



Theses and Dissertations

---

2003-10-08

## Large Eddy Simulation Based Turbulent Flow-induced Vibration of Fully Developed Pipe Flow

Matthew Thurlow Pittard  
Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Mechanical Engineering Commons](#)

---

### BYU ScholarsArchive Citation

Pittard, Matthew Thurlow, "Large Eddy Simulation Based Turbulent Flow-induced Vibration of Fully Developed Pipe Flow" (2003). *Theses and Dissertations*. 103.  
<https://scholarsarchive.byu.edu/etd/103>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact [scholarsarchive@byu.edu](mailto:scholarsarchive@byu.edu), [ellen\\_amatangelo@byu.edu](mailto:ellen_amatangelo@byu.edu).

LARGE EDDY SIMULATION BASED TURBULENT FLOW-INDUCED  
VIBRATION OF FULLY DEVELOPED PIPE FLOW

by

Matthew T. Pittard

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

Brigham Young University

December 2003

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Matthew T. Pittard

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

\_\_\_\_\_  
Date

\_\_\_\_\_  
Jonathan D. Blotter, Chair

\_\_\_\_\_  
Date

\_\_\_\_\_  
R. Daniel Maynes

\_\_\_\_\_  
Date

\_\_\_\_\_  
Scott D. Sommerfeldt

BRIGHAM YOUNG UNIVERSITY

As chair of the candidates graduate committee, I have read the thesis of Matthew T. Pittard in its final form and have found that (1) its format, citations, and Bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

---

Date

---

Jonathan D. Blotter  
Chair, Graduate Committee

Accepted for the Department

---

Brent L. Adams  
Graduate Coordinator

Accepted for the College

---

Douglas M. Chabries  
Dean, College of Engineering and Technology

## ABSTRACT

### LARGE EDDY SIMULATION BASED TURBULENT FLOW-INDUCED VIBRATION OF FULLY DEVELOPED PIPE FLOW

Matthew T. Pittard

Department of Mechanical Engineering

Masters of Science

Flow-induced vibration caused by fully developed pipe flow has been recognized, but not fully investigated under turbulent conditions. This thesis focuses on the development of a numerical Fluid-Structure Interaction (FSI) model that will help define the relationship between pipe wall vibration and the physical characteristics of turbulent flow. Commercial FSI software packages are based on Reynolds Averaged Navier-Stokes (RANS) fluid models, which do not compute the instantaneous fluctuations in turbulent flow. This thesis presents an FSI approach based on Large Eddy Simulation (LES) flow models, which do compute the instantaneous fluctuations in turbulent flow. The results based on the LES models indicate that these fluctuations contribute to the pipe vibration. It is shown that there is a near quadratic relationship between the standard deviation of the pressure field on the pipe wall and the flow rate. It is also shown that a strong relationship between pipe vibration and flow rate exists. This research has a direct impact on the geothermal, nuclear, and other fluid transport industries.

## ACKNOWLEDGEMENTS

Two years ago, I was informed by a friend and colleague that a new professor was joining the faculty here at BYU. He also informed me that this professor was looking for someone to take on a challenging project for a Masters degree. At the time, I was finishing my undergraduate degree and contemplating continuing my education and the project description seemed to be just my “cup of tea.” It must have taken great faith for Dr. Jonathan Blotter to hand over this research on the limited information he had about my abilities and me. Yet how thankful I am for the chance I have had to work with him on this exciting and interesting project. I thank him for his guidance, direction, enthusiasm and tutoring in the theory behind this project and most of all for the countless hours of editing he has done for me. I know that he has spent many long hours away from his family in the office reading and revising papers and chapters of this thesis.

In addition to Dr. Blotter, there has also been time and energy sacrificed by other faculty members here at BYU. To the rest of my committee, Dr. Dan Maynes and Dr. Scott Sommerfeldt, thank you for your input and suggestions. Appreciation is also Dr. Russell Daines, a former teacher who has since moved on to work for NASA, who gave me the first guidance on this journey in CFD modeling of turbulent flow. Also, the statistical analysis used in this thesis was greatly aided with the help of Dr. Wayne Larsen.

Last year, funding was granted for me to attend a FLUENT Users Group Meeting in Manchester, NH. It came at a good time because I was very frustrated with my progress in being able to get the software packages to work as they were intended. At the conference, I made the acquaintance of a Dr. Graham Goldin, a FLUENT engineer who graciously showed me some valuable tips. Without these priceless gems, I don't believe I would have been able to accomplish this work.

Finally, I do not know how I would have finished this thesis without the love and support of my dear sweet wife, Keri. It isn't very often that you see a man and a woman who are both fascinated with the sciences of the world fall in love. Yet to my delight, I have found a girl that I can talk with about anything from Reynolds numbers to romance. I have truly been blessed to find a companion that will go to the ends of the earth for me—even so far as to spend hours trying to understand a subject well enough to edit a technical journal paper! I can't thank her enough for the innumerable hours she spent with me writing and revising my thesis. In fact, during the time towards the end of my revisions, I would introduce my wife as “editor and chief.”

Finally yet importantly, I need to say a special thank you to the Department of Energy and the Idaho National Engineering and Environmental Laboratory, which funded this research.

## TABLE OF CONTENTS

<b>TITLE PAGE</b> .....	<b>i</b>
<b>SIGNATURE PAGES</b> .....	<b>ii</b>
<b>ABSTRACT</b> .....	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>v</b>
<b>TABLE OF CONTENTS</b> .....	<b>vii</b>
<b>LIST OF TABLES</b> .....	<b>ix</b>
<b>LIST OF FIGURES</b> .....	<b>x</b>
<b>LIST OF SYMBOLS</b> .....	<b>xiv</b>
<b>1 INTRODUCTION</b> .....	<b>1</b>
1.1 THE PROBLEM .....	1
1.2 OBJECTIVE AND GOALS .....	4
1.3 HYPOTHESIS .....	4
1.4 WORK SCOPE AND THESIS OUTLINE.....	6
<b>2 BACKGROUND</b> .....	<b>9</b>
2.1 FSI PHYSICS .....	9
2.2 FLOW-INDUCED PIPE VIBRATION .....	10
2.2.1 Analytical.....	10
2.2.2 Experimental.....	11
2.3 FSI MODELING.....	13
2.4 LES PIPE FLOW .....	14
2.5 LES AND STRUCTURAL COUPLING.....	15
<b>3 FLOW MODELING</b> .....	<b>17</b>
3.1 TURBULENCE THEORY .....	17
3.2 DIRECT NUMERICAL SIMULATION (DNS) .....	20
3.3 REYNOLDS AVERAGED NAVIER-STOKES (RANS).....	22
3.4 LARGE EDDY SIMULATION (LES) .....	26
<b>4 LES MODELING IN FLUENT</b> .....	<b>33</b>
4.1 MODEL CONSTRUCTION (GAMBIT) .....	33
4.1.1 Geometric Domain and Characteristic Length .....	34
4.1.2 Procedure .....	36
4.2 SOLVING THE LES MODEL (FLUENT) .....	40
4.2.1 Getting Started .....	41
4.2.2 Setting up the Model.....	43

4.2.3	Initializing the Flow .....	46
4.2.4	Iterating .....	47
4.3	MODEL VERIFICATION .....	49
4.3.1	Establishing Grid-Independence .....	49
4.3.2	Examining the General Characteristics of the Flow .....	55
4.3.3	Examining Pressure Gradients .....	56
4.4	LIMITATIONS OF LES MODELING IN FLUENT .....	58
<b>5</b>	<b>FEA MODELING IN ANSYS .....</b>	<b>61</b>
5.1	BASICS .....	61
5.2	MODEL CONSTRUCTION .....	63
5.2.1	Shell Elements .....	63
5.2.2	Boundary Conditions .....	65
5.2.3	Pressure Mapping.....	69
5.2.4	Running the Transient Analysis.....	70
5.3	MODEL VERIFICATION .....	71
5.4	ACCELERATION EXTRACTION.....	74
5.5	LIMITATIONS OF FEA IN ANSYS.....	76
<b>6</b>	<b>RESULTS .....</b>	<b>77</b>
6.1	STANDARD DEVIATIONS VS. FLOW RATE .....	77
6.1.1	Pressure .....	77
6.1.2	Acceleration .....	79
6.2	SURFACE ROUGHNESS CORRECTION .....	83
6.3	SIGNIFICANT FACTORS .....	85
6.3.1	Material and Diameter .....	85
6.3.2	Statistical Analysis.....	89
6.4	NORMALITY OF PRESSURE DISTRIBUTIONS .....	97
<b>7</b>	<b>CONCLUSION .....</b>	<b>99</b>
7.1	SUMMARY .....	99
7.1.1	Fluid Model.....	99
7.1.2	Structure.....	99
7.1.3	Results.....	100
7.2	RECOMMENDATIONS .....	101
7.3	PUBLICATIONS.....	102
<b>8</b>	<b>REFERENCES.....</b>	<b>103</b>
<b>9</b>	<b>APPENDIX.....</b>	<b>107</b>
9.1	ISOTROPIC TURBULENCE FORTRAN CODE .....	107
9.2	INITIALIZATION OF FLOW USING AN INTERPOLATION FILE.....	133
9.3	PRESSURE EXTRACTION .....	137
9.4	PRESSURE MAPPING MACROS .....	140
9.5	IMECE 2003 CONFERENCE PROCEEDING .....	143

## LIST OF TABLES

Table 1-1	Outline of thesis goals.....	4
Table 3-1	Equations of motion for a Newtonian fluid with constant properties.....	21
Table 3-2	Strengths and weaknesses of popular closure models .....	25
Table 4-1	Pressure gradient verification of theory vs. FLUENT .....	58
Table 5-1	Theory vs. ANSYS for static internal pressure.....	72
Table 6-1	Justification for scaling deflections for PVC calculations.....	86
Table 6-2	List of results for various pipe diameters and flow rates .....	87
Table 6-3	Multivariate regression: $\ln(P')$ versus $D, Q$ .....	91
Table 6-4	Response surface regression: $\ln(A')$ versus $D, Q$ .....	95
Table 9-1	Macro to export the pressures in FLUENT.....	138
Table 9-2	UDF for pressure table exporting for parallel processing.....	138
Table 9-3	Macro for mapping CFD pressures to a transient structural model .....	140

## LIST OF FIGURES

Figure 1-1	Contribution to the study of turbulent flow induced vibration in fully developed pipe flow .....	3
Figure 1-2	Viscous sublayer thickness versus flow rate.....	5
Figure 1-3	Solution procedure if deflections are below viscous sublayer.....	5
Figure 1-4	Proposed solution procedure.....	6
Figure 2-1	Acceleration standard deviation vs. flow rate results from Evans fully developed pipe experiment for 3-inch diameter pipe (used with permission).....	12
Figure 3-1	Large vs. small scale eddies of smoke flow.....	19
Figure 3-2	Velocity field models of turbulent flow in a pipe, a) RANS based model, b) LES based model.....	30
Figure 3-3	Velocity profile comparison of RANS and LES based models.....	30
Figure 3-4	Pressure fluctuations along the length of the pipe as computed by RANS and LES based models .....	31
Figure 4-1	Characteristic length of mesh given in microns for a 3-inch pipe .....	35
Figure 4-2	Geometry Creation.....	37
Figure 4-3	Virtual line creation .....	37
Figure 4-4	Face linking.....	37
Figure 4-5	Mesh Check .....	39
Figure 4-6	Boundary Types.....	39
Figure 4-7	Front and side views of discretized flow domain .....	40
Figure 4-8	Reading in a case file .....	42
Figure 4-9	Checking the mesh.....	42

Figure 4-10	Turning on the LES solver .....	43
Figure 4-11	Appropriate selection of fluid properties .....	44
Figure 4-12	Boundary condition definition .....	44
Figure 4-13	Fluid selection .....	45
Figure 4-14	Periodic settings .....	45
Figure 4-15	Solution control settings .....	46
Figure 4-16	Grid independence study .....	50
Figure 4-17	Velocity profiles with boundary layer included .....	51
Figure 4-18	Comparison of the Reichardt equation and the LES model for the lower Reynolds number ( $Re_D \approx 83,000$ ) .....	53
Figure 4-19	Velocity profile comparison for lower Reynolds number ( $Re_D \approx 83,000$ ) using common, non-dimensional parameters .....	53
Figure 4-20	Comparison of the Reichardt equation and the LES model for the upper Reynolds number ( $Re_D \approx 415,000$ ) .....	54
Figure 4-21	Velocity profile comparison for upper Reynolds number ( $Re_D \approx 415,000$ ) using common, non-dimensional parameters .....	54
Figure 4-22	LES velocity contour .....	55
Figure 4-23	Fluctuating pressure contour .....	56
Figure 4-24	Fluctuating pressure field for channel flow .....	56
Figure 5-1	Setting the element type to Shell93 .....	64
Figure 5-2	Geometric representation of a Shell93 element .....	64
Figure 5-3	Setting node thicknesses .....	65
Figure 5-4	Nodal deployment with applied boundary conditions .....	67
Figure 5-5	Creating areas by skinning .....	68
Figure 5-6	Structural model with mapped pressure .....	69
Figure 5-7	VonMises stress on 3-inch pipe to a 70-kPa internal load .....	73
Figure 5-8	Recreation of continuous deflections vs. time .....	75

Figure 5-9	Central difference vs. spline-Ridder technique for estimating acceleration .....	75
Figure 6-1	Standard deviation of the pressure fluctuations on the pipe surface versus flow rate for a 3-inch diameter pipe .....	78
Figure 6-2	Coefficient of pressure vs. Reynolds number for 3-inch diameter data ...	79
Figure 6-3	Standard deviation of the acceleration on the pipe surface at a point versus flow rate for a 3-inch diameter pipe .....	80
Figure 6-4	Pressure vs. acceleration standard deviations .....	81
Figure 6-5	Comparison of experimental data and numerically simulated data for 3-inch pipe .....	82
Figure 6-6	Ratio of numerical to experimental curve fits—as the flow rate increases, the contribution of the pressure fluctuations to the overall vibrational response also increases .....	83
Figure 6-7	Pressure fluctuation standard deviation vs. flow rate .....	88
Figure 6-8	Acceleration standard deviation vs. flow rate.....	88
Figure 6-9	All pressure vs. acceleration standard deviations .....	89
Figure 6-10	Normal probability plot of residuals for $\ln(P')$ .....	92
Figure 6-11	Contour of $\ln(P')$ model—as the diameter decreases and the flow rate increases, the pressure fluctuations also increase .....	93
Figure 6-12	Coefficient of pressure vs. Reynolds number .....	94
Figure 6-13	Normal Probability plot of residuals for $(A')$ .....	96
Figure 6-14	Contour of $\ln(A')$ model—as the diameter decreases and the flow rate increases, the accelerations also increase.....	96
Figure 6-15	Pressure field on the pipe surface, a) positive pressure field, b) negative pressure field .....	98
Figure 6-16	Distribution of the turbulent pressure fluctuations at each interior point of a pipe wall.....	98
Figure 9-1	Interpolation beginnings .....	134
Figure 9-2	Select the values to write .....	135
Figure 9-3	Name the file.....	135

Figure 9-4	Read in the file .....	136
Figure 9-5	Comparison of streamlines .....	136
Figure 9-6	Convergence comparison.....	137
Figure 9-7	Execute command window .....	137

## LIST OF SYMBOLS

### *Acronyms*

<b>Acronym</b>	<b>Definition</b>
2-D	Two-Dimensional
3-D	Three-Dimensional
AISI	American Iron and Steel Institute
ANOVA	Analysis of Variance
ASME	American Society of Mechanical Engineers
BYU	Brigham Young University
CFD	Computational Fluid Dynamics
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
DOF	Degree(s) of Freedom
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Method
FSI	Fluid-Structure Interaction
GB	Gigabytes
GS	Grid Scale
GUI	Graphical User Interface
ISU	Idaho State University
LES	Large-Eddy Simulation
N-S	Navier-Stokes
PDE	Partial Differential Equation
PISO	Pressure-Implicit with Splitting of Operators
PVC	Polyvinyl Chloride
RANS	Reynolds Averaged Navier-Stokes
RNG	Renormalized Group
RSM	Reynolds Stress Model
SGS	Subgrid Scale
SIMPLE	Semi-Implicit Pressure Linked Equations
TUI	Text User Interface

### *Roman alphabet*

<b>Symbol</b>	<b>Unit(s)</b>	<b>Definition</b>
$A'$	g	Standard deviation of the acceleration
$A_{sum}$	m/s <sup>2</sup>	Magnitude of the AX, AY, and AZ direction vectors

$AX$	$m/s^2$	Acceleration in the x-direction
$AY$	$m/s^2$	Acceleration in the y-direction
$AZ$	$m/s^2$	Acceleration in the z-direction
$B$	$kg/s$	Structural damping coefficient
$C_f$	—	Skin friction coefficient
$C_p$	—	Coefficient of pressure
$C_s$	—	Smagorinsky-Lilly turbulent SGS viscosity model constant
$C_\mu$	—	k- $\epsilon$ turbulent viscosity model constant
$df$	—	Degrees of freedom
$D$	m or in	Pipe diameter
$E$	Pa	Young's Modulus
$f$	—	Moody friction factor
$F$	multiple	Force vector
$g$	—	Acceleration due to gravity ( $1g = 9.81m/s^2$ )
$G$	—	Spatial filter function
$k$	$kg \cdot m^2/s^2$	Kinetic energy
$k_s$	m	Surface roughness constant
$K$	multiple	Stiffness matrix
$l$	m	The characteristic length of a turbulent eddy
$L_{pipe}$	m	Length of the pipe model (either fluid or structure)
$L_s$	m	Mixing length of the SGS
$M$	kg	Mass matrix
$N$	—	Number of computational elements
$Nu$	—	Nussult number – non-dimensional heat transfer coefficient
$P$	Pa	Pressure
$P'$	Pa	Standard deviation of pressure
$Pr$	—	Prandtl number
$Q$	$m^3/s$ or liters/min	Pipe flow rate
$r$	m	Variable pipe radius
$R$	m	Pipe radius
$R_{ij}$	Pa	RANS stress tensor
$R^2$	—	Correlation coefficient
$R_i$	m	Inside diameter
$R_o$	m	Outside diameter
$Re_D$	—	Reynolds number based on diameter
$Re_l$	—	Turbulent Reynolds number
$S_{ij}$	1/s	Rate of strain tensor
$t$	m	Thickness
$u$	m/s	Characteristic turbulent velocity/Fluid velocity vector
$u'$	m/s	The fluctuating component about the mean of the velocity
$\bar{u}$	m/s	Mean velocity
$u_\tau$	m/s	Friction velocity
$U$	multiple	Displacement vector

$U_0$	m/s	Mean velocity
$U^+$	—	Non-dimensional velocity normalized by the friction velocity
$U_{sum}$	m	Magnitude of the UX, UY, and UZ direction vectors
$UX$	m	Displacement in the x-direction
$UY$	m	Displacement in the y-direction
$UZ$	m	Displacement in the z-direction
$V$	$m^3$	Volume of the computational cell
$\mathbf{x}$	m	Spatial location
$\mathbf{x}'$	m	Spatial vector
$y$	m	Distance from the wall
$y^+$	m	Dimensional parameter commonly used in turbulence analysis

*Greek Alphabet*

<b>Symbol</b>	<b>Unit(s)</b>	<b>Definition</b>
$\delta_{ij}$	—	Kronecker Delta
$\Delta P$	Pa	Pressure gradient
$\Delta t$	s	Time-step
$\Delta x$	m	Characteristic computational cell size
$\varepsilon$	$m^2/s^3$	Viscous dissipation rate
$\varepsilon$	m/m	Strain
$\phi$	—	Any flow variable (velocity, pressure, density, etc.)
$\phi'$	—	The SGS scale flow variable (velocity, pressure, density, etc.)
$\tilde{\phi}$	—	The GS variable (velocity, pressure, density, etc.)
$\eta$	m	Kolmogorov length scale
$\kappa$	—	von Karmen constant
$\mu$	Pa·s	Dynamic viscosity
$\mu_t$	Pa·s	SGS eddy viscosity
$\mu_{turb}$	Pa·s	Turbulent viscosity
$\nu$	—	Poisson's ratio
$\nu$	$m^2/s$	Kinematic viscosity
$\rho$	$kg/m^3$	Density
$\sigma_{(1,2,3)}$	Pa	(Principle) Stress
$\sigma_{vm}$	Pa	VonMises Stress
$\tau$	s	Kolmogorov time scale
$\tau_{ij}$	Pa	LES stress tensor
$\upsilon$	m/s	Kolmogorov velocity scale

*Note on Units:* All models were created using SI units; however, experimental data used English units. It is easier to refer to the diameters in terms of their English value instead of their SI equivalent since they are round numbers. Therefore, all diameters will be given in terms of inches and flow rates in terms of liters/minute. Everything else is given in SI units.

# 1 INTRODUCTION

## 1.1 THE PROBLEM

In a world of cause and effect, it becomes natural to study the way different mediums interact. Since the tragic failure of the Tacoma Narrows Bridge on November 7, 1940<sup>1</sup>, the study of the interaction between fluid and structure has become an important area of scientific research. Fluid-Structure Interaction (FSI) is the field of study that investigates this physical phenomenon. Within the realm of FSI there exists a subset called flow-induced vibration. Research in this field attempts to quantify the vibration of a structure caused by a fluid flowing past or through it.

In general, flow-induced vibration is divided into three main mechanisms: *turbulence-induced vibration*—as seen in fluttering pipes, *vorticity shedding-induced vibration*—the phenomenon that destroyed the Tacoma Bridge, and *fluidelastic instability*—a unique form of flow-induced vibration that is most commonly seen in nuclear heat exchangers after the tube velocity reaches a critical value.<sup>2</sup> Of these, the turbulence-induced phenomenon will be the focus of this research—specifically the vibrations of a pipe containing fully developed turbulent fluid flow.

The vibration of a pipe transporting fluid has been recognized by researchers and quantified using analytical, numerical or experimental techniques. In the past, researchers such as Saito<sup>3</sup>, Evans<sup>4</sup>, Durant<sup>5,6</sup>, Brevart<sup>7</sup> and Kim<sup>8</sup> investigated and attempted to quantify the relationship between fluid flow rate and pipe vibration. Although results vary, each researcher proposed that pipe vibration was a direct result of the pressure fluctuations at the pipe wall inherent in turbulent flow.

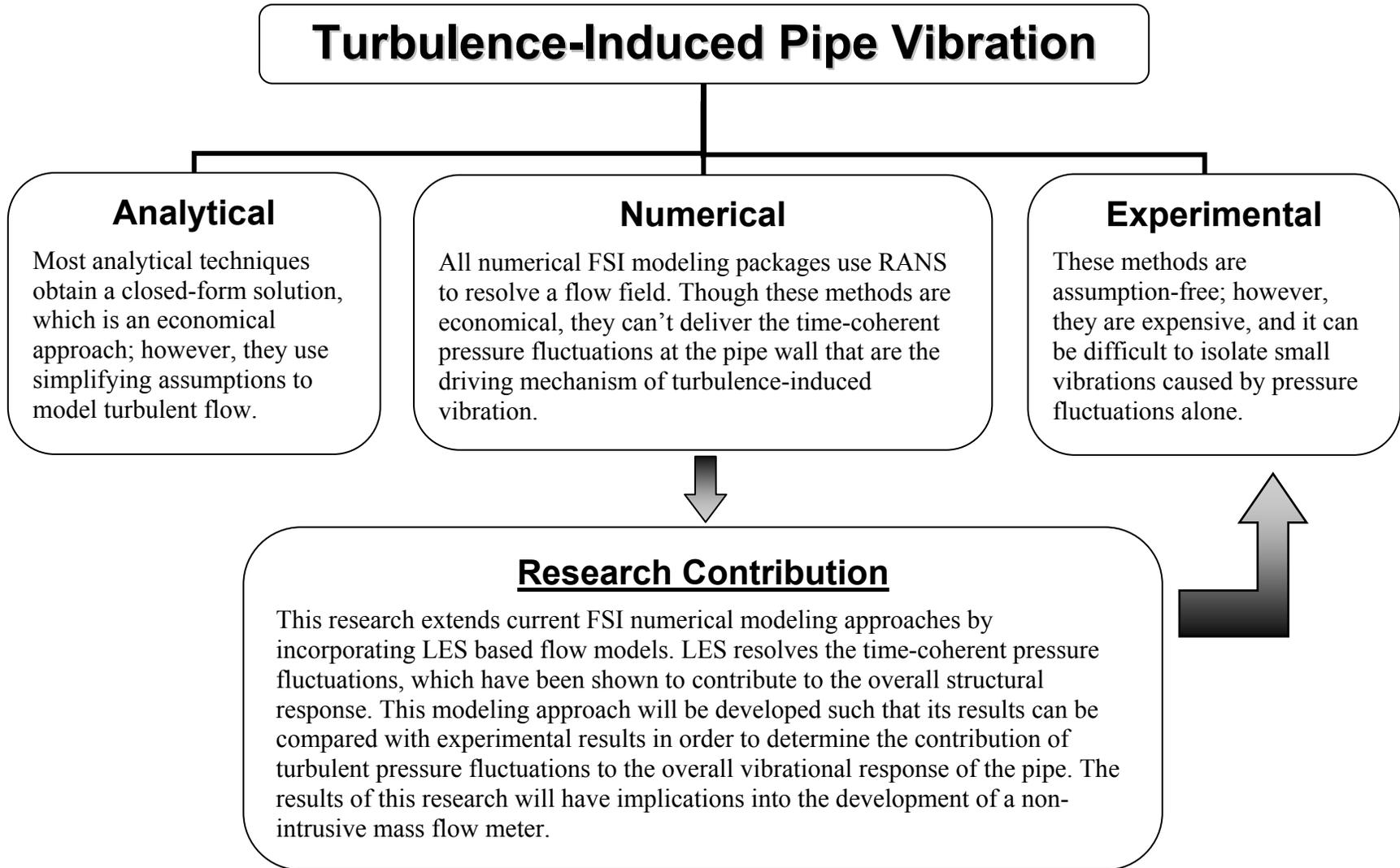
Researchers face challenges unique to their method of solving this FSI problem through analytical, numerical or experimental means. Current analytical and numerical techniques model fluid flow using simplifying assumptions, typically based on time-

averaged equations, which do not provide instantaneous values. Even commercially available numerical codes for this type of analysis are insufficient. Commercial FSI codes use Reynolds Averaged Navier-Stokes (RANS) based turbulent models. However, these codes will not produce pressure variations at the fluid-structure interface; hence, they will not accomplish the purpose of this study. Experimental solutions can be time-consuming and expensive. It can also be difficult to isolate the vibrations induced by pressure fluctuations alone. Because of these challenges, accurately quantifying the vibrations induced by pressure fluctuations alone has not yet been accomplished.

Although the methods described above are insufficient to solve the FSI problem of fully developed turbulent pipe flow, there are numerical techniques that are sufficient to model the fluid alone. These techniques are based on what is known as Large Eddy Simulation (LES). LES models the fluid flow by spatially filtering the governing flow equations and solving for a local-averaged velocity rather than a time-averaged velocity, which will produce the pressure variations desired. There are also commercial analysis packages with the ability to accurately model the structural response alone. Therefore, coupling an LES-based fluid model with a structural model will provide the capability necessary to analyze the turbulent-induced phenomenon.

Numerically analyzing fully developed turbulent pipe flow by coupling a fluid model with a structural model will help determine the contribution of pressure variations to the overall pipe vibration, as determined from experiment. Such an analysis has application to the eventual development of a vibration sensor as a non-intrusive mass flow meter. A model development of this kind will also provide a benchmark and method for investigating similar FSI problems where experimental data would be difficult to achieve.

Figure 1-1 illustrates current developments and challenges in the area of turbulence-induced FSI. As shown, the goal of this thesis is to build on current numerical modeling capabilities and to develop a FSI model based on LES, which accounts for the instantaneous pressure fluctuations in the flow. This model will then be used to determine the relationship between flow noise as measured by the pipe acceleration and the flow rate in the pipe.



**Figure 1-1 Contribution to the study of turbulent flow induced vibration in fully developed pipe flow**

## 1.2 OBJECTIVE AND GOALS

The objective of this research is to numerically model the structural vibrations caused by fully developed, internal turbulent pipe flow. The flow model will be based on an LES formulation and will be coupled with a structural model to obtain the FSI solution. This research will be accomplished by achieving the goals listed in Table 1-1.

**Table 1-1 Outline of thesis goals**

1	Develop an LES based, numerical model of turbulent pipe flow, which accounts for the instantaneous pressure fluctuations at the wall
2	Couple the fluid and structural models and determine the contribution pressure fluctuations have on the dynamic response of a pipe

## 1.3 HYPOTHESIS

The main objective of this thesis is to numerically model the structural vibrations induced by fully developed pipe flow. This objective will be obtained by breaking the problem into two sections: fluid and structure. It is first necessary to develop an LES fluid model. The pressure fluctuations at the pipe wall obtained from the fluid model will be imported as a distributed load on a structural model. From here, three scenarios of investigation are possible:

*Case I: Pressure variations do not deform the pipe.*

It will be determined that the pipe vibration was not induced by turbulent flow.

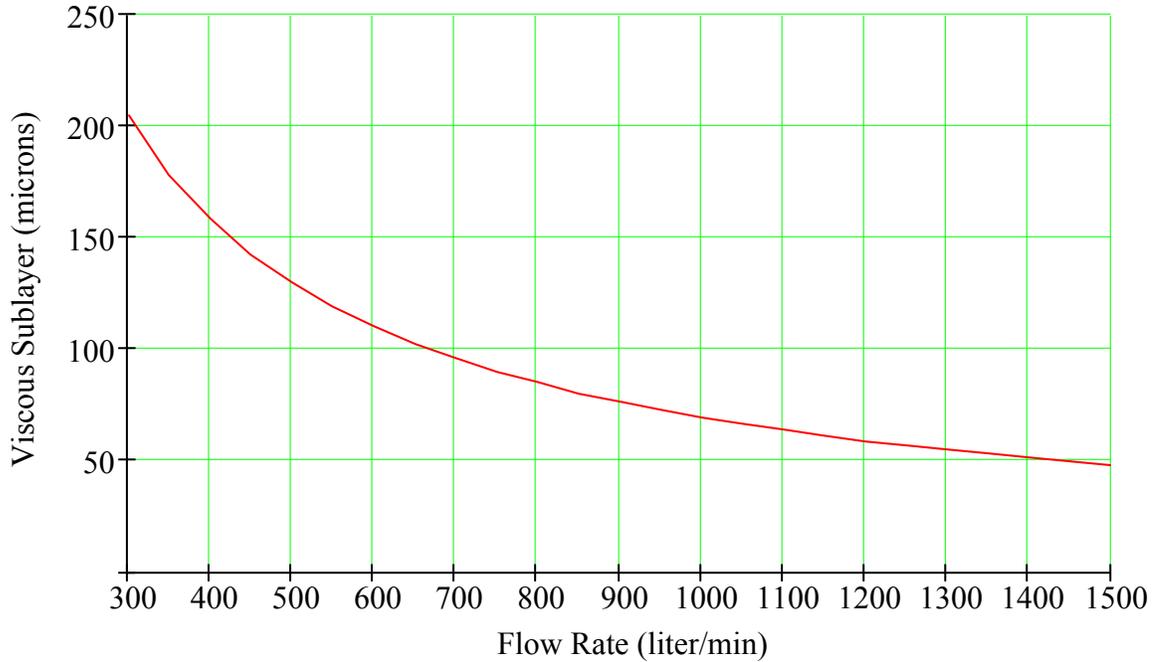
*Case II: Pressure variations deform the pipe, but do not alter the structure of the flow.*

If the maximum deflection obtained from a static loading of the pressure fluctuations is smaller than the size of the viscous sublayer, it can be assumed that the changes in the pipe geometry do not significantly influence the flow solution.<sup>a</sup> For this

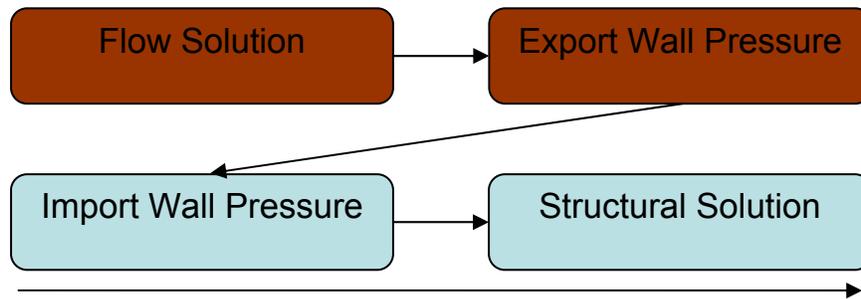
---

<sup>a</sup> The viscous sub-layer is a very small laminar region adjacent to the wall; since flow is laminar in this region, deformation of the pipe wall should not induce changes in the overall structure of the flow.

comparison, a plot of the viscous sublayer as a function of flow rate is shown in Figure 1-2. At the highest flow rate, a maximum displacement of 50 microns is required before a fully coupled solution procedure is necessary. A one-way coupling procedure, as shown in Figure 1-3, will be employed if deflections remain within the viscous sublayer. In other words, a structural update of the deformed pipe geometry will not be needed between time-steps.



**Figure 1-2 Viscous sublayer thickness versus flow rate<sup>b</sup>**

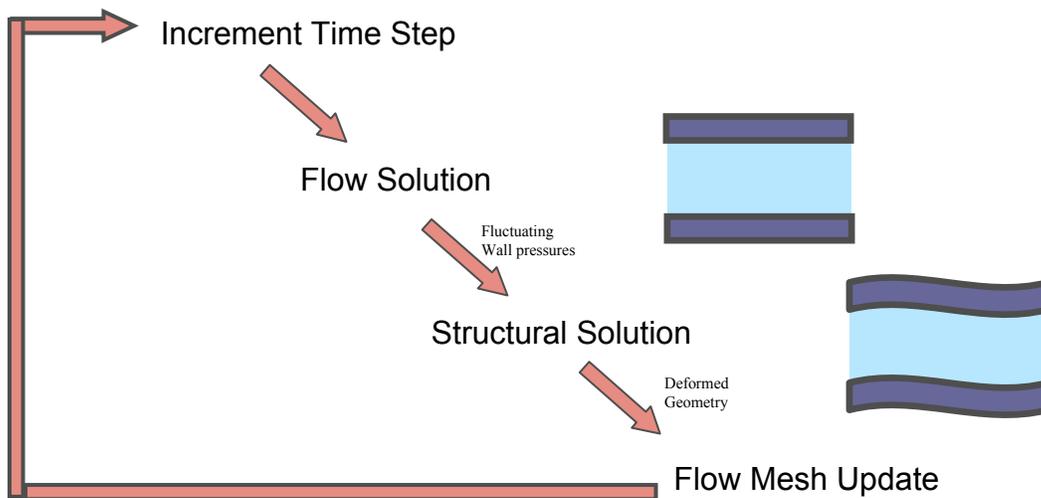


**Figure 1-3 Solution procedure if deflections are below viscous sublayer**

<sup>b</sup> This figure is from the results of a 3-inch pipe only. Similar plots could be given for other diameters.

*Case III: Pressure fluctuations deform the pipe and alter the structure of the flow.*

If a maximum deflection of the pipe wall induced by the turbulent pressure variations is larger than the viscous sublayer, it will be assumed that the changes in the pipe geometry alter the structure of the flow. If this is the case, the iterative coupling procedure outlined in Figure 1-4 will be employed. For each time step, pressure variations from the fluid solver will be imported into the structural solver. The geometry and mesh will be updated to reflect the change in pipe shape, and be imported back into the fluid solver, where another time-step can be generated. This will be repeated until the desired number of time-step solutions is obtained. Collected data will be analyzed to determine the contribution of the pressure fluctuations to the overall vibratory response.



**Figure 1-4 Proposed solution procedure**

## **1.4 WORK SCOPE AND THESIS OUTLINE**

The goals stated in Table 1-1 will be met by investigating six different flow rates between 300 and 1500 liters/min for a 3-inch pipe, as well as three flow rates for each of 1.5 and 4-inch pipes. Primary focus will be on schedule 40 steel pipes with a secondary investigation on the effect of changing the material to PVC.

This thesis will follow the recommended outline as prescribed by the BYU Mechanical Engineering Department guidelines. Chapter 2 contains a literature review and outlines the work that has been done in LES pipe flow, FSI physics, flow-induced

pipe vibrations, FSI modeling, and LES and structural coupling. Since the LES-based fluid model represents a unique aspect of this research, Chapter 3 is completely devoted to a discussion on this modeling approach. Chapter 4 presents the details of developing a suitable fluid model using LES theory. In Chapter 5, a basic overview of structural modeling is given, along with a description of the structural model used for this research. The results of the combined FSI model are presented and discussed in Chapter 6. Finally, Chapter 7 summarizes the conclusions of this research and discusses possible areas of future research. References are listed in Chapter 8, and the appendix in Chapter 1 contains important details to the re-creation of the presented work.



## 2 BACKGROUND

It was once said, “The interactive phenomena between fluid and body motion represent one of the most difficult problems in the field of fluid dynamics.”<sup>9</sup> Despite its difficulty, the study of the interaction between fluid and structure entertains a growing audience, due largely to the increased speed and efficiency of today’s numerical techniques. This chapter provides an overview of the research previously conducted to solve these complicated FSI problems, including analytical, experimental and numerical procedures. This literature review is summarized in the following order: FSI physics, flow-induced pipe vibrations, FSI modeling, LES pipe flow, and LES and structural coupling.

### 2.1 FSI PHYSICS

This section presents a physical explanation of how energy is transferred at the fluid-structure interface in fully developed turbulent pipe flow. Intuition may theorize that energy is transferred because fluid particles “hit” the pipe wall, like marbles dropping on a tin roof. However, internally flowing fluids do not behave this way. The water molecules adjoining the pipe wall do not move (no-slip condition); in other words, they have no velocity, and consequently no kinetic energy. However, molecules that are approaching the wall, such as in turbulent eddies, do have velocity and kinetic energy. This kinetic energy must be converted to another form of energy as the molecule reaches the pipe wall, according to the first law of thermodynamics. Some of the kinetic energy is converted to heat as turbulent eddies dissipate, but most is converted into potential energy in the form of pressure. This can be verified by integrating the r-momentum equation from the cylindrical form of the Navier-Stokes (N-S) equations at the wall.<sup>10,11</sup> Turbulent flow is characterized by the chaotic formation and dissipation of eddies, which cause pressure fluctuations at the pipe wall.<sup>12</sup>

It has been shown that the pipe will vibrate in response to the turbulent pressure fluctuations.<sup>3</sup> This phenomenon can be experienced by placing your hand on a water faucet or hose and feeling the pulsations as water flows through. The response of piping structure from pressure fluctuations is affected by several factors, including the elastic modulus of the material, structural damping, structural mass/density, and boundary conditions.

## **2.2 FLOW-INDUCED PIPE VIBRATION**

As mentioned in the introduction, a number of studies have been conducted to assess the relationship between the flow rate through a pipe and its accompanying dynamic response. These researchers approached this problem using analytical and/or experimental techniques. This next section discusses what has been done in each of these areas and the limitations to each approach.

### **2.2.1 Analytical**

Most analytical studies use a theoretical wave perspective to analyze the FSI phenomenon by studying the way waves propagate through a pipe when excited by an outside force. However, these studies employ potential flow theory, which does not accurately describe turbulent flow. Three of these studies include those conducted by Cuschieri and Leyrat,<sup>13</sup> Brevart and Fuller,<sup>7</sup> and Gorman et al.<sup>14</sup> Cuschieri and Leyrat conducted theoretical studies on the vibrational influence of a fluid-loaded pipe using potential flow theory and the wave equation. The study developed an equation of motion of an infinitely long pipe shell influenced by a moving internal fluid. In 1993, Brevart and Fuller analyzed the effect of internal flow on the wave propagation along an infinite cylinder using potential flow theory and the Flügge model. They showed that the internal flow of a pipe would in fact cause the wave number to change in the axial direction. Gorman et al. investigated the effect of annular two-phase flow on the natural frequencies of a pipe and concluded that the phase in contact with the pipe has the greatest effect, also using potential flow theory and the Flügge model. As impressive as these studies may be, they are not completely accurate because turbulent flow does not exactly behave according to potential flow theory.

Durant et al.<sup>5</sup> also used an analytical perspective, but they refined the previous methods when they characterized the vibro-acoustic response of the pipe to random excitation by a cross-power spectral density, given as a 2-D integral over the domain occupied by the structure. These fluctuations were analyzed after cancellation of contaminating background noise. The high velocity study concluded only a few decibels difference between a numerical prediction based on a Corcos-like model of wall pressure and experimental data. Prior to this study, Durant performed similar experiments on the mass flow rate of a single-component, turbulent gas using pipe vibrations.<sup>6</sup>

Kim and Kim<sup>8</sup> took another approach by using wave decomposition theory to analyze the pipe vibrations. Here, they reported on estimated flow rates using three accelerometers and an excitation signal on the outside of the pipe wall. Hibiki<sup>15</sup> noted that the flow-induced vibrations due to a two-phase mixture flowing in a loop were proportional to the gas and liquid flow rate. Again, these researchers used analytical techniques, which still fall subject to their limitations of simplifying assumptions.

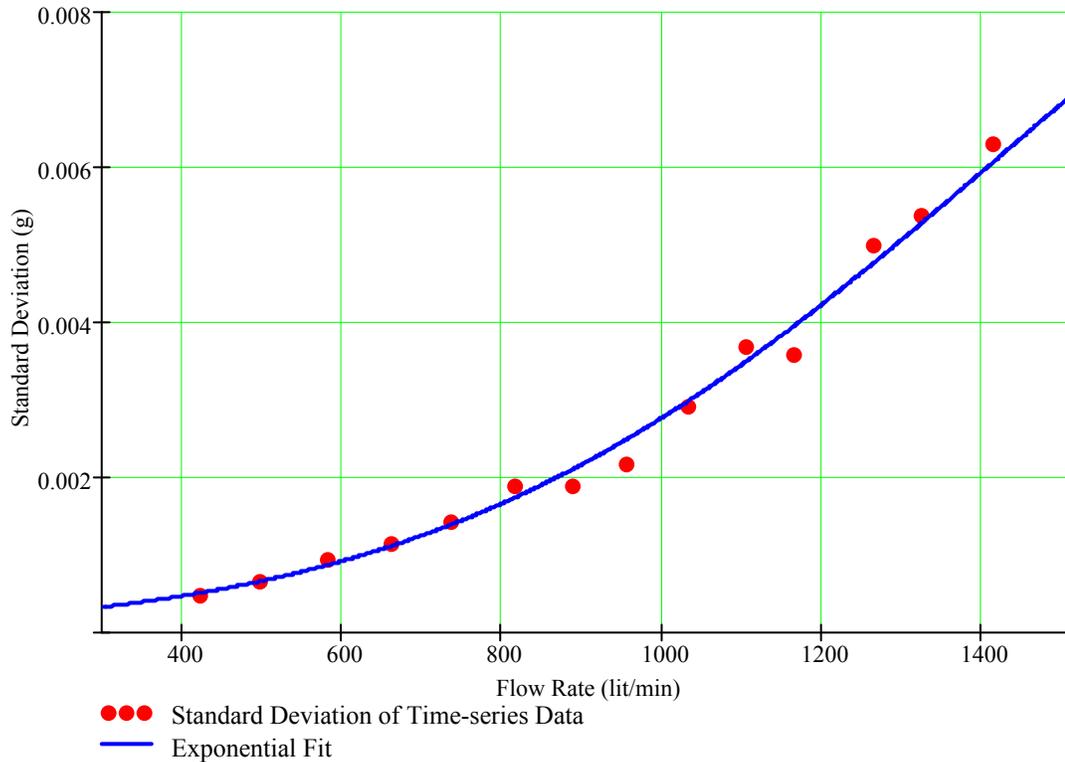
In all aforementioned cases, simplifying assumptions had to be made about the way the turbulent flow behaves, so they cannot suitably model the way the fluid pressure forces excite the pipe.

