly Consistent Mass Matrix, Part 2: Implementation, *International Journal for Numerical Methods in Fluids*, Vol. 11, John Wiley & Sons, Ltd., pp. 621-659, 1990.
- [21] Ramaswamy, B., Jue, T. C., Akin, J. E., Semi-Implicit and Explicit Finite element Schemes for Coupled Fluid/Thermal Problems, *International Journal for Numerical Methods in Engineering*, Vol. 34, John Wiley & Sons, Ltd., pp. 675-696, 1992.
- [22] Gresho, P. M., A Modified Finite Element Method for Solving the Incompressible Navier-Stokes Equations, *Lectures in Applied Mathematics*, Vol. 22, Part 1, American Mathematical Society, Providence, RI, pp. 193-240, 1985.
- [23] Chorin, A.J., 1969, "On the Convergence of Discrete Approximations to the Navier-Stokes Equations," *Mathematics of Computation*, Vol. 23, American Mathematical Society, Providence, R.I., pp 341-353.
- [24] Chorin, A. J., Manderson, J.E., *A Mathematical Introduction to Fluid Mechanics*, 3rd edition, Springer-Verlag, N.Y.,NY, pp.36-39, 1969.
- [25] Marchioro and Pulvirenti, *Mathematical Theory of Incompressible Nonviscous Fluids*, Springer-Verlag, N.Y., NY, pp. 49-50, 1994.
- [26] Pepper, D.W., Modeling of Natural Convection in Three Dimensions using a Time-Split Finite Element Method, *Numerical Heat Transfer*, Vol. 11, pp. 31-55, 1987.
- [27] Heinrich, J. C. and Yu, C.-c., Finite Elements Simulations of Bouyancy-Driven Flows with Emphasis on Natural Convection in a Horizontal Circular Cylinder, *Computational Methods and Applications in Mechanical Engineering*, Vol. 69, pp. 1-27, 1988.

- [28] Brooks, A.N., Hughes, T. J. R., Streamline Upwind/Petrov Galerkin Formulations for Convective Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations, Computer Methods in applied Mathematics and Engineering, Vol. 32, North-Holland, pp. 199-259, 1982.
- [29] Hughes, T. J. R., Recent Progress in the Development and Understanding of SUPG Methods with Special Reference to the Compressible Euler and Navier-Stokes Equations, International Journal for Numerical Methods in Engineering, Vol. 7, John Wiley & Sons, Ltd., pp. 1261-1275, 1987.
- [30] Fletcher, C. A. J., Computational Techniques for Fluid Dynamics: Volume 1- Fundamentals and General Techniques, 2nd edition, Springer-Verlag, N.Y., NY., 1991.
- [31] Hoffman, K. A., and Chiang, S. A. Computational Fluid Dynamics for Engineers, Vol. 1, Engineering Education Systems, Klaus A. Hoffman, 1993.
- [32] Hindmarsh, A. C., Gresho, P. M., and Griffiths, D. F., The Stability of Explicit Euler Time-Integration for Certain Finite Difference Approximations of the Multi-Dimensional Advection-Diffusion Equation, International Journal for Numerical Methods in Fluids, Vol. 4, John Wiley & Sons, Ltd., pp. 853- 897, 1984.
- [33] J. Smagorinsky, General circulation experiments with the primitive equations, Mon Weather Rev, vol. 91, pp. 99-164, 1963.
- [34] Oden, J. T., W. H. Wu, and M. Ainsworth, Three-Step h-p Adaptive Strategy for the Incompressible Navier-Stokes Equations, The IMA Volume, Modeling, Mesh Generation, and Adaptive Numerical.
- [35] Haug, H.-C. and Usmani, A. S., 1994 Finite Element Analysis for Heat Transfer, Theory and Software, Springer-Verlag, London, U.K.
- [36] Zienkiewicz O.C, Zhu, J. Z., 1987, "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis," International Journal for Numerical Methods in Engineering, Vol. 24, John Wiley & Sons, Ltd., pp. 337-357.
- [37] Ghia, Ghia, and Shin (1982), "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method", Journal of Computational Physics, Vol. 48, pp. 387-411.
- [38] Los Alamos National Labs. KIVA. < <https://lanl.gov/projects/feynman-center/deploying-innovation/intellectual-property/software-tools/kiva/index.php>>

Curriculum Vitae

5540 Del Rey Ave
Las Vegas, NV 89146

David Fyda

(702) 325-8759
david.fyda@gmail.com

Education

University of Nevada, Las Vegas

- Bachelor of Science in Mechanical Engineering 3.58 GPA Dec 2012
 - Dean's List for 2009 - 2012
- Masters of Science in Mechanical Engineering 3.52 GPA May 2016

Experience

CCSD – Findlay Middle School

Substitute Teacher Aug 2015 – Present

Prepare lesson plans appropriate to district curriculum aligned with Nevada standards. Ensure the opportunity for all students to learn in a positive environment. Maintain student progress and provide appropriate instruction. Work professionally with administrators, staff, parents, and community to promote a positive learning environment.

UNLV – Mechanical Engineering Department

Graduate Research Assistant Jan 2013 – Jan 2015

Developed, analyzed and optimized finite element models for large eddy simulation (LES) modeling using Navier-Stokes equations for fluid dynamics inside internal combustion engines. Validation and verification of linear and non-linear fluid dynamic models on combustion engines using heat transfer modeling. Worked with the KIVA and REACT, a transient, multi-phase, multi-component combustion modeling software that is widely used in the automotive industry. Analyzed variables including temperature, pressure, density, velocity, and concentration of chemical species.

Projects

NASA

Pipe Design Modeling Summer 2013

Modeled a theoretical design for a gas pipeline that is able to withstand the extreme conditions of space. Modeling was done as a proof of concept for the theoretical pipe design. The design needed to withstand huge amounts of pressure and be wrapped around a 4'' radius drumroll without failure. The finite element model developed shown the pipe design met the required criteria.

Guest Speaker

International Conference on Inverse Engineering Problems May 2014

Spoke about the benefits of coding a meshless 3-D model to characterize the three dimensional wind velocity field across an existing topography. A mountainous region north of Las Vegas was used to verify the model accuracy by using 25 existing towers sparsely located across the region to provide real data.

Skills

- Proficient in AUTOCAD, MATLAB, COMSOL, ANSYS, CFX, SOLIDWORKS, HYPERMESH, and FORTRAN
- Engineer in Training Certification - 0T6810 - Nevada