## **2.2.2 Experimental**

One of the first experimental studies for flow-induced vibration of a pipe due to internal flow was done by Saito, et al. in 1990.<sup>3</sup> They quantified their findings by plotting the root mean square pressure and acceleration values against flow velocity. However, measurements were taken immediately after the fluid passed through an orifice, which altered the pipe diameter; hence, the flow was not fully developed. In addition, no distinction was made between the vibration caused by the fluid “hitting” the orifice and the vibration caused by the turbulence.

In 1999, Evans noted a similar relationship between flow velocity and vibration, which eventually led to a patent.<sup>4,16</sup> In his study, he recorded accelerometer data on the outside of a pipe carrying fully developed flow. He quantified this relationship plotting standard deviation of the time series accelerometer data against the flow rate, as shown in Figure 2-1. His studies concluded that there is a strong relationship between the

amplitude of the vibrations and the mass flow through the pipe. This phenomenon may be experienced by placing your hand on a hose or faucet and feeling the motion of vibration increase as flow increases. Thus, it is no surprise that Evans also theorized that the vibrations were a direct result of the amplitude of the pressure fluctuations at the pipe wall. Even though Evans made efforts to eliminate all other causes of vibration, his studies are still unclear whether turbulent pressure variations alone caused the vibrations.



**Figure 2-1 Acceleration standard deviation vs. flow rate results from Evans fully developed pipe experiment for 3-inch diameter pipe (used with permission)**

Every experimental case studied inherently has trouble isolating the vibrations due to pressure fluctuations alone. Uncontrollable factors such as pump noise, clamps, bends in the pipe and irregularities in the cylindrical geometry all contribute to the overall vibration sensed by the accelerometer. It is also difficult to eliminate variation in the flow rate at high velocities—which affects the fully developed nature of the flow. Consequently, it is difficult to determine the effect fully developed flow has on the vibrations of a pipe.

## 2.3 FSI MODELING

Since analytical solutions to the FSI problem involve many assumptions, a better understanding of this phenomenon is most likely to be achieved using numerical models. As far as the author can tell, the first modeling of the FSI phenomena for water pipe flow occurred during the second phase of the work performed by Saito et al. To confirm their experimental data, Saito imported the experimental measured pressures from the pipe wall into a NASTRAN<sup>®</sup> FEA model.<sup>3</sup> This resulted in a low spatial resolution model that seemed like a reasonable approach but required experimental data. Saito's work was one of the first attempts to model the behavior of a structure from forces exerted by a fluid.

Commercial codes have been developed that model the interaction between fluid and structure, e.g. FIDAP<sup>®</sup>, ALGOR<sup>®</sup>, ADINA<sup>®</sup>, ANSYS<sup>®</sup>, STRACO<sup>®</sup>, SYSNOISE<sup>®</sup>, and IFSAS<sup>®</sup>. These codes for FSI analysis are based on the principle of energy transfer mentioned in Section 2.1 and in theory would take Saito's problem from start to finish. These packages have a large market, and it is foreseen that they will be used more heavily as their development becomes more sophisticated and user friendly. Many exciting results have been concluded from FSI studies using these packages. For example, in 1998, Ortega used FSI to model a cerebral aneurysm. Since damage to the vessel wall is believed to be caused by the shear stress from the flow, an FSI model is perfectly adapted in this situation. Ortega's research now predicts when that will happen.<sup>17</sup> Many other researchers have used FSI to study external flows around bluff bodies (like prisms and square cylinders), flows inducing vibration in bridges<sup>18</sup>, and even print head cartridges.<sup>19</sup>

From these studies, it is clear to see that FSI modeling will play an important role in engineering design in the future. However, as sophisticated as all these programs may be, they still have limitations in the resolution of their flow field. All commercial codes use RANS equations for their turbulent models; however, such models will not produce pressure variations at the fluid-structure interface needed for analysis of fully developed turbulent pipe flow. Since these pressure variations are what generate pipe vibrations, these commercial codes will not accomplish the purposes of this study.

## 2.4 LES PIPE FLOW

Accurately predicting the interaction between fluid and structure requires great care in the selection of a turbulence model. Most turbulence models use time-averaged N-S equations as their basic governing equations and consequently solve for average velocity and pressure. Even for the unsteady class of problems, this method solves for average values at a particular time-step; hence, RANS based methods are insufficient for this application. The clarification of this is presented in Section 3.3 in a detailed discussion of RANS modeling. Our unsteady problem distinguishes itself not by the average characteristics of the flow, but by the instantaneous ones. This eliminates any option of using RANS turbulence models.

For flows where unsteady RANS is unsuitable, LES is the recommended alternative. This method has been deemed by many experts in the CFD world as one of the most powerful computational tools available today for the calculation of turbulent flows. The name of this method reflects its very essence: whereas large-scale flow structures are calculated or resolved explicitly, small-scale processes—that are below the limits of numerical resolution, are parameterized using models of various degrees of complexity. In other words, large eddies are directly simulated and the smaller ones are modeled.

Of the LES research for turbulent flow in a pipe, Eggels' PhD. thesis resides as the most concise and complete work on LES modeling of cylindrical pipe geometry.<sup>20</sup> Eggels studied DNS<sup>c</sup> and LES simulations of flows in non-rotating and rotating pipes using a staggered/structured grid along with a modified Smagorinsky subgrid scale model. Due to the limitations in the computing resources in 1994, his studies were limited to lower Reynolds numbers than we wish to investigate; however, most of the methodology still holds true and will be the guiding basis for the development of an accurate turbulent pipe model. Our LES model will also be compared against empirical relationships such as the Reichardt equation.

---

<sup>c</sup> Direct Numerical Simulation—This is the most accurate of methods to resolve fluid flow and will be discussed in greater detail in Chapter 3.

## 2.5 LES AND STRUCTURAL COUPLING

Now that economical methods for modeling instantaneous turbulence have been created, the door for modeling the FSI has been opened much wider. Now problems that require instantaneous pressures at the fluid structure interface are feasible with LES. The trick now is to determine how to couple fluid models together with structural models.

Great headway in this direction started in 1994 when Davis and Hassan<sup>21</sup> developed the LES code using a finite element method (FEM). Historically, CFD codes have been written using a finite volume approach.<sup>22</sup> However, at the time, finite volume discretizations required a structured grid, which produced a very large number of elements for complex geometries. On the other hand, the FEM allows the use of an unstructured grid. Davis and Hassan capitalized on the flexibility of the FEM to develop a LES code to solve a steam-generator flow problem. Since most of the structural analysis codes are written using FEM, their research was a great step in the right direction of coupling the fluid and structural solvers. However, the author was unable to find anyone that has merged the FEM-LES and structural codes.

More recently, Murakami et al. investigated the effect on the flow field due to the oscillations of a square cylinder using an LES model.<sup>9</sup> They studied the flow field by first forcing the cylinder to oscillate and then by allowing the wind to induce free oscillation of the cylinder. The study produced interesting results—especially during the investigation of the wind-induced oscillations. The oscillations of the cylinder were driven by the net lift coefficient calculated at each time-step.

It is a continuation of studies like these that I wish to pursue, namely the influence of pressure fluctuations generated by an LES fully developed pipe flow model on a thin-shelled cylinder. Here LES will be coupled with finite element analysis (FEA) to investigate these effects.



## 3 FLOW MODELING

Chapter 3 provides an overview of the details pertinent to choosing an appropriate turbulence model. Accurate analysis of FSI requires great care in the area of flow modeling. If resources and indefinite time were available, ideally, a Direct Numerical Simulation (DNS) approach would be superlative (see Section 3.2). However, spatial and temporal refinements would need to be approximately to the Kolmogorov<sup>d</sup> scale<sup>23</sup> for a solution to be accurate. Therefore, even with today's supercomputing power, DNS is limited to simple geometries with low Reynolds numbers. On the other hand, RANS methods are suited for most engineering problems; however, they model the statistical time-average of the flow and do not produce instantaneous values of pressure—the main item of interest to this problem. LES gives us a compromise between these two extremes, allowing one to resolve the flow to the size of the mesh and locally model the effects of the smaller scales of turbulence, which provides the instantaneous pressures of interest at a reasonable computational cost. To support the claim that LES modeling is the most appropriate method for this application, a brief background on turbulence and the different approaches to its modeling is given in this chapter.

### 3.1 TURBULENCE THEORY

The purpose of this section is to provide basic background on turbulence, the definition of the turbulent Reynolds number, and its relation to the fundamental scales of turbulence. This discussion will provide a foundation and understanding as to why the LES approach is the most suitable for this situation.

---

<sup>d</sup> The Kolmogorov scale is the smallest scale of turbulence; in other words, no simplification of turbulence can be made. Andrei Nikolaevich Kolmogorov (1903-1987), a 19<sup>th</sup> century Russian mathematician, made significant contributions to many branches of science including the understanding of turbulent fluid flow.

Although researchers have studied turbulence for many years, it is not yet possible to characterize turbulence from a purely theoretical standpoint.<sup>24</sup> Though the governing equations of motion are still too complex, many important characteristics of turbulence are well known and found in most of the literature.<sup>11,12</sup> A brief discussion is made here; however, the reader is referred to the published literature referred to in this section for a more detailed understanding.

Turbulence is time-dependent, 3-D, highly non-linear, and contrary to intuition is not a random phenomenon. Detailed studies have shown that turbulent flows contain definite spatial (coherent) structures that develop in time—these are what are usually referred to as eddies. The dynamic and geometrical properties of the largest eddies are closely related to the corresponding properties of the mean flow field. On average, the length ( $l$ ) of the large eddies is defined by a fraction<sup>e</sup> of a geometrical characteristic length. For pipe flow, the characteristic length would be considered the pipe diameter ( $l \sim .1D$ ).<sup>20</sup> Along with average length, these eddies also possess a characteristic velocity taken as a fraction<sup>f</sup> of the mean velocity ( $u \sim 0.05U_0$ ).<sup>12</sup> These scales, along with the kinematic viscosity ( $\nu$ ) of the fluid, define the turbulent Reynolds number shown by Eq. (3.1).

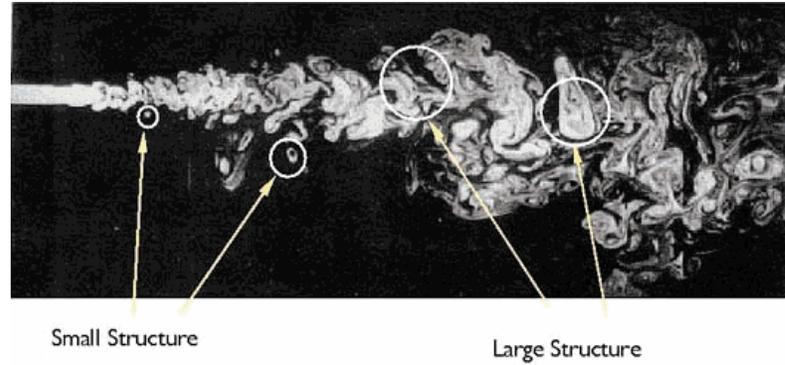
$$\text{Re}_t = \frac{ul}{\nu} \quad (3.1)$$

Turbulence theory states that the eddies also vary in size. This is illustrated by the large and small scales shown in Figure 3-1.<sup>25</sup> The largest eddies break down into smaller eddies, which break down into even smaller eddies. This process of eddy breakdown transfers kinetic energy from the mean flow to progressively smaller scales of motion. This process is known as the *energy cascade*. At the smallest scales of turbulent motion, the kinetic energy is converted to heat by means of *viscous dissipation*.

---

<sup>e</sup> 1/10 is used for most length fractions

<sup>f</sup> 1/20 is used for most velocity fractions



**Figure 3-1 Large vs. small scale eddies of smoke flow**

As a matter of necessity, the dissipation rate is fully determined by the characteristic scales defined for the large eddies. This is interesting since the dissipation rate is associated with the microstructure. The time and length scales of the smallest turbulent eddies are many orders of magnitude greater than the time scales and free paths of molecular motion. As a result, the processes of viscous dissipation are statistically independent of molecular motion. Through dimensional analysis, the relation between  $u$  and  $l$  as given by Eq. (3.2) defines an expression for the rate of dissipation ( $\varepsilon$ ).<sup>20</sup>

$$\varepsilon \sim \frac{u^3}{l} \quad (3.2)$$

From the dissipation rate and the molecular property of kinematic viscosity, dimensional analysis yields expressions for the Kolmogorov scales, which are the fundamental scales of turbulence length ( $\eta$ ), velocity ( $\nu$ ) and time ( $\tau$ ) shown by Eqs. (3.3)-(3.5) respectively.

$$\eta = \sqrt[4]{\left(\frac{\nu^3}{\varepsilon}\right)} \quad (3.3)$$

$$\nu = \sqrt[4]{\nu\varepsilon} \quad (3.4)$$

$$\tau = \sqrt{\frac{\nu}{\varepsilon}} \quad (3.5)$$

Substituting the viscous dissipation rate into Eqs. (3.3)-(3.5), the following non-dimensional turbulence scales are found to relate to the turbulent Reynolds number given by Eqs. (3.6)-(3.8).

$$\frac{\eta}{l} = \text{Re}_l^{-\frac{3}{4}} \quad (3.6)$$

$$\frac{v}{u} = \text{Re}_l^{-\frac{1}{4}} \quad (3.7)$$

$$\frac{\tau \cdot u}{l} = \text{Re}_l^{-\frac{1}{2}} \quad (3.8)$$

These Reynolds relationships will be used to enunciate the complications regarding the numerical simulations of flow fields and is a starting point for our next discussion on modeling.

### **3.2 DIRECT NUMERICAL SIMULATION (DNS)**

L. M. H. Navier and G. G. Stokes<sup>g</sup> independently derived the general equations governing Newtonian fluid motion almost 150 years ago. These equations, known as the Navier-Stokes (N-S) equations, coupled with the continuity equation provide a full description of the motion of a Newtonian fluid at any unique instant in time. Any techniques used to model boundary layer flow must be based on these equations. Since our situation involves an incompressible fluid, constant viscosity and density can be assumed, yielding the relations shown in Table 3-1 (given here in rectangular coordinates).

---

<sup>g</sup> Navier—French engineer, 1785-1836 and Stokes—English mathematician, 1819-1903. The Navier Stokes equations are a subject that captures the interest of many scientists, from different perspectives. According to the database Web of Science (Science Citation Index), there is an average of 15-20 published papers per week dedicated to the subject.

**Table 3-1 Equations of motion for a Newtonian fluid with constant properties**

(continuity)	$\frac{\partial}{\partial x}(u_x) + \frac{\partial}{\partial y}(u_y) + \frac{\partial}{\partial z}(u_z) = 0$	<b>(3.9)</b>
(x-momentum)	$\rho \left( \frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} + u_z \frac{\partial u_x}{\partial z} \right) = \mu \left( \frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \right) - \frac{\partial P}{\partial x} + \rho g_x$	<b>(3.10)</b>
(y-momentum)	$\rho \left( \frac{\partial u_y}{\partial t} + u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} + u_z \frac{\partial u_y}{\partial z} \right) = \mu \left( \frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2} \right) - \frac{\partial P}{\partial y} + \rho g_y$	<b>(3.11)</b>
(z-momentum)	$\rho \left( \frac{\partial u_z}{\partial t} + u_x \frac{\partial u_z}{\partial x} + u_y \frac{\partial u_z}{\partial y} + u_z \frac{\partial u_z}{\partial z} \right) = \mu \left( \frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2} \right) - \frac{\partial P}{\partial z} + \rho g_z$	<b>(3.12)</b>
These equations are simplified by using index notation		
(continuity)	$\frac{\partial u_i}{\partial x_i} = 0$	<b>(3.13)</b>
(momentum)	$\rho \left( \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) = \mu \frac{\partial^2 u_i}{\partial x_j^2} - \frac{\partial P}{\partial x_i} + \rho g_i$	<b>(3.14)</b>

It would only seem logical that resolving the flow field would be nothing more than simply discretizing the above equations, applying the boundary conditions and then iterating until convergence. This is precisely what happens with DNS and in fact would provide results equivalent to experimental data. The problem then arises in the magnitude of the discretizations required by this type of simulation. In Section 3.1, the smallest length and time scales of turbulent flow were defined. Therefore, in order for the N-S equations to yield an accurate solution, the discretizations need to be on the same order of magnitude as the smallest length and time scales. The number of discretizations ( $N$ ) in one direction can be determined from Eq. (3.15).<sup>20</sup>

$$N \approx \frac{D}{\Delta x} \approx \frac{10 \cdot l}{\eta} \approx .1 \text{Re}_l^{\frac{3}{4}} \quad (3.15)$$

Since it was noted earlier that turbulence is a 3-D phenomena, accurately modeling the flow requires a mesh approximately  $N^3$ . For most of our application problems, that would mean a model with 3 million to 30 million elements! The time scale resolution needs to be on the same order of magnitude as well if one wishes to capture all information in the temporal direction.

Solving problems using DNS demands relentless patience, since problems with the simplest geometry require hundreds or even thousands hours of computational effort.<sup>26</sup> Nevertheless, the complications enumerated above apply only to the flow aspect. It is easy to see how coupling flow and structural problems create insurmountable problems. Therefore, DNS is not a viable option and we must look to a method that is less computationally expensive.

### 3.3 REYNOLDS AVERAGED NAVIER-STOKES (RANS)

The most common remedy to the excessive number of elements required by DNS is through an ensemble averaging technique. By doing this, one models the behavior of the smaller scales and hopes to capture the gross characteristics of the flow. This is the general idea behind the RANS based equations referred to in earlier sections. The derivation is quite simple and the implementation accurate enough for most applications. This section outlines the derivation in general terms, and then discusses the advantages and limitations to this approach.

In developing the equations to describe turbulent flow, there exist fluctuations in the flow. The overall velocity vector ( $u$ ) shown in Eq. (3.16), can therefore be defined by a mean velocity ( $\bar{u}$ ) and a fluctuating component about the mean ( $u'$ ).

$$u = \bar{u} + u' \quad (3.16)$$

This expression for the fluctuations can be substituted into the relations found in Table 3-1. A simplified treatment of this derivation is for a 2-D boundary layer over a flat plate of turbulent flow. Eliminating the appropriate terms<sup>h</sup> yields Eq. (3.17).

$$\rho \left( \frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} \right) = \mu \frac{\partial^2 u_x}{\partial y^2} \quad (3.17)$$

Time-averaging the result and neglecting the lower order terms yields Eq. (3.18).

$$\rho \left( \bar{u}_x \frac{\partial \bar{u}_x}{\partial x} + \bar{u}_y \frac{\partial \bar{u}_x}{\partial y} \right) = \mu \frac{\partial^2 \bar{u}_x}{\partial y^2} - \rho \frac{\partial}{\partial y} \overline{u_x u_y'} \quad (3.18)$$

---

<sup>h</sup> From the x-momentum equation,  $u_z$  is eliminated due to the 2D assumption, gravity is not a source, and the second derivative of  $u_x$  with respect to  $x$  and  $z$  as well as the pressure gradient are negligible.

One may wonder why the averaged velocity fluctuations ( $u'_x$  and  $u'_y$ ) are still considered significant. It is true that the average of the velocity fluctuations will always be zero; however, the average of the *product* of these fluctuations is not necessarily equal to zero. Conversely, this product can contribute significantly to the overall motion of the flow. The difficulty in solving this form of the equation of motion is developing an analytical expression for the term  $\rho \overline{u'_x u'_y}$ , known as the Reynolds stress. Since the Reynolds stress is unknown because the velocity fluctuations are not computed directly, there are an insufficient number of equations for all the unknowns. Determining a way to model the Reynolds stress is called the closure problem.

The most common way to deal with the closure problem is to model the kinetic energy ( $k$ ) and dissipation rate ( $\varepsilon$ ). This is known as the  $k$ - $\varepsilon$  method and is widely used for most engineering applications. It is the most popular and widely tested of the closure models and provides reasonable accuracy for a wide range of flow geometries. However, the model is weak to inadequate for adverse pressure gradients that produce boundary layer separation. It also performs poorly when dealing with strong curvature, swirl and rotations.

In simple terms, the first step in treating the closure problem using the  $k$ - $\varepsilon$  method is to use the Boussinesq approximation shown in Eq. (3.19).

$$-\rho \overline{u'_x u'_y} = \mu_{turb} \frac{\partial \overline{u_x}}{\partial y} \quad (3.19)$$

Next, the turbulent viscosity is approximated with values for the kinetic energy of the flow ( $k$ ) and kinetic energy dissipation rate ( $\varepsilon$ ), with  $C_\mu$  as an empirical viscosity constant described in Eq. (3.20).

$$\mu_{turb} = \frac{C_\mu k^2}{\varepsilon} \quad (3.20)$$

In Eq. (3.20), the values of the kinetic energy ( $k$ ) and the dissipation rate ( $\varepsilon$ ) come from separate partial differential transport equations. Substituting (3.19) and (3.20) into the equation of motion (3.18) yields Eq. (3.21).

$$\bar{u}_x \frac{\partial \bar{u}_x}{\partial x} + \bar{u}_y \frac{\partial \bar{u}_x}{\partial y} = \nu \frac{\partial^2 \bar{u}_x}{\partial y^2} + \frac{\partial}{\partial y} \frac{C_\mu k^2}{\varepsilon} \frac{\partial \bar{u}_x}{\partial y} \quad (3.21)$$

Eq. (3.21) can be solved using numerical methods since all of the terms involve time-averaged values. The constant value of  $C_\mu$  most generally comes from empirical data or from DNS databases.

Many techniques have been developed to deal with this closure problem using some other defining algebraic, differential and/or empirical relation in the flow. Unfortunately, a set of relations that are applicable for every situation does not exist. This is partly because a closure model consists of an unavoidable empirical constant ( $C_\mu$ ) that may change for every flow case. In other words, no matter how many additional relations are incorporated, there will still be more unknowns than equations. The closure methods vary in level of sophistication and each is suitable for different types of problems. A list of a few closure models are given in Table 3-2 and are ranked in degree of complexity from the least to the greatest.<sup>27</sup> Since an ultimate RANS closure model does not exist, there are also many other models available in addition to the suite given.<sup>28</sup>

**Table 3-2 Strengths and weaknesses of popular closure models**

Method	Strengths	Weaknesses
Spalart-Allmaras	A one-equation model, which provides less computational effort than most other models. Produced for external flow over airfoils but is increasing in popularity for turbo machinery applications. Performs well for attached wall-bounded flows with weakly complex boundary layers.	Weak for adverse pressure gradients that produce boundary layer separation. Since it is relatively new, it has a lack of submodels available.
RNG k- $\epsilon$	Possesses many of the same characteristics as the standard k- $\epsilon$ , but uses mathematical group theory to determine the previously empirical constants. It performs better for moderately complex flows like jet impingement.	Subject to limitations due to isotropic eddy viscosity assumptions.
Reynolds Stress Model (RSM)	Highly rooted in the physics by solving a transport equation for each Reynolds stress.	Requires much more computational effort than any other technique

The appropriate closure model depends on the flow geometry and the desired accuracy. An advantage to using a time-averaged approach is that the computational time is generally low. The k- $\epsilon$  model generally takes a fraction of the time to arrive at a solution when compared with the DNS model. This is why it is the method of choice when it comes to parametric and engineering design. Re-running a model that takes weeks to converge after changing a parameter is too costly. This allows reasonable approximations to a flow system when the turbulent fluctuations are not important to the solution. On the other hand, the major deficiency of the time-averaged approach is that turbulent fluctuations are not described. Since the driving function in our problem lies in the fluctuations of pressure and not the mean, RANS modeling is not a viable option.

### 3.4 LARGE EDDY SIMULATION (LES)

In contrast to a time-averaged approach, LES computes the instantaneous velocity and pressure fields without the high cost of DNS. On a superficial level, LES could be thought of as the middle ground between the range of DNS and RANS. On one end, numerically modeling the flow using DNS produces the fluctuations yet is computationally infeasible. RANS modeling is computationally feasible, yet doesn't compute the fluctuations. LES captures the transient nature of the flow by spatially averaging and modeling on the subgrid. In this way, LES provides the instantaneous results as does DNS, yet it also incorporates empirical modeling like RANS to be more efficient.

LES solves the same N-S equations as DNS but the equations are “spatially filtered” to the size of the grid. Filtering the N-S equations means that the flow is resolved to a characteristic scale, usually taken to be the size of the grid, and then modeled on the smaller scales. The motivation for this comes from the fact that large eddies possess anisotropic behavior and need to be resolved. The smaller eddies possess a more universally isotropic behavior and like the RANS models can be treated from a statistical standpoint. Typically, the grid spacing is such that most of the total turbulent kinetic energy contained is in the large eddies and is directly calculated.<sup>26</sup> The remaining fraction of the kinetic energy must then be modeled for the flow to be physically realistic.

As a result, LES suffers from the high cost of simulating flows at high Reynolds numbers. At high Reynolds numbers, there is a wide range between the largest and smallest dissipative scales. Even though the smallest dissipative scales are modeled, there is still a wide range of flow scales that must be calculated directly, which can be computationally intensive.

In addition to low Reynolds flows, LES is good for unbounded flows where viscosity serves to set the scale of dissipative eddies.<sup>29</sup> Because of the coarse grid spacing in LES, near-wall turbulence effects are modeled to accurately account for the effects of the boundary.<sup>30</sup>

Progress in the numerical simulation of turbulence has been rapid since the 1990s. New techniques both for the numerical approximation of the N-S equations and for the subgrid-scale models used in LES have emerged and are being widely applied for both

fundamental and applied engineering studies. Most of the recent developments to LES have been in developing sub-grid models. Some of these include the structure-function, selective structure-function, filtered structure-function, scale-similarity, mixed, and dynamic models.<sup>30</sup>

In order to apply a sub-grid model, smaller scales of turbulence must be eliminated from the calculation through a filter. The sub-grid model is an important mode of storing and transferring kinetic energy. One of the most important functions of the filter is to distinguish between the modeled sub-grid flow and the calculated large eddy flow. This allows transfer of kinetic energy from the calculated scale to the modeled scale.

In LES, the instantaneous quantities are resolved to the size of the grid. Each variable in the flow field ( $\phi$ ) is then broken into its large scale ( $\tilde{\phi}$  or Grid Scale (GS)) and small scale ( $\phi'$  or subgrid scale (SGS)) components as shown in Eq. (3.22). Note the similarity to the velocity Eq. (3.16) in the RANS derivation.

$$\phi = \tilde{\phi} + \phi' \quad (3.22)$$

The GS component is defined by the moving average equation where  $\mathbf{x}'$  is a spatial vector and  $G$  is a function of the cell volume shown here in Eqs. (3.23) and (3.24).

$$\tilde{\phi}(\mathbf{x}) = \int_D \phi(\mathbf{x}') G(\mathbf{x}, \mathbf{x}') d\mathbf{x}' \quad (3.23)$$

where

$$G(\mathbf{x}, \mathbf{x}') = \begin{cases} 1/V & \text{for } \mathbf{x}' \in \Delta V \\ 0 & \text{otherwise} \end{cases} \quad (3.24)$$

Therefore, the expression for the moving average becomes the expression shown in Eq. (3.25).

$$\tilde{\phi}(\mathbf{x}) = \frac{1}{V} \int_V \phi(\mathbf{x}') d\mathbf{x}', \quad \mathbf{x}' \in V \quad (3.25)$$

Applying this filter to the N-S equations found in Table 3-1, spatially or locally averaged values are obtained instead of time-averaged quantities. The governing equations for LES flow are shown in index notation Eqs. (3.26) and (3.27).

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0 \quad (3.26)$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\tilde{u}_i \tilde{u}_j) = -\frac{1}{\rho} \frac{\partial}{\partial x_i} \tilde{P} + \nu \frac{\partial^2 \tilde{u}_i}{\partial x_j^2} + \frac{\partial \tau_{ij}}{\partial x_j} \quad (3.27)$$

Filtering the N-S equations produces SGS Reynolds stresses that are much smaller than the size of the mesh.  $\tau_{ij}$  is the stress tensor that represents the SGS contributions to the overall GS velocity. It is a term similar to the  $R_{ij}$  stress in RANS and is defined as the difference of the local average of the product of the instantaneous velocities and the product of the local averages as shown in Eq. (3.28).

$$\tau_{ij} = \widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j \quad (3.28)$$

$\tau_{ij}$  is modeled on the SGS and the accuracy of the model falls on the assumption that velocities smaller than the size of the grid are indeed homogeneous and accurately modeled. This results in restrictions on the grid size. A finer grid will produce a flow with minimal modeling as compared to a coarse grid. The accuracy of LES is largely a function of the resolution of the large eddies. When flows increase in Reynolds number, so does the spectrum of eddies which lends itself to finer meshes to capture all the large-scale kinetic energy. When the Reynolds number increases, the amount of modeling increases. The goal of LES is to resolve most of the flow and model very little of it. Therefore, with LES there is a trade off between grid size and model accuracy. However, if various constraints are followed a good balance can be obtained.

As with the Reynolds stress, the SGS stress,  $\tau_{ij}$ , is modeled since there are no governing equations to compute the local average of the velocity products. It is mathematically computed by relating the subgrid stress with the turbulent viscosity and strain rate shown in Eq. (3.29).

$$\tau_{ij} - \frac{1}{3} \delta_{ij} \tau_{kk} = -2\mu_t S_{ij} \quad (3.29)$$

In Eq. (3.29),  $\delta$  is the Kronecker delta,  $S_{ij}$  represents the rate of strain tensor and  $\mu_t$  is the SGS eddy viscosity.

The most common SGS eddy viscosity model is the Smagorinsky-Lilly model.<sup>30</sup> In this model, the eddy viscosity is proportional to a sub-grid mixing length ( $L_s$ ) and the strain rate tensor as defined in Eq. (3.30).

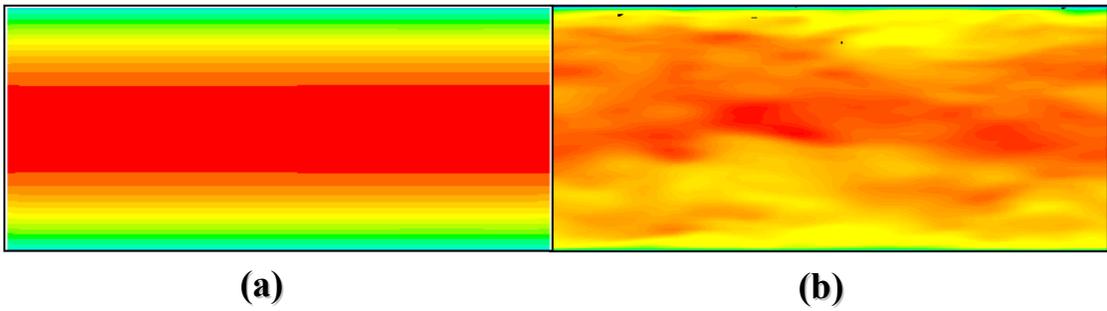
$$\mu_t = \rho L_s^2 \sqrt{2S_{ij}S_{ij}} \quad (3.30)$$

Overall, Smagorinsky's model is good for isotropic flows but usually breaks down near boundaries unless near wall treatment is employed since the contribution of turbulent viscosity at the wall should be zero. Therefore, accurately accounting for the wall boundary condition requires modifications to the mixing length. The method used in FLUENT<sup>®</sup> is shown in Eq. (3.31).<sup>31</sup>

$$L_s = \min \left( \kappa \cdot y, C_s \cdot V^{\frac{1}{3}} \right) \quad (3.31)$$

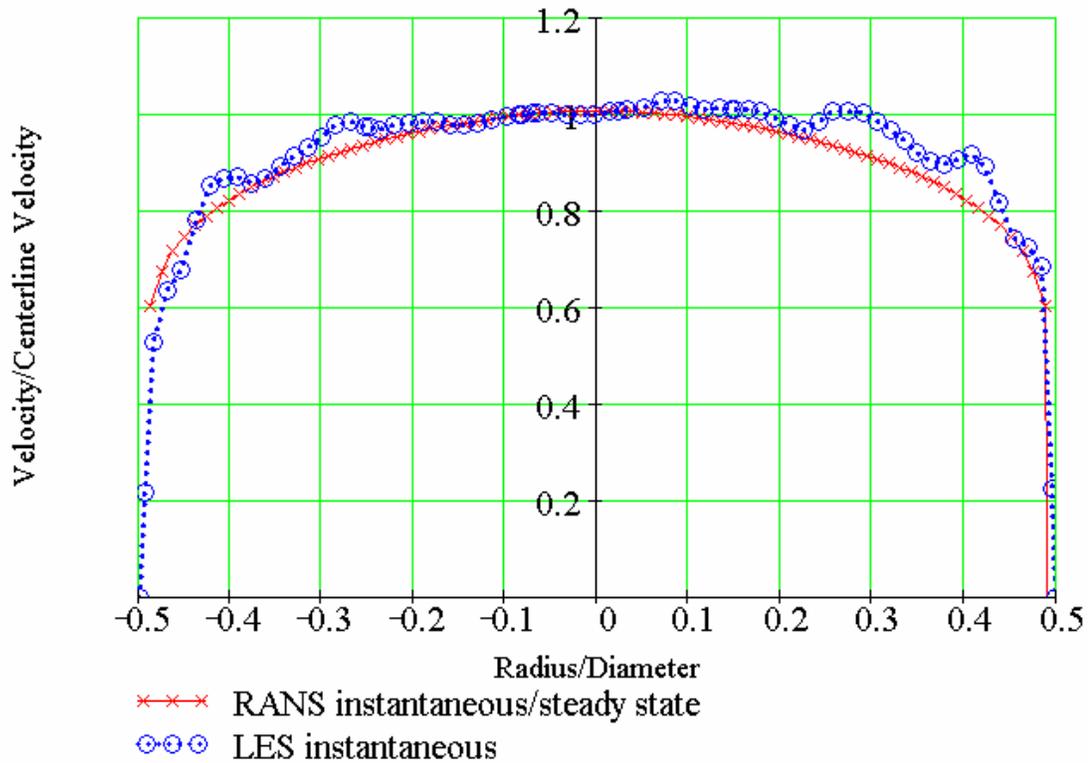
In Eq. (3.31),  $\kappa$  is the von Karmen constant ( $\kappa = 0.42$ ),  $y$  is the distance to the closest wall,  $C_s$  is the Smagorinsky constant and  $V$  is the volume of the computational cell. In general,  $C_s = 0.1$  yields the best results for a wide range of flows and will be used in this research.<sup>30</sup> Recent advances in LES have focused on ways to model sub-grid scales and account for turbulent energy transport between the modeled turbulence and the calculated turbulence using the renormalized group theory.<sup>31</sup> These developments have mainly improved LES modeling for low Reynolds flows and still lack a treatment for near wall effects. Therefore, in this research, the Smagorinsky-Lilly model will be the SGS model of choice.

To illustrate the differences between RANS and LES, the flow field for turbulent flow in a pipe was modeled using both approaches. The longitudinal section of the velocity field for the RANS and LES based models are shown in Figure 3-2. Figure 3-2a shows the average nature of a RANS model while Figure 3-2b illustrates the instantaneous (non-steady) result obtained by a LES approach.



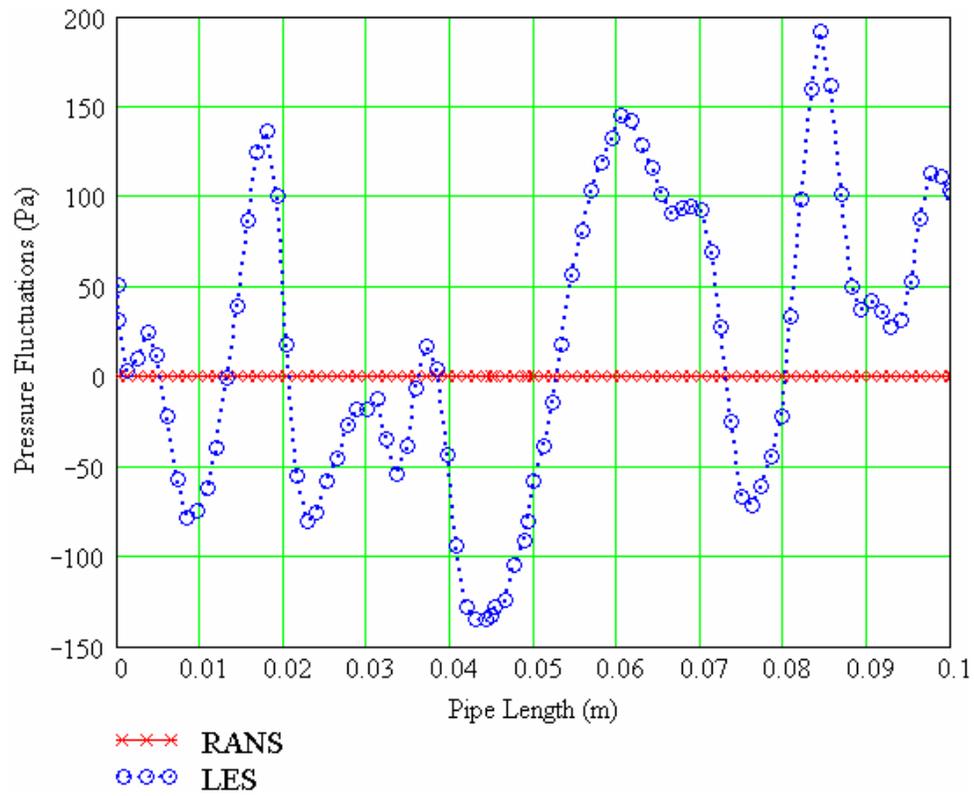
**Figure 3-2 Velocity field models of turbulent flow in a pipe, a) RANS based model, b) LES based model**

Figure 3-3 provides a plot along a cross-section of the above velocity profiles, which further illustrates the differences between RANS and LES. A plot of the pressure fluctuations<sup>i</sup> along the length of the pipe as computed by the models is shown in Figure 3-4. These figures graphically distinguish between the average values computed in RANS models and the fluctuations computed in LES models.



**Figure 3-3 Velocity profile comparison of RANS and LES based models**

<sup>i</sup> The pressure fluctuations are equal to the static pressure minus the gradient.



**Figure 3-4 Pressure fluctuations along the length of the pipe as computed by RANS and LES based models**



## 4 LES MODELING IN FLUENT

In this chapter, the development of a fluctuating turbulent fluid model using FLUENT<sup>®</sup> is presented by describing the necessary steps for obtaining an accurate LES model. The chapter is organized in the following manner:

- Model Construction (GAMBIT)
- Solving the LES Model (FLUENT)
- Model Verification
- Limitations of LES modeling in FLUENT

### 4.1 MODEL CONSTRUCTION (GAMBIT)

One underlying difficulty with any CFD model lies in the construction of the grid or mesh. As mentioned earlier, LES resolves the flow to the size of the grid and models the isotropic behavior of turbulence on the subgrid. Therefore, the accuracy of an LES model is dependent upon the degree of resolution. Ultimately, one could create a grid for an LES model on the order of the Kolmogorov scale, which would mean that the flow would be completely resolved, and no modeling would be done. On the other hand, a very coarse grid could be employed, which only resolve the very large eddies, and too much modeling would occur. The first example would be the equivalent of performing a DNS simulation, which has already been demonstrated in Section 3.2 to be too computationally expensive, and the second example would yield erroneous results. Therefore, with LES there is a trade-off between grid size and model accuracy. However, if various constraints are followed, a good balance can be obtained. The next two sections discuss the constraints to be followed in obtaining an accurate LES mesh and the practical application of these principles using GAMBIT.

### 4.1.1 Geometric Domain and Characteristic Length

Egges<sup>20</sup> suggests that the pipe domain for an LES model should be five diameters (5D) in length, with a resolution of  $y^+ < 1$  near the wall.<sup>j</sup> The implementation of LES using FLUENT was done slightly different. First, only slight correlation errors were noted in the pressure or velocity fields when a 4/3D length was used as opposed to a 5D length. The differences were not considered significant enough to change the result, so the smaller pipe length was used to expedite the solution time. Second, FLUENT handles the wall boundary condition using a law-of-the-wall approach, which means that there are no computational restrictions on the near-wall spacing. It is also recommended that each computational cell be approximately the same size to capture the developing and dissipating eddies.<sup>31</sup> Because eddies are modeled at the grid scale, very fine grid resolutions can be required for all cells in the domain. To estimate the characteristic cell length in terms of the  $y^+$  value, the equations for skin friction,  $C_f$ , friction velocity,  $u_\tau$ , and Reynolds number,  $Re_D$ , average velocity,  $U_0$  and  $y^+$ , (Eqs. (4.1)-(4.5)) must be algebraically manipulated to yield Eq. (4.6) and approximated to Eq. (4.7).<sup>32</sup>

$$\frac{C_f}{2} = 0.023 Re_D^{-0.2} \quad (4.1)$$

$$u_\tau = U_0 \sqrt{\frac{C_f}{2}} \quad (4.2)$$

$$Re_D = \frac{U_0 D}{\nu} \quad (4.3)$$

$$U_0 = \frac{Q}{\pi R^2}; \quad R = \frac{D}{2} \quad (4.4)$$

$$y^+ = \frac{u_\tau \Delta x}{\nu} \quad (4.5)$$

$$\Delta x = \frac{y^+ \nu \pi R^2}{Q \sqrt{0.023 \left( \frac{2Q}{\pi R \nu} \right)^{-0.2}}} \quad (4.6)$$

---

<sup>j</sup>  $y^+$ , or inner coordinate, is a common non-dimensional parameter to describe turbulence. It is defined as the product of a length scale and the friction velocity divided by the kinematic viscosity. See equation (4.5)

$$\Delta x = 11.416 \cdot 10^9 y^+ R \left( \frac{\nu \pi R}{Q} \right)^{0.9} \quad (4.7)$$

Based on these equations, the characteristic length of the cell,  $\Delta x$  ( $\mu m$ ), is a linear function of  $y^+$  since the radius of the pipe,  $R$  ( $m$ ), the volumetric flow rate,  $Q$  ( $liters/min$ ), and the kinematic viscosity,  $\nu$  ( $m/s$ ) are all constant for an individual flow. The larger the  $y^+$  value, the larger the length of the cell. Therefore, trial and error is required to find an acceptable medium between the total number of elements, which directly affects computation time, and the model accuracy. Figure 4-1 is a contour plot of the characteristic length equation where the contours are the size of the characteristic length given in microns. This plot can be used as a guideline for determining the size of the characteristic length.

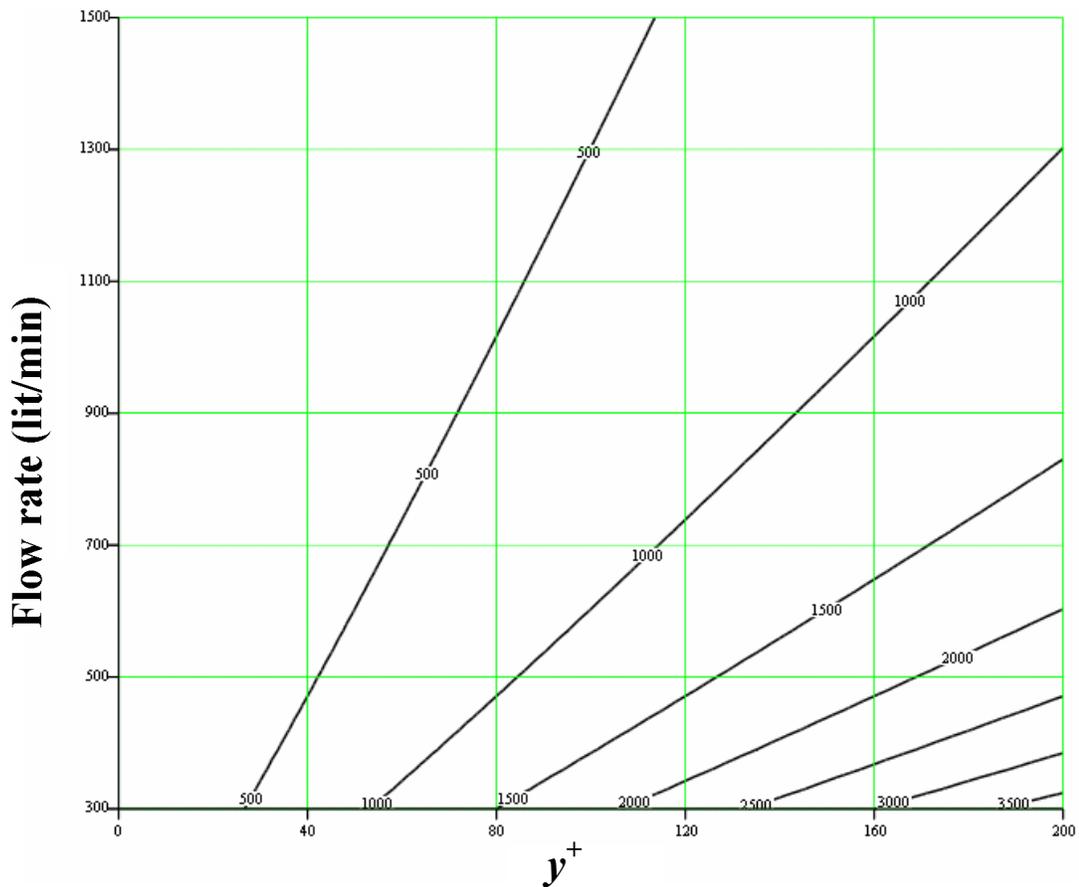


Figure 4-1 Characteristic length of mesh given in microns for a 3-inch pipe

Since the Reynolds numbers for the spectrum of flow rates ranged between 83,000 and 415,000, an extremely fine grid resolution would have been required if  $y^+ \approx 1$  was used (i.e. 5-20 microns). Grid independence studies of the velocity profiles showed that using a  $y^+$  value in the lower half of the log-law<sup>k</sup> layer ( $y^+ \approx 20-250$ ) produced adequate results and did not significantly change the fluctuating pressure fields. The organized instructions on creating this flow domain/mesh can be found in the next section.

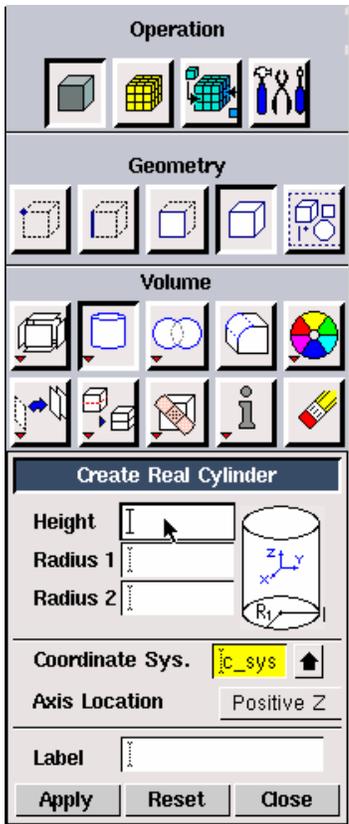
#### 4.1.2 Procedure

In this section, a detailed step-by-step outline is given for the creation of the mesh domain. Since GAMBIT was used in the creation of this geometry and mesh, the steps are directly referenced using this program.

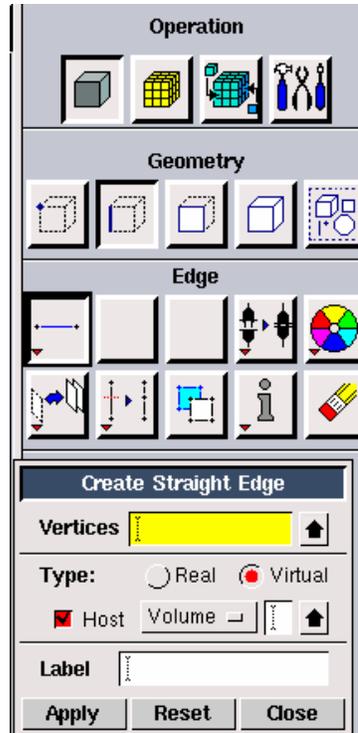
- A. Start GAMBIT from a UNIX prompt using the following command:  
    %gambit2 (use `-driver x11` as a flag if you are working over CITRIX)
- B. Create a cylindrical geometry as shown in Figure 4-2.
- C. Create a virtual line shown in Figure 4-3 from one vertex to the other. This gives the ability to specify the discretizations along the pipe wall. Choose the two vertices as end points for the virtual line. A virtual line is a line that is used solely for the purpose of node deployment. If a real line had been created, it would show up in the actual model, whereas virtual lines do not.
- D. Link the inlet and outlet faces for periodic use as shown in Figure 4-4. This step is necessary prior to meshing because in order for periodic boundary conditions to work, the inlet cells must be the same as the exiting cells. Since GAMBIT has its own algorithm for creating a mesh, it will not automatically choose the same discretization on the inlet as it does the outlet. They need to be constrained to do so—periodic boundary conditions cannot be applied in GAMBIT unless they have been linked prior to meshing.

---

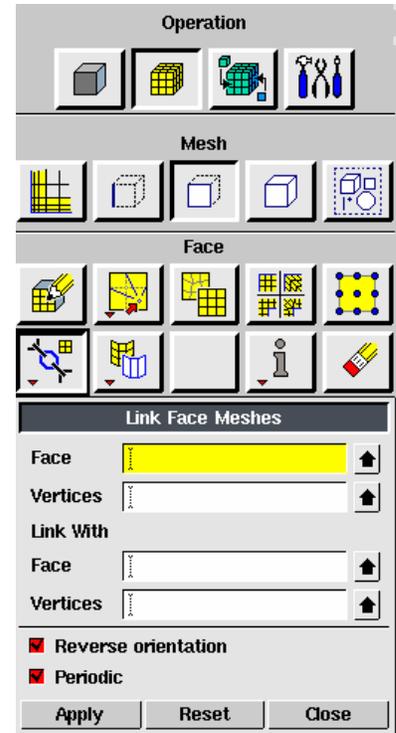
<sup>k</sup> The log-law layer is the region above the laminar viscous sublayer. Strictly speaking, another region called the buffer layer exists between the viscous sublayer and log-law region, but it is not as well defined.



**Figure 4-2 Geometry Creation**



**Figure 4-3 Virtual line creation**



**Figure 4-4 Face linking**

- E. Deploy the nodes on the edges and the wall. It seems to work best if elements are as close to being square as possible. This can be done by making the nodes equidistant and not biasing them to one side. The number of nodes deployed on the front edge and side of the pipe will be proportional according to the circumference/length ratio. To determine the number of nodes required for the virtual line and circular lines, follow the following equations given in the previous section to determine the characteristic length and then use Eqs. (4.8) and (4.9). Essentially, a rectangle is being constructed by opening up the pipe from a 3-D model to a 2-D one with the circumference of the pipe constituting the width of the rectangle.

$$N_{length} \cong \text{round}\left(\frac{L_{pipe}}{\Delta x}, 0\right) \quad (4.8)$$

$$N_{circumference} \cong \text{round}\left(\frac{\pi D}{L_{pipe}} N_{length}, 0\right) \quad (4.9)$$

- F. Mesh using the map/cooper scheme. Other schemes were tested, however, the best meshing scheme tended to be the mapped ones. Tet meshes produced meshes with far too many elements. GAMBIT should automatically select the most appropriate meshing scheme.
- G. To ensure that the generated mesh is of good quality, check the aspect ratio and equiangle skew range. Each range should have the approximate look shown in the histogram near the bottom of Figure 4-5. If both of those qualities do not appear distributed as such, re-meshing using a different scheme is necessary.
- H. Select FLUENT 5/6 as your solver: **Solve → FLUENT 5/6**
- I. Select the third icon at the top of the right hand menu. This zone sets the boundary conditions. Apply the following boundary conditions as shown in Figure 4-6.
- i. Periodic on inlet and outlet faces (make sure you select both faces before applying the periodic condition)
  - ii. Select the “Wall” boundary condition for the wall face.
  - iii. You may select the volume icon and set the fluid as “Water” if it will help you recognize it later on in the process; however, it is unnecessary.
- J. Export the mesh with an appropriate name: **File → Export → Mesh**

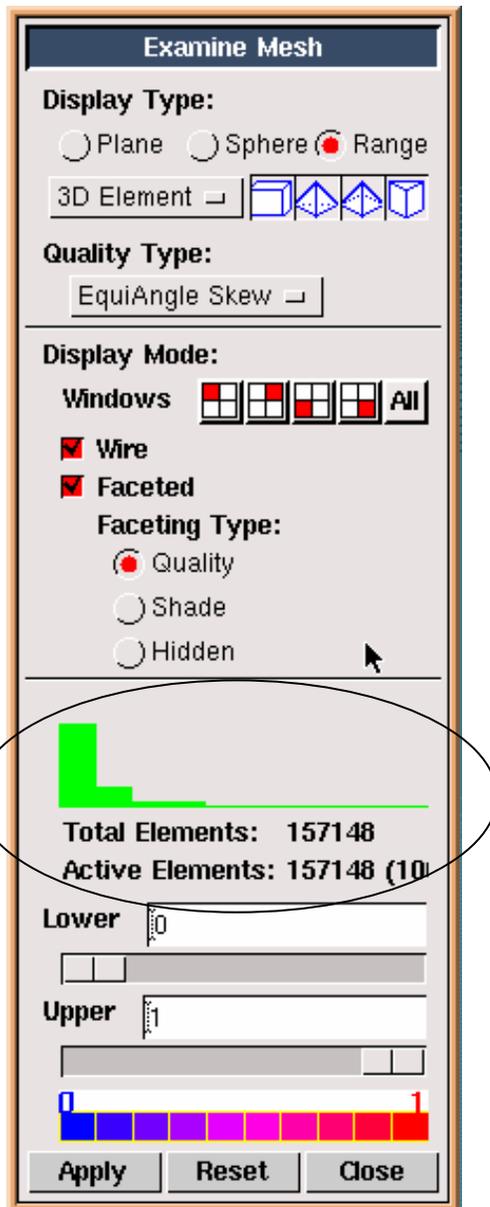


Figure 4-5 Mesh Check

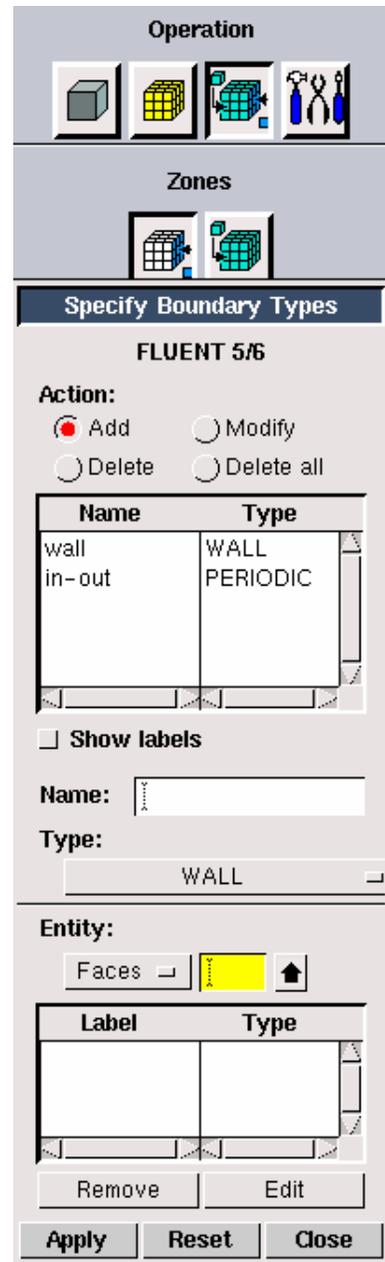
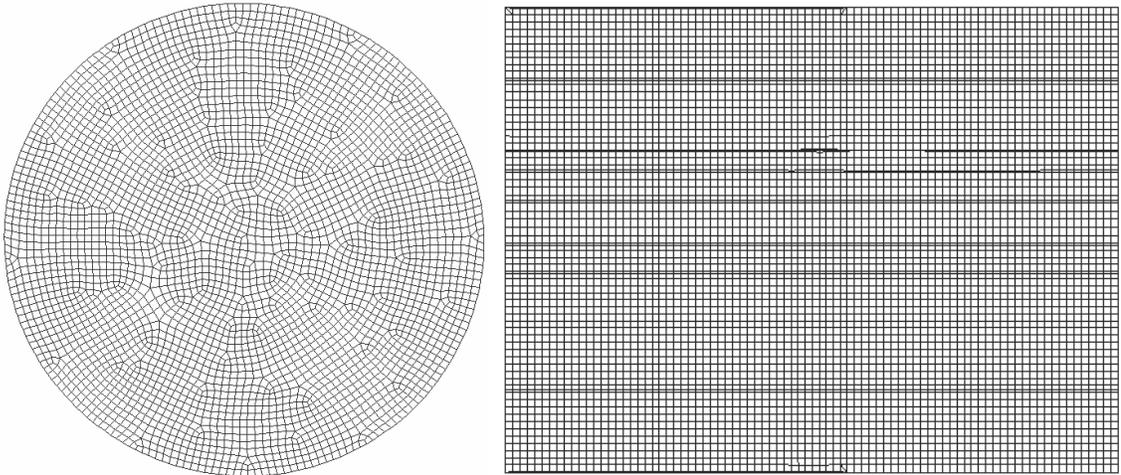


Figure 4-6 Boundary Types

The  $y^+$  and pipe domain values used are modest compared to some suggested values but provide a feasible computation time and a reasonable flow model. The final LES model used in the analysis for the 3-inch pipe includes the following fundamental characteristics graphically displayed in Figure 4-7:

- Periodic pipe length ( $L_{pipe}$ ) = 10cm
- Characteristic length ( $\Delta x$ ) = 500 $\mu$ m

- Number of elements ( $N$ )  $\approx$  300,000
- Periodic boundary conditions in the streamwise direction and no slip at the wall.



**Figure 4-7 Front and side views of discretized flow domain**

## **4.2 SOLVING THE LES MODEL (FLUENT)**

Once the domain and mesh have been imported into FLUENT, further details must be followed to obtain a LES turbulent flow solution. This section outlines the systematic procedure used to solve the LES simulation as incorporated by FLUENT and is given next as a continuation of Section 4.1.2. It is intended to serve as both a tutorial as well as necessary considerations for the construction of an LES model. The GUI for FLUENT may not look exactly like the presented pictures in the future; however, it is the author's intent to discuss the underlying principles and items of note that most likely will continue to be important considerations for many years to come. Application of these principles should transfer easily to any CFD package. Like the instructions given above, this section is accompanied with many figures to aid in the explanation of the process. For ease in repeating this procedure and reference, subheadings will be included throughout the instructions—which may be a little unorthodox, but it helps communicate the important points.

### 4.2.1 Getting Started

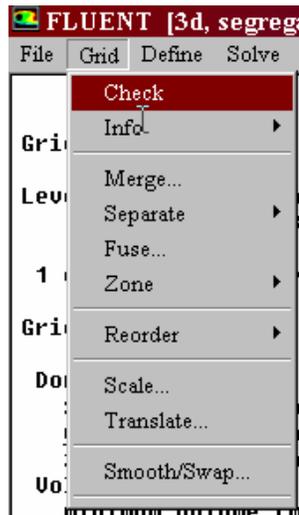
- K. FLUENT must be started differently depending on the environment and/or OS being used:
- i. Over a CITRIX client, start FLUENT using the following command line entry: `%fluent6 -driver x11 3d`. The `-driver x11` flag indicates FLUENT to use X window drivers as opposed to Open GL drivers. For some reason, over CITRIX, using Open GL drivers simply messes up all the color schemes.
  - ii. Using a UNIX or LINUX workstation: `%fluent6 3d`
  - iii. After logging onto the supercomputers Marylou or Marylou2 using `%ssh username@marylou(2): %fluent6 -driver x11 3d -t(# of processors to use to parallel process-through trial and error 16 seemed to provide adequate increase in speed as well as a sensitivity to other users. It should be noted that the use of additional processors increases the number of overall equations that need to be computed since boundary condition information is passed from processor to processor. This additional information would not exist if serial process was used)`—it is important to use the `-driver x11` flag here as well.
  - iv. If using a PC, select 3d as the mode option in the window that pops up after executing FLUENT.

- L. Import the mesh generated by GAMBIT in step J: **File** → **Read** → **Case** (Figure 4-8)



**Figure 4-8** Reading in a case file

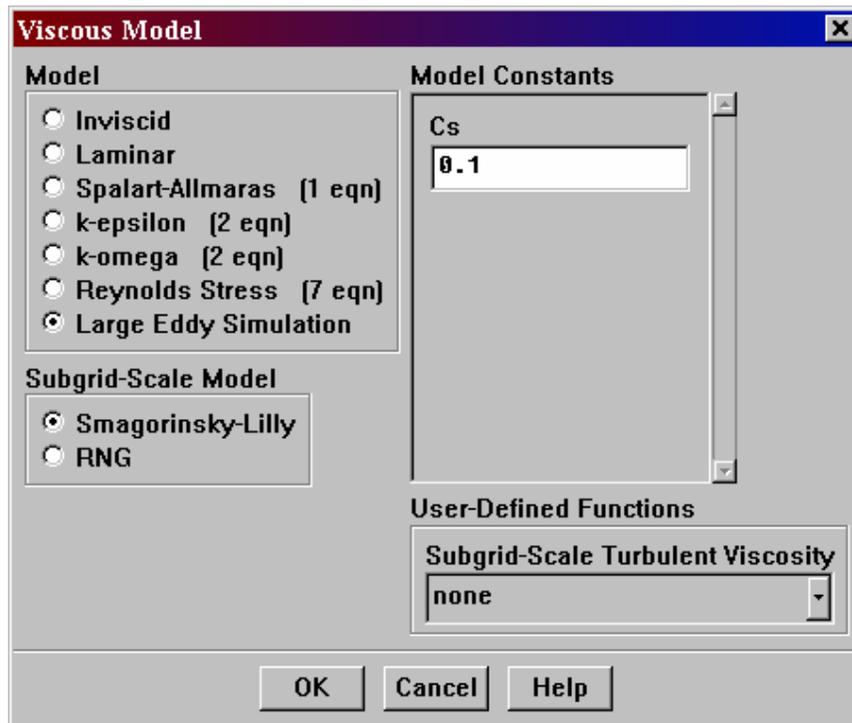
- M. Check the mesh for negative values of volume. *If the minimum volume calculated is a negative number, then you must regenerate the mesh:* **Grid** → **Check** (Figure 4-9)



**Figure 4-9** Checking the mesh

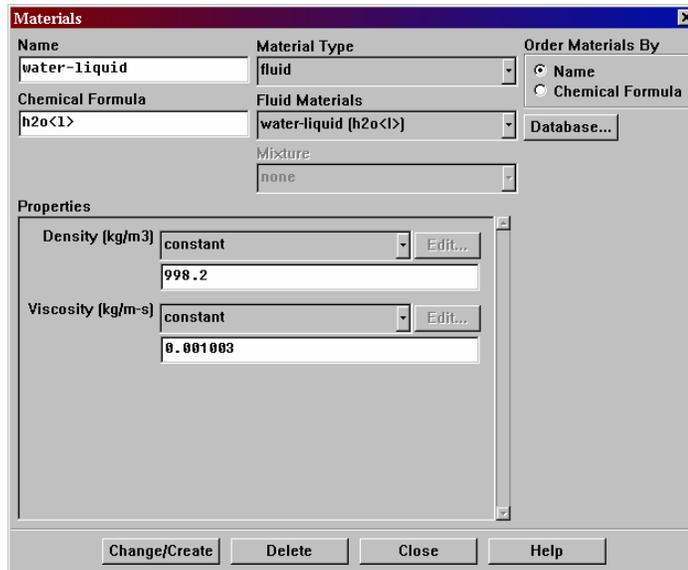
## 4.2.2 Setting up the Model

- N. Turn on the LES solver by selecting the radio button: **Define** → **Models** → **Viscous (then select Large Eddy Simulation)** The Smagorinsky-Lilly constant is recommended to stay at the default 0.1 for most applications. However, the turbulent shear stress contribution can be increased by increasing this value. Experiments have shown that a constant of 0.1 produces adequate results. (Figure 4-10)



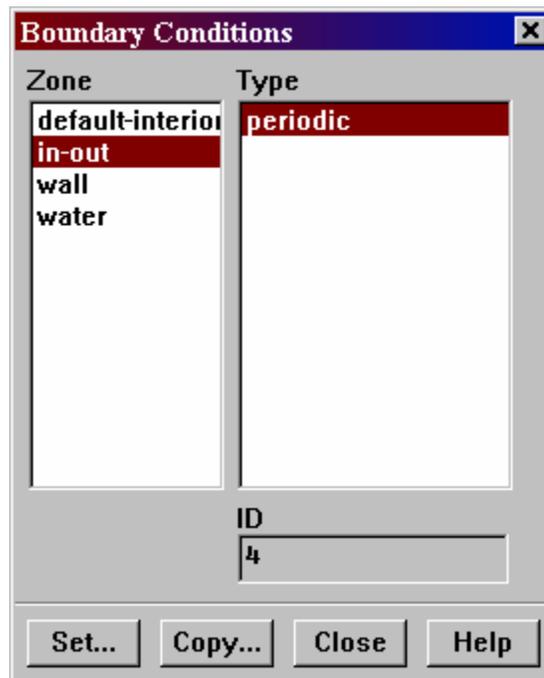
**Figure 4-10** Turning on the LES solver

- O. Select the type of fluid to be used in the simulation: **Define** → **Materials**. Either enter in the specific properties of water, or choose from a database of values by selecting h2o<liquid>. (Figure 4-11)



**Figure 4-11 Appropriate selection of fluid properties**

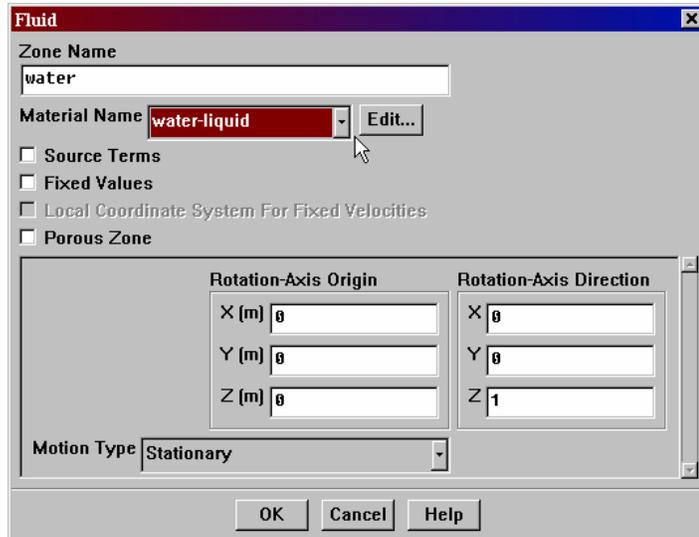
- P. Check the boundary conditions and make sure they were transferred over correctly from GAMBIT. **Define** → **Boundary Conditions** (Figure 4-12)



**Figure 4-12 Boundary condition definition**

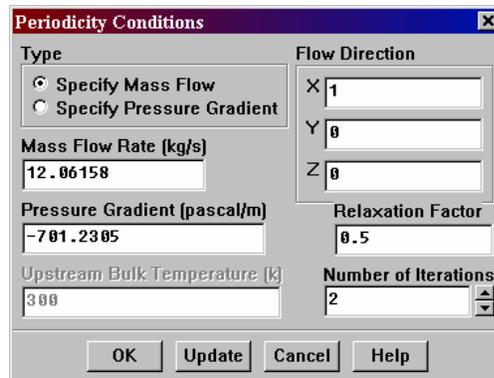
- Q. It is also important to make sure that the fluid is set correctly. The default fluid is set to be air and needs to be set to water. It is a common mistake to overlook this

by thinking that the working fluid has already been set in step O. FLUENT will use air as the working fluid if this is not done. Do this by pressing “Set” after selecting the water zone. (Figure 4-13)



**Figure 4-13 Fluid selection**

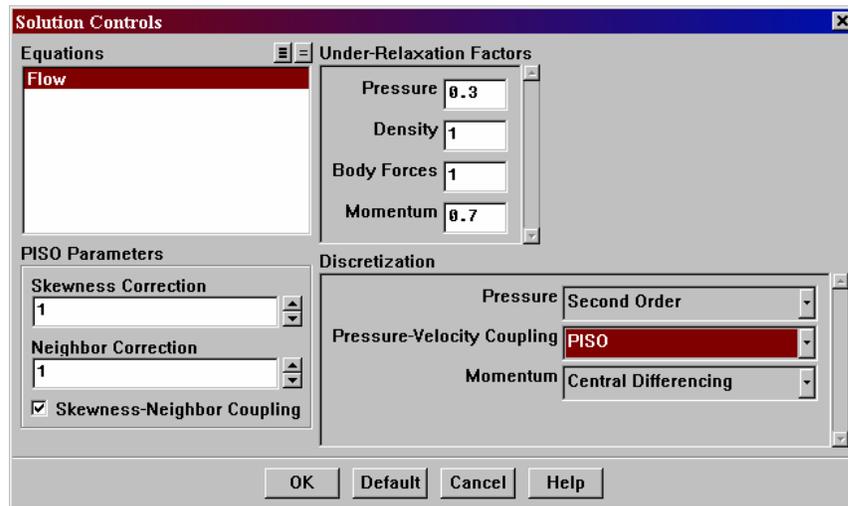
- R. Next, set the periodic conditions accordingly. The two options to choose from are to specify the mass flow rate or the pressure gradient. **Define → Periodic Conditions** (Figure 4-14)



**Figure 4-14 Periodic settings**

- S. Steps O-R should completely define the model. After this point, the remaining steps are to initialize the flow field, set solution parameters and iterate to periodic convergence. To begin, enter the solution controls window and make the following changes to the settings as seen below. **Solve → Controls → Solution.**

It is important to use central differencing momentum discretization, which may seem contrary to intuition. With Reynolds numbers in the range of interest, it would only seem logical that the downwind cells would have no influence on the upwind ones. However, LES studies show that the downwind cell affects the momentum as much as the upwind cell and it is suggested to avoid using upwind schemes.<sup>a</sup> In addition, it may be an advantage to use PISO scheme for the Pressure-Velocity coupling. It provides faster convergence for unsteady flows than the standard SIMPLE approach. It also may be an advantage to start with a first order Pressure scheme and then move to a second order scheme after a few hundred time-steps. If the model is having difficulty converging, reducing the under-relaxation factors 0.1 at a time may help. (Figure 4-15)



**Figure 4-15 Solution control settings**

### 4.2.3 Initializing the Flow

This next step is critical to the convergence of an LES model. In some circles, the initialization of the flow field is considered an art. The author could not agree more. LES models are extremely sensitive to the starting point for the solver and the similar models could be initialized the same way and one could diverge and the other converge.

<sup>a</sup> For a detailed discussion of upwind and downwind cells and schemes, see Patankar.<sup>22</sup>

Initialization can be accomplished in about four different ways, each of which may or may not converge to a correct solution.

The first initialization method is to simply set all the nodes to zero. In most CFD models, this is the typical method of choice and the solution is iterated until residual convergence. The second method is to calculate the mean flow and initialize all cells to that single value. FLUENT handles this by initializing all cells except for the nodes next to the walls. A small boundary layer is assumed and the nodes passing are linearly interpolated from zero at the wall to the mean flow value. The third initialization method is to define a velocity profile using user-defined functions. For pipe flow, a time-averaged  $1/7^{\text{th}}$  power law velocity profile would be a reasonable way to initialize.

The fourth, and perhaps most effective, method is to read in an interpolated data file. This can be done a couple ways. One way is to use the results of a similar simulation; however, one will face a cyclic problem in attempting to use this method at the beginning of a simulation since a solution to a similar problem may not exist. Alternatively, if a mathematical representation of turbulence were assumed, an existing simulation would not need to be available. Isotropic turbulence has been mathematically modeled using the spectrum of energy of a turbulent flow. Isotropic turbulence can be commonly seen in a turbulent flow field far away from any walls. G. Goldin,<sup>33</sup> a FLUENT engineer, has developed a FORTRAN code that creates an interpolation file of isotropic turbulence and is used with permission to first initialize the flow. The complete code is given in the appendix, Section 9.1, for anyone wishing to replicate this data. The isotropic initialization starts the swirling and eddies expected in a turbulent flow and LES flows initialized with this method have been found to reach fully developed conditions much more quickly than flows initialized with single values. A tutorial on using interpolation files is also given in the appendix, Section 9.2.

- T. To read the interpolation file choose **File**→**Interpolation**→**Read** and choose the file created by the FORTRAN code.

#### **4.2.4 Iterating**

The time-step is set by the average time it takes for one particle to travel across one-hundredth of the domain, shown in Eq. (4.10). Though it is also important to use an

implicit solver, the reason for making the time-step so small is not for stability, but to capture the transient nature and energy of the turbulent flow. Too large of a time-step can result in a loss of significant transient information. Large time steps can also lead to low wavenumber resolution, which will result in modeling less energy than actually exists in the flow. A wavenumber is defined by the angular sampling frequency over the mean velocity and is usually non-dimensionalized by the size of the boundary layer.<sup>34</sup> Using the Nyquist criterion, the non-dimensional wavenumber resolution for all flows will be 18.75 given by Eq. (4.11). This value captures approximately 97% of the energy by comparing this with known energy cascade plots.<sup>34</sup>

The model should be solved for approximately 5000 time-steps or until the iterations converge in under 20 iterations. Once the flow is repeatedly converging with the same number of time-steps, gather samples of time statistics to capture unsteady statistical information for comparison to known steady state solutions.

$$\Delta t = \frac{\pi R^2 L}{100Q} \quad (4.10)$$

$$(k\delta)_{nyquist} = \frac{R}{2\Delta t U_0} = 50 \frac{R}{L} = 18.75 \quad (4.11)$$

- U. Set the time step according to Eq. (4.10).
- V. After the model has been created, boundary conditions set and flow initialized, the next step is to simply iterate on the governing equations until residual convergence. It is important to realize that the residuals may not be the only way to tell if a flow field has converged. It is also important to set up monitors of other physical quantities such as the ensemble wall shear stress, average velocity across a plane or pressure gradient. The author has found that the continuity equation always seems to be the equation least satisfied and this could be because LES simulations introduce some random perturbations at the inlet to also induce the mixing nature of turbulent flow, however, the residual should be approximately 0.001.
- W. After the solution is periodically converging in the same number of time-steps per iteration, turn on the “Data Sampling for Time Statistics” to capture data for comparison against known empirical relations such as the Reichardt equation.<sup>32</sup>

- X. With the velocity profiles confirmed, it will be assumed that the instantaneous flow simulations are accurate and the wall pressures will be extracted using macros at each time-step. These tabular sets of data provide the load for the structural model.

### 4.3 MODEL VERIFICATION

Step W above alluded to the necessity of confirming that the simulated flow is accurate. There are a number of ways to provide confidence in the flow solution. Some of these methods include establishing grid-independence, examining the general characteristics of the flow, and examining pressure gradients.

#### 4.3.1 Establishing Grid-Independence

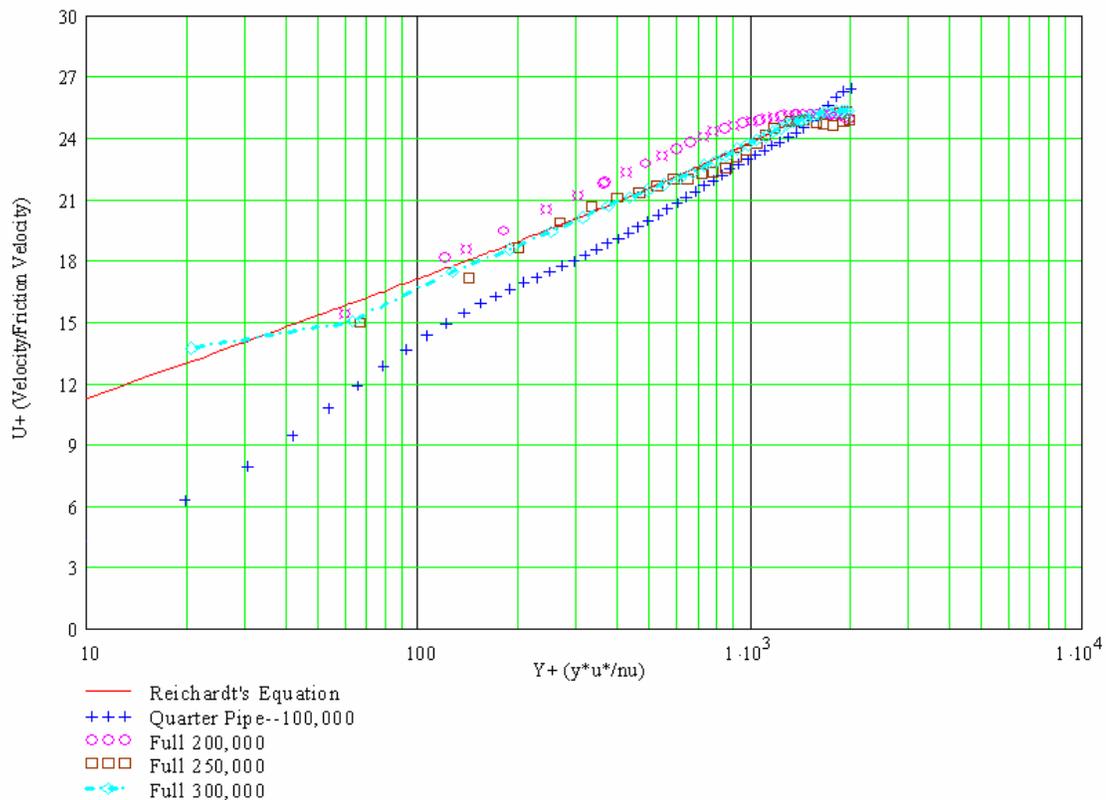
Classical numerical theory suggests the use of a grid independence study as a technique used to verify the convergence of a model. In an LES model, this can be particularly difficult since there is no real reason that the instantaneous velocities should converge to a particular value. However, time-averaged results should converge to a particular value, and should align with theory in order for a grid to be considered independent. A good benchmark for fully developed turbulent velocity profiles is the Reichardt equation, Eq. (4.12), where  $u$  is the velocity,  $u_\tau$  is the friction velocity (Eq. (4.2)), and  $y^+$  is the inner coordinate (Eq. (4.5)). Note that the characteristic length is now the distance from the wall ( $y$ ) ( $\Delta x = y = R - r$ ),  $r$  is the radial position, and  $R$  is the pipe radius.

$$\frac{u}{u_\tau} = 2.5 \left[ y^+ \left( \frac{1.5(1 + \frac{r}{R})}{1 + 2(\frac{r}{R})^2} \right) \right] + 5.5 \quad (4.12)$$

In Figure 4-16, velocity profiles are compared to each other and a plot of the Reichardt equation.

As seen in Figure 4-16, the flow field converges to the Reichardt equation as the number of elements increases. Thought was given to making a gross assumption of a 2-D axisymmetric problem; however, turbulence is a highly 3-D problem, especially when trying to capture the transient nature of the pressure fluctuations. In attempting to capture

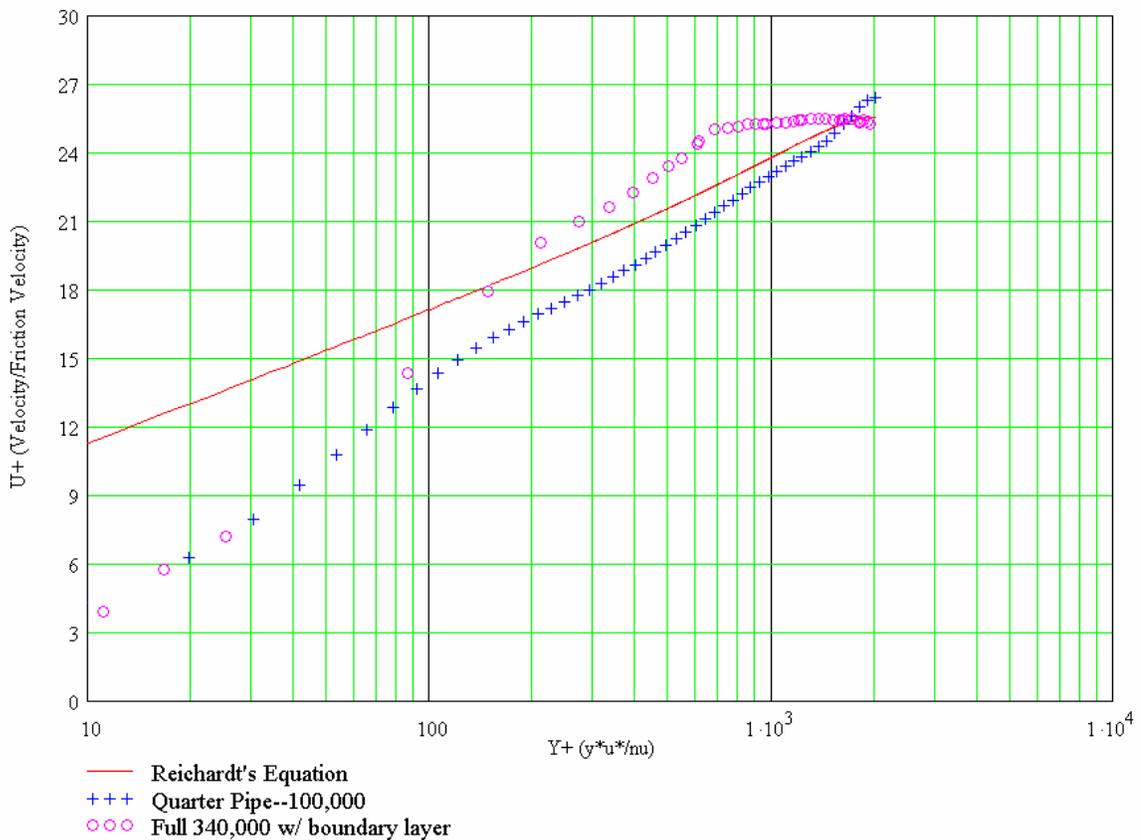
the pressure fluctuations at the wall with as little computational effort as possible, first versions of the grid simply modeled a quarter of the pipe using periodic boundaries on the two quartered sides.<sup>b</sup> Though this was an assumption, it was felt that it could still capture the fluctuations if the flow was correct. Unfortunately, the velocity profiles did not match theory as noted by the ‘+++’ plot in the above figure. The deployment of the nodes was such that a few nodes were placed in the transition layer—which we hoped would provide good resolution at the wall. Unfortunately, the grid seemed to produce a velocity profile with too large of a viscous and transitional layer. By looking at the profile, the turbulent shear stress contribution was not as high as it needed to be near the wall for the velocity to align with theory. It was felt that the boundary conditions on the quarter sides yielded such a behavior since the eddies exiting the model at one of the quarter sides did not necessarily enter on the other. This was especially seen on the centerline of the pipe, where the velocity was much too high.



**Figure 4-16 Grid independence study**

<sup>b</sup> The pipe was also modeled using symmetric boundaries; however, the results were grossly erroneous, and are hence not shown.

To eliminate geometric assumption concerns about the flow, a full 3-D pipe was developed. The designs ranged from 200,000 to 300,000 computational cells and yielded the profiles above in Figure 4-16. As it is noted, the 300,000-cell model produced the best result. The model was attempted to be improved upon using a boundary layer creation, however, it seemed to develop a behavior similar to that of the quarter pipe simulation as shown in Figure 4-17. Even though it was the most densely packed simulation, it still leads to incorrect results. The 300,000 node model velocity profiles aligned most closely with empirical data and so it was considered to be grid independent.



**Figure 4-17 Velocity profiles with boundary layer included**

It is also interesting to note that although each model behaved so differently in terms of velocity profiles, the pressure fluctuations at the wall of each model were on the same order of magnitude, and visually the location of the pressure fluctuations were the same. This is most likely because the largest of the eddies are the main contributors to the pressure fluctuations. In fact, the standard deviation of the pressure field from the

300,000 to the 340,000 cell models differed by less than five percent. This suggests that the main contribution to the value of the pressure fluctuation occurs in the log-law layer of the flow, which is reasonable since most of the chaotic activity happens in this layer.<sup>35</sup> Since pressure data was the scalar of interest, if the visual location and amplitude of the pressure fluctuations were used for grid-independence, the 100,000-node quarter model would be independent. However, since the velocity profiles on the 300,000-node full model aligned with the Reichardt equation and possessed the same pressure fluctuations, it was most conservative to use that model.

Since the cost of validating each and every flow rate with its corresponding diameter was too high given the time constraints, it will be assumed that if the range of models are valid for the 3-inch pipe following the specified instructions that any other diameter would be valid by following those same instructions since the models only parameter that was changed was the diameter. The velocity profile comparison of the Reichardt equation and the resulting 300,000 LES based model was excellent (less than 2% maximum error) as shown in Figure 4-18 and Figure 4-19 using inner coordinates and common non-dimensional parameters respectively.<sup>c</sup> Of course, there are some slight errors in the velocity profile induced by the SGS model, especially near the wall, but realize that the SGS model is most widely used for isotropic turbulent conditions where near wall effects are not considered. Granted, FLUENT provides a near wall modification to the SGS model; however there are still modeling assumptions that make it difficult to capture the viscous sublayer as noted earlier. Figure 4-18 and Figure 4-19 represent the velocity profile validation for the lower Reynolds numbers and Figure 4-20 and Figure 4-21 represent the velocity profiles for the highest Reynolds number. The higher Reynolds number simulations possessed 5% maximum error from the Reichardt equation.

---

<sup>c</sup> The velocity profiles shown here represent the 300 liter/min flow scenario. The characteristic  $y^+$  length used here is approximately 20. Similar plots exist for the higher flow rates, yet are redundant. The higher flow rates yielded a  $y^+$  value of 250, and all three validations were met.

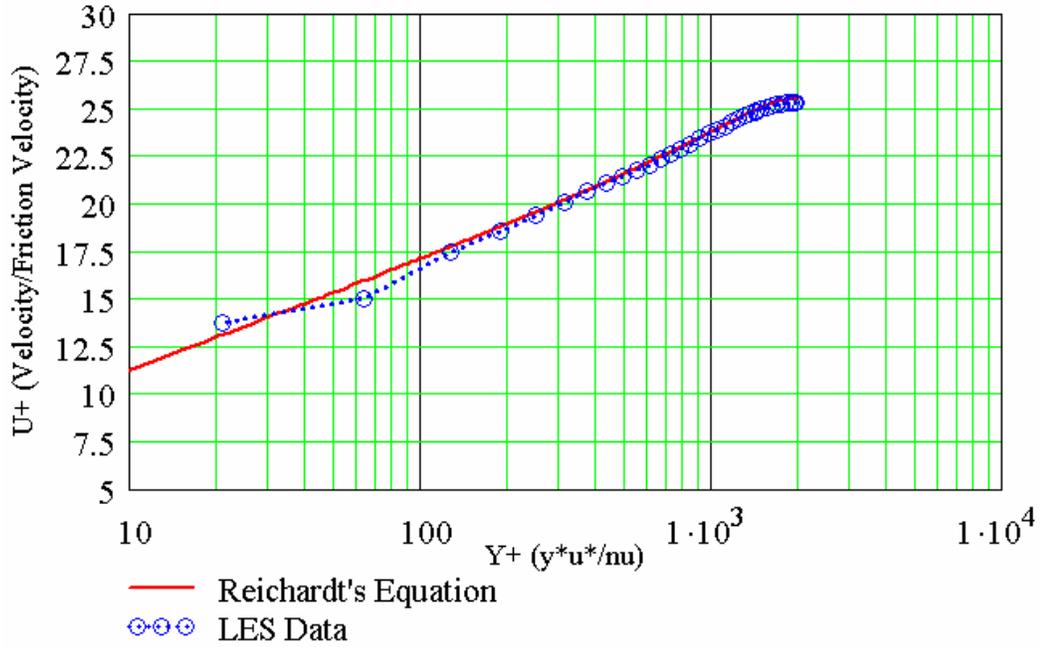


Figure 4-18 Comparison of the Reichardt equation and the LES model for the lower Reynolds number ( $Re_D \approx 83,000$ )

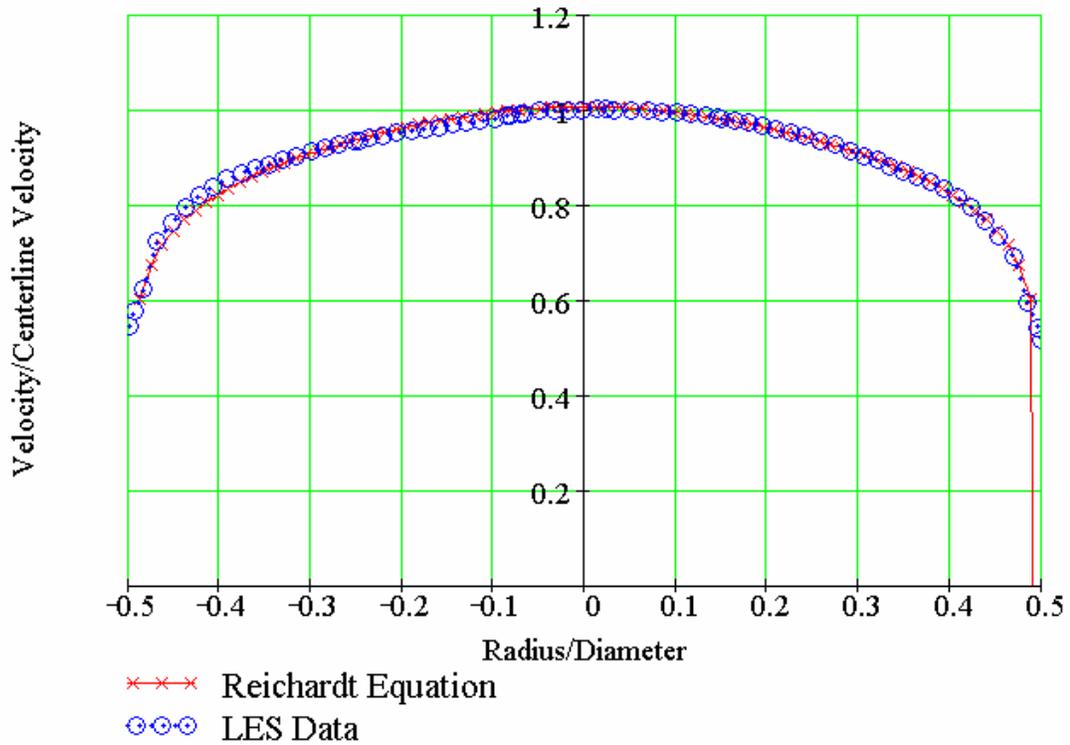


Figure 4-19 Velocity profile comparison for lower Reynolds number ( $Re_D \approx 83,000$ ) using common, non-dimensional parameters

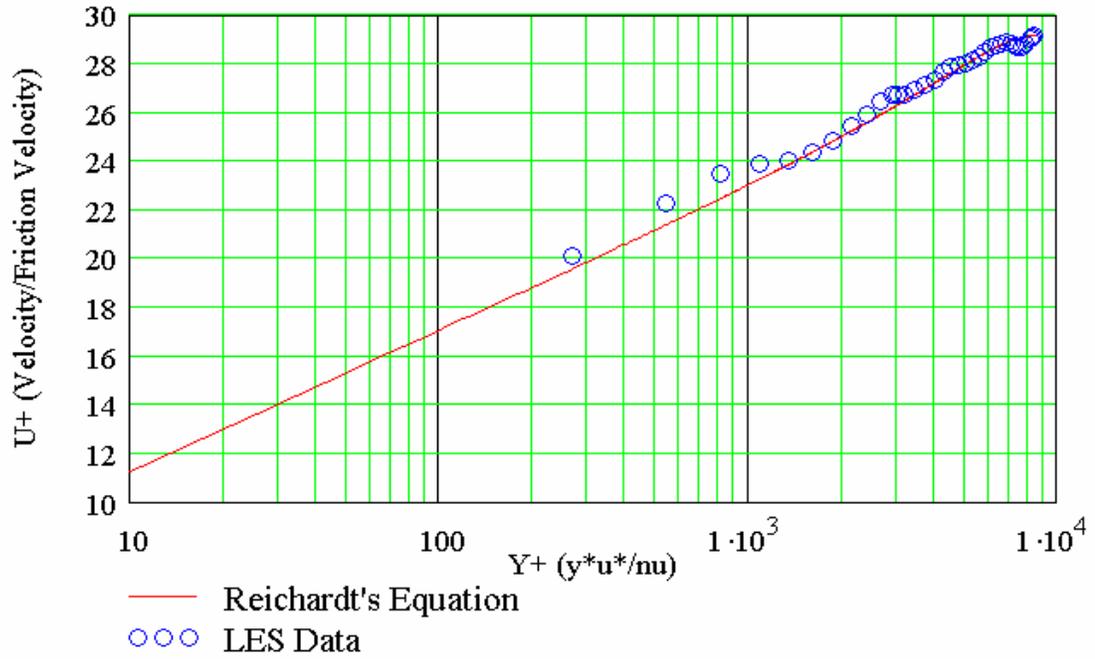


Figure 4-20 Comparison of the Reichardt equation and the LES model for the upper Reynolds number ( $Re_D \approx 415,000$ )

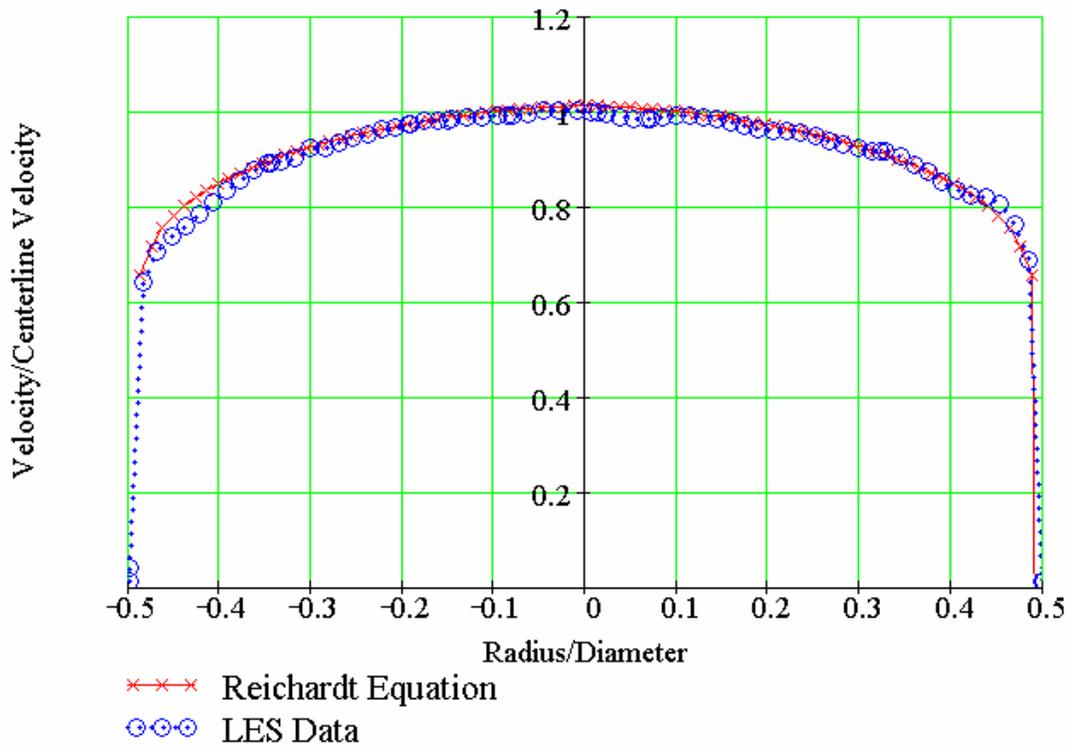
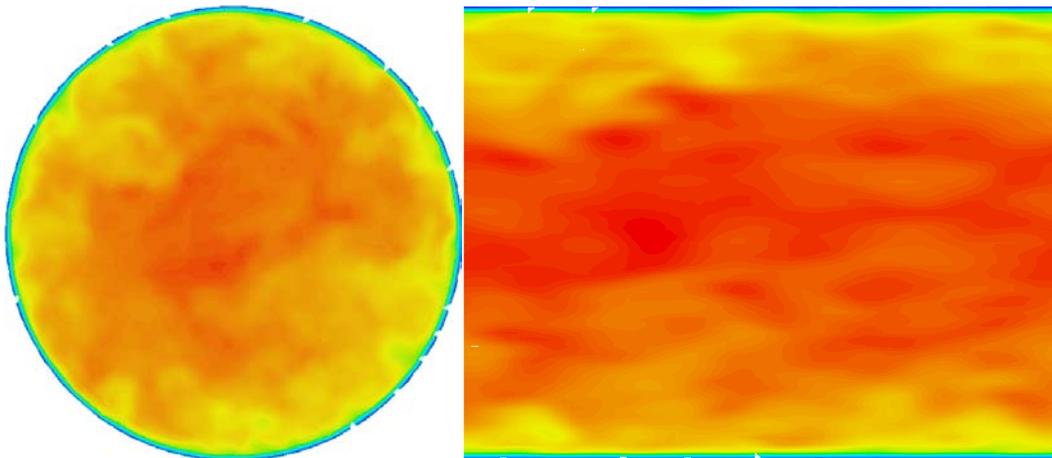


Figure 4-21 Velocity profile comparison for upper Reynolds number ( $Re_D \approx 415,000$ ) using common, non-dimensional parameters

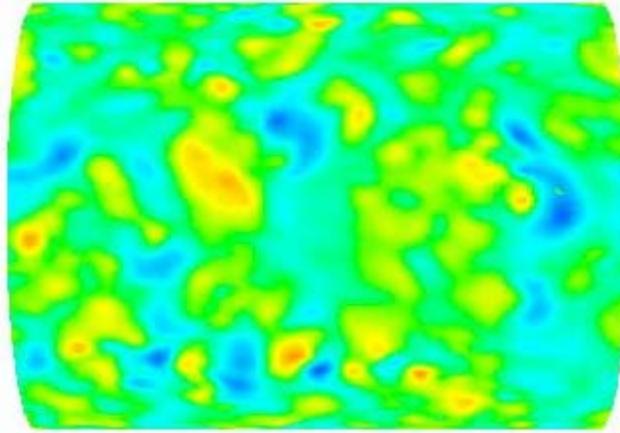
### 4.3.2 Examining the General Characteristics of the Flow

An obvious, but often overlooked, method of validation is to examine the general characteristics of the flow. If this is truly a turbulent flow, then it should appear so, with varying velocity and pressure fields. A common problem faced by LES simulations is that they take a long time to begin the random motion without proper initialization. Unless the LES flows are initialized with the isotropic turbulent conditions spoken of in the previous section, the pipe flow has difficulty initiating its mixing process. In fact, even after several thousands of iterations, it may appear that there is no turbulent shear stress contribution at all. The time-averaged velocity profiles may look parabolic or even like slug flow when this phenomenon occurs. Therefore, it is important to check for proper initialization by examining the cross sectional velocity contours and the pressure contours of the wall.

Once the model is grid independent, the first step in confirming the validity of an LES simulation is through visual techniques. Figure 4-22 represents a typical cross-sectional velocity profile and Figure 4-23 is a plot of the pressure fluctuations along the pipe wall. These results visually confirm the concept behind LES and the fluctuating velocities and pressure is obvious. Each flow rate and diameter was visually confirmed this way.

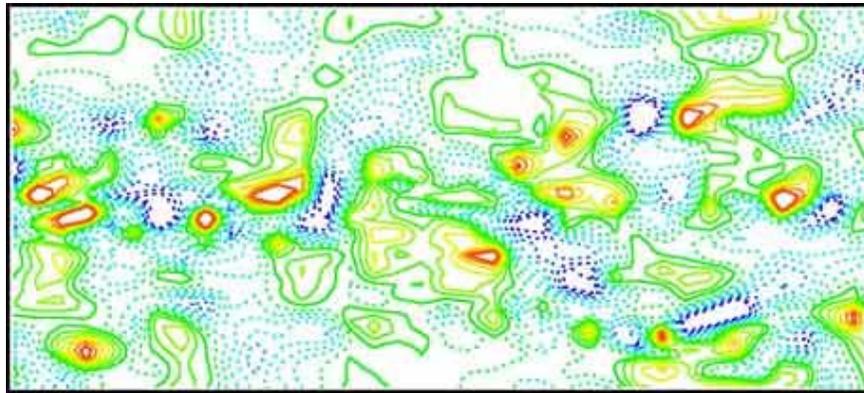


**Figure 4-22** LES velocity contour



**Figure 4-23 Fluctuating pressure contour**

The pressure fluctuations are also validated by existing LES simulations of channel flow. Figure 4-24 shows LES data collected by Chang.<sup>35</sup> The same basic structure exists in the pressure fields determined in pipe flow. The red regions represent positive pressures and the blue represent negative pressures. The footprints left in the LES channel flow are similar to the ones noted in the pipe flow simulation.



**Figure 4-24 Fluctuating pressure field for channel flow**

### 4.3.3 Examining Pressure Gradients

A third method of validation is to check the pressure gradients computed by FLUENT against the pressure gradient as calculated by the Colebrook equation, Eq. (4.13) and Eq. (4.14), where  $k_s$  is the equivalent roughness,  $D$  is the diameter,  $Re$  is the Reynolds number,  $f$  is the friction factor,  $\Delta P$  is the pressure drop,  $\rho$  is the density of the fluid,  $L_{pipe}$  is the pipe length and  $U_0$  is the average velocity in the pipe. It should be noted

that the Colebrook equation is only accurate to  $\approx 10\%$  as a general rule of thumb.<sup>10</sup> It is therefore expected that the LES solution will deliver results within that range of calculated pressure gradients.

$$\frac{1}{\sqrt{f}} = -0.869 \ln \left( \frac{k_s}{3.7D} + \frac{2.523}{\text{Re}_D \sqrt{f}} \right) \quad (4.13)$$

$$\frac{\Delta P}{L_{pipe}} = f \frac{\rho \cdot U_0^2}{2D} \quad (4.14)$$

It should also be noted here that there is no way at this point to incorporate surface roughness effects into the model since LES modeling in FLUENT assumes perfectly smooth pipe (i.e.:  $k_s = 0$ ). Surface roughness increases the shear stress on the pipe, which also increases the pressure gradient. Surface roughness also decreases the viscous sublayer to zero in the limiting case of a fully rough condition. It is then conceded that if a fully rough surface were considered, the one-way procedure assumption would never work since a structural perturbation will penetrate into the fully turbulent region. However, surface roughness will increase the size of the pressure fluctuations and will be given minor consideration in the end of Chapter 6

The pressure gradients were on average about 9% below the calculation from the Colebrook equation. Since the Colebrook equation is an empirical model of the turbulent portion of the Moody data, a probable explanation for this difference could be that every pipe possesses a surface roughness to some degree. Since surface roughness cannot be considered in an LES model, it would likely under predict the pressure gradient. The individual results are listed in Table 4-1. As stated before, the Colebrook equation is still an approximation based on empirical data and is only accurate to within 10%.

**Table 4-1 Pressure gradient verification of theory vs. FLUENT**

Diameter	Flow Rate (liters/min)	Theory (Pa/m)	FLUENT (Pa/m)	% Error
3-inch	300	-147	-136	-7%
	500	-368	-350	-5%
	750	-764	-694	-9%
	1000	-1284	-1195	-7%
	1250	-1924	-1764	-8%
	1500	-2679	-2460	-8%
1.5-inch	150	-1178	-1009	-14%
	300	-4086	-3685	-10%
	750	-21428	-18035	-16%
4-inch	300	-37	-33	-11%
	1000	-322	-298	-7%
	1500	-671	-610	-9%
Average Error				-9%

From the visual velocity and pressure fields, the time-averaged velocity profiles and the bench marked pressure gradients, it is concluded that the fluid models represent adequate approximations of the pressure profiles to be used in the structural model, which completes Goal 1 from Table 1-1.

#### 4.4 LIMITATIONS OF LES MODELING IN FLUENT

As ideal as it sounds to complete an exhaustive number of flow rates between the range of flow rates, the FLUENT models typically consist of approximately 300,000 cells and require approximately 5000 time-steps to converge. To put the amount of computational power in perspective, each cell requires four equations to be solved per iteration and each time-step takes approximately 30 iterations to converge. In other words, each simulation requires 190 billion equations to be solved for a converged solution—each equation requiring many multiplications (which are the most costly operation for a computer to perform). To reduce computational cost, the use of a supercomputer or some type of parallel processing machine is suggested; however, most

models take multiple days and weeks of straight computation to reach a fully developed solution.

In addition to the unavoidable computational costs required, there are many limitations to the scope imposed by the FLUENT software itself.<sup>31</sup> The following is a list of the LES constraints with periodic boundary conditions:

- The flow must be incompressible.
- The geometry must be translationally periodic.
- If one of the coupled solvers is used, you can specify only the pressure jump; for the segregated solver, you can specify either the pressure jump or the mass flow rate.
- No net mass addition through inlets/exits or extra source terms is allowed.
- Species can be modeled only if inlets/exits (without net mass addition) are included in the problem. Reacting flows are not permitted.
- Discrete phase and multiphase modeling are not allowed.
- Surface roughness effects cannot be incorporated.

These limitations have helped define the scope of this thesis. With this in mind, the problem will be limited to the study of fully developed water in a pipe at various discrete flow rates, as stated in Section 1.4. Two-phase, compressible flow and surface roughness effects will not be considered.



## 5 FEA MODELING IN ANSYS

In this chapter, the development of a structural model using ANSYS<sup>®</sup> is presented by describing the necessary steps for obtaining an accurate structural response to the pressure fluctuations obtained from the flow solution. The chapter is organized in the following manner:

- Basics
- Model Construction
- Model Verification
- Acceleration Extraction
- Limitations of FEA in ANSYS

### 5.1 BASICS

Finite element analysis (FEA) is one of several numerical techniques for solving boundary value problems and is used universally to calculate stress and deflection of mechanical structures. It begins with a model of the part, its material properties, and its boundary conditions. A computer program then uses this information to break the model into smaller fractions, called “finite elements.” These elements are analogous to the “cells” referred to in CFD. The type of finite element (FE) depends on the problem to be solved. The behavior of each element is readily predicted by a set of mathematical equations. This set, made up of literally thousands of equations, is essentially a matrix version of Hooke's Law.<sup>36</sup> In 1678, Robert Hooke set down the basis for modern finite element stress which states an elastic body stretches in proportion to its applied force. In other words, finite element analysis is nothing more than solving an enormous system of interconnected springs. In mathematical terms,  $F = KU$ , where  $F$  represents force,  $K$  is a proportional constant, and  $U$  is the linear displacement.<sup>37</sup> This law is also expressed as  $\sigma = E\varepsilon$ , where the stress on an element is proportional to the strain by Young's modulus.

On a simplistic level, an FEA program applies a known force ( $F$ ) to a part of known material and geometry (yielding  $K$ ) and solves for  $U$  by premultiplying the force vector by the inverse of the stiffness matrix ( $U = K^{-1}F$ ). For dynamic problems, such as the one of interest in this case, the static spring problem simply becomes a dynamic one. As such, the classic spring-mass differential equation of motion is used as the constitutive relation ( $M\ddot{U} + B\dot{U} + KU = F$ , where  $M$  is the mass,  $B$  the damping). These types of problems are solved using various approaches such as modal methods, component mode synthesis, or condensation algorithms.<sup>38</sup> As was the case with LES, the computer is required because of the astronomical number of calculations needed to analyze a substantial structure. FEA, as well as CFD, maintains growing popularity since the power and low cost of modern computers has facilitated the use of such methods.

The word ‘finite’ in FEA comes from the idea that there are a finite, or countable, number of elements in a finite element model. Just as the set of elements would be joined together to build the whole structure, the equations describing the behaviors of the individual elements are joined into an extremely large set of equations that describe the behavior of the whole structure. A computer then accounts for all the individual behaviors to predict the behavior of the actual object by solving a set of simultaneous equations. From the solution, the computer extracts the behavior of the individual elements, which provides enough information to calculate the stress and deflection of all the parts of the structure.

The solution procedure in FEA is very similar to the one employed in CFD with a few subtle differences. Primarily, FEA linear analysis problems have one unique solution to the system. Unlike CFD solvers, only one iteration is required to satisfy the system of equations since the system is linear. Multiple iterations are only employed in FEA when large plastic deflections are considered to be relevant. Second, FEA uses finite element methods as opposed to finite difference methods employed by CFD. Finite element methods utilize integral techniques, such as Rayleigh-Ritz, for solving the governing partial differential equations. This provides more freedom in the creation of the unstructured shapes than with the finite difference method. Therefore, finite element analysis is a way to deal with structures that are more complex than can be dealt with

analytically using partial differential equations. Naturally then, FEA deals with complex boundaries better than finite difference equations.

To determine the structural response of our pipe due to the extracted pressures, an FEA program called ANSYS will be employed. ANSYS has been developed using the above theory and is an internationally recognized, reliable FEA software package.<sup>39</sup> The remainder of this section discusses the FEA model, which is the second part of the procedure in determining the dynamic response of a pipe transporting turbulent flow. The model construction procedure will be followed by a discussion of the FE model validation. The extraction procedure of the accelerations from the ANSYS model will then be presented. Finally, the limitations of the FE modeling capabilities will be discussed.

## **5.2 MODEL CONSTRUCTION**

Analogous to Sections 4.1 and 4.2, this section contains the details required for the successful construction of the structural model of the pipe. This section also contains subheadings that represent significant milestones in the model creation and can be easily referenced should the reader wish to return to this section for replication of work. Again, it should be noted that as this section can be used as a guided tour, its main purpose is to discuss underlying principles in the model creation.

The steps to the model construction are listed and discussed below:

- A. Open ANSYS and make sure the working directory is set correctly.
- B. Select the Preferences and make sure the structural box is checked. This will turn off all other options associated with the other solution methods, customizing the GUI for a structural problem. Make sure the discipline is chosen as an h-type.<sup>d</sup>

### **5.2.1 Shell Elements**

Shell elements were originally developed to efficiently represent thin sheets or plates of steel or aluminum, both flat and curved surfaces. They include out-of-plane

---

<sup>d</sup> h-type (linear interpolation) and p-type (polynomial interpolation) are both available and can be chosen according to preference. Since it is expected and confirmed that the loading will produce very small deflections, which justifies the use of the linear interpolation. For more information, see Ansys users manual.

bending effects in their fundamental formulation, as well as transferring shear, tension, and compression in the plane. They are also much more computationally efficient than a solid element. These reasons make it ideally suited for our problem.

- C. Choose the element type to be Shell93 (Figure 5-1). Of all the shell elements, Shell93 is particularly well suited to model curved surfaces and seems to be the most robust and suited for this type of linear dynamic problem because it possesses four nodes on each corner and four mid-side nodes as in Figure 5-2. It also allows for all six degrees of freedom at every node that allows for quadratic deformations in the plane of the shell. It contains the highest number of nodes possible for a shell element, which will become very important later in the discussion of the pressure mapping.

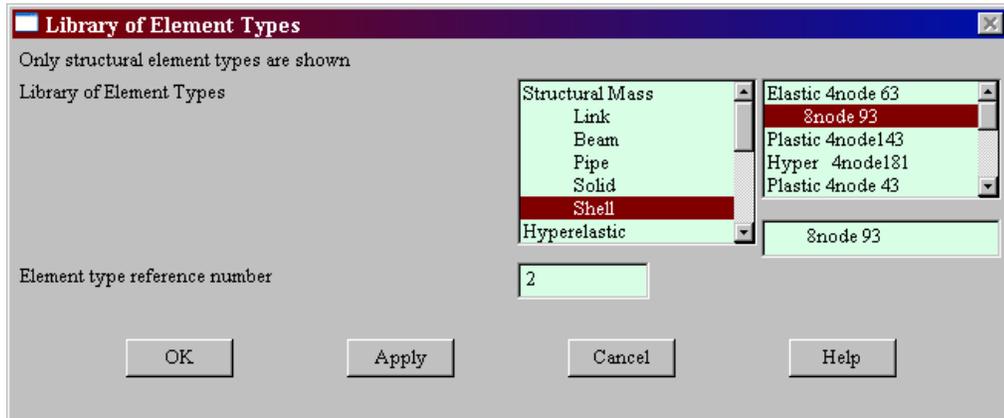


Figure 5-1 Setting the element type to Shell93

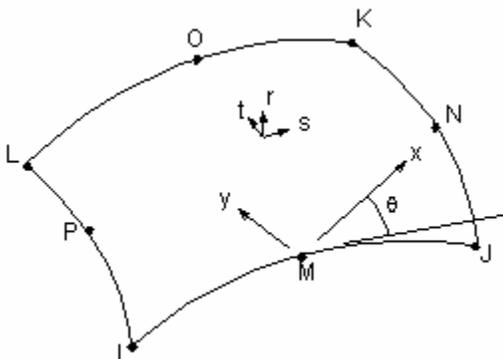
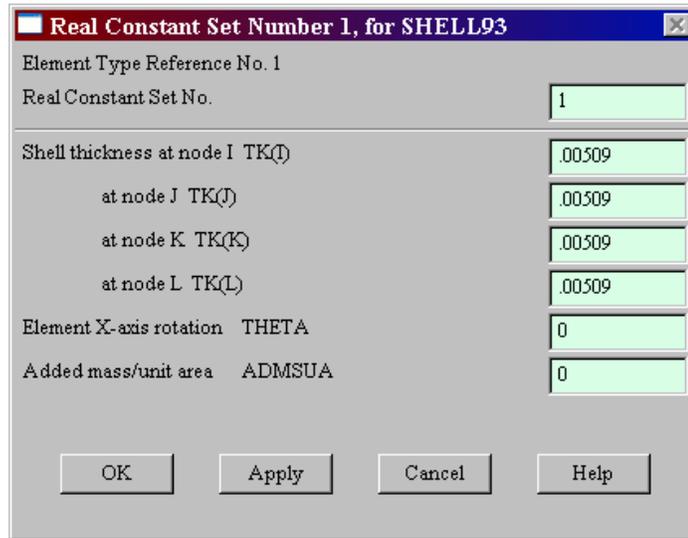


Figure 5-2 Geometric representation of a Shell93 element

- D. Add the thickness to each node according to the specified thickness of the chosen pipe. The thickness at each node should be the overall thickness of the pipe unless attempting to model a surface of changing thickness. In most cases, a constant thickness should be entered at each node as in Figure 5-3.



**Figure 5-3** Setting node thicknesses

- E. Input the material properties. In most cases, common material properties should already exist in the database.

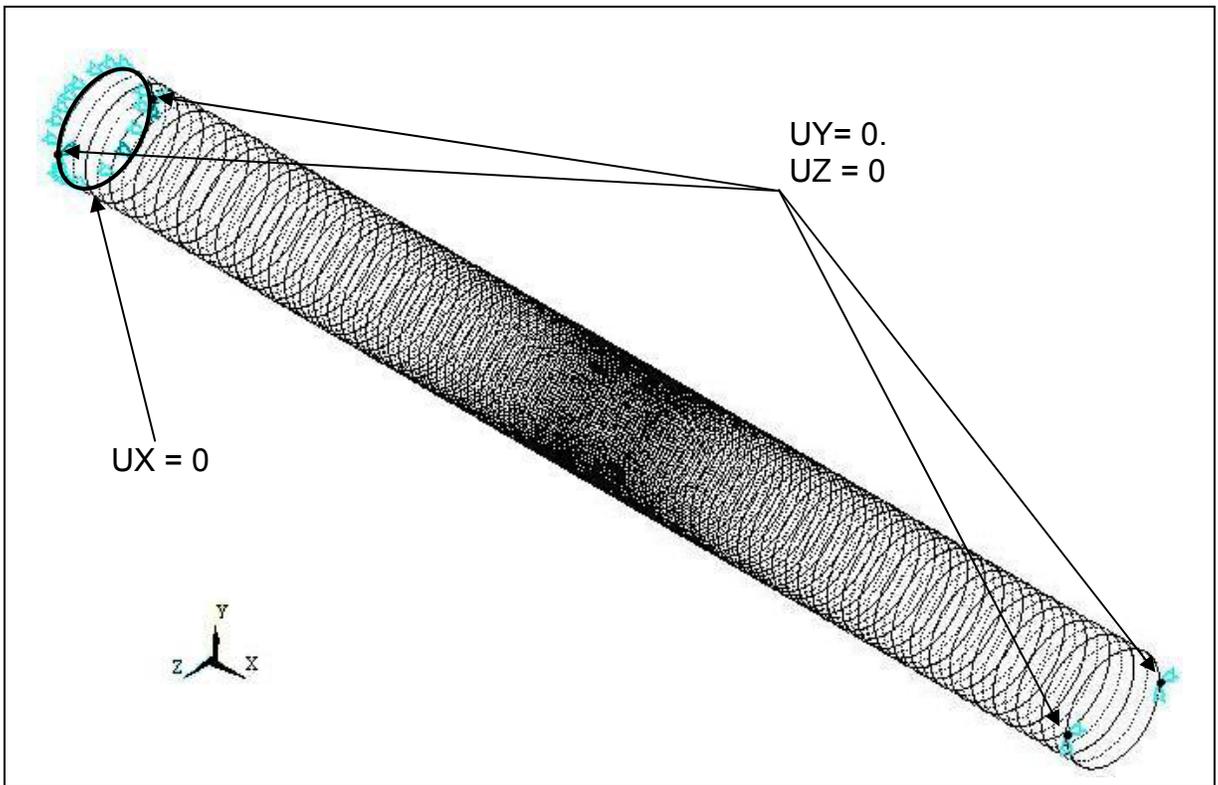
## 5.2.2 Boundary Conditions

Modeling the boundary conditions for our situation became an exceptionally involved problem. The experimental test setup at Idaho State University (ISU)<sup>°</sup> contained a five-meter-long near-fixed support set up with intermediate hangers along the pipe to provide support. Keep in mind that the LES model required a length of only  $L_{pipe} = 4/3D$ . The spacing of the grid points along this small section produced 15,000-18,000 pressure points. Ideally, every one of those points needs to be imported into the structural model. To get the entire pressure field required for the actual experimental results would require a structural model to consist of over 1,000,000 nodes! Unfortunately, structural models are not made to be that large—in fact, the license of ANSYS here at BYU will only allow

<sup>°</sup> ISU is the location of the experimental air-water flow loop and is where Evans collected most of the experimental data.

100,000 nodes. Even a model this size would cripple the supercomputer since a serial version is all that is afforded to us here at BYU. To deal with this issue, model assumptions needed to be made in order to accommodate the limitations on the structural model. For this problem, many iterations of boundary conditions were suggested and attempted to obtain reasonable experimental conditions. The first iterations of the structural model consisted of a model of the same dimension as the fluid model (4/3D). If the right boundary conditions were applied, it was hoped that this model would work. The worst-case situation would be found if the small pipe was simply allowed to respond to the pressure load free of any boundary conditions. It was proposed to simply fix a node on the bottom of the pipe and allow everything else to remain free. This, however, posed a problem since the solution produced dominant stress concentrations around the fixed node, acting as a sink for the energy. The single fixed node destroyed all the energy transferred by the fluid to the structure. It then became clear that alternative boundary conditions needed to be employed. The entire list of the different options will not be listed, however it is sufficient to say that many possibilities were investigated to model the 4/3D section of pipe. All options tried were deemed insufficient in capturing the underlying physics of the problem.

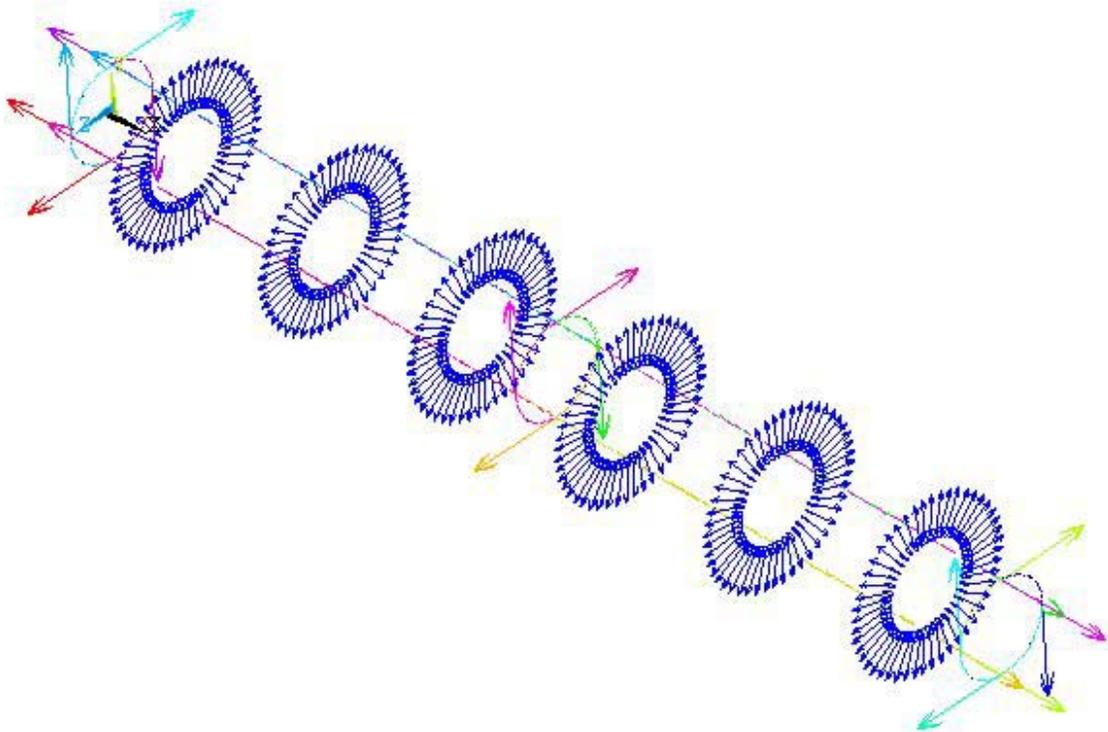
It was noted that the accelerometer mounted on the experimental setup rested in the middle of two hanging supports 1.1 meters apart. Since it was virtually impossible to model the entire 5 meters of pipe with fixed ends, this set up may indeed provide a reasonable approximation. Therefore, one end of the pipe was secured in the direction of the pipe along its circumference ( $UX = 0$ ) and then two nodes on each end were secured in the radial and transverse directions as a simulation of the hangers ( $UY = 0$ ,  $UZ = 0$ ). No rotations were fixed since the hanging pipe supports would allow rotation in all directions. These boundary conditions are visually represented in Figure 5-4.



**Figure 5-4 Nodal deployment with applied boundary conditions**

F. Since the boundary conditions must be appropriately applied as stated in the earlier sections and the area of interest is in the center of the pipe, uneven grid placement helps facilitate the issue with incorporating the appropriate boundary conditions as well as capturing the necessary information at the center of the pipe. Even though the pipe model has been reduced from 5 meters to one, this still requires a model with approximately 200,000 nodes to capture all available pressure information. Unfortunately, this is also too large of a model to solve. To deal with this issue, it was determined that clustering the nodes near the middle of the pipe would be the best approach to including the boundary conditions yet applying enough nodes at the area of interest. To do this, two different sections of the pipe must be created: a course deployment of nodes at the boundary and a fine deployment in the middle. The model will then be mirrored around the center plane of symmetry.

G. Create the geometry by first creating a circle and then copying that image twice along the pipe—once for the middle and end. Create the surface of the pipe using a skinning operation and then “glue” them all together using the glue operation so there is one cohesive pipe entity. It is critical that the area directions are visible because the applied loads will act in the same direction as the area direction. To ensure that the area directions are heading in the outward normal direction, turn on the area direction lines in the symbols menu. The first half of the geometric model should look something like Figure 5-5. It is also critical that the line directions in the lengthwise direction of the pipe are noted but not necessarily aligned with a certain direction. The implementation of step F is accomplished by unevenly distributing the node deployment—larger spaces near the boundary to fine spacing near the center. The circular lines will have nodes evenly deployed and direction is not a necessary consideration.



**Figure 5-5** Creating areas by skinning

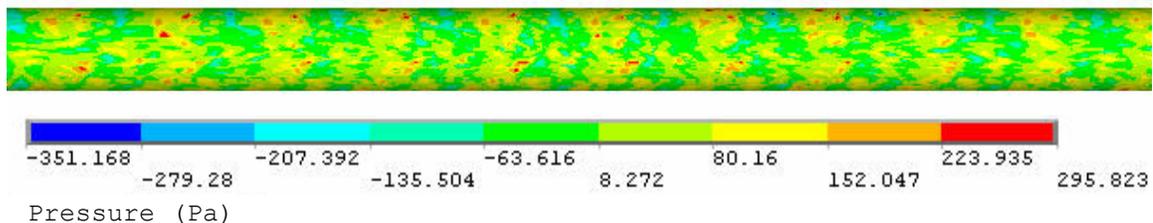
H. To create the mesh, space the nodes such that a majority of the nodes lie towards the middle of the pipe. To do this, section the boundaries of the pipe using a

fewer number of nodes than the middle. Experience has shown that a 1/10 ratio works fairly well. Next, discretize the edge lines unevenly by placing more nodes toward the center and mesh using a Quad/Mapped 3- or 4-side scheme. Other schemes may be used as long as the number of nodes is about 30,000 per model. Experience has also shown that if errors or warnings exist in the mesh, their will also exist errors in the solution and the program will most likely crash during the solve phase. The final nodal deployment should look similar to Figure 5-4.

- I. Apply the boundary conditions as described above by first selecting the nodes on one side of the model and setting all  $UX = 0$ . Pick the middle nodes on both of the ends and set  $UY = 0$  and  $UZ = 0$  as shown in Figure 5-4.

### 5.2.3 Pressure Mapping

- J. Once the model has been created with the appropriate boundary conditions delicate care needs to be taken to import the pressures extracted from FLUENT into ANSYS. First, periodically copy the 4/3D pressure profiles to cover the full structural domain. Next, read the tabular data into the structure and interpolate between the pressure and structure data points. Repeat this same procedure for each time-step. An example of the mapped pressure on the inside pipe surface at a given instant in time is shown in Figure 5-6, where the legend units are given in pascals. Here it is easy to see the repeating pattern from the periodic replication of the FLUENT pressure fields. The macros used to import the data and write a load case file for each time-step can be found in Section 9.4.



**Figure 5-6** Structural model with mapped pressure

## 5.2.4 Running the Transient Analysis

There are three different types of transient analysis options in ANSYS: full, reduced, and mode superposition.<sup>40</sup> The full is the most general and makes no simplifying assumptions about the model. Faster solution times could be arrived at using the reduced or modal methods. However, these methods require a premonition of how the model will likely behave when loaded so that certain degrees of freedom (DOF) are specified. Since the varying pressures will likely produce deflections and rotations in an unpredictable pattern, again, the most conservative approach was to use a full transient solution procedure.

K. Given the magnitude of the pressure fluctuations, large deflections were not expected, however, there is a guideline that if the out-of-plane deflection of a flat plate or shell is greater than half the thickness, then membrane forces become significant in resisting the applied load. In ANSYS, this calls for activating a Large Displacement solution (i.e. geometric nonlinearity). Ignoring this can result in missing the inherent strength of the design. With this in mind, the solution procedure could change if the deflections generated are larger than half of the shell thickness. However, small deflections were assumed and verified and 50 time-steps were loaded to determine the structural response of the pipe. The number of time-steps was not dictated by the amount of pressure data available from the FLUENT model, but the size of the files generated by the ANSYS solution procedure. Fat32 file system will only support files that are less than two gigabytes (GB). To solve a transient analysis, the structure stiffness matrix  $K$  and the mass matrix  $M$  must be constructed. This yields a matrix file that is approximately 1.8 GB. When the solution is complete using 50 time-steps, the results file is the same size. This is the extent of the storage capacity of a file and thus our model. With the limited number of time-steps available for use and the time-step constraint on the LES model, a very short actual simulation time proved to be yet another problem. This issue was aided by taking every fourth LES pressure field, ramping the load steps to gradually merge together and using 5 substeps in every time-step but only writing out the values at the end of the

time-step. This way more information could be extracted from the 50 time-step constraint.

### 5.3 MODEL VERIFICATION

Unfortunately, finite element analysis can easily produce misleading or erroneous results if the problem is incorrectly formulated. Validation of the model is just as critical for the structural model as it is for the fluid models. The model can be validated by applying a constant internal pressure and comparing four quantities against well-documented theory for an infinitely long pipe: the VonMises stresses at the inner and outer diameter, and the change in length and diameter. The VonMises stresses are given in Eqs. (5.1)-(5.4) where  $\sigma_1$ ,  $\sigma_2$  and  $\sigma_3$  are the principle stresses in the axial, tangential and radial directions respectively, and  $\sigma_{vm}$  is the VonMises stress.  $P$  is the internal pressure,  $t$  is the thickness,  $R_i$  and  $R_o$  is the inner and outer radius, and  $r$  is the radial position of interest. The change in length of the pipe is given by Eq. (5.5) where  $\nu$  is Poisson's ratio,  $L_{pipe}$  is the original length and  $E$  is the modulus of elasticity. Finally, the change in radius is given by Eq. (5.6).<sup>41</sup>

$$\sigma_1(r) = 0 \quad (5.1)$$

$$\sigma_2(r) = \frac{P \cdot R_i^2 (R_o^2 + r^2)}{r^2 (R_o^2 - R_i^2)} \quad (5.2)$$

$$\sigma_3(r) = \frac{P \cdot R_i^2 (r^2 - R_o^2)}{r^2 (R_o^2 - R_i^2)} \quad (5.3)$$

$$\sigma_{vm}(r) = \sqrt{\frac{(\sigma_1(r) - \sigma_2(r))^2 + (\sigma_1(r) - \sigma_3(r))^2 + (\sigma_2(r) - \sigma_3(r))^2}{2}} \quad (5.4)$$

$$\Delta L_{pipe} = \frac{-P \cdot \nu \cdot L_{pipe} (R_o + R_i)}{2E \cdot t} \quad (5.5)$$

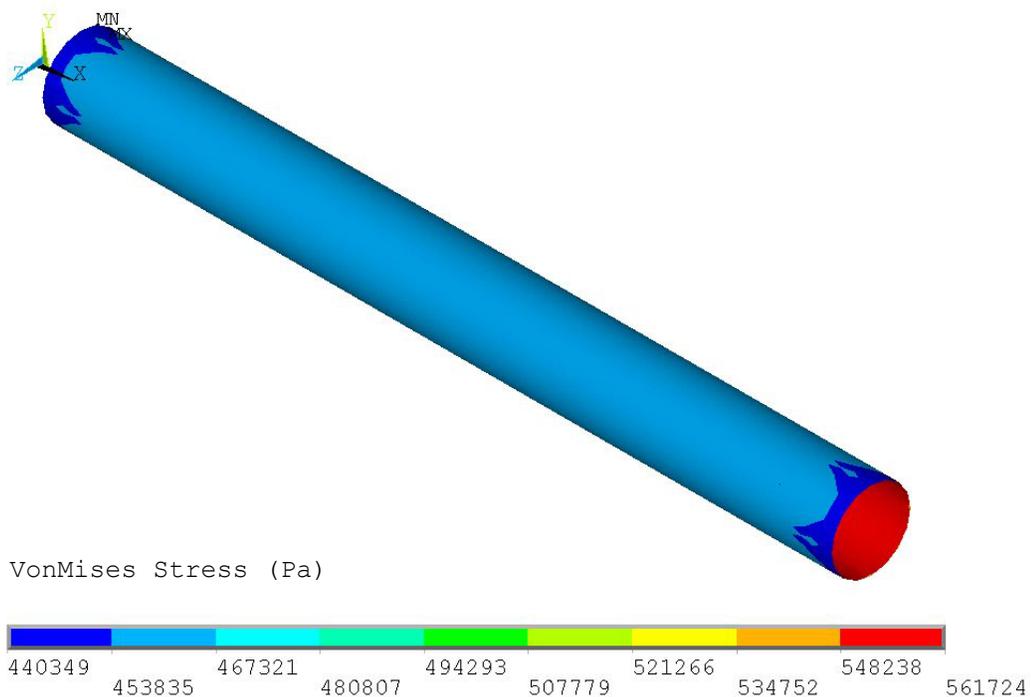
$$\Delta R = \frac{2P \cdot R_o \cdot R_i^2}{E (R_o^2 - R_i^2)} \quad (5.6)$$

For the validation model, the boundary conditions consist of a simple fixed node at the middle of the pipe to model the appropriate boundary conditions, and the results in Table 5-1 are given for all the diameters at the 70-kPa and 100-Pa internal pressure loads. It cannot be assumed that if the 3-inch model is valid, all other diameter models are also valid since the different structural models, unlike the flow models, are not just scales of the 3-inch diameter model. The distance between boundary conditions was a fixed value at 1.1 meters. Therefore, the ratio of pipe diameter to length will be different for each model and should be validated.

**Table 5-1 Theory vs. ANSYS for static internal pressure**

Pressure	Diameter	Criteria	Theory	ANSYS	% Error
70-kPa	3-inch	$\sigma_{vm}(R_o)$ (MPa)	0.45314	0.4543	-0.256%
		$\sigma_{vm}(R_i)$ (MPa)	0.56143	0.5592	0.397%
		$\Delta l$ ( $\mu\text{m}$ )	-0.8608	-0.802	6.830%
		$\Delta R$ ( $\mu\text{m}$ )	0.10234	0.0975	4.729%
	1.5-inch	$\sigma_{vm}(R_o)$ (MPa)	0.33015	0.33014735	0.002%
		$\sigma_{vm}(R_i)$ (MPa)	0.43936	0.4357755	0.815%
		$\Delta l$ ( $\mu\text{m}$ )	-0.6563	-0.66	-0.564%
		$\Delta R$ ( $\mu\text{m}$ )	0.03889	0.0378	2.798%
	4-inch	$\sigma_{vm}(R_o)$ (MPa)	0.55766	0.558943	-0.230%
		$\sigma_{vm}(R_i)$ (MPa)	0.66542	0.66301	0.362%
		$\Delta l$ ( $\mu\text{m}$ )	-1.0342	-0.976	5.626%
		$\Delta R$ ( $\mu\text{m}$ )	0.16418	0.157	4.373%
<i>Average % Error</i>					<i>2.07%</i>
100-Pa	3-inch	$\sigma_{vm}(R_o)$ (Pa)	647.3493	649.128	-0.275%
		$\sigma_{vm}(R_i)$ (Pa)	802.0386	799.0138	0.377%
		$\Delta l$ (nm)	-1.2297	-1.144	6.969%
		$\Delta R$ (nm)	0.14621	0.14333	1.970%
	1.5-inch	$\sigma_{vm}(R_o)$ (Pa)	471.64867	471.7772	-0.027%
		$\sigma_{vm}(R_i)$ (Pa)	627.65203	622.6637	0.795%
		$\Delta l$ (nm)	-0.93756	-0.876	6.566%
		$\Delta R$ (nm)	0.05555	0.0543	2.250%
	4-inch	$\sigma_{vm}(R_o)$ (Pa)	796.6507	798.3915	-0.219%
		$\sigma_{vm}(R_i)$ (Pa)	950.6038	947.0136	0.378%
		$\Delta l$ (nm)	-1.47741	-1.394	5.646%
		$\Delta R$ (nm)	0.23454	0.223	4.920%
<i>Average % Error</i>					<i>2.45%</i>
<b><i>Overall % Error</i></b>					<b><i>2.26%</i></b>

From the data above, the model matches theory within 1% of the stress computations, and 7% for changes in length for all pipe diameters investigated. The only discrepancies between the model and ANSYS occur at the boundaries—typical for all models. The graphical representation of the VonMises stress (Figure 5-7) shows that at the outer edges where the pipe is not fixed the model produces inaccurate results. The inappropriate boundary conditions and lack of elements are speculated to be the cause of the breakdown of the equations. However, in the region where the accelerometer is to be placed, the ANSYS model aligns with theory as shown in Table 5-1.

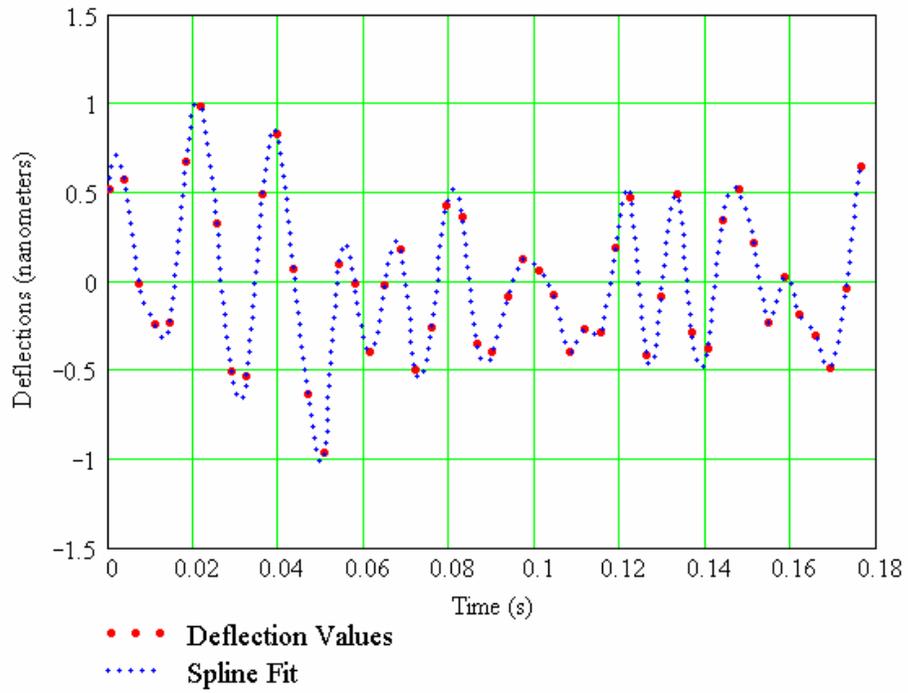


**Figure 5-7 VonMises stress on 3-inch pipe to a 70-kPa internal load**

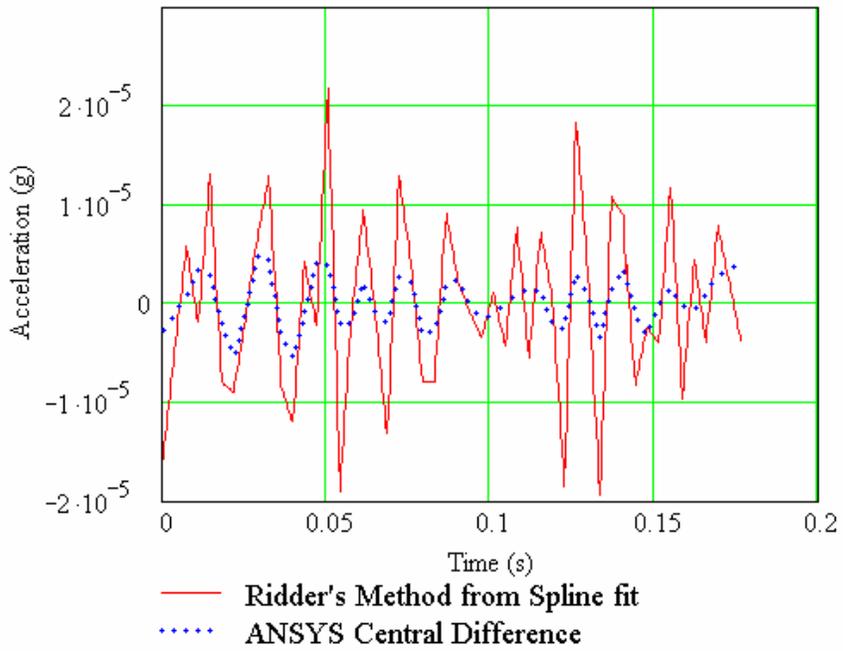
A full transient grid independent solution is too computationally expensive given the size of the files created and the time it would take. Also, recall that only a fraction of the fluid model pressure data can be incorporated into the model and 30,000 is the maximum number of nodes allowed. Therefore, it is likely that even a static solution will be different if more or fewer nodes than the given model are used with the extracted pressure fields. Therefore, confidence in the model will be assumed using the static analysis provided above for the two internal pressures.

## 5.4 ACCELERATION EXTRACTION

With the transient model created, the end goal can be determined by reading the acceleration of a node at the middle of the pipe—approximately where an experimental accelerometer would be placed. The accelerations can be computed using finite difference techniques from the time history values of deflection. This can be done within ANSYS, or using commercial software packages such as Mathcad.<sup>®</sup> Mathcad's spline-fitting techniques create a continuous function from the given discrete points. Figure 5-8 demonstrates how a spline fit recreates the probable motion of the pipe using the given data. The first and second temporal derivatives of the position can be computed using a number of finite difference techniques that range from simple first order to higher order. The simplest of these is the central difference technique employed by ANSYS and the higher order is Ridder's method used by Mathcad. With Ridder's algorithm, the first derivative is expected to be accurate to within seven or eight significant digits, if the value of the derivative is not too close to a singularity of the function. The accuracy tends to decrease by one significant digit for each increase in the order of the derivative.<sup>42</sup> Comparison of these two methods can produce very different results (Figure 5-9); however, since a spline fit with Ridder's method seems to be a more accurate way to represent the transient response, it will be assumed that this method will yield the more acceptable prediction of the standard deviation of the acceleration. Although the central difference technique provides a more visually pleasing acceleration plot, it severely underestimates accelerations produced by the sharp peaks and valleys shown in the position plot in Figure 5-8.



**Figure 5-8** Recreation of continuous deflections vs. time



**Figure 5-9** Central difference vs. spline-Ridder technique for estimating acceleration

With the acceleration vs. time plots, standard deviations are established using the same 50 discrete time locations obtained from the original displacement plot. Results from this exercise along with a comparison between the ANSYS and Mathcad calculations of the accelerations are found in Section 6.1.2.

## **5.5 LIMITATIONS OF FEA IN ANSYS**

To reiterate the limitations mentioned in this section, we cannot solve this problem using more than 50 time-steps due to the 2-GB file size constraints imposed by the Fat32 file type. The 30,000-node model is set by the computation time and file size. There is also a limitation on the physics of the Shell93 elements in that they do not consider the effects of system damping.<sup>40</sup> However, for the materials of interest, the damping is generally insignificant.

## 6 RESULTS

This chapter presents and discusses the results obtained after following the procedures described in Chapters 4 and 5. Section 6.1 presents the results through a comparison of the standard deviations of the pressure fluctuations and accelerations to the flow rate for a 3-inch pipe. Section 6.2 addresses possible corrections to account for surface roughness effects. Section 6.3 builds on the results presented in Section 6.1 by illustrating the effects of changing the pipe material and diameter. Section 6.4 concludes by describing the pressure distributions on the inside surface of the pipe.

### 6.1 STANDARD DEVIATIONS VS. FLOW RATE

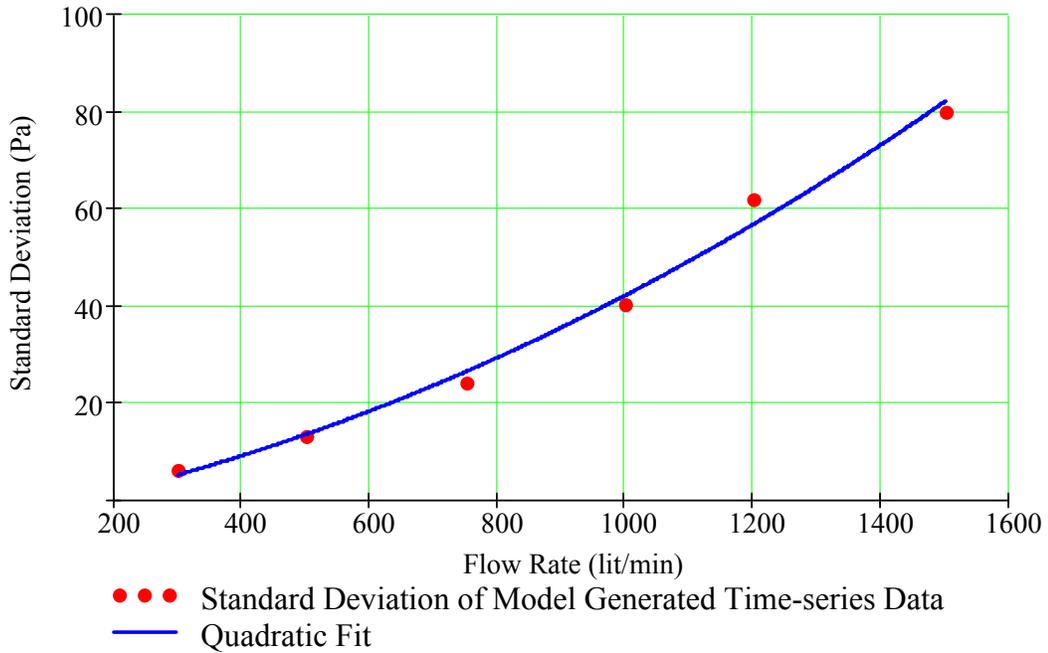
One important contribution to the development of a non-intrusive flow meter these results have to offer is the conformation of the trend noted by Evans<sup>4</sup> (Figure 2-1). This section presents the relationship between the flow rate and standard deviations of the pressure fluctuations and acceleration.

#### 6.1.1 Pressure

At each time-step, the pressure fluctuations at each point on the pipe wall were extracted from the fluid solution. The descriptive statistics of this data show that as the flow rate increases, the standard deviation of the pressure fluctuations also increases. The standard deviation of the pressure field vs. flow rate for the 3-inch pipe is plotted in Figure 6-1. A second order polynomial given by Eq. (6.1), where  $P'$  is the standard deviation of the pressure field and  $Q$  is the flow rate in lit/min, fit the data with an  $R^2 = 0.997$ . However, the quadratic term improves the fit by less than 2 %, suggesting that a

linear relationship may be a more realistic representation.<sup>f</sup> It could then be argued that the standard deviation of the pressure fluctuations could simply be scaled by the Reynolds number. Since pressure is a source of energy transfer between the fluid and the structure, a similar relationship between flow rate and pipe acceleration is expected. Section 6.1.2 verifies this possibility.

$$P'(Q) = 2.376 \cdot 10^{-5} Q^2 + 2.012 \cdot 10^{-2} Q - 2.477 \quad R^2 = 0.997 \quad (6.1)$$

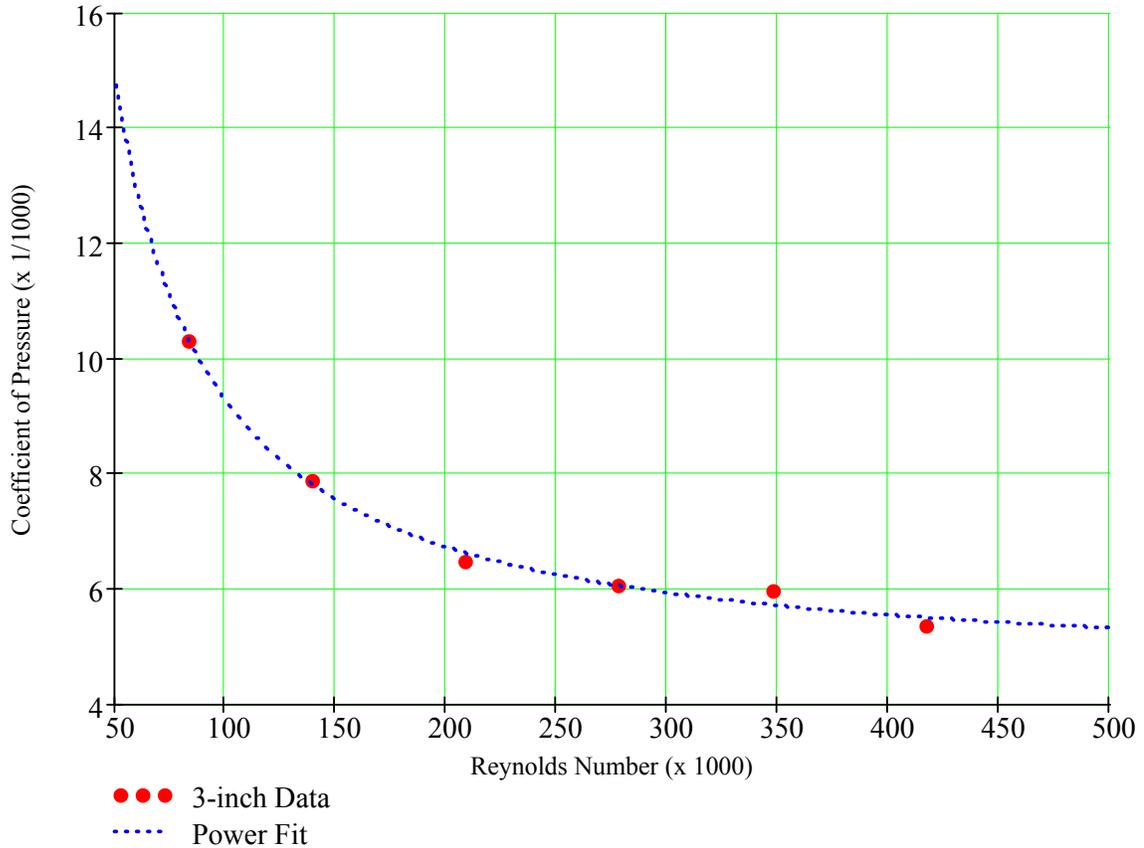


**Figure 6-1 Standard deviation of the pressure fluctuations on the pipe surface versus flow rate for a 3-inch diameter pipe**

Alternatively, the use of common non-dimensional parameters might help better quantify the relationship between pressure and flow rate by plotting the coefficient of pressure ( $C_p = \frac{2P_\sigma}{\rho U_0^2}$ ) vs. Reynolds number ( $Re_D = \frac{U_0 D}{\nu}$ ) shown in Figure 6-2. This is a common method of representing the data and though less intuitive, these non-dimensional parameters can provide the simple relationship given in Eq. (6.2).

$$C_p(Re_D) = 0.8138 Re_D^{-0.3895} \quad R^2 = 0.965 \quad (6.2)$$

<sup>f</sup>  $R^2$  values generally increase with the order of the polynomial regression; however, it is not necessarily the most realistic. If a higher order polynomial improves the  $R^2$  value by very little, the lower order fit is most likely realistic.



**Figure 6-2 Coefficient of pressure vs. Reynolds number for 3-inch diameter data**

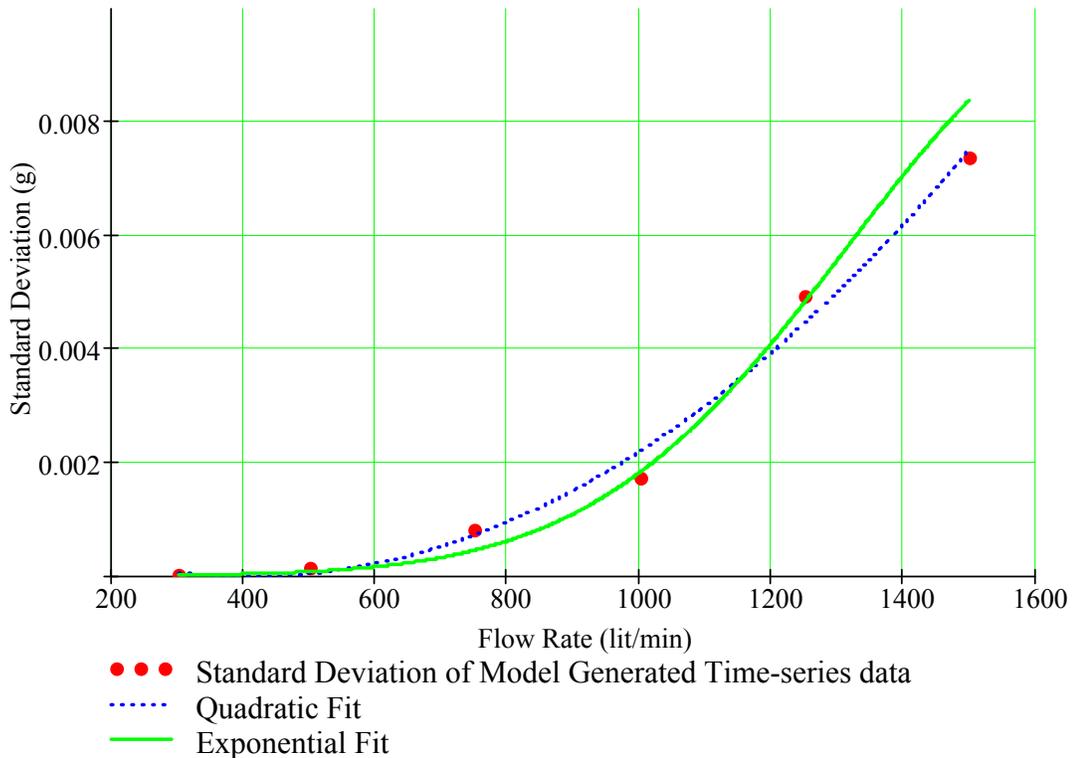
### 6.1.2 Acceleration

Goal 2 listed in Table 1-1 is to determine the contribution pressure fluctuations have on the dynamic response of a pipe. This is accomplished by quantifying the standard deviation of the acceleration signal versus flow rate. The maximum static deflection of the pipe at the flow rate of 1500 lit/min was approximately 100 nanometers. This is roughly 500 times smaller than the size of the viscous sublayer, suggesting that updating the flow model with the deformed pipe geometry at each time step would not change the pressure characteristics of the flow, and therefore would not provide any additional insight. A one-way coupling procedure, as described in Section 1.3 (*Case II*) was therefore used.

Once the deflections for the various flow rates were determined using 50 time-steps, the acceleration of the pipe at a point was determined by computing the second derivative of the position with respect to time. As mentioned in Section 5.4, the use of

Ridder's method on the spline fit of the position data most accurately calculates acceleration.

The standard deviations of three-inch pipe wall accelerations are plotted against flow rate in Figure 6-3. As alluded to in the previous section, a quadratic relationship similar to that for pressure exists for acceleration. This is confirmed by the curve fit given by Eq. (6.3), where  $A'$  is the standard deviation of the acceleration in g, and  $Q$  is the flow rate in lit/min. Though the fit is remarkable, the vertex for the quadratic fit occurs at approximately 400 lit/min, which is not physically possible. Perhaps a more realistic fit would be an exponential one used by Evans<sup>4</sup> in his statistical analysis given in Eq. (6.4). This is also given in Figure 6-3 and does not contain the unrealistic behavior in the bounds of flow rates, however, it still possesses an unrealistic constant at  $Q=0$ , as does the quadratic fit. It should be noted that these acceleration values are within the measurement range and resolution of many piezoelectric accelerometers and confirm the relationship presented by Evans.



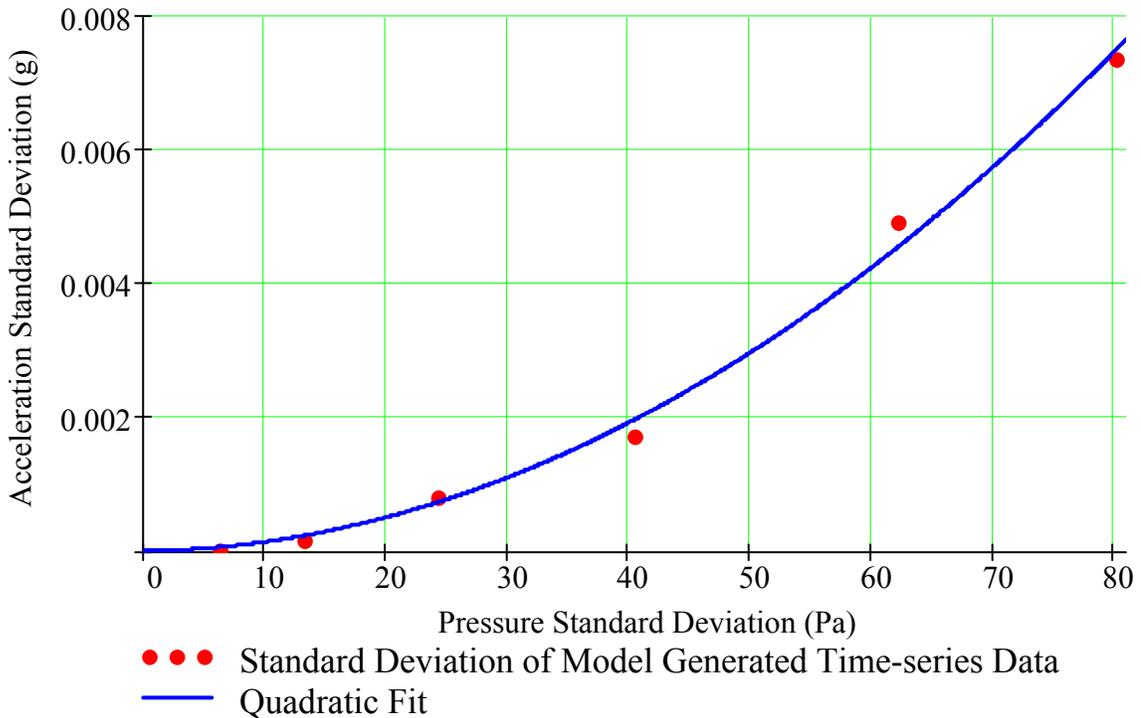
**Figure 6-3 Standard deviation of the acceleration on the pipe surface at a point versus flow rate for a 3-inch diameter pipe**

$$A'_{quadratic}(Q) = 6.323 \cdot 10^{-9} Q^2 - 5.171 \cdot 10^{-6} Q + 1.041 \cdot 10^{-3} \quad R^2 = 0.990 \quad (6.3)$$

$$A'_{exponential}(Q) = 5.437 \cdot 10^{-7} e^{1.149 \cdot 10^{-2} Q - 3.375 \cdot 10^{-6} Q^2} \quad R^2 = 0.974 \quad (6.4)$$

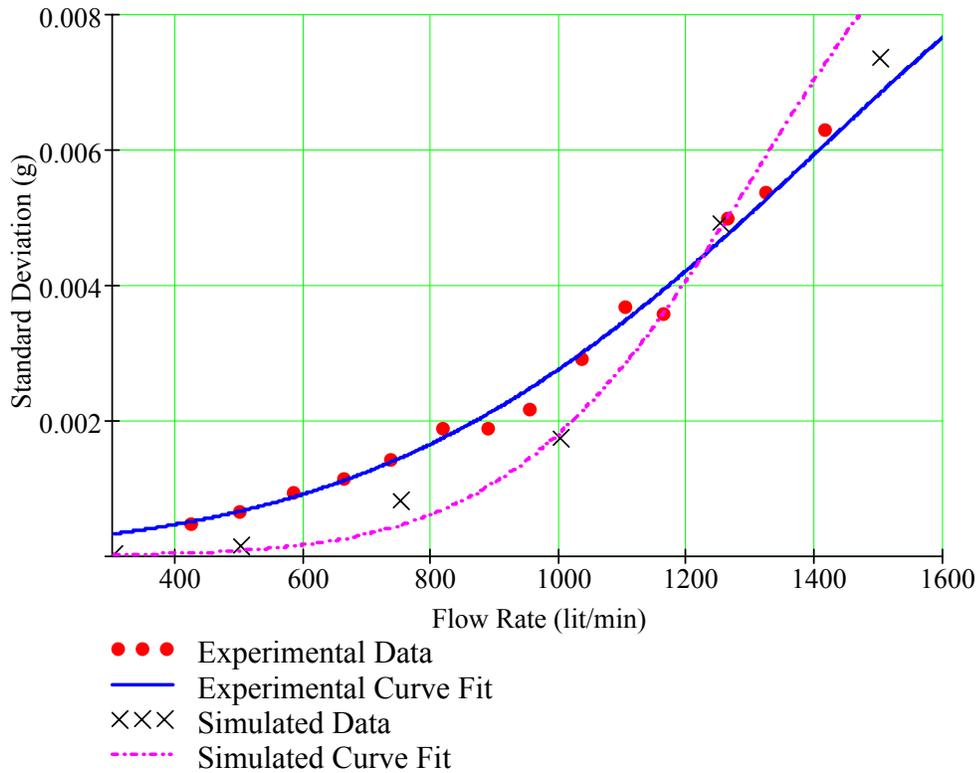
Now that both the accelerations and pressure fluctuations have been related to the flow rate, the final statistical analysis is to relate the accelerations and pressure fluctuations to each other (Figure 6-4). If the energy at the fluid-structure interface can be represented as the pressure fluctuations, then Figure 6-4 illustrates the relationship between the energy at the interface and the structure's response to that energy. The relation between the two standard deviations, given in Eq. (6.5), has a more realistic fit than the flow rate relations do. Since the fits given by Eqs. (6.1) and (6.3) contain an unrealistic constant for the case of  $Q = 0$ , they are restricted to flow rates between 300 and 1500 lit/min. However, the relation between the pressure and acceleration makes physical sense for a zero-pressure fluctuation value.

$$A'(P') = 1.133 \cdot 10^{-6} P'^2 + 3.040 \cdot 10^{-6} P' \quad R^2 = 0.995 \quad (6.5)$$

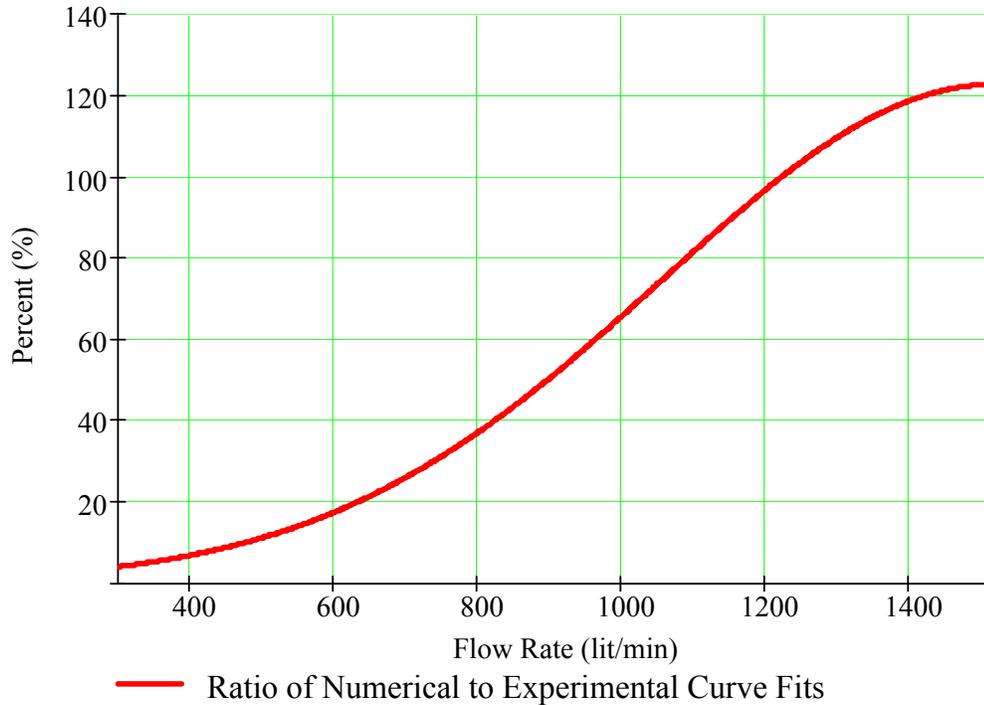


**Figure 6-4 Pressure vs. acceleration standard deviations**

The contribution of the pressure fluctuations to the overall vibration response recorded by Evans can be shown by comparing Figure 6-3 with the data from Evans shown in Figure 6-5. The contribution can be determined by computing the average of the ratio of the two sets of data. This procedure yields a value of approximately 50%. In other words, the pressure fluctuations contribute to about 50% of the overall observed acceleration. This value is not constant and the pressure fluctuations make a larger contribution at relatively higher flow rates as noted in Figure 6-6. The curve fit seems to over-predict the simulated data and under-predict the experimental data at high flow rates, which explains the contribution being over 100%. Realistically, the numerical pressure fluctuations should never be greater than the experimental. Though the comparison is remarkable and the magnitudes extremely close, the important idea is that the basic trends are similar for both the experimental and numerical results. The author makes no claim that this model is depicting the exact experimental scenario due to various parameters such as boundary conditions and material properties.



**Figure 6-5 Comparison of experimental data and numerically simulated data for 3-inch pipe**



**Figure 6-6 Ratio of numerical to experimental curve fits—as the flow rate increases, the contribution of the pressure fluctuations to the overall vibrational response also increases**

## 6.2 SURFACE ROUGHNESS CORRECTION

Surface roughness may account for a portion of the differences between the experimental and numerical simulations, since it is known that a rougher pipe will produce greater pressure fluctuations.<sup>g</sup> Accounting for surface roughness effects can be extremely complex and is typically modeled using RANS techniques at this time. However, simple scaling arguments can be used to quantify the standard deviations of the pressure fluctuations by relating the ratio of rough to smooth pressure fluctuations to the corresponding Moody friction factors. A similar relationship for non-dimensional heat transfer (Nusselt number) is given in Eq. (6.6), where Pr is the Prandtl number ( $Pr = \frac{\nu}{\alpha}$ ). Since the skin friction coefficient  $C_f$  is related to the Moody friction factor  $f$  by  $C_f = \frac{f}{4}$ ,

<sup>g</sup> All the results published here have not surface roughness effects and its addition has been included as an afterthought to this thesis. The inclusion of these effects were addressed as a result of the thesis defense and the results from this inclusion would take months to complete. Therefore, they are addressed here in theory.

the non-dimensional heat transfer coefficient  $Nu$  could be related to the non-dimensional pressure drop  $f$ .<sup>32</sup>

$$\frac{Nu_{rough}}{Nu_{smooth}} \cong \left( \frac{C_{f,rough}}{C_{f,smooth}} \right)^n \quad n = 0.68 \text{Pr}^{0.215} \quad (6.6)$$

Roughness effects on the overall pressure fluctuations can be handled by considering the shear stress  $\tau = \mu \frac{du}{dy} - \overline{\rho u'v'}$  where the second term represents the turbulent stress due to the fluctuating components. The turbulent shear stress should be scaled by the time-averaged velocity fluctuation products. The shear stress is also defined by the density, average velocity and friction factor. This now gives a relationship for the friction factor as a function of the fluctuating and mean velocities shown in Eqs. (6.7) to (6.9).

$$\tau_{fluctuating} \sim -\overline{\rho u'v'} \quad (6.7)$$

$$\frac{1}{2} \rho U_0^2 f \sim \overline{\rho u'v'} \quad (6.8)$$

$$f \sim \frac{\overline{u'v'}}{U_0^2} \quad (6.9)$$

Assuming that  $\overline{u'v'} \sim \overline{u'^2}$ , and that the pressure fluctuations are directly proportional to the velocity fluctuations through Bernoulli's equation (i.e.:  $P' \sim \overline{u'^2}$ ); combining with Eq. (6.9), yields a scaling relation for the pressure fluctuations  $P'$  shown in Eq. (6.10). From this, a correction factor for the pressure standard deviation can be determined by ratioing the rough pipe by the smooth pipe, which does resemble the heat transfer relation given in Eq. (6.11).<sup>h</sup>

$$P' \sim f U_0^2 \quad (6.10)$$

$$\frac{P'_{rough}}{P'_{smooth}} \cong \frac{f_{rough}}{f_{smooth}} \quad (6.11)$$

---

<sup>h</sup> This same relation can also be determined by a ratio of the average velocities, which turns out to be a ratio of the friction velocities for the smooth and wholly turbulent pipes. This leads to the ratio of Moody friction factors.

It should be mentioned that this is nothing more than a scaling argument relation, and experimental data should be used to determine the exact relationship between the fluctuating pressure and velocity components. Also, this merely provides a quantification for the standard deviation of the pressure fluctuations. Surface roughness will also affect the pressure field spatially; however, it may provide a more accurate representation of the actual pressure field.

According to Eqs. (6.5), (6.11) and (4.13), corrections to numerical data due to roughness effects are as follows:  $P'$  would increase by 2-6% and  $A'$  by 1-2%, with increasing flow rate. [Equation (4.13) with  $k_s = 0$  and  $k_s = .0015\text{mm}$  (surface roughness value for the drawn steel tubing used in the experimental setup) provides  $f_{smooth}$  and  $f_{rough}$ .

For  $Q = 300$  lit/min,  $\frac{f_{rough}}{f_{smooth}} = 1.02 \cong \frac{P'_{rough}}{P'_{smooth}}$ , and for  $Q = 1500$  lit/min,

$\frac{f_{rough}}{f_{smooth}} = 1.06 \cong \frac{P'_{rough}}{P'_{smooth}}$ , or in other words,  $P'$  increases by 2-6% with increasing flow rate.

Then, according to Eq. (6.5) and Figure 6-4, if  $P'$  increases by 2% at  $Q = 300$  lit/min,  $A'$  increases by 1%, and if  $P'$  increases by 6% at  $Q = 1500$  lit/min,  $A'$  increases by 2%.]

### 6.3 SIGNIFICANT FACTORS

The scope of this thesis is not only to determine the relationship between the dynamic response of the pipe and the flow rate, but also to determine the effect of changing pipe diameter and material. This section discusses these effects and statistically models the pressure and acceleration standard deviations.

#### 6.3.1 Material and Diameter

Changing pipe materials yields deflections that are scaled by the ratio of the elastic moduli of the pipe materials, assuming transient effects are negligible. It is true that the structural analysis is a function of other material properties such as Poisson's ratio and the density; however, our model neglects gravity and the deflections are so small that those properties have a minor effect. Poisson's ratio enters the equation predominately in the off-diagonal terms in the stiffness matrix ( $K$ ) and is typically a small contributor to the radial component of deflection. If the loading were larger, Poisson

effects would possibly need consideration. To verify this assumption, the highest loading magnitude (i.e. 1500 lit/min pressure fluctuations) was applied to identically meshed pipes with the two different material property sets. The static results for the deflection ( $U_{sum}$ )<sup>i</sup> at each node were exported and compared. The ratio of the  $U_{sum}$  for PVC to the  $U_{sum}$  of steel is almost identical to the ratio of the elastic moduli of steel to PVC. Table 6-1 shows less than a 1% difference between the simple ratio of moduli and the mean of the ratio of deflections. There could be a case for using the mode in the error calculation given the number of data points, which would decrease the error to less than a tenth of a percent. These relationships were expected and can be used as a further model verification.

**Table 6-1 Justification for scaling deflections for PVC calculations**

ANSYS Ratio Results (no. nodes = 33248)		Moduli Ratio	
<u>Parameter</u>	$\frac{U_{sum}(PVC)}{U_{sum}(Steel)}$	Steel Modulus ( $E_{steel}$ )	1.93E+11 Pa
Mean <sup>j</sup>	<b>71.99</b>	PVC Modulus ( $E_{PVC}$ )	2.70E+09 Pa
Standard Deviation	1.33	Ratio $\frac{E_{steel}}{E_{PVC}}$	<b>71.48</b>
Percent Diff from Mean			0.71%

The response from changing the material to PVC (or any material) will be computed by scaling the deflections for the steel by the ratio of moduli. This constant factors out of the calculation of the derivative since  $\frac{d^2}{dt^2} \left( \frac{E_{steel}}{E_{pvc}} U_{sum}(t) \right) = \frac{E_{steel}}{E_{pvc}} \frac{d^2}{dt^2} U_{sum}(t)$ . Subsequently, it can also be shown that the standard deviation of any data scaled by a factor is equivalent to the product of the constant and the original standard deviation as shown in Eqs. (6.12) and (6.13), where  $AY$  is the vector of data.<sup>k</sup> In other words, assuming transient effects are negligible, changing

<sup>i</sup>  $U_{sum}$ =square root of the sum of the squares for the three deflections UX, UY and UZ.

<sup>j</sup> The median = 71.94 and mode = 71.43

<sup>k</sup> In our case, the vector of data  $AY$  represents the acceleration of  $UY$

materials simply scales the standard deviation of the acceleration values by the ratio of moduli.

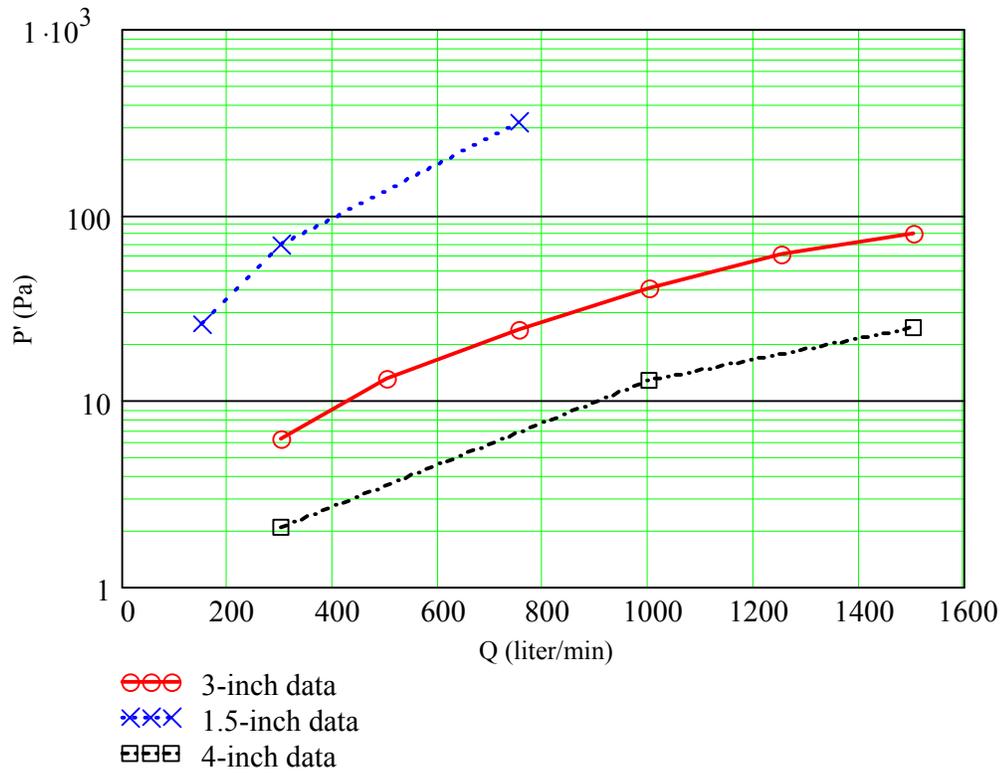
$$\frac{E_{steel}}{E_{pvc}} \overline{AY} = \frac{\frac{E_{steel}}{E_{pvc}} AY_1 + \dots + \frac{E_{steel}}{E_{pvc}} AY_n}{n} = \frac{E_{steel}}{E_{pvc}} \frac{AY_1 + \dots + AY_n}{n} = \frac{E_{steel}}{E_{pvc}} \overline{AY} \quad (6.12)$$

$$\sigma_{\frac{E_{steel}}{E_{pvc}} AY} = \sqrt{\sum_{i=1}^n \left( \frac{\frac{E_{steel}}{E_{pvc}} AY_i - \frac{E_{steel}}{E_{pvc}} \overline{AY}}{n} \right)^2} = \frac{E_{steel}}{E_{pvc}} \sqrt{\sum_{i=1}^n \left( \frac{AY_i - \overline{AY}}{n} \right)^2} = \frac{E_{steel}}{E_{pvc}} \sigma_{AY} \quad (6.13)$$

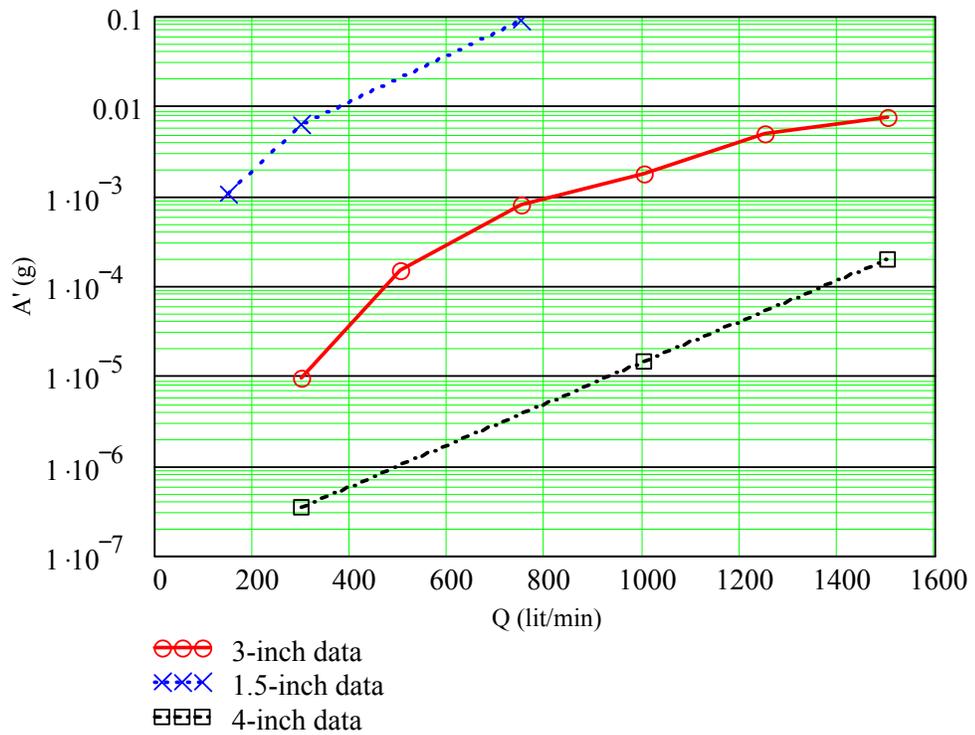
The various diameters were modeled in FLUENT and ANSYS following the exact same procedures as outlined in Chapters 4 and 5. Table 6-2 lists these results, and Figure 6-7 to Figure 6-9 give a graphical representation of the results. Disparities between the individual diameter values require semi-log plots to provide a clear depiction of the data.

**Table 6-2 List of results for various pipe diameters and flow rates**

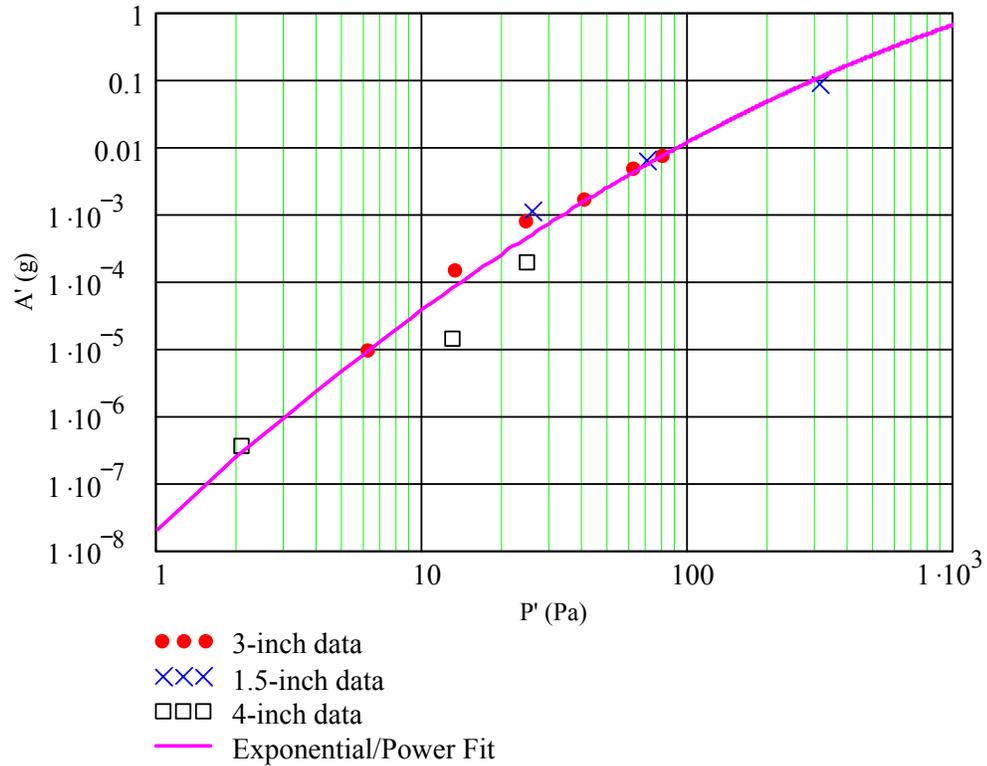
Diameter	Q (lit/min)	P' (Pa)	A' (g)
3-inch	300	6.19E+00	9.41E-06
	500	1.31E+01	1.48E-04
	750	2.42E+01	8.03E-04
	1000	4.04E+01	1.73E-03
	1250	6.21E+01	4.92E-03
	1500	8.01E+01	7.36E-03
1.5-inch	150	2.60E+01	1.07E-03
	300	6.97E+01	6.24E-03
	750	3.11E+02	8.70E-02
4-inch	300	2.09E+00	3.48E-07
	1000	1.29E+01	1.45E-05
	1500	2.48E+01	1.98E-04



**Figure 6-7 Pressure fluctuation standard deviation vs. flow rate**



**Figure 6-8 Acceleration standard deviation vs. flow rate**



**Figure 6-9 All pressure vs. acceleration standard deviations**

Figure 6-9 represents the correlation between the standard deviations of pressure and acceleration, and the regression equation for the data is given in Eq. (6.14). Although data for the 3-inch diameter pipe behave quadratically, data for all the diameters do not. Rather, an exponential/power fit regression seems more realistic when all data points are considered. This relation is not guaranteed for all pipe diameters, but will supply a good approximation of the acceleration response given the standard deviation of the pressure.

$$A'(P') = 2.097 \cdot 10^{-8} P'^{-0.1639 \ln(P') + 3.6306} \quad R^2 = 0.962 \quad (6.14)$$

### 6.3.2 Statistical Analysis

From the information given in Table 6-2, a statistical analysis can provide additional insight into the effect of pipe diameter ( $D$ ) and flow rate ( $Q$ ) on the overall response of  $P'$  and  $A'$ . Since the data contain continuous design variables, they are best analyzed using a response surface in a statistical software package such as NCSS.<sup>®</sup> A full quadratic regression model was fit for the pressure and acceleration. Interestingly, the

best fit for the model first required a logarithmic transformation, which is typically done for standard deviation responses.<sup>43</sup> In fact, the logarithmic transform on the pressure and acceleration standard deviations increased the correlation coefficient for the model by 10% and 30% respectively. This section first discusses the statistical model of the pressure fluctuations by giving the ANOVA table, the normal probability plot of the residuals, and a contour plot of the statistical regression. Each item is followed by a brief explanation of the significant details necessary to understand the analysis. The statistical model of the acceleration data is discussed in like manner.

### 6.3.2.1 Pressure

The analysis of the multivariate regression of the transformed pressure is given in Table 6-3, with significant values to the interpretation of the analysis given in bold type. First, the hierarchical model summary lists the number of removed and remaining terms, the  $R^2$  cutoff value, and the  $R^2$  and adjusted  $R^2$  values of the final model. The removed terms make such an insignificant contribution to the response that they can be eliminated. The  $R^2$  value represents the amount of data described by the model and is a good indication of how well a model fits the data, whereas the adjusted  $R^2$  value accounts for the sample size. Second, the sequential ANOVA table identifies the significance and contribution of the linear, quadratic and interaction terms to the overall model by providing a p-value<sup>1</sup> and incremental  $R^2$  value. In this case, 94.6% of the data could be described using a simple linear model. The addition of the quadratic increases this by 4.6% , and the interaction term makes an insignificant contribution. Next, the ANOVA section presents the contribution each individual factor (pipe diameter  $D$  and flow rate  $Q$ ) has on the response. The small p-values (less than 0.001) in this case indicate that both factors are important to the model and could significantly describe most of the data given their terms alone. Finally, the estimation section breaks up the model into the individual components and gives a p-value for each term. In this case, little improvement to the fit is obtained by including the  $D^2$  and interaction term ( $D \cdot Q$ ).

---

<sup>1</sup> A p-value indicates the statistical strength or evidence a term has of being a significant contributor to the overall response (i.e. evidence against the null hypothesis). The number represents the chance that a more one would **not** find the same term significant if another random trial collected a data point. Therefore, the lower the p-value, the stronger the evidence.

A p-value is an accurate method for predicting significance if the error term is normally distributed in the model. A normal probability plot of the residuals<sup>m</sup> finds the normal score of each residual and compares it with the residual itself. The data are plotted against a theoretical normal distribution in such a way that the points should form an approximate straight line banded by a 95% confidence interval. Departures from this straight line indicate departures from normality and less confidence in the p-values. Figure 6-10 confirms the fact that the p-values accurately provide evidence against the null hypothesis. For more information on the ANOVA and Residual plots, see Keuhl.<sup>43</sup>

**Table 6-3      Multivariate regression:  $\ln(P')$  versus  $D$ ,  $Q$ <sup>n</sup>**

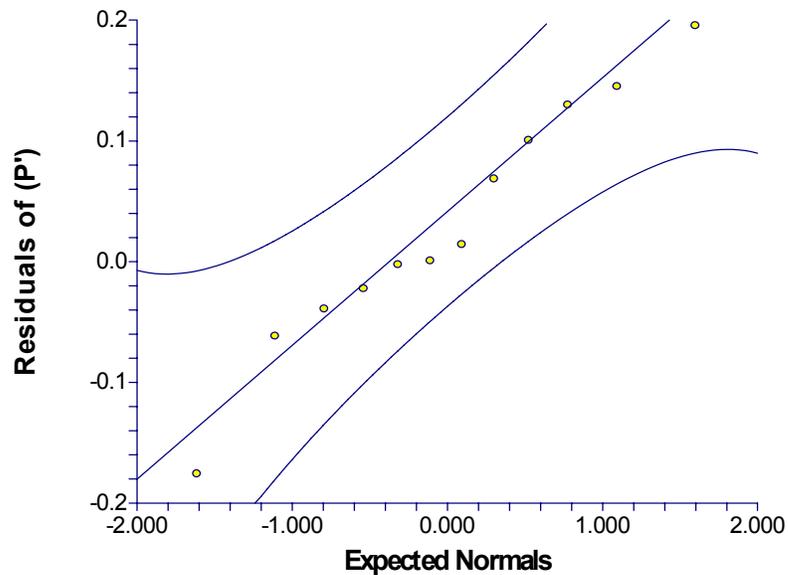
<u>Hierarchical Model Summary Section</u>						
Number of Terms Removed						0
<b>Number of Terms Remaining</b>						<b>5</b>
R-Squared Cutoff Value						0.001
<b>R-Squared of Final Model</b>						<b>0.995</b>
<b>R-Squared adjusted of Final Model</b>						<b>0.981</b>
<u>Sequential ANOVA Section</u>						
<u>Source</u>	<u>df</u>	<u>Sequential Sum-Squares</u>	<u>Mean Square</u>	<u>F-Ratio</u>	<u>P-value</u>	<u>Incremental R-Squared</u>
<b>Regression</b>	5	18.62636	3.725272	248.66	<b>0.000001</b>	<b>0.995197</b>
<b>Linear</b>	2	17.71203	8.856015	591.13	<b>0</b>	<b>0.946345</b>
<b>Quadratic</b>	2	0.869141	0.434571	29.01	<b>0.000823</b>	<b>0.046438</b>
Interaction	1	4.52E-02	4.52E-02	3.02	0.1331	0.002415
Total Error	6	8.99E-02	1.50E-02			0.004803
<u>ANOVA Section</u>						
<u>Factor</u>	<u>df</u>	<u>Last Sum-Squares</u>	<u>Mean Square</u>	<u>F-Ratio</u>	<u>P-value</u>	<u>Term R-Squared</u>
<b><math>D</math></b>	3	15.35943	5.119809	341.74	<b>0</b>	<b>0.820647</b>
<b><math>Q</math></b>	3	11.06353	3.687843	246.16	<b>0.000001</b>	<b>0.591119</b>
Total Error	6	8.99E-02	1.50E-02			0.004803

continued on next page...

<sup>m</sup> A residual is the difference between the actual and predicted values. It is a representation of the error in a model.

<sup>n</sup> The analysis was done using uncoded units.

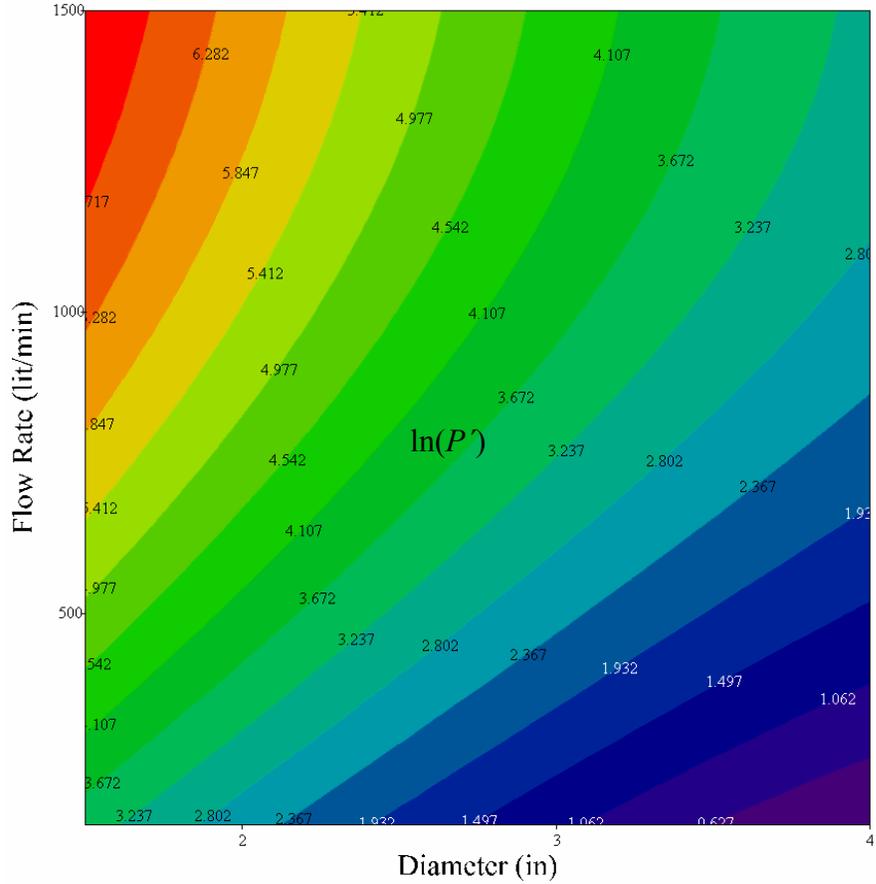
Estimation Section						
Parameter	df	Regression Coefficient	Standard Error	T-Ratio	P-value	Last R-Squared
Intercept	1	5.964302				
$D$	1	-2.50213	0.284131	-8.81	0.000119	0.062075
$Q$	1	5.36E-03	4.44E-04	12.09	0.000019	0.117026
$D^2$	1	0.225077	5.61E-02	4.01	0.007021	0.012885
$Q^2$	1	-1.373E-06	2.39E-07	-5.74	0.001211	0.026408
$D \cdot Q$	1	-2.315E-04	1.33E-04	-1.74	0.1331	0.002415



**Figure 6-10** Normal probability plot of residuals for  $\ln(P')$

Figure 6-11 is a contour plot of the multivariate regression given in Eq. (6.15). Flow rate and diameter are axes values and the natural log of the pressure standard deviations are the contours. Figure 6-11 is significant in that over the range of flow rates and diameters investigated the pressure fluctuations can be predicted with 98% confidence. Furthermore, for a given diameter pipe, the standard deviation of the pressure fluctuations could be experimentally measured and the mass flow rate inside the pipe could be predicted by manipulating Eq. (6.15).

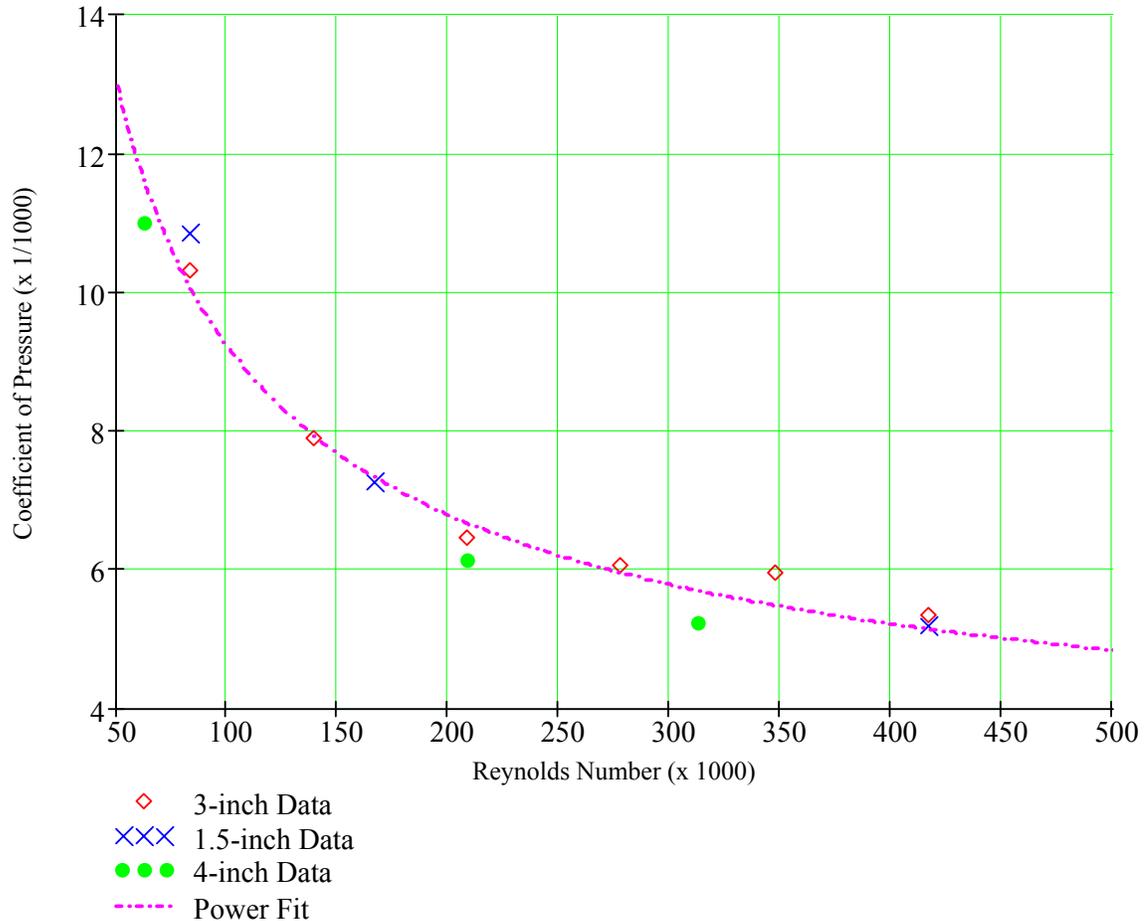
$$\ln(P') = 5.964 - 2.502D + 5.36 \cdot 10^{-3}Q + 0.225D^2 - 1.373 \cdot 10^{-6}Q^2 - 2.315 \cdot 10^{-4}D \cdot Q \quad (6.15)$$



**Figure 6-11 Contour of  $\ln(P)$  model—as the diameter decreases and the flow rate increases, the pressure fluctuations also increase**

Alternatively, the same non-dimensional parameters of coefficient of pressure and Reynolds number used in Figure 6-2 can be used as shown in Figure 6-12. These non-dimensional parameters can provide an easier relationship to model given in Eq. (6.16).

$$C_p(\text{Re}_D) = 1.186 \text{Re}_D^{-0.4219} \quad R^2 = 0.953 \quad (6.16)$$



**Figure 6-12 Coefficient of pressure vs. Reynolds number**

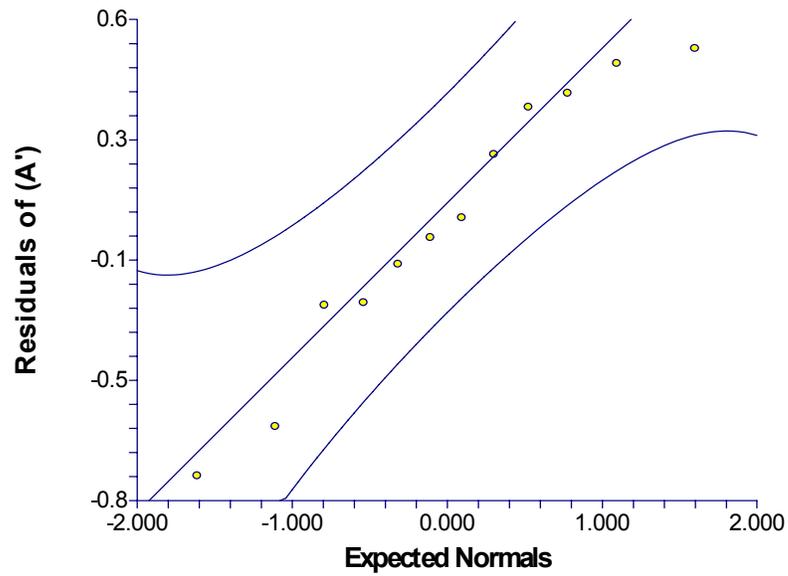
### 6.3.2.2 Acceleration

The statistical analysis for the acceleration response leads to similar regression results noted in Section 6.1.2. From Table 6-4, the general prediction equation is given in Eq. (6.17). Figure 6-13 gives the normal probability plot of the residuals, and Figure 6-14 is a contour of the regression given in Eq. (6.17). Figure 6-14 is a fundamental result in that it relates flow rate acceleration and pipe diameter. Therefore, given pipe diameter, the acceleration on the surface of a pipe could be experimentally measured and used to predict the flow rate inside the pipe with 98% confidence. This provides significant insight to the development of a non-intrusive, accelerometer based flow sensor.

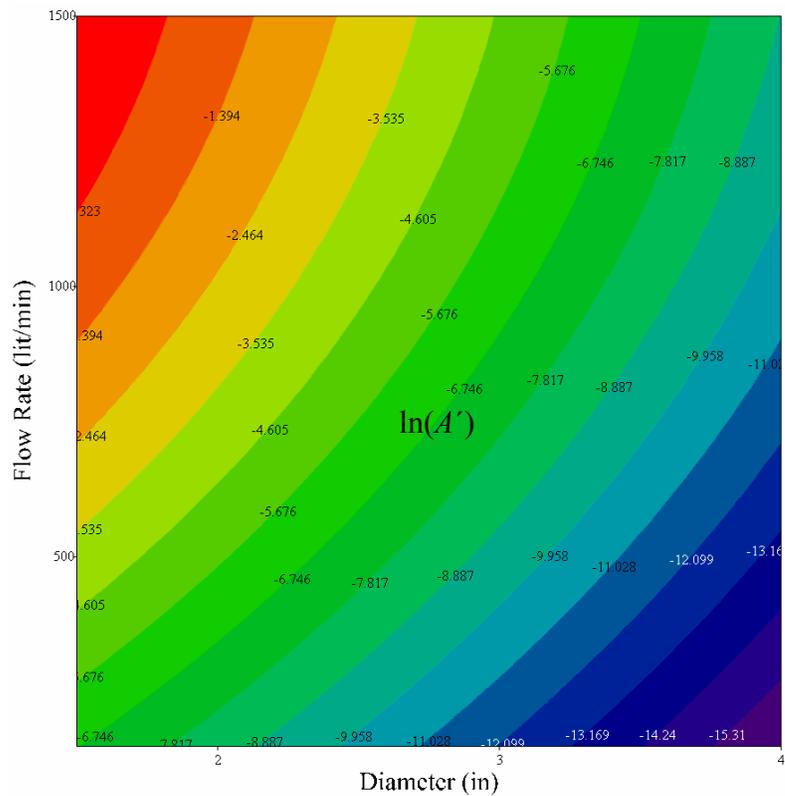
**Table 6-4 Response surface regression:  $\ln(A')$  versus  $D, Q$**

<u>Hierarchical Model Summary Section</u>						
Number of Terms Removed						1
<b>Number of Terms Remaining</b>						<b>4</b>
R-Squared Cutoff Value						0.001
<b>R-Squared of Final Model</b>						<b>0.986</b>
<b>R-Squared adjusted of Final Model</b>						<b>0.975</b>
<u>Sequential ANOVA Section</u>						
<u>Source</u>	<u>df</u>	<u>Sequential Sum-Squares</u>	<u>Mean Square</u>	<u>F-Ratio</u>	<u>P-value</u>	<u>Incremental R-Squared</u>
<b>Regression</b>	4	129.2513	32.31281	126.56	0.000001	0.986362
<b>Linear</b>	2	125.9549	62.97746	246.67	0	0.961206
Quadratic	2	3.296323	1.648161	6.46	0.025764	0.025155
Interaction	7	1.78716	0.255309			0.013638
Total Error	4	129.2513	32.31281	126.56	0.000001	0.986362
<u>ANOVA Section</u>						
<u>Factor</u>	<u>df</u>	<u>Last Sum-Squares</u>	<u>Mean Square</u>	<u>F-Ratio</u>	<u>P-value</u>	<u>Term R-Squared</u>
<b>D</b>	2	114.7602	57.38012	224.75	0	0.875776
<b>Q</b>	2	59.45251	29.72626	116.43	0.000004	0.453703
Total Error	7	1.78716	0.255309			0.013638
<u>Estimation Section</u>						
<u>Parameter</u>	<u>df</u>	<u>Regression Coefficient</u>	<u>Standard Error</u>	<u>T-Ratio</u>	<u>P-value</u>	<u>Last R-Squared</u>
Intercept	1	-3.832193				
$D$	1	-2.507952	1.14231	-2.2	0.064155	0.009392
$Q$	1	0.0098915	1.53E-03	6.46	0.000348	0.081206
$D^2$	1	-0.2463281	0.2061	-1.2	0.27093	0.002783
$Q^2$	1	-2.6408E-06	8.77E-07	-3.01	0.019591	0.017682

$$\ln(A') = -3.832 - 2.508D + 9.892 \cdot 10^{-3}Q - 0.246D^2 - 2.641 \cdot 10^{-6}Q^2 \quad (6.17)$$



**Figure 6-13** Normal Probability plot of residuals for (A')



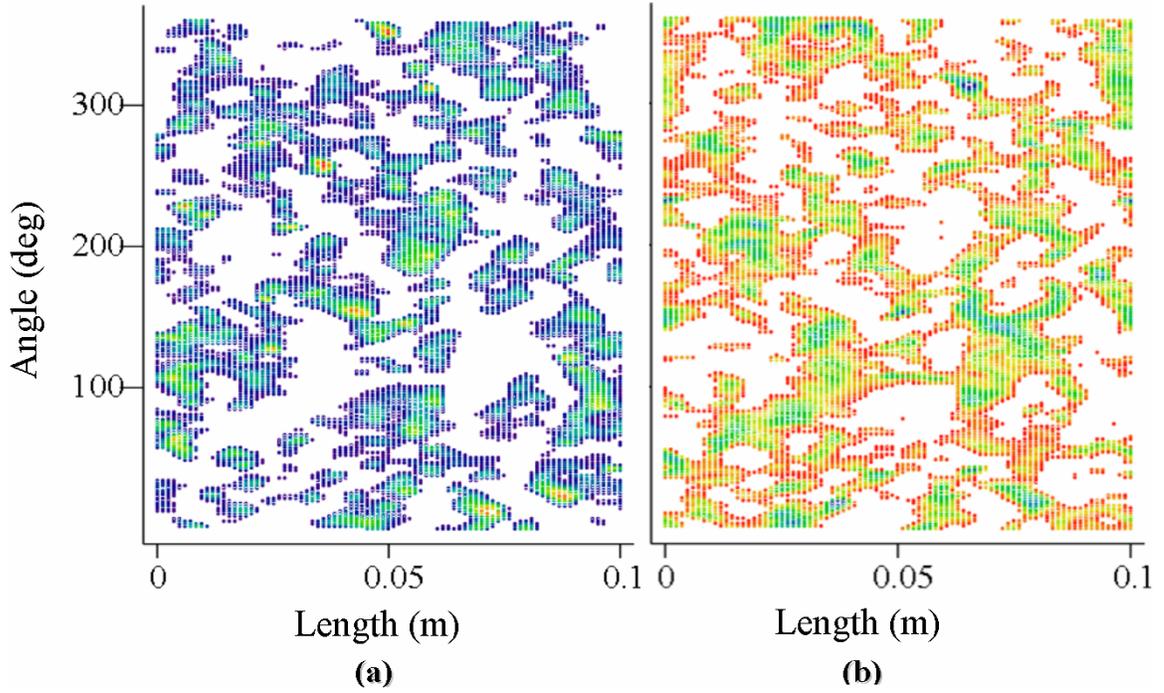
**Figure 6-14** Contour of  $\ln(A')$  model—as the diameter decreases and the flow rate increases, the accelerations also increase

## 6.4 NORMALITY OF PRESSURE DISTRIBUTIONS

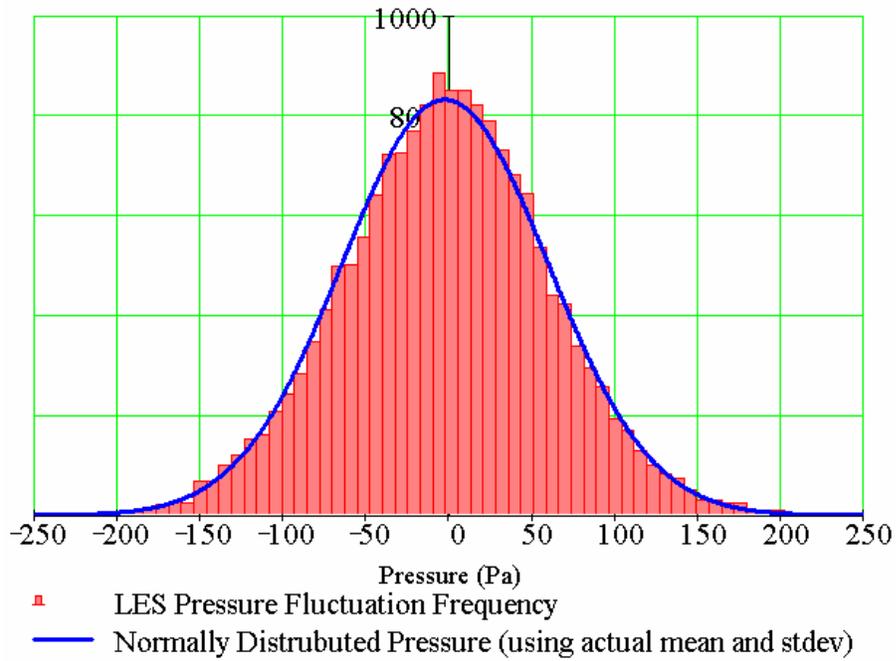
The main goal of the CFD simulation is to obtain the source of energy transfer from the fluid to the structure, which, as stated in Section 2.1, are the wall pressure fluctuations. Avenues to obtain these pressures need not be limited to the CFD approach used here but could include other approximations or experimental methods. One potential approximation method could be based on the normality of the wall pressure fluctuations for a turbulent flow as determined from this study. If the pressure fluctuations could be statistically characterized, computationally expensive techniques to produce the pressure field could be circumvented since all other details obtained from the flow simulation such as the instantaneous velocities are unnecessary. Statistically characterizing the pressure fluctuations is beyond the scope of this paper, yet the accomplishment of such could provide a significant contribution. The remainder of this section briefly discusses a platform for such a method.

At each time-step in the flow solution, the pressure fluctuations were exported for use in the structural model. The positive and negative pressure fluctuations on the surface of the pipe for one time-step are illustrated in Figure 6-15. This provides insight to the spatial distributions along the pipe wall. The vertical axis is the angle measurement of the pipe (i.e. the circumferential length) and the horizontal axis represents the length of the CFD model. In reference to these pressures, it is noted that the maximum peaks and valleys leave “footprints” in the flow most likely caused by turbulent eddies. These footprints are a characteristic of turbulent pressure fields and should always be seen.<sup>44</sup>

Although the pressure fields shown in Figure 6-15 may appear random, it has been well documented that turbulence is not a random phenomena.<sup>26,27,32</sup> Recall the pressure data discussed in Section 6.1.1. A histogram of the pressure fluctuations on the pipe wall produced a near Gaussian distribution (Figure 6-16). Therefore, the seeming chaos of turbulence actually possesses a semblance of order. Knowing that the pressure fluctuations always follow a Gaussian distribution concedes the possibility of statistically characterizing the pressure fields. Such a method could then circumvent the expensive CFD techniques.



**Figure 6-15 Pressure field on the pipe surface, a) positive pressure field, b) negative pressure field**



**Figure 6-16 Distribution of the turbulent pressure fluctuations at each interior point of a pipe wall**

# 7 CONCLUSION

## 7.1 SUMMARY

This thesis developed and presented a method for investigating FSI problems based on LES flow. The method involved two major milestones: the development of the fluid and the structural models.

### 7.1.1 Fluid Model

The fluid model was developed in FLUENT using an LES approach. The LES approach computes the instantaneous pressure and velocity values required to predict the pipe vibration caused by turbulent flow. Most commercial FSI software packages are RANS based, which do not compute the instantaneous values. The fluid model captures 97% of the total energy with a 2 to 6% error in the velocity profiles at the low and high Reynolds number respectively. The pressure gradients are 9% below the theoretical calculation, and the fluctuating pressure fields visually behave as expected with high pressure footprints being followed by low pressure ones.

### 7.1.2 Structure

The purpose of the structural model is to compute the pipe accelerations based on the pressure fluctuations from the flow model. The structural modeling was done in the commercial package ANSYS. The structural model consists of a 1.1m schedule 40 pipe with three different inside diameters. Each pipe is modeled by thin shell elements with approximately 30,000 elements. The structural model contains 2% error from a theoretical static solution scenario. A Mathcad program was written to map and transform the pressure data computed in the flow model to the structural model. Fifty time steps

were used to determine the structural displacement as a function of time. Ridder's method was used to determine the dynamic response of the spline fit displacement values.

### 7.1.3 Results

Primary focus for this study is on six flow rates between the range of 300 and 1500 lit/min for a 3-inch schedule 40 steel pipe. A definite relationship between the acceleration of the pipe (pipe vibration) and flow rate exists, and turbulent flow contributes to pipe vibration to an extent that can be measured with an accelerometer. The results show that as the flow rate increases, both the pressure and acceleration standard deviations increase at either a quadratic or an exponential rate. Statistical curve fits result in  $R^2$  values of 0.997 and 0.99 for the quadratic models for pressure standard deviation vs. flow rate, and acceleration standard deviation vs. flow rate, respectively. Non-dimensional parameters (coefficient of pressure and Reynolds number) universalize the pressure results and also provide an excellent fit ( $R^2=0.965$ ). From experimental data, a more realistic relationship for the acceleration in the pipe vs. flow rate may be exponential, in which case the statistical curve fit explains 97.4% of the error in the model. By comparison with other experimental results, the contribution of the turbulent flow to the pipe vibration at low flow rates is a rather small component (3%) of the overall vibration. However, as the flow rate increases, the turbulent flow induced vibration becomes a more significant component ( $\approx 100\%$ ) of the total response. These last two concepts open possible avenues for the development of a non-intrusive mass flow sensor.

Material effects were determined to be the ratio of moduli if transient effects were negligible; however, additional studies need to be completed to investigate the validity of this assumption. The deflections were so small that it was felt that the pressure force would be a forcing function for a series of static solutions.

In addition to the 3-inch diameter, two additional diameters and various flow rates between the range of 150 and 1500 lit/min were investigated to determine the effect diameter has on the pressure and acceleration standard deviations. The results showed that as the flow rate increases and the diameter decreases, the pressure fluctuations and subsequently the acceleration standard deviations also increase. Statistical models fit to

the data confirm this finding. Logarithmic transformations of the standard deviation terms increase the multivariate regressions for pressure and acceleration by 10% and 30% respectively, which supports the exponential relations noted by the experimental data. The multivariate pressure and acceleration regressions were fit with  $R^2$  values of 0.981 and 0.975 respectively. As with the 3-inch results, non-dimensional terms could also be used as an alternative method of describing the pressure data. Instead of a multivariate regression, the pressure coefficient and Reynolds number transformations of the data collapse the data onto a single power fit line with an  $R^2=0.953$ .

## 7.2 RECOMMENDATIONS

Basic conclusions from this research presented a technique for determining the vibrational response of a structure to turbulent fluid flow. It is therefore recommended that the incorporation of LES into existing commercial FSI software packages would provide the greatest contribution. The difficulty may arise in mapping the fluid model to the structure since current FSI packages require an interface node between the fluid and structure. It has been demonstrated here that LES requires many more nodes at the fluid structure interface than is required by a structure. Therefore, an intelligent way of coupling the LES and structural model will need to be accomplished as a part of its implementation.

Future numerical studies of fully developed turbulent pipe flow may be conducted using an alternate turbulence model called Detached Eddy Simulation (DES), which will soon be incorporated into FLUENT. DES simulates instantaneous flow using a hybrid of RANS and LES and is growing in popularity among the aerodynamics fields for modeling flow over an airfoil. In this method, RANS is used for the near wall treatment in the log-law and viscous sublayer, and LES is used outside those regions. Since LES is ineffective in modeling the sublayer anyway (refer to Section 4.3), DES could be a viable solution to modeling fully developed pipe flow and may yield a shorter solution time than LES alone.

Since pipe wall pressure fluctuation fields in fully developed turbulent flow tend to follow statistical patterns, a great contribution to FSI could be made by statistically characterizing the pressure fluctuation field in terms of both magnitude and visual

pattern. FSI problems of this nature could then circumvent the expense of developing a fluid model.

### **7.3 PUBLICATIONS**

Two publications of work done through this thesis have been submitted to separate conferences. The latest one may be found in Section 9.5, and serves as an extended abstract to this thesis. The first publication is a four-page document submitted to the 2003 FLUENT Users Group Meeting and focuses on the main aspects of the application of FLUENT in the development of the LES model.<sup>45</sup> In November 2003, a similar paper will be submitted to the general proceedings of the 2003 ASME International Mechanical Engineering Congress & Exposition.<sup>46</sup> The paper was written to the general engineering audience. A third publication is intended to combine this research with the experimental counterpart and is planned to be submitted to the *Journal of the Acoustical Society of America*.

## 8 REFERENCES

1. Koobus, B., Farhat, C., Tran, H., "Computation of unsteady viscous flows around moving bodies using the k- $\epsilon$  turbulence model on unstructured dynamic grids," *Computer Methods in Applied Mechanics and Engineering*, 190 (n11-12), December 2000, p 1441-1466.
2. Weaver, D., Ziada, S., Au-Yang, M., Chen, S., Paidoussis, M., Pettigrew, M., "Flow-induced vibrations in power and process plant components - progress and prospects," *Journal of Pressure Vessel Technology, Transactions of the ASME*, 123 (n3), August 2000, p 339-348.
3. Saito, N., Miyano, H., Furukawa, S., "Study on vibration response of pipes induced by internal flow," *1990 Pressure Vessels and Piping Conference*, Nashville, TN, ASME Pressure Vessels Piping Div Publ PVP, June 17-21 1990, p 233-238.
4. Evans, R., "Two-phase mass flow measurement using noise analysis," *45<sup>th</sup> International Instrumentation Symposium*, Albuquerque, NM, ISA - International Society for Measurement and Controls, May 1999.
5. Durant, C., Robert, G., Filippi, P., Mattei, P., "Vibroacoustic response of a thin cylindrical shell excited by a turbulent internal flow: comparison between numerical prediction," *Journal of Sound and Vibration*, 229 (n5), 2000, p 1115-1155.
6. Durant, C., Robert, G. "Vibroacoustic response of a pipe excited by a turbulent internal flow," *Flow, Turbulence and Combustion*, 61 (n1), 1998, p 55-69.
7. Brevart B., Fuller, C., "Effect of an internal flow on the distribution of vibration energy in an infinite fluid filled thin cylindrical elastic shell," *Journal of the Sound and Vibration*, 167, October 8, 1993, p 149-163.
8. Kim, Y.K., Kim, Y.H., "Three accelerometer method for the measurement of flow rate in pipe," *Journal of the Acoustical Society of America*, 100 (n2 pt 1), August 1996, p 717-726.

9. Murakami, S., Mochida, A., Sakamoto, S., "CFD analysis of wind-structure interaction for oscillating square cylinders," *Journal of Wind Engineering and Industrial Aerodynamics*, 72 (n 1-3), Nov-Dec, 1997, p 33-46.
10. Munson, B., Young, D., Okiishi, T., *Fundamentals of Fluid Mechanics 3<sup>rd</sup> Edition*, John Wiley & Sons, Inc. New York, NY, 1998.
11. Panton, R., *Fundamentals of Fluid Mechanics 2<sup>nd</sup> Edition*, John Wiley & Sons, Inc. New York, NY, 1996.
12. Tennekes, H., Lumley, J., *A First Course in Turbulence*, The MIT Press, Cambridge, MA, 1972.
13. Cuschieri, J., Leyrat, G., "Vibration transmission through a fluid loaded pipe shell with internal non zero flow velocity," *Winter Annual Meeting of the American Society of Mechanical Engineers*, Dallas, TX, ASME Pressure Vessels Piping Division Publication PVP, Nov 25-30 1990, 202, p 31-35.
14. Gorman, D., Liu, M., Horachek, J. "The effect of annular two-phase flow upon the natural frequencies of a pipe," *Engineering Structures*, 20 (n 8), August 1998, p 726-731.
15. Hibiki, T., Ishii, M., "Effect of flow-induced vibration on local flow parameters of two-phase flow," *Nuclear Engineering and Design*, 185, 1998, p 113-125.
16. Evans, R., Wilkins, C., Goodrich, L., Blotter, J, *Method and apparatus for measuring the mass flow rate of a fluid*, United States Patent 6,412,352, July 2, 2002.
17. Ortega, H., "Technical report. Computer simulation helps predict cerebral aneurysms," *Journal of Medical Engineering & Technology*, 22 (n 4), Jul-Aug, 1998, p 179-181.
18. Koobus, B., Farhat, C., Tran, H., "Computation of unsteady viscous flows around moving bodies using the k- $\epsilon$  turbulence model on unstructured dynamic grids," *Computer Methods in Applied Mechanics and Engineering*, 190 (n 11-12), December 2000, p 1441-1466.
19. Xiong, G., Huang, P., Saunders, C., Heink, P., "CFD simulation of laser printhead - a case study on noise reduction," *Transactions of the Aeronautical and Astronautical Society of the Republic of China*, 33, March 2001, p 1-6.

20. Eggels, J. Direct and Large Eddy Simulation of Turbulent Flow in a Cylindrical Pipe Geometry, Delft University Press, Delft, Netherlands, 1994.
21. Davis, F., Hassan, Y., "Two-dimensional finite element method large eddy simulation for application to turbulent steam generator flow," *Nuclear Technology*, 106, April 1994, p 83-99.
22. Patankar, S., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Pub. Corp, New York, NY, 1980.
23. Galperin, B., Orszag, S. *Large Eddy Simulation of Complex Engineering and Geophysical Flows*, Cambridge University Press, New York, NY, 1993.
24. Wilcox, D., *Turbulence modeling for CFD*, DCW Industries, La Cañada, CA, 2000.
25. FLUENT instructional video, *Advanced Turbulence Modeling*, Corporate Training, Lebanon, NH, 2000.
26. Pope, S., *Turbulent Flows*, Cambridge University Press, New York, NY, 2000.
27. FLUENT v5.5 *Users Manual*, Ch 9.
28. Rumsey, C., Vatsa, V., "A Comparison of the Predictive Capabilities of Several Turbulence Models Using Upwind and Central-difference Computer Codes." *AIAA-Fluid Mechanics Division*, 1993, p. 1-16.
29. Rogallo, R., Moin, P., "Numerical Simulation of Turbulent Flows." *Annual Review of Fluid Mechanics* 16, 1984, p 99-137.
30. Lesieur, M., Metais, O., "New Trends in Large-Eddy Simulations of Turbulence." *Annual Review of Fluid Mechanics*, 28, 1996, p 45-82.
31. FLUENT v5.5 *Users Manual*, Ch 9, pp. 40-60.
32. Kays, W., Crawford, M. *Convective Heat and Mass Transfer 3<sup>rd</sup> edition*, McGraw Hill, New York, New York, 1993.
33. Goldin, G, FLUENT, June 2002.

34. Ligrani, P. M. "Structure of turbulent boundary layers." Aerodynamics and Compressible Flows, Encyclopedia of Fluid Mechanics (Editor: N. Chermisinoff), Vol. 8, pp. 111-189, Gulf Publishing, October 1988.
35. Chang, P. A. "Relationships between wall pressure and velocity field sources." University of Maryland, 1998.
36. Balling, R. *Fundamental Methods of Structural Analysis*, BYU professor's book to be published soon.
37. Hibbeler, R. C. *Mechanics of Materials 3<sup>rd</sup> edition*, Prentice Hall, Upper Saddle River, New Jersey, 1997.
38. Cook, R.D., Malkus, D.S., Plesha, M.E., *Concepts and Applications of Finite Element Analysis 3<sup>rd</sup> edition*, Wiley, New York, New York, 1989.
39. ANSYS website, [www.ansys.com](http://www.ansys.com), 06/03.
40. ANSYS v6.1. *Users Manual*.
41. Roark, R., Young, W., *Formulas for Stress and Strain 5<sup>th</sup> edition*, McGraw-Hill, New York, New York, 1975.
42. Mathcad v11, *Users Manual*.
43. Kuehl, R. O. *Design of experiments : statistical principles of research design and analysis*, Duxbury, Pacific Grove, CA, 2000.
44. Snarski, S.R. "Relation Between the Fluctuating Wall Pressure and the Turbulent Structure of a Boundary Layer on a Cylinder in Axial Flow", NUWC-NPT Technical Report 10,223, Naval Undersea Warfare Center Detachment, New London, CT, August 1993.
45. Pittard, M., Blotter, J. "Fluid structure interaction using LES models developed in FLUENT," *2003 FLUENT USERS GROUP MEETING*, Manchester, NH, May 2003.
46. Pittard, M., Blotter, J., "Numerical Modeling of LES based turbulent flow induced vibration," *2003 ASME IMECE*, Washington, D.C., November 2003.

## 9 APPENDIX

### 9.1 ISOTROPIC TURBULENCE FORTRAN CODE

This is the code generously given to the author by Graham Goldin, a FLUENT Engineer.<sup>33</sup> It is used to initialize a flow with isotropic turbulence conditions as stated in Section 4.2.3. This code is used with permission and any use of this should be done after contacting Graham at [gmg@fluent.com](mailto:gmg@fluent.com).

```

      program ranfld
c     implicit none

      integer
      .  nx, ny, nz, nxh, nyh, nzh

      parameter (nx = 48, ny = 48, nz = 48)
      parameter (nxh = nx / 2, nyh = ny / 2, nzh = nz / 2)

      integer
      .  iseed, i, j, k

      real
      .  dx, dy, dz, uref, urms, lnscl, pi, xlen, ylen, zlen, sigma,
      .  umx, umn, vmx, vmn, wmx, wmn,
      .  umean, vmean, wmean, vrms, wrms, ratio

      real
      .  wk(2*nx), wk2(2*nx)

      real
      .  up(nx,ny,nz), vp(nx,ny,nz), wp(nx,ny,nz),
      .  rad(nx,ny,nz), exx(nx,ny,nz),
      .  ph1(nx,ny,nz), ph2(nx,ny,nz), ph3(nx,ny,nz)

      real
      .  rannum

      complex
      .  tmp1(nx,ny,nz), tmp2(nx,ny,nz), tmp3(nx,ny,nz), tmp4(nx,ny,nz)

      double precision drandm
      double precision dseed

      common/rndo /dseed

      pi = 4.0 * atan(1.0)

c     print*, ' Enter Urms:'
c     read(5,*) urms
c     print*, ' Enter Uref:'
```

```

c      read(5,*) uref
c      print*, ' Enter Iseed:'
c      read(5,*) izeed
c      izeed = 1000000

      dx = 2.0 * pi / float(nx)
      dy = 2.0 * pi / float(ny)
      dz = 2.0 * pi / float(nz)

      xlen = float(nx) * dx
      ylen = float(ny) * dy
      zlen = float(nz) * dz

C-----
C---- warm up the random number generator
C-----
      dseed  = 256.d0
      do i = 1,1000
        rannum = drandm(dseed)
      end do
C-----

      call rand3d
      . (xlen, ylen, zlen, uref, urms, ratio,
      .  lnscl, izeed, up, vp, wp, ph1, ph2, ph3, sigma, rad, exx, wk,
      .  tmp1, tmp2, tmp3, tmp4)
cgmng
c      open(unit=50,file='qphys.dat',form='unformatted')
c      write(50) nx, ny, nz
c      write(50) ((up(i,j,k),i=1,nx),j=1,ny),k=1,nz),
c      .          ((vp(i,j,k),i=1,nx),j=1,ny),k=1,nz),
c      .          ((wp(i,j,k),i=1,nx),j=1,ny),k=1,nz)
c      close(50)

      print*, 'Enter Velocity Scale'
      read(5,*) u_scale
      print*, 'Enter x_min'
      read(5,*) x_min
      print*, 'Enter x_max'
      read(5,*) x_max
      print*, 'Enter y_min'
      read(5,*) y_min
      print*, 'Enter y_max'
      read(5,*) y_max
      print*, 'Enter z_min'
      read(5,*) z_min
      print*, 'Enter z_max'
      read(5,*) z_max

      open(unit=50,file='qphys.ip',form='formatted')
      write(50,*) '2'
      write(50,*) '3'
      write(50,*) nx*ny*nz
      write(50,*) '3'
      write(50,*) 'x-velocity'
      write(50,*) 'y-velocity'
      write(50,*) 'z-velocity'
      do i = 1,nx
        do j = 1,ny
          do k = 1,nz
            !write(50,*) xlen*float(i-1)/float(nx) + 0.5*dx
            write(50,*) x_min + (x_max-x_min)*float(i-1)/float(nx)
          end do
        end do
      end do
      do i = 1,nx
        do j = 1,ny
          do k = 1,nz
            !write(50,*) ylen*float(j-1)/float(ny) + 0.5*dy
            write(50,*) y_min + (y_max-y_min)*float(j-1)/float(ny)
          end do
        end do
      end do

```

```

        end do
    end do
end do
do i = 1,nx
    do j = 1,ny
        do k = 1,nz
            !write(50,*) zlen*float(k-1)/float(nz) + 0.5*dz
            write(50,*) z_min + (z_max-z_min)*float(k-1)/float(nz)
        end do
    end do
end do
do i = 1,nx
    do j = 1,ny
        do k = 1,nz
            !write(50,*) up(i,j,k)
            write(50,*) up(i,j,k)*u_scale
        end do
    end do
end do
do i = 1,nx
    do j = 1,ny
        do k = 1,nz
            !write(50,*) vp(i,j,k)
            write(50,*) vp(i,j,k)*u_scale
        end do
    end do
end do
do i = 1,nx
    do j = 1,ny
        do k = 1,nz
            !write(50,*) wp(i,j,k)
            write(50,*) wp(i,j,k)*u_scale
        end do
    end do
end do
close(50)
cgmrg

c --- calculate velocity statistics
c
urms = 0.0
vrms = 0.0
wrms = 0.0
umean = 0.0
vmean = 0.0
wmean = 0.0
umx = -10.0e10
vmx = -10.0e10
wmx = -10.0e10
umn = 10.0e10
vmn = 10.0e10
wmn = 10.0e10
do k = 1,nz
    do j = 1,ny
        do i = 1,nx
            umean = umean + up(i,j,k)
            vmean = vmean + vp(i,j,k)
            wmean = wmean + wp(i,j,k)
            urms = urms + up(i,j,k)**2
            vrms = vrms + vp(i,j,k)**2
            wrms = wrms + wp(i,j,k)**2
            umx = max( umx,up(i,j,k) )
            vmx = max( vmx,vp(i,j,k) )
            wmx = max( wmx,wp(i,j,k) )
            umn = min( umn,up(i,j,k) )
            vmn = min( vmn,vp(i,j,k) )
            wmn = min( wmn,wp(i,j,k) )
        end do
    end do
end do
umean = umean / (float(nx)*float(ny)*float(nz))

```

```

vmean = vmean / (float(nx)*float(ny)*float(nz))
wmean = wmean / (float(nx)*float(ny)*float(nz))
urms = sqrt( urms / (float(nx)*float(ny)*float(nz)) )
vrms = sqrt( vrms / (float(nx)*float(ny)*float(nz)) )
wrms = sqrt( wrms / (float(nx)*float(ny)*float(nz)) )

write(6,*) '   umean,vmean,wmean: ',umean,vmean,wmean
write(6,*) '   urms,vrms,wrms  : ',urms,vrms,wrms
write(6,*) '   umx,vmx,wmx    : ',umx,vmx,wmx
write(6,*) '   umn,vmn,wmn    : ',umn,vmn,wmn
c
stop
end

C*****
double precision function drandm(dl)
C*****

double precision dl
dl = dmod(16807.0d0*dl,2147483647.0d0)
drandm = dl * 4.6566128752458d-10
C-----
return
end
c
subroutine rand3d
. (xlen, ylen, zlen, uref, urms, ratio,
. lnscl, iseed, up, vp, wp, ph1, ph2, ph3, sigma, rad, exx, wk,
. tmp1, tmp2, tmp3, tmp4)

c implicit none

integer
. nx, ny, nz, nxh, nyh, nzh, nxhp, nyhp, nzhp, nsum

parameter (nx = 48, ny = 48, nz = 48)
parameter (nxh = nx / 2, nyh = ny / 2, nzh = nz / 2)
parameter (nxhp= nx/2+1, nyhp= ny/2+1, nzhp= nz/2+1)
parameter (nsum = nxh*nxh + nyh*nyh + nzh*nzh)

integer
. i, j, k,
. irkmx, iseed, ii, jj, kk, im, ip, jm, jp, km, kp,
. ishift, jshift, kshift, kseed,
. im1, im2, ip1, ip2, jm1, jm2, jp1, jp2,
. km1, km2, kp1, kp2

real
. xlen, ylen, zlen, uref, urms, lnscl, pi, dkx, dky, dkz,
. dkx2, dky2, dkz2,
. rkmax, fac, akx, aky, akz,
. etot, enorm, sigma,
. umcon, sp1, rand1, phmax, ur, ui, ratio,
. ran, rk, temp, rk0, x, y, z, vr, vi, dx, dy, dz, sp2,
. rand2, sp3, rand3, wr, wi, r1, rfac, r12dx, r12dy, r12dz

real
. wk(2*nx)

real
. up(nx,ny,nz), vp(nx,ny,nz), wp(nx,ny,nz),
. rad(nx,ny,nz), exx(nx,ny,nz),
. ph1(nx,ny,nz), ph2(nx,ny,nz), ph3(nx,ny,nz)

real cpeak,a,b,c,d

real
. sn(0:nxh),esh(0:nxh)

real
. wtr(nx,ny,nz),rindex(228),rindex2(228),rkcut

```

```

complex
. ai

complex
. tmp1(nx,ny,nz), tmp2(nx,ny,nz), tmp3(nx,ny,nz), tmp4(nx,ny,nz)

double precision drandm
double precision dseed

common/rndo /dseed

c Set up scaling factor (so that up() will have correct urms value)

do k = 1,nz
do j = 1,ny
do i = 1,nx
exx(i,j,k) = 0.0
end do
end do
end do

c write random data input info into tape 44

write(6,60)
write(6,61) nx, ny, nz, xlen, ylen, zlen
write(6,64) nxh, nyh, nzh
write(6,62) uref, urms, lnscl

60 format(10x, 'The Turbulent Initialization Conditions',/)
61 format(1x, 'nkx=', i4, 3x, 'nky=', i4, 3x, 'nkz=', i4, 3x,
. 'xlen=', f8.5, 3x, 'ylen=', f8.5, 3x, 'zlen=', f8.5)
62 format(10x, 'uref,urms,lnscl',5x,3e13.5)
64 format(1x, 'nkxh=', i4, 3x, 'nkyh=', i4, 3x, 'nkzh=', i4)

do i=1,nx
rindex(i) =float(i)-1.
rindex2(i)=float(i-1)*float(i-1)
end do

pi = 4.0 * atan(1.0)

dkx = 2.0 * pi / xlen
dky = 2.0 * pi / ylen
dkz = 2.0 * pi / zlen
dkx2=dkx*dkx
dky2=dky*dky
dkz2=dkz*dkz
rkcut=float(nxh)*sqrt(8./9.)

do k=1,nzh+1
do j=1,nyh+1
do i=1,nxh+1
wtr(i,j,k)=sqrt(rindex2(i)*dkx2+rindex2(j)*dky2
1 +rindex2(k)*dkz2)
end do
end do
end do
do k = 1,nzh+1
do j = 1,nyh+1
do i = 2,nxh
ii = nx - i + 2
wtr(ii,j,k)=sqrt(rindex2(i)*dkx2+rindex2(j)*dky2
1 +rindex2(k)*dkz2)

end do
end do
end do

do k = 1,nzh+1
do j = 2,nyh

```

```

do i = 2,nxh
  ii = nx - i + 2
  jj = ny - j + 2
  wtr(ii,jj,k)=sqrt(rindex2(i)*dkx2+rindex2(j)*dky2
1      +rindex2(k)*dkz2)

end do
end do
end do

do k = 2,nzh
do j = 1,nyh+1
do i = 2,nxh
  ii = nx - i + 2
  kk = nz - k + 2
  wtr(ii,j,kk)=sqrt(rindex2(i)*dkx2+rindex2(j)*dky2
1      +rindex2(k)*dkz2)

end do
end do
end do

do k = 2,nzh
do j = 2,nyh
do i = 2,nxh
  ii = nx - i + 2
  jj = ny - j + 2
  kk = nz - k + 2
  wtr(ii,jj,kk)=sqrt(rindex2(i)*dkx2+rindex2(j)*dky2
1      +rindex2(k)*dkz2)

end do
end do
end do

do k = 1,nzh+1
do j = 2,nyh
do i = 1,nxh+1
  jj = ny - j + 2
  wtr(i,jj,k)=sqrt(rindex2(i)*dkx2+rindex2(j)*dky2
1      +rindex2(k)*dkz2)

end do
end do
end do

do k = 2,nzh
do j = 1,nyh+1
do i = 1,nxh+1
  kk = nz - k + 2
  wtr(i,j,kk)=sqrt(rindex2(i)*dkx2+rindex2(j)*dky2
1      +rindex2(k)*dkz2)

end do
end do
end do

do k = 2,nzh
do j = 2,nyh
do i = 1,nxh+1
  jj = ny - j + 2
  kk = nz - k + 2
  wtr(i,jj,kk)=sqrt(rindex2(i)*dkx2+rindex2(j)*dky2
1      +rindex2(k)*dkz2)

end do
end do
end do

do k=1,nz
do j=1,ny

```

```

do i=1,nx
if (wtr(i,j,k).le.0.0001)wtr(i,j,k)=0.
if (wtr(i,j,k).gt.0.0001.and.wtr(i,j,k).le.rkcut)wtr(i,j,k)=1.
if (wtr(i,j,k).gt.rkcut)wtr(i,j,k)=0.
end do
end do
end do

c   write(6,*) 'cpeak='
c   read(5,*) cpeak
c   rk0 = cpeak * max(dkx,dky,dkz)

do i = 0,nxh
    sn(i) = 0.
end do

do k = 1,nz
    kk = k
    if(k.gt.nzhp) kk = nz - k + 2
    km = kk-1
    do j = 1,ny
        jj = j
        if(j.gt.nyhp) jj = ny - j + 2
        jm = jj-1
        do i = 1,nx
            ii = i
            if(i.gt.nxhp) ii = nx - i + 2
            im = ii-1
            iks= im*im+jm*jm+km*km
            rk = sqrt(float(iks))
            do m = 0,nxh
                if(rk.gt.float(m)-.5.and.rk.le.float(m)+.5
1          .and.rk.le.float(nxh)*sqrt(8./9.))
1          sn(m) = sn(m) + 1.
            end do
        end do
    end do
end do

c   tsum = 0
c   do i = 0,nxh
c       write(*,*) i,sn(i)
c       tsum = tsum + sn(i)
c   end do
c   write(*,*) tsum

do i = 0,nxh
    rk = float(i)
    if(rk.lt.2.021) then
        a = 1.617
        c = 2.411e-3
        b = 2.021
        d = 4.352e-3
        a = alog10(a)
        b = alog10(b)
        c = alog10(c)
        d = alog10(d)
        rk= alog10(rk+1.e-10)
        esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
        go to 999
    else if(rk.lt.2.426) then
        a = 2.021
        c = 4.352e-3
        b = 2.426
        d = 6.093e-3
        a = alog10(a)
        b = alog10(b)
        c = alog10(c)
        d = alog10(d)
        rk= alog10(rk)
        esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)

```

```

go to 999
else if(rk.lt.3.234) then
a = 2.426
c = 6.093e-3
b = 3.234
d = 8.231e-3
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.4.043) then
a = 3.234
c = 8.231e-3
b = 4.043
d = 8.647e-3
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.5.660) then
a = 4.043
c = 8.647e-3
b = 5.660
d = 7.190e-3
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.8.085) then
a = 5.660
c = 7.190e-3
b = 8.085
d = 5.109e-3
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.12.128) then
a = 8.085
c = 5.109e-3
b = 12.128
d = 3.179e-3
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.16.170) then
a = 12.128
c = 3.179e-3
b = 16.170
d = 2.271e-3
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)

```

```

esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.20.213) then
a = 16.170
c = 2.271e-3
b = 20.213
d = 1.684e-3
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.24.255) then
a = 20.213
c = 1.684e-3
b = 24.255
d = 1.330e-3
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.32.340) then
a = 24.255
c = 1.330e-3
b = 32.340
d = 8.893e-4
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.48.510) then
a = 32.340
c = 8.893e-4
b = 48.510
d = 4.674e-4
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.64.681) then
a = 48.510
c = 4.674e-4
b = 64.681
d = 2.384e-4
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.80.851) then
a = 64.681
c = 2.384e-4
b = 80.851
d = 1.404e-4
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)

```

```

rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.101.063) then
a = 80.851
c = 1.404e-4
b = 101.063
d = 7.493e-5
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.121.276) then
a = 101.063
c = 7.493e-5
b = 121.276
d = 4.409e-5
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else if(rk.lt.141.489) then
a = 121.276
c = 4.409e-5
b = 141.489
d = 2.535e-5
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
go to 999
else
a = 141.489
c = 2.535e-5
b = 161.701
d = 1.514e-5
a = alog10(a)
b = alog10(b)
c = alog10(c)
d = alog10(d)
rk= alog10(rk)
esh(i) =10.**((d-c)/(b-a)*(rk-a)+c)
end if
999 continue
end do

esh(0) = 0.
do m = 1,nxh
if(float(m).gt.float(nxh)*sqrt(8./9.)) esh(m) = 0.
end do

eres = 0.

do i = 0,nxh

rk = sqrt(float(i))

if(rk.gt.0..and.rk.le.float(nxh)*sqrt(8./9.))
1 eres=eres+esh(i)

end do

write(*,*) 'eres = ',eres

```

```

do k = 1,nzh+1
  km = k-1
  do j = 1,nyh+1
    jm = j-1
    do i = 1,nxh+1
      im = i-1

      iks= im*im+jm*jm+km*km
      rk = sqrt(float(iks))

      do m = 0,nxh
        if(rk.gt.float(m)-.5.and.rk.le.float(m)+.5
1      .and.rk.le.float(nxh)*sqrt(8./9.)) then
          ik = m
          exx(i,j,k) = esh(ik)/sn(ik)
          tmp4(i,j,k) = esh(ik)/sn(ik)
          go to 99
          end if
        end do
        exx(i,j,k) = 0.
        tmp4(i,j,k) = 0.
99      continue
      end do
    end do
  end do

  exx(1,1,1) = 0.
  tmp4(1,1,1) = 0.

do k = 1,nzh+1
  do j = 1,nyh+1
    do i = 2,nxh

      ii = nx - i + 2

      exx(ii,j,k) = exx(i,j,k)
      tmp4(ii,j,k) = tmp4(i,j,k)

    end do
  end do
end do

do k = 1,nzh+1
  do j = 2,nyh
    do i = 2,nxh

      ii = nx - i + 2
      jj = ny - j + 2

      exx(ii,jj,k) = exx(i,j,k)
      tmp4(ii,jj,k) = tmp4(i,j,k)

    end do
  end do
end do

do k = 2,nzh
  do j = 1,nyh+1
    do i = 2,nxh

      ii = nx - i + 2
      kk = nz - k + 2

      exx(ii,j,kk) = exx(i,j,k)
      tmp4(ii,j,kk) = tmp4(i,j,k)

    end do
  end do
end do

```

```

do k = 2,nzh
  do j = 2,nyh
    do i = 2,nxh

      ii = nx - i + 2
      jj = ny - j + 2
      kk = nz - k + 2

      exx(ii,jj,kk) = exx(i,j,k)
      tmp4(ii,jj,kk) = tmp4(i,j,k)

    end do
  end do
end do

do k = 1,nzh+1
  do j = 2,nyh
    do i = 1,nxh+1

      jj = ny - j + 2

      exx(i,jj,k) = exx(i,j,k)
      tmp4(i,jj,k) = tmp4(i,j,k)

    end do
  end do
end do

do k = 2,nzh
  do j = 1,nyh+1
    do i = 1,nxh+1

      kk = nz - k + 2

      exx(i,j,kk) = exx(i,j,k)
      tmp4(i,j,kk) = tmp4(i,j,k)

    end do
  end do
end do

do k = 2,nzh
  do j = 2,nyh
    do i = 1,nxh+1

      jj = ny - j + 2
      kk = nz - k + 2

      exx(i,jj,kk) = exx(i,j,k)
      tmp4(i,jj,kk) = tmp4(i,j,k)

    end do
  end do
end do

c  do k=1,nz
c  do j=1,ny
c  do i=1,nx
c    exx(i,j,k) = wtr(i,j,k)*exx(i,j,k)
c    tmp4(i,j,k) = wtr(i,j,k)*tmp4(i,j,k)
c  end do
c  end do
c  end do

c  write(6,*) '   EXX TEST '
c  write(6,121) ((exx(i,j,2),i=1,8),j=1,8)
c 121 format (1x,8(1pe12.5))

c  Normalize target array

  etot = 0.0

```

```

do k = 1,nz
  do j = 1,ny
    do i = 1,nx
      etot = etot + exx(i,j,k)
    end do
  end do
end do

do k = 1,nz
  do j = 1,ny
    do i = 1,nx
c      exx(i,j,k) = exx(i,j,k) / etot *.5/3.
      exx(i,j,k) = exx(i,j,k) /3.
    end do
  end do
end do

c  Print out target spectrum

c      write(6,1769)
c1769  format(20x,'exx(i,j,2)')

c      do j = 1,ny
c          write(6,1760) (exx(i,j,2),i=1,nxh)
c      end do

1760  format(1x,9e13.5)

c  Calculate total energy in target spectrum

enorm = 0.0
do k = 1,nz
  do j = 1,ny
    do i = 1,nx
      enorm = enorm + exx(i,j,k)
    end do
  end do
end do

c  Convert target spectrum into physical velocities

do k = 1,nz
  do j = 1,ny
    do i = 1,nx
      tmp1(i,j,k) = cmplx(0.0,0.0)
      tmp2(i,j,k) = cmplx(0.0,0.0)
      tmp3(i,j,k) = cmplx(0.0,0.0)
    end do
  end do
end do

etot = 0.0
phmax = pi

do k = 1,nzh+1
  do j = 1,nyh+1
    do i = 1,nxh+1

c  Calculate velocity magnitude

      umcon = sqrt(2.0 * exx(i,j,k))

c  Add in random phase information

c      sp1 = ran(iseed)
c      sp2 = ran(iseed)
c      sp3 = ran(iseed)

      sp1 = drandm(dseed)
      sp2 = drandm(dseed)

```

```

        sp3 = drandm(dseed)
c --- range -1 to +1
        rand1 = 2.0 * sp1 - 1.0
        rand2 = 2.0 * sp2 - 1.0
        rand3 = 2.0 * sp3 - 1.0

        ph1(i,j,k) = rand1 * phmax
        ph2(i,j,k) = rand2 * phmax
        ph3(i,j,k) = rand3 * phmax

        if((i.eq.1.and.j.ne.1.and.k.ne.1).or.
1         (i.ne.1.and.j.eq.1.and.k.ne.1).or.
1         (i.ne.1.and.j.ne.1.and.k.eq.1).or.
1         (i.eq.1.and.j.eq.1.and.k.eq.1)) then
            if(ph1(i,j,k).ge.0.) then
                ph1(i,j,k) = 0.
            else
                ph1(i,j,k) = pi
            end if
            if(ph2(i,j,k).ge.0.) then
                ph2(i,j,k) = 0.
            else
                ph2(i,j,k) = pi
            end if
            if(ph3(i,j,k).ge.0.) then
                ph3(i,j,k) = 0.
            else
                ph3(i,j,k) = pi
            end if
        end if

        ur = umcon * cos(ph1(i,j,k))
        ui = -umcon * sin(ph1(i,j,k))
        tmp1(i,j,k) = cmplx(ur,ui)

        vr = umcon * cos(ph2(i,j,k))
        vi = -umcon * sin(ph2(i,j,k))
        tmp2(i,j,k) = cmplx(vr,vi)

        wr = umcon * cos(ph3(i,j,k))
        wi = -umcon * sin(ph3(i,j,k))
        tmp3(i,j,k) = cmplx(wr,wi)

        etot = etot + 0.5 * umcon**2

    end do
end do

tmp1(1,1,1) = 0.
tmp2(1,1,1) = 0.
tmp3(1,1,1) = 0.

do k = 1,nzh+1
    do j = 1,nyh+1
        do i = 2,nxh

            ii = nx - i + 2

            umcon = sqrt(2.0 * exx(i,j,k))
            ur = umcon * cos(ph1(i,j,k))
            ui = -umcon * sin(ph1(i,j,k))
            vr = umcon * cos(ph2(i,j,k))
            vi = -umcon * sin(ph2(i,j,k))
            wr = umcon * cos(ph3(i,j,k))
            wi = -umcon * sin(ph3(i,j,k))

            tmp1(ii,j,k) = cmplx(ur,-ui)
            tmp2(ii,j,k) = cmplx(vr,-vi)

```

```

        tmp3(ii,j,k) = cmplx(wr,-wi)
        etot = etot + 0.5 * umcon**2
    end do
end do
end do

do k = 1,nzh+1
    do j = 2,nyh
        do i = 2,nxh

            ii = nx - i + 2
            jj = ny - j + 2

            umcon = sqrt(2.0 * exx(i,j,k))
            ur = umcon * cos(ph1(i,j,k))
            ui = -umcon * sin(ph1(i,j,k))
            vr = umcon * cos(ph2(i,j,k))
            vi = -umcon * sin(ph2(i,j,k))
            wr = umcon * cos(ph3(i,j,k))
            wi = -umcon * sin(ph3(i,j,k))

            tmp1(ii,jj,k) = cmplx(ur,ui)
            tmp2(ii,jj,k) = cmplx(vr,vi)
            tmp3(ii,jj,k) = cmplx(wr,wi)
            etot = etot + 0.5 * umcon**2

        end do
    end do
end do

do k = 2,nzh
    do j = 1,nyh+1
        do i = 2,nxh

            ii = nx - i + 2
            kk = nz - k + 2

            umcon = sqrt(2.0 * exx(i,j,k))
            ur = umcon * cos(ph1(i,j,k))
            ui = -umcon * sin(ph1(i,j,k))
            vr = umcon * cos(ph2(i,j,k))
            vi = -umcon * sin(ph2(i,j,k))
            wr = umcon * cos(ph3(i,j,k))
            wi = -umcon * sin(ph3(i,j,k))

            tmp1(ii,j,kk) = cmplx(ur,ui)
            tmp2(ii,j,kk) = cmplx(vr,vi)
            tmp3(ii,j,kk) = cmplx(wr,wi)
            etot = etot + 0.5 * umcon**2

        end do
    end do
end do

do k = 2,nzh
    do j = 2,nyh
        do i = 2,nxh

            ii = nx - i + 2
            jj = ny - j + 2
            kk = nz - k + 2

            umcon = sqrt(2.0 * exx(i,j,k))
            ur = umcon * cos(ph1(i,j,k))
            ui = -umcon * sin(ph1(i,j,k))
            vr = umcon * cos(ph2(i,j,k))
            vi = -umcon * sin(ph2(i,j,k))
            wr = umcon * cos(ph3(i,j,k))
            wi = -umcon * sin(ph3(i,j,k))

```

```

        tmp1(ii,jj,kk) = cmplx(ur,-ui)
        tmp2(ii,jj,kk) = cmplx(vr,-vi)
        tmp3(ii,jj,kk) = cmplx(wr,-wi)
        etot = etot + 0.5 * umcon**2

    end do
end do

do k = 1,nzh+1
    do j = 2,nyh
        do i = 1,nxh+1

            jj = ny - j + 2

            umcon = sqrt(2.0 * exx(i,j,k))
            ur = umcon * cos(ph1(i,j,k))
            ui = -umcon * sin(ph1(i,j,k))
            vr = umcon * cos(ph2(i,j,k))
            vi = -umcon * sin(ph2(i,j,k))
            wr = umcon * cos(ph3(i,j,k))
            wi = -umcon * sin(ph3(i,j,k))

            tmp1(i,jj,k) = cmplx(ur,-ui)
            tmp2(i,jj,k) = cmplx(vr,-vi)
            tmp3(i,jj,k) = cmplx(wr,-wi)
            etot = etot + 0.5 * umcon**2

        end do
    end do
end do

do k = 2,nzh
    do j = 1,nyh+1
        do i = 1,nxh+1

            kk = nz - k + 2

            umcon = sqrt(2.0 * exx(i,j,k))
            ur = umcon * cos(ph1(i,j,k))
            ui = -umcon * sin(ph1(i,j,k))
            vr = umcon * cos(ph2(i,j,k))
            vi = -umcon * sin(ph2(i,j,k))
            wr = umcon * cos(ph3(i,j,k))
            wi = -umcon * sin(ph3(i,j,k))

            tmp1(i,j,kk) = cmplx(ur,-ui)
            tmp2(i,j,kk) = cmplx(vr,-vi)
            tmp3(i,j,kk) = cmplx(wr,-wi)
            etot = etot + 0.5 * umcon**2

        end do
    end do
end do

do k = 2,nzh
    do j = 2,nyh
        do i = 1,nxh+1

            jj = ny - j + 2
            kk = nz - k + 2

            umcon = sqrt(2.0 * exx(i,j,k))
            ur = umcon * cos(ph1(i,j,k))
            ui = -umcon * sin(ph1(i,j,k))
            vr = umcon * cos(ph2(i,j,k))
            vi = -umcon * sin(ph2(i,j,k))
            wr = umcon * cos(ph3(i,j,k))
            wi = -umcon * sin(ph3(i,j,k))

            tmp1(i,jj,kk) = cmplx(ur,ui)

```

```

                tmp2(i,jj,kk) = cmplx(vr,vi)
                tmp3(i,jj,kk) = cmplx(wr,wi)
                etot = etot + 0.5 * umcon**2

            end do
        end do
    end do

    do k=1,nz
    do j=1,ny
    do i=1,nx
        tmp1(i,j,k) = wtr(i,j,k)*tmp1(i,j,k)
        tmp2(i,j,k) = wtr(i,j,k)*tmp2(i,j,k)
        tmp3(i,j,k) = wtr(i,j,k)*tmp3(i,j,k)
    end do
    end do
    end do

    do k = 1,nz
        do j = 1,ny
            do i = 1,nx
                exx(i,j,k) = tmp4(i,j,k)
            end do
        end do
    end do

c   Print out total energies

        write(6,71) enorm, etot
71   format(' Exx total = ',e15.7,' Energy in U (Spectral)= ',e15.7)

c   Calculate velocity in physical space

        call ifft3d(tmp1, up, nx, ny, nz, tmp4, wk)
        call ifft3d(tmp2, vp, nx, ny, nz, tmp4, wk)
        call ifft3d(tmp3, wp, nx, ny, nz, tmp4, wk)

c   call fft3d(up, tmp1, nx, ny, nz, wk)
c   call ifft3d(tmp1, up, nx, ny, nz, tmp4, wk)
c   compute energy in physical space

        etot = 0.0

    do k = 1,nz
        do j = 1,ny
            do i = 1,nx
                etot = etot+.5*(up(i,j,k)**2+vp(i,j,k)**2+wp(i,j,k)**2)
            end do
        end do
    end do

        etot = etot / (float(nx)*float(ny)*float(nz))
        print*, ' Energy (physical)=', etot

        ratio = 3.*enorm / etot
        write(6,*) '   ratio = ',ratio
c   write(*,*) 'sigma ='
c   read(*,*)   sigma

c   Scale by SIGMA

c   ratio = sqrt(ratio)

c   do k = 1,nz
c   do j = 1,ny
c   do i = 1,nx
c       up(i,j,k) = sigma*ratio * up(i,j,k)
c       vp(i,j,k) = sigma*ratio * vp(i,j,k)
c       wp(i,j,k) = sigma*ratio * wp(i,j,k)
c   end do
c   end do

```

```

c      end do

      return
      end

c
      subroutine fft3d(v, vht, il, j1, kl, z2)

c This routine performs an FFT on the real array v and places the
c result in the complex array vht

c      implicit none

      integer id, il, j1, kl, i, j, k, ilm1, jlm1, klm1, nn(3)

      real tke
      real v(il,j1,kl)

      complex vht(il,j1,kl), z2(il)

c      Convert v to spectral space

      tke = 0.0

      do k = 1,kl
        do j = 1,j1
          do i = 1,il
            vht(i,j,k) = cmplx(v(i,j,k),0.0)
            tke = tke + 0.5 * v(i,j,k)**2
          end do
        end do
      end do

      tke = tke / float(il * j1 * kl)

      print*, ' FFT3D internal check-'
      print*, ' TKEave (physical) =', tke

      nn(1) = il
      nn(2) = j1
      nn(3) = kl

      call fourt(vht,nn,3,-1,1,z2)

      do k = 1,kl
        do j = 1,j1
          do i = 1,il
            vht(i,j,k) = vht(i,j,k) / nn(1) / nn(2) / nn(3)
          end do
        end do
      end do

      tke = 0.0

      do k = 1,kl
        do j = 1,j1
          do i = 1,il
            tke = tke + 0.5 * real(vht(i,j,k) * conjg(vht(i,j,k)))
          end do
        end do
      end do

      print*, ' TKEave (spectral) =', tke

      return
      end

c
      subroutine ifft3d(vht, v, il, j1, kl, vhtmp, z2)

c This routine performs an inverse 3-D FFT

      integer il, j1, kl, i, j, k, nn(3)

```

```

real tke,tkep,ratio
real v(il,jl,kl)

complex z2(il)
complex vht(il,jl,kl), vhtmp(il,jl,kl)

tke = 0.0

do k = 1,kl
  do j = 1,jl
    do i = 1,il
      tke = tke + 0.5 * real(vht(i,j,k) * conjg(vht(i,j,k)))
    end do
  end do
end do

print*, ' IFFT3D internal check-'
print*, ' TKEave (spectral) =', tke

nn(1) = il
nn(2) = jl
nn(3) = kl

do k = 1,kl
  do j = 1,jl
    do i = 1,il
      vhtmp(i,j,k) = vht(i,j,k) * nn(1) * nn(2) * nn(3)
    end do
  end do
end do

call fourt(vhtmp,nn,3,1,1,z2)

tkep = 0.0

do k = 1,kl
  do j = 1,jl
    do i = 1,il
      v(i,j,k) = real(vhtmp(i,j,k)) / nn(1) / nn(2) / nn(3)
      tkep = tkep + 0.5 * v(i,j,k)**2
    end do
  end do
end do

tkep = tkep / float(il * jl * kl)

print*, ' TKEave (physical) =', tkep

return
end

c
c subroutine fourt(data,nn,ndim,isign,iform,work)
c
c the cooley-tukey fast fourier transform in usasi basic fortran
c
c transform(j1,j2,,,,) = sum(data(i1,i2,,,,)*w1**((i2-1)*(j2-1))
c                      *w2**((i2-1)*(j2-1))*,,,,),
c where i1 and j1 run from 1 to nn(1) and w1=exp(isign*2*pi=
c sqrt(-1)/nn(1)), etc. there is no limit on the dimensionality
c (number of subscripts) of the data array. if an inverse
c transform (isign=+1) is performed upon an array of transformed
c (isign=-1) data, the original data will reappear.
c multiplied by nn(1)*nn(2)*,,, the array of input data must be
c in complex format. however, if all imaginary parts are zero (i.e.
c the data are disguised real) running time is cut up to forty per-
c cent. (for fastest transform of real data, nn(1) should be even.)
c the transform values are always complex and are returned in the
c original array of data, replacing the input data. the length
c of each dimension of the data array may be any integer. the
c program runs faster on composite integers than on primes, and is

```

```

c      particularly fast on numbers rich in factors of two.
c
c      timing is in fact given by the following formula.  let ntot be the
c      total number of points (real or complex) in the data array, that
c      is, ntot=nn(1)*nn(2)*...  decompose ntot into its prime factors,
c      such as 2**k2 * 3**k3 * 5**k5 * ...  let sum2 be the sum of all
c      the factors of two in ntot, that is, sum2 = 2*k2.  let sumf be
c      the sum of all other factors of ntot, that is, sumf = 3*k3*5*k5*..
c      the time taken by a multidimensional transform on these ntot data
c      is t = t0 + ntot*(t1+t2*sum2+t3*sumf).  on the cdc 3300 (floating
c      point add time = six microseconds), t = 3000 + ntot*(600+40*sum2+
c      175*sumf) microseconds on complex data.
c
c      implementation of the definition by summation will run in a time
c      proportional to ntot*(nn(1)+nn(2)+...).  for highly composite ntot
c      the savings offered by this program can be dramatic.  a one-dimen-
c      sional array 4000 in length will be transformed in 4000*(600+
c      40*(2+2+2+2+2)+175*(5+5+5)) = 14.5 seconds versus about 4000*
c      4000*175 = 2800 seconds for the straightforward technique.
c
c      the fast fourier transform places three restrictions upon the
c      data.
c      1.  the number of input data and the number of transform values
c          must be the same.
c      2.  both the input data and the transform values must represent
c          equispaced points in their respective domains of time and
c          frequency.  calling these spacings deltat and deltaf, it must be
c          true that deltaf=2*pi/(nn(i)*deltat).  of course, deltat need not
c          be the same for every dimension.
c      3.  conceptually at least, the input data and the transform output
c          represent single cycles of periodic functions.
c
c      the calling sequence is--
c      call fourt(data,nn,ndim,isign,iform,work)
c
c      data is the array used to hold the real and imaginary parts
c      of the data on input and the transform values on output.  it
c      is a multidimensional floating point array, with the real and
c      imaginary parts of a datum stored immediately adjacent in storage
c      (such as fortran iv places them).  normal fortran ordering is
c      expected, the first subscript changing fastest.  the dimensions
c      are given in the integer array nn, of length ndim.  isign is -1
c      to indicate a forward transform (exponential sign is -) and +1
c      for an inverse transform (sign is +).  iform is +1 if the data are
c      complex, 0 if the data are real.  if it is 0, the imaginary
c      parts of the data must be set to zero.  as explained above, the
c      transform values are always complex and are stored in array data.
c      work is an array used for working storage.  it is floating point
c      real, one dimensional of length equal to twice the largest array
c      dimension nn(i) that is not a power of two.  if all nn(i) are
c      powers of two, it is not needed and may be replaced by zero in the
c      calling sequence.  thus, for a one-dimensional array, nn(1) odd,
c      work occupies as many storage locations as data.  if supplied,
c      work must not be the same array as data.  all subscripts of all
c      arrays begin at one.
c
c      example 1.  three-dimensional forward fourier transform of a
c      complex array dimensioned 32 by 25 by 13 in fortran iv.
c      dimension data(32,25,13),work(50),nn(3)
c      complex data
c      data nn/32,25,13/
c      do 1 i=1,32
c      do 1 j=1,25
c      do 1 k=1,13
c 1  data(i,j,k)=complex value
c      call fourt(data,nn,3,-1,1,work)
c
c      example 2.  one-dimensional forward transform of a real array of
c      length 64 in fortran ii,
c      dimension data(2,64)
c      do 2 i=1,64

```

```

c   data(1,i)=real part
c 2  data(2,i)=0.
c   call fourt(data,64,1,-1,0,0)
c
c   there are no error messages or error halts in this program.  the
c   program returns immediately if ndim or any nn(i) is less than one.
c
c   program by norman brenner from the basic program by charles
c   rader, june 1967.  the idea for the digit reversal was
c   suggested by ralph alter.
c
c   this is the fastest and most versatile version of the fft known
c   to the author.  a program called four2 is available that also
c   performs the fast fourier transform and is written in usasi basic
c   fortran.  it is about one third as long and restricts the
c   dimensions of the input array (which must be complex) to be powers
c   of two.  another program, called four1, is one tenth as long and
c   runs two thirds as fast on a one-dimensional complex array whose
c   length is a power of two.
c
c   reference--
c   ieee audio transactions (june 1967), special issue on the fft.
c   dimension data(*),nn(1),ifact(32),work(1)
c   data twopi/6.2831853071796/,rthlf/0.70710678118655/
c   data nprev/0/,np0/0/
c the following call is for gathering statistics on library use at ncar
c call q8qst4( 4hplib      , 5hfourt      ,5hfourt      ,10hversion 9)
c   if(ndim-1)920,1,1
1   ntot=2
   do 2 idim=1,ndim
   if(nn(idim))920,920,2
2   ntot=ntot*nn(idim)
c
c   main loop for each dimension
c
   np1=2
   do 910 idim=1,ndim
   n=nn(idim)
   np2=np1*n
   if(n-1)920,900,5
c
c   is n a power of two and if not, what are its factors
c
5   m=n
   ntwo=np1
   if=1
   idiv=2
10  iquot=m/idiv
   irem=m-idiv*iquot
   if(iquot-idiv)50,11,11
11  if(irem)20,12,20
12  ntwo=ntwo+ntwo
   ifact(if)=idiv
   if=if+1
   m=iquot
   go to 10
20  idiv=3
   inon2=if
30  iquot=m/idiv
   irem=m-idiv*iquot
   if(iquot-idiv)60,31,31
31  if(irem)40,32,40
32  ifact(if)=idiv
   if=if+1
   m=iquot
   go to 30
40  idiv=idiv+2
   go to 30
50  inon2=if
   if(irem)60,51,60
51  ntwo=ntwo+ntwo

```

```

        go to 70
60    ifact(if)=m
    c
    c    separate four cases--
    c        1. complex transform or real transform for the 4th, 9th,etc.
    c            dimensions.
    c        2. real transform for the 2nd or 3rd dimension. method--
    c            transform half the data, supplying the other half by con-
    c            jugate symmetry.
    c        3. real transform for the 1st dimension, n odd. method--
    c            set the imaginary parts to zero.
    c        4. real transform for the 1st dimension, n even. method--
    c            transform a complex array of length n/2 whose real parts
    c            are the even numbered real values and whose imaginary parts
    c            are the odd numbered real values. separate and supply
    c            the second half by conjugate symmetry.
    c
70    icode=1
    ifmin=1
    ilrng=np1
    if(idim-4) 71,100,100
71    if(iform) 72,72,100
72    icode=2
    ilrng=np0*(1+nprev/2)
    if(idim-1) 73,73,100
73    icode=3
    ilrng=np1
    if(ntwo-np1) 100,100,74
74    icode=4
    ifmin=2
    ntwo=ntwo/2
    n=n/2
    np2=np2/2
    ntot=ntot/2
    i=1
    do 80 j=1,ntot
    data(j)=data(i)
80    i=i+2
    c
    c    shuffle data by bit reversal, since n=2**k. as the shuffling
    c    can be done by simple interchange, no working array is needed
    c
100   if(ntwo-np2) 200,110,110
110   np2hf=np2/2
    j=1
    do 150 i2=1,np2,np1
    if(j-i2) 120,130,130
120   ilmax=i2+np1-2
    do 125 i1=i2,ilmax,2
    do 125 i3=i1,ntot,np2
    j3=j+i3-i2
    tempr=data(i3)
    tempi=data(i3+1)
    data(i3)=data(j3)
    data(i3+1)=data(j3+1)
    data(j3)=tempr
125   data(j3+1)=tempi
130   m=np2hf
140   if(j-m) 150,150,145
145   j=j-m
    m=m/2
    if(m-np1) 150,140,140
150   j=j+m
    go to 300
    c
    c    shuffle data by digit reversal for general n
    c
200   nwork=2*n
    do 270 i1=1,np1,2
    do 270 i3=i1,ntot,np2
    j=i3

```

```

do 260 i=1,nwork,2
  if(icase-3)210,220,210
210 work(i)=data(j)
   work(i+1)=data(j+1)
   go to 230
220 work(i)=data(j)
   work(i+1)=0.
230 ifp2=np2
   if=ifmin
240 ifp1=ifp2/ifact(if)
   j=j+ifp1
   if(j-i3-ifp2)260,250,250
250 j=j-ifp2
   ifp2=ifp1
   if=if+1
   if(ifp2-np1)260,260,240
260 continue
   i2max=i3+np2-np1
   i=1
   do 270 i2=i3,i2max,np1
     data(i2)=work(i)
     data(i2+1)=work(i+1)
270 i=i+2
c
c main loop for factors of two. perform fourier transforms of
c length four, with one of length two if needed. the twiddle factor
c  $w = \exp(i \text{sign} * 2 * \pi * \sqrt{-1} * m / (4 * \text{mmax}))$ . check for  $w = i \text{sign} * \sqrt{-1}$ 
c and repeat for  $w = w * (1 + i \text{sign} * \sqrt{-1}) / \sqrt{2}$ .
c
300 if(ntwo-np1)600,600,305
305 npltw=np1+np1
   ipar=ntwo/np1
310 if(ipar-2)350,330,320
320 ipar=ipar/4
   go to 310
330 do 340 i1=1,ilrng,2
   do 340 k1=i1,ntot,npltw
     k2=k1+np1
     tempr=data(k2)
     tempi=data(k2+1)
     data(k2)=data(k1)-tempr
     data(k2+1)=data(k1+1)-tempi
     data(k1)=data(k1)+tempr
     data(k1+1)=data(k1+1)+tempi
340
350 mmax=np1
360 if(mmax-ntwo/2)370,600,600
370 lmax=max0(npltw,mmax/2)
   do 570 l=np1,lmax,npltw
     m=1
     if(mmax-np1)420,420,380
380 theta=-twopi*float(l)/float(4*mmax)
     if(isign)400,390,390
     theta=-theta
390 wr=cos(theta)
     wi=sin(theta)
410 w2r=wr*wr-wi*wi
     w2i=2.*wr*wi
     w3r=w2r*wr-w2i*wi
     w3i=w2r*wi+w2i*wr
420 do 530 i1=1,ilrng,2
     kmin=i1+ipar*m
     if(mmax-np1)430,430,440
430 kmin=i1
     kdif=ipar*mmax
440 kstep=4*kdif
     if(kstep-ntwo)460,460,530
460 do 520 k1=kmin,ntot,kstep
     k2=k1+kdif
     k3=k2+kdif
     k4=k3+kdif
     if(mmax-np1)470,470,480

```

```

470  u1r=data(k1)+data(k2)
     u1i=data(k1+1)+data(k2+1)
     u2r=data(k3)+data(k4)
     u2i=data(k3+1)+data(k4+1)
     u3r=data(k1)-data(k2)
     u3i=data(k1+1)-data(k2+1)
     if(isign)471,472,472
471  u4r=data(k3+1)-data(k4+1)
     u4i=data(k4)-data(k3)
     go to 510
472  u4r=data(k4+1)-data(k3+1)
     u4i=data(k3)-data(k4)
     go to 510
480  t2r=w2r*data(k2)-w2i*data(k2+1)
     t2i=w2r*data(k2+1)+w2i*data(k2)
     t3r=wr*data(k3)-wi*data(k3+1)
     t3i=wr*data(k3+1)+wi*data(k3)
     t4r=w3r*data(k4)-w3i*data(k4+1)
     t4i=w3r*data(k4+1)+w3i*data(k4)
     u1r=data(k1)+t2r
     u1i=data(k1+1)+t2i
     u2r=t3r+t4r
     u2i=t3i+t4i
     u3r=data(k1)-t2r
     u3i=data(k1+1)-t2i
     if(isign)490,500,500
490  u4r=t3i-t4i
     u4i=t4r-t3r
     go to 510
500  u4r=t4i-t3i
     u4i=t3r-t4r
510  data(k1)=u1r+u2r
     data(k1+1)=u1i+u2i
     data(k2)=u3r+u4r
     data(k2+1)=u3i+u4i
     data(k3)=u1r-u2r
     data(k3+1)=u1i-u2i
     data(k4)=u3r-u4r
520  data(k4+1)=u3i-u4i
     kdif=kstep
     kmin=4*(kmin-i1)+i1
     go to 450
530  continue
     m=m+lmax
     if(m-mmax)540,540,570
540  if(isign)550,560,560
550  tempr=wr
     wr=(wr+wi)*rthlf
     wi=(wi-tempr)*rthlf
     go to 410
560  tempr=wr
     wr=(wr-wi)*rthlf
     wi=(tempr+wi)*rthlf
     go to 410
570  continue
     ipar=3-ipar
     mmax=mmax+mmax
     go to 360

c
c  main loop for factors not equal to two.  apply the twiddle factor
c   $w = \exp(i\text{sign} * 2 * \pi * \sqrt{-1} * (j1-1) * (j2-j1) / (\text{ifp1} + \text{ifp2}))$ , then
c  perform a fourier transform of length ifact(if), making use of
c  conjugate symmetries.
c
600  if(ntwo-np2)605,700,700
605  ifp1=ntwo
     if=inon2
     nplhf=np1/2
610  ifp2=ifact(if)*ifp1
     jlmin=np1+1
     if(jlmin-ifp1)615,615,640

```

```

615 do 635 j1=j1min,ifp1,np1
theta=-twopi*float(j1-1)/float(ifp2)
if(isign)625,620,620
620 theta=-theta
625 wstpr=cos(theta)
wstpi=sin(theta)
wr=wstpr
wi=wstpi
j2min=j1+ifp1
j2max=j1+ifp2-ifp1
do 635 j2=j2min,j2max,ifp1
ilmax=j2+ilrng-2
do 630 i1=j2,ilmax,2
do 630 j3=i1,ntot,ifp2
tempr=data(j3)
data(j3)=data(j3)*wr-data(j3+1)*wi
630 data(j3+1)=tempr*wi+data(j3+1)*wr
tempr=wr
wr=wr*wstpr-wi*wstpi
635 wi=tempr*wstpi+wi*wstpr
640 theta=-twopi/float(ifact(if))
if(isign)650,645,645
645 theta=-theta
650 wstpr=cos(theta)
wstpi=sin(theta)
j2rng=ifp1*(1+ifact(if)/2)
do 695 i1=1,ilrng,2
do 695 i3=i1,ntot,np2
j2max=i3+j2rng-ifp1
do 690 j2=i3,j2max,ifp1
j1max=j2+ifp1-np1
do 680 j1=j2,j1max,np1
j3max=j1+np2-ifp2
do 680 j3=j1,j3max,ifp2
jmin=j3-j2+i3
jmax=jmin+ifp2-ifp1
i=1+(j3-i3)/np1hf
if(j2-i3)655,655,665
655 sumr=0.
sumi=0.
do 660 j=jmin,jmax,ifp1
659 sumr=sumr+data(j)
660 sumi=sumi+data(j+1)
work(i)=sumr
work(i+1)=sumi
go to 680
665 iconj=1+(ifp2-2*j2+i3+j3)/np1hf
j=jmax
sumr=data(j)
sumi=data(j+1)
oldsr=0.
oldsi=0.
j=j-ifp1
670 tempr=sumr
tempi=sumi
sumr=twowr*sumr-oldsr+data(j)
sumi=twowr*sumi-oldsi+data(j+1)
oldsr=tempr
oldsi=tempi
j=j-ifp1
if(j-jmin)675,675,670
675 tempr=wr*sumr-oldsr+data(j)
tempi=wi*sumi
work(i)=tempr-tempi
work(iconj)=tempr+tempi
tempr=wr*sumi-oldsi+data(j+1)
tempi=wi*sumr
work(i+1)=tempr+tempi
work(iconj+1)=tempr-tempi
680 continue
if(j2-i3)685,685,686

```

```

685  wr=wstpr
      wi=wstpi
      go to 690
686  tempr=wr
      wr=wr*wstpr-wi*wstpi
      wi=tempr*wstpi+wi*wstpr
690  twowr=wr+wr
      i=1
      i2max=i3+np2-np1
      do 695 i2=i3,i2max,np1
        data(i2)=work(i)
        data(i2+1)=work(i+1)
695  i=i+2
      if=if+1
      ifp1=ifp2
      if(ifp1-np2) 610,700,700

c
c  complete a real transform in the 1st dimension, n even, by con-
c  jugate symmetries.
c
700  go to (900,800,900,701),icase
701  nhalf=n
      n=n+n
      theta=-twopi/float(n)
      if(isign) 703,702,702
702  theta=-theta
703  wstpr=cos(theta)
      wstpi=sin(theta)
      wr=wstpr
      wi=wstpi
      imin=3
      jmin=2*nhalf-1
      go to 725
710  j=jmin
      do 720 i=imin,ntot,np2
        sumr=(data(i)+data(j))/2.
        sumi=(data(i+1)+data(j+1))/2.
        difr=(data(i)-data(j))/2.
        difi=(data(i+1)-data(j+1))/2.
        tempr=wr*sumi+wi*difr
        tempi=wi*sumi-wr*difr
        data(i)=sumr+tempr
        data(i+1)=difi+tempi
        data(j)=sumr-tempr
        data(j+1)=-difi+tempi
720  j=j+np2
      imin=imin+2
      jmin=jmin-2
      tempr=wr
      wr=wr*wstpr-wi*wstpi
      wi=tempr*wstpi+wi*wstpr
725  if(imin-jmin) 710,730,740
730  if(isign) 731,740,740
731  do 735 i=imin,ntot,np2
735  data(i+1)=-data(i+1)
740  np2=np2+np2
      ntot=ntot+ntot
      j=ntot+1
      imax=ntot/2+1
745  imin=imax-2*nhalf
      i=imin
      go to 755
750  data(j)=data(i)
      data(j+1)=-data(i+1)
755  i=i+2
      j=j-2
      if(i-imax) 750,760,760
760  data(j)=data(imin)-data(imin+1)
      data(j+1)=0.
      if(i-j) 770,780,780
765  data(j)=data(i)

```

```

      data(j+1)=data(i+1)
770  i=i-2
      j=j-2
      if(i-imin)775,775,765
775  data(j)=data(imin)+data(imin+1)
      data(j+1)=0.
      imax=imin
      go to 745
780  data(1)=data(1)+data(2)
      data(2)=0.
      go to 900

c
c  complete a real transform for the 2nd or 3rd dimension by
c  conjugate symmetries.
c
800  if(i1rng-np1)805,900,900
805  do 860 i3=1,ntot,np2
      i2max=i3+np2-np1
      do 860 i2=i3,i2max,np1
        imin=i2+i1rng
        imax=i2+np1-2
        jmax=2*i3+np1-imin
        if(i2-i3)820,820,810
810  jmax=jmax+np2
820  if(idim-2)850,850,830
830  j=jmax+np0
        do 840 i=imin,imax,2
          data(i)=data(j)
          data(i+1)=-data(j+1)
840  j=j-2
850  j=jmax
        do 860 i=imin,imax,np0
          data(i)=data(j)
          data(i+1)=-data(j+1)
860  j=j-np0
      c
      c  end of loop on each dimension
      c
900  np0=np1
      np1=np2
910  nprev=n
920  return
      end

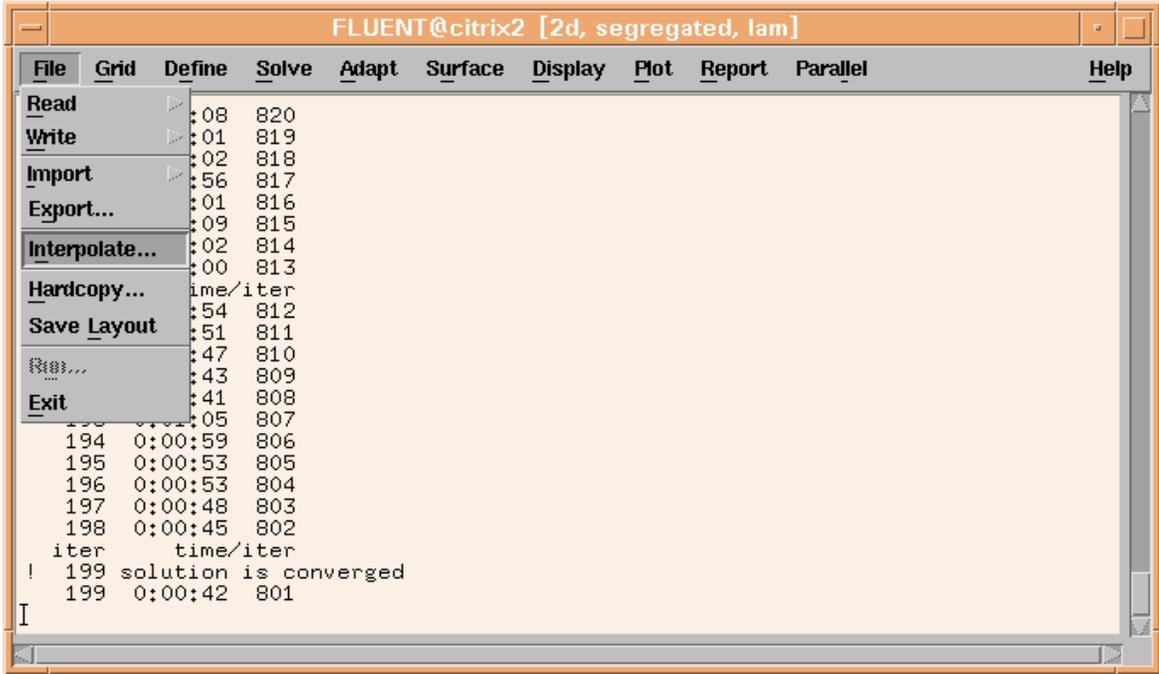
```

## 9.2 INITIALIZATION OF FLOW USING AN INTERPOLATION FILE

With the interpolation file created using the FORTAN code, the next step is to read the file into the model. FLUENT contains functionality to both produce and read existing interpolation files. An interpolation file is a list of numbers that contain the location and scalar values of pressure, velocity, temperature, etc. for a given model. FLUENT uses a zero order interpolation scheme that essentially assigns a cell in the new model the closest value in the model from which the interpolation file came from. It is much like the interpolation scheme used by ANSYS in Section 5.2.3 except no linear interpolation is done. The old cell values are simply mapped to the closest new cell, which is why it is referred to as a zero order scheme. This next section is a tutorial for reading and writing interpolation files in FLUENT. The use of the interpolation files

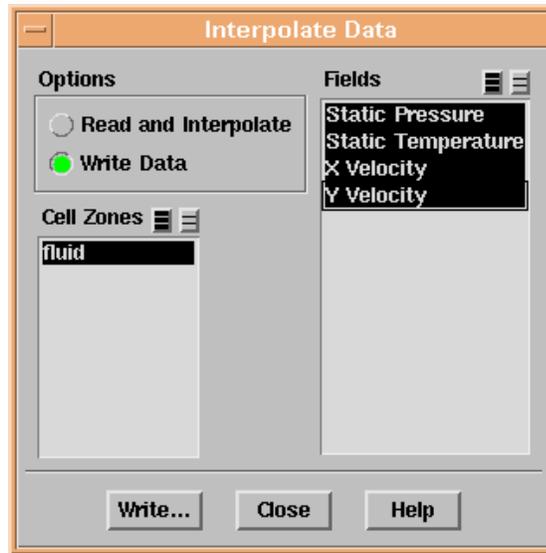
expedites grid independence studies when the flow is initialized with a converged solution from a coarse grid. The example used here is for the classic skewed-lid heat transfer problem:

- A. Write the interpolation file by choosing **File → Interpolate** (Figure 9-1)



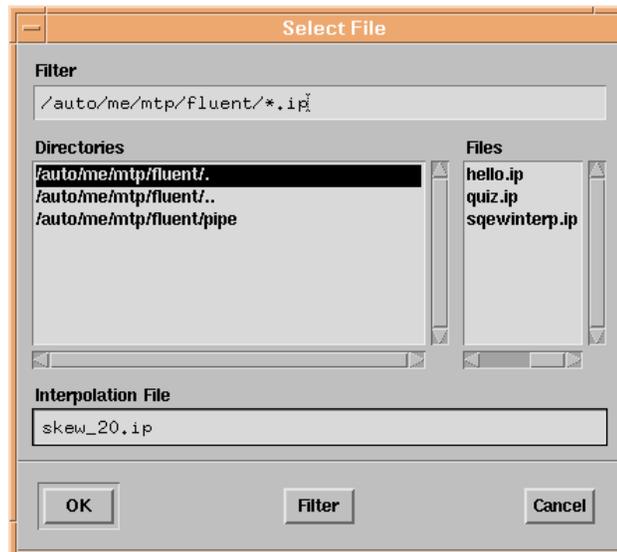
**Figure 9-1 Interpolation beginnings**

- B. Select the fluid entity that you would like to write as an interpolation file (usually there is only one fluid). Then select the **Write Data** radio button, and then select all the profiles you want to export (usually you will select them all.)



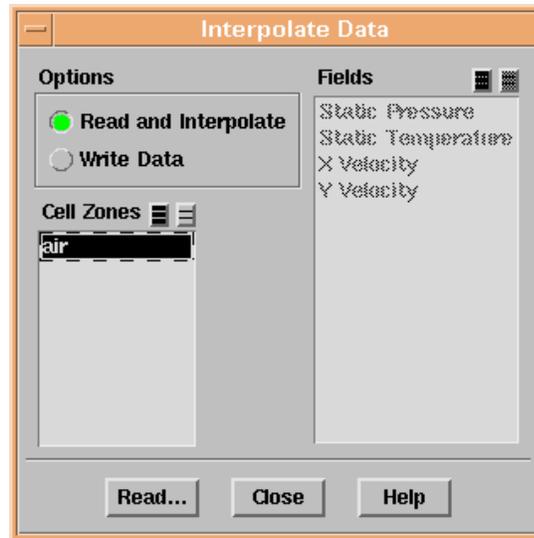
**Figure 9-2** Select the values to write

- C. Make sure you name it with an extension of \*.ip. When you read it in, it will only display the \*.ip extensions. This example is named skew\_20.ip.



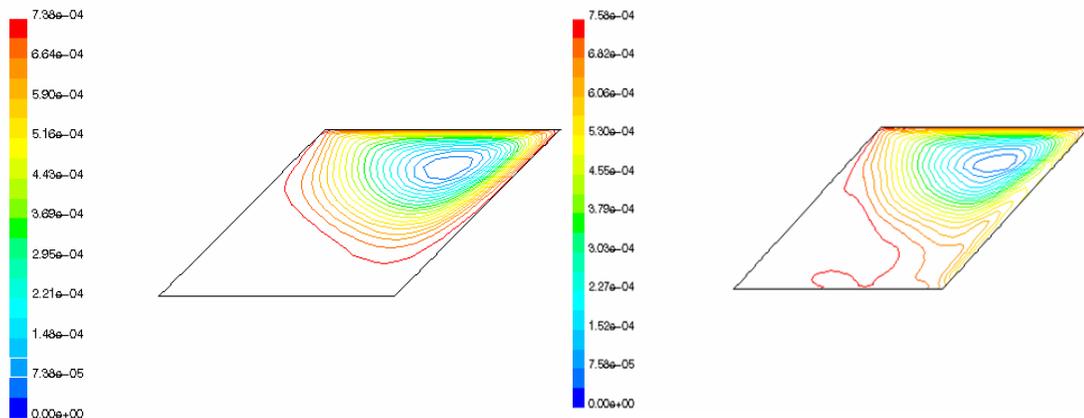
**Figure 9-3** Name the file

- D. Now create the finer mesh in Gambit and read that file in as usual. Then set up your FLUENT model as usual doing everything but the initialization of the profiles.
- E. Instead of initializing your profiles to be constants, read in the interpolation file that was created in step 4. Choose **Read → then select the file (i.e.: skew\_20.ip)**



**Figure 9-4 Read in the file**

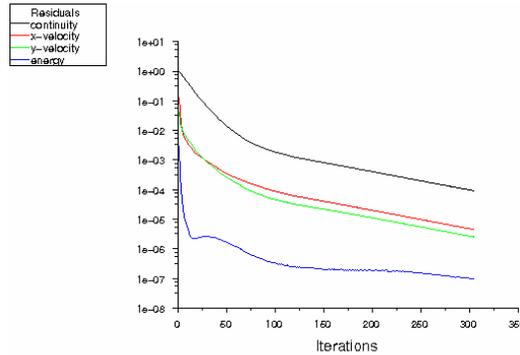
F. The FLUENT main display should let you know that it has initialized the values. Of course, one can check to make sure this has happened by plotting a contour of the streamlines, or anything else. Figure 9-5 displays a course mesh streamlines compared with the finer mesh streamlines after interpolation.



**Figure 9-5 Comparison of streamlines**

G. Start iterating until convergence... Figure 9-6 shows the difference in convergence iteration number/speed. The model initialized with a constant value took over 500 iterations to converge while the interpolated model took a little over 300 to converge.

## Residuals with Interpolation



## Residuals without Interpolation

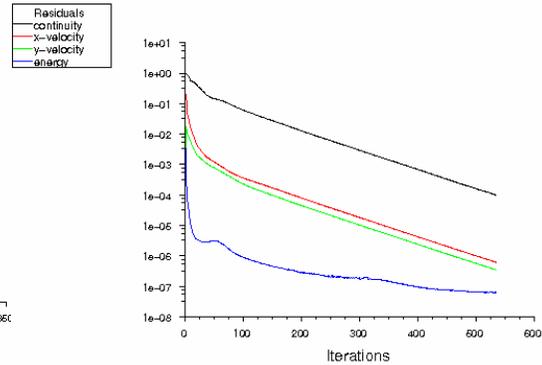


Figure 9-6 Convergence comparison

## 9.3 PRESSURE EXTRACTION

The pressure extraction process is performed after the fluid LES model is sufficiently converged. Once converged, the simulation is saved and opened on a serial version of FLUENT since the parallel version lacks the exporting functionality. After the simulation is open, create a journal file of the following list of Text User Interface (TUI) commands and type the following line in the execute command window found under **Solve** → **Execute Command**. In this case, the file containing the TUI commands is called *press.jou*.

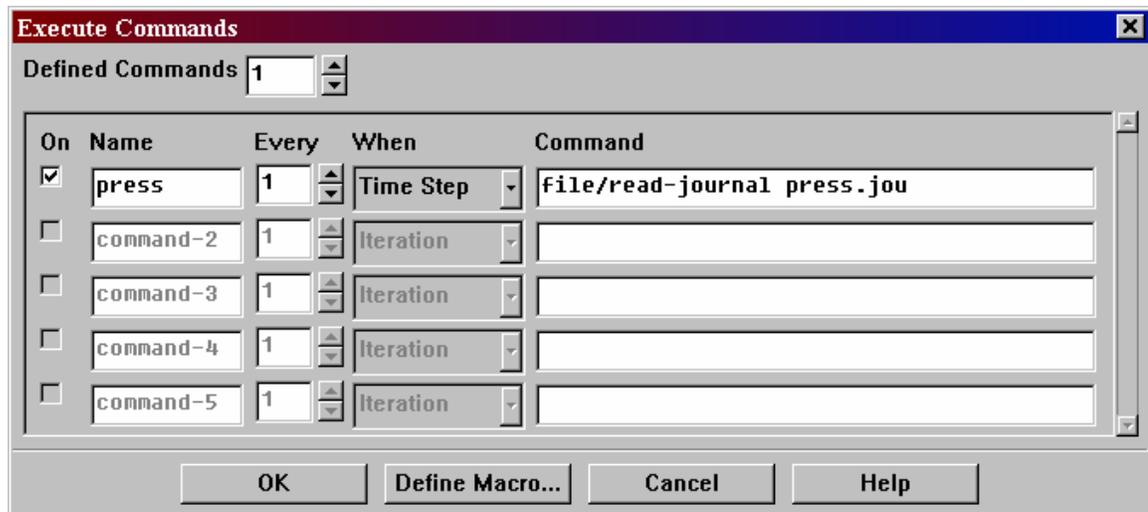


Figure 9-7 Execute command window

The string of TUI commands are listed in Table 9-1. The left column contains the text to be entered into the file and the right column contains the comments to the text. There is not a way to comment journal files that the author knows about, so the text must be copied into a file without the comments.

**Table 9-1 Macro to export the pressures in FLUENT**

<b>Journal Macro</b>	<b>Comment</b>
file/export	%enter the export menu
ascii	%the file will be written in ASCII format
pressure-%t.csv	%the name of the file will be a function of the time flow time-step number and written in a comma separated value format.
wall	%this specifies the region of interest. In our case, it is the wall
()	%this specifies the end of the regions to be written
no	%no loads are to be written at the boundaries
yes	%the default delimiter is a space character; type yes for comma
pressure	separated values
q/	%specify the scalar value to be written. In our case it is the pressure
no	%choose no to the cell centered option
q/	%this quits the export menu and returns to the home menu

As the end of this project was approaching, Bill Wangard, Ph. D., another FLUENT engineer provided assistance in the development of a parallel User Defined Function (UDF) for exporting the pressure on the supercomputers given in Figure 9-2. It exports the pressure on a face zone to file "export". The zone ID is hard-wired into the source. Recompile it for a different zone id, or define a SCHEME variable in the FLUENT GUI and call it from the UDF. This procedure is described in the UDF manual for v6.1.

**Table 9-2 UDF for pressure table exporting for parallel processing**

```
#include "udf.h"

# define WALLID 5

DEFINE_ON_DEMAND(export)
{
#if !RP_HOST /* Host will do nothing in this udf */
    face_t f;
    Thread *tf;
    real x[ND_ND];
    Domain *domain = Get_Domain(1);
    FILE * fp;
```

```

int i, dummy=0;

/* Node 0 will open a NEW file while node 1, 2, ... will wait. After
node 0 finishes,
Node 1 will open the SAME file while node 2, 3 ... will wait.
This goes on. */

/* Here is the signal to start */
#if RP_NODE
if (! I_AM_NODE_ZERO_P) PRF_CRECV_INT(myid - 1, &dummy, 1, myid - 1);
#endif

/* Open the output file */
fp = fopen("export", "a");

tf=Lookup_Thread(domain, WALLID);

begin_f_loop(f, tf) /* loops over faces in a face thread */
if PRINCIPAL_FACE_P(f, tf)
{
F_CENTROID(x, f, tf);
for(i=0;i<ND_ND;i++)
fprintf(fp, "%12.4e ", x[i]);
fprintf(fp,"%12.4e \n", F_P(f, tf));

}
end_f_loop(f, tf);

Message0("\n\nExport is complete. \n");

/* After the node finishes, it will close the file and send a signal
saying I am done so that the next node can start */
#if RP_NODE
if (! I_AM_NODE_LAST_P) PRF_CSEND_INT(myid + 1, &dummy, 1, myid);
#endif

fclose (fp);

#endif
}

```

As noted in the above figure, the macro can be executed at any number of time-steps and will write a pressure file for each specified time-step. The result can take up a considerable amount of disk space since each file can range anywhere between one and two megabytes depending on the number of nodes at the wall.

The files need to be periodically copied and formatted in such a way for ANSYS to read them. This task is accomplished using a worksheet created in Mathcad that sequentially reads the data file into the program, periodically copies the 4/3D section to cover the 1.1 meter section of structural model and writes it in a floating point column

format that contains 4 columns with 8 characters in 12 spaces (namely X, Y, Z, Pressure respectively). A text file written in any other format will be unreadable in ANSYS using the macros defined in the next section. There is nothing special or unique about the Mathcad program used to generate these files. Any program that will create a text file that contains the X, Y, and Z location with its associated pressure value will work.

## 9.4 PRESSURE MAPPING MACROS

As with FLUENT, ANSYS has the ability to read macros of text interface functions that can mitigate the time-consuming process of repeated commands. Since 50 files of pressure data need to be imported into the model, a series of short macros are written to accomplish this purpose. The alternative is to input the correct values from the table at each time-step totaling approximately 2 million data points by hand, an obviously unfeasible choice.

The first step is to take advantage of a few of the ANSYS functions that allow the mapping and interpolation of existing data onto the structure. In the event that there exists fewer data points than structural nodes, the function is designed to interpolate between the given set of data to provide an approximation for the pressure at the structural node. If there happens to be more data than structural nodes, as in our case, the program still interpolates if a data point doesn't coincide with a node, but it essentially extracts the few data points that the model needs. The macro used to map the pressure is given in Table 9-3 and is named *pressuremap*.<sup>o</sup> This macro maps the pressure for one time-step. Since the pipe problem involves multiple time-steps, an additional macro expedites the creation of multiple load steps and is also given in the table entitled *tpressload*.

**Table 9-3 Macro for mapping CFD pressures to a transient structural model**

<p>MACRO: <i>pressuremap</i>, ARG1          ARGUMENTS: 1) file number          ANSYS Version: 6.1</p>
---

<p><b>Description:</b> This macro will maps pressures from a different analysis</p>
---

<p><b>Dependencies:</b> Requires a file in ASCII floating point format with 4 columns 12</p>
--

<sup>o</sup> ANSYS allows for commenting inside the macro through the use of an (!) and is done so to provide and easy reference to its location.

characters long with 8 numbers, which represent the point locations and pressures to be mapped.

**Usage:** This macro is used to map pressure from a CFD mesh onto a structural mesh. The file name and extension must be specified in line 49; it is preferable to place a number after each file name. This number becomes the argument placed after the macro name.

```
! define arrays for the pressure mapping onto your structural model

/nerr,-0,200000000

!Define the total number of rows in the array
ppts=17000*11

!asel,s,,,3,6
!nsla,s,1
nsl,all
cm,psurf,node

!Type the path after the /cwd command
/cwd,/auto/grp2/mflow/ansys/smallpipe

!Enter the preprocessor stage--this won't work unless you are in this
stage
/prep7

!
!           GET DATA
!
*dim,data,,ppts,4
!the name of the file to be read is the second argument in the *vread
!function, ARG1 is the first argument input in the command line after
!the macro name. Therefore, it is recommended that the files be named
!with a number corresponding to their time-step.
*vread,data(1,1),1500litmin-%ARG1%,txt,,JIK,4,ppts,,0
(4f12.8)

*del,cfdxyz,,nopr
*del,cfdpres,,nopr
*dim,cfdxyz,,ppts,3
*voper,cfdxyz(1,1),data(1,1),add,0
*voper,cfdxyz(1,2),data(1,2),add,0
*voper,cfdxyz(1,3),data(1,3),add,0

*dim,cfdpres,,ppts
*voper,cfdpres(1),data(1,4),add,0

*del,data

cmsel,s,psurf
*get,nnum,node,,count
*get,nmax,node,,num,max
*del,nlis1,,nopr
*del,nlis2,,nopr
```

```

*del, pmapxyz, , nopr
*del, pmap2xyz, , nopr
*del, pout, , nopr
*del, pout2, , nopr
*del, nmask, , nopr
!
*dim, nlis1, array, nmax, 1
*dim, nlis2, array, nnum, 1
*dim, pmapxyz, array, nmax, 3
*dim, pmap2xyz, array, nnum, 3
*dim, pout, array, nmax, 1
*dim, pout2, array, nnum, 1
*dim, nmask, array, nmax
*vget, nmask(1), node, 1, nsel
!
! The mapping operations needs compressed arrays (no gaps)
! but the *VGET fill such that row number is node number
! So we get the data then we compress it before mapping
! Then we expand it again in order to use the SFFUN
!
*vfill, nlis1(1,1), ramp, 1, 1
*vget, pmapxyz(1,1), node, 1, loc, x
*vget, pmapxyz(1,2), node, 1, loc, y
*vget, pmapxyz(1,3), node, 1, loc, z
!
*vmask, nmask(1)
*vfun, nlis1(1), comp, nlis1(1)
*vfun, nlis2(1), copy, nlis1(1)
!
*vmask, nmask(1)
*vfun, pmapxyz(1,1), comp, pmapxyz(1,1)
*vmask, nmask(1)
*vfun, pmapxyz(1,2), comp, pmapxyz(1,2)
*vmask, nmask(1)
*vfun, pmapxyz(1,3), comp, pmapxyz(1,3)
!
*mfun, pmap2xyz(1,1), copy, pmapxyz(1,1)
!
! Perform mapping
!
*moper, pout2(1,1), pmapxyz(1,1), MAP, cfdpres(1), cfdxyz(1,1), 3, .01
!
*voper, pout(1), pout2(1), SCAT, nlis2(1)
!
! make sure you have the elements attached to these nodes in the active
set
!
esln, a
!
! apply pressure
!
sffun, pres, pout(1)
sf, all, pres
sffun
nplot
/replot

```

MACRO: tpressload, ARG1, ARG2, ARG3, ARG4

ARGUMENT: 1) beginning time 2) number of load steps, 3) time step 4) skip

ANSYS Version: 6.1

**Description:** This macro imports the pressure data into the ANSYS model and writes each time-step to the model.

**Dependencies:** You need to make sure the pressuremap.mac is in the same directory as this macro and they are both updated to fit your model. Make sure you have all the necessary pressure load files in the proper format and numbered starting at 1 with a base name and \*.txt extension. The format should look a something like this: basenamel.txt.

**Usage:** You will pass in the first time value, the number of load steps, time step for the FLUENT model and the number of files you have skipped.

```
!Type the path after the /cwd command
/cwd,/auto/grp2/mflow/ansys/1.5inch/300litmin/

!import the first pressure load
pressuremap,1

!set the end of the first time step.
begtime=ARG1
time,begtime

!write this load step
LSWRITE,1

!enter the loop to have the pressure imported at different time
timesteps
nloads=ARG2
timestep=ARG3
skip=ARG4
*do,i,2,nloads,1
    time, begtime+(i-1)*timestep*skip
    pressuremap,i
    LSWRITE,i
*enddo
```

## 9.5 IMECE 2003 CONFERENCE PROCEEDING

Found on next page.

## IMECE2003-42541

### NUMERICAL MODELING OF LES BASED TURBULENT FLOW INDUCED VIBRATION

Matthew T. Pittard

Jonathan D. Blotter

Brigham Young University  
Dept. of Mechanical Engineering  
Provo, UT 84602

T: 801-422-7820, F: 801-422-0516, jblotter@byu.edu

#### ABSTRACT

Flow-induced vibration caused by fully developed pipe flow has been recognized, but not fully investigated under turbulent conditions. This paper focuses on the development of a numerical, fluid-structure interaction (FSI) model that will help define the relationship between pipe wall vibration and the physical characteristics of turbulent flow. Commercial FSI software packages are based on Reynolds Averaged Navier-Stokes (RANS) fluid models which do not compute the instantaneous fluctuations in turbulent flow. This paper presents an FSI approach based on Large Eddy Simulation (LES) flow models that compute the instantaneous fluctuations in turbulent flow. The results based on the LES models indicate that these fluctuations contribute to the pipe vibration. It is shown that there is a near quadratic relationship between the standard deviation of the pressure field on the pipe wall and the flow rate. It is also shown that a strong relationship between pipe vibration and flow rate exists. This research has a direct impact on the geothermal, nuclear, and other fluid transport industries.

#### INTRODUCTION

Flow-induced vibration can be divided into three categories: *turbulence-induced vibration* - as seen in fluttering pipes, *vorticity shedding-induced vibration* - the phenomena that destroyed the

Tacoma Narrows Bridge, and *fluid elastic instability* - a unique form of flow-induced vibration that is most commonly seen in nuclear heat exchangers after the tube velocity reaches a critical value [1]. Of these, it is the turbulence-induced phenomena that will be the focus of this research.

Keywords: Large eddy simulation, fluid-structure interaction

The vibration of a pipe transporting fluid has been recognized by researchers and quantified using numerical, analytical and experimental techniques. In the past, researchers such as Saito [2], Evans [3], Durant [4,5], Brevbart [6], and Kim [7] have presented relationships between fluid flow rate and pipe vibration. Although results vary, each researcher proposed that pipe flutter was a direct result of the pressure fluctuations at the pipe wall which are inherent to turbulent flow.

Although some results have been presented as listed above, researchers still face basic challenges in attempting to solve this problem. The current numerical and analytical solutions model the pressure fluctuations using simplifying assumptions about the fluid flow. These simplified models use time-averaged equations, which do not provide instantaneous values. Commercial FSI codes use Reynold's Averaged Navier-Stokes (RANS) equations to

model the turbulent flow. These models do not compute the pressure variations at the fluid-structure interface. Since it has been hypothesized that these pressure variations contribute to the pipe vibrations, these commercial codes will not accomplish the purpose of this study. Experimental solutions can be time consuming and expensive. It can also be difficult to isolate the vibrations induced by the pressure fluctuations alone. Because of these two concerns, accurately quantifying the vibrations induced only by the pressure fluctuations has yet to be presented.

This paper presents a method for computing the pipe vibration using Large Eddy Simulation (LES) based turbulent flow models. LES models spatially filter the governing Navier-Stokes equations and compute the instantaneous pressure fluctuations in the pipe. The technique presented in this paper couples a LES based fluid model with a structural solver to provide a method for analyzing this turbulence-induced phenomena. This modeling approach will help determine the contribution of the pressure variations to the overall vibration of the pipe. A model development of this kind will also provide a benchmark and method for investigating future applications where experimental data would be difficult to obtain.

This paper first presents a background and review of previous analytical and experimental developments. An overview of RANS and LES theory is then presented followed by a detailed description of the FSI procedure developed in this paper. Finally, the results indicating the effects of the turbulent flow on the vibration of the pipe are presented and discussed.

## **BACKGROUND**

This section provides a brief background on the energy transfer in a pipe containing a flowing fluid. Also, as mentioned in the introduction, some research has been done to determine the relationship between the flow rate through a pipe and the accompanied dynamic response. These researchers approached this problem using numerical, analytical and/or experimental techniques. This section also briefly discusses what has been done in these areas and the limitations to each approach.

### **Energy Transfer**

The understanding of how energy is transferred at the fluid-structure interface in fully developed turbulent pipe flow is fundamental to developing an accurate model. Intuition may

theorize that energy is transferred because fluid particles “hit” the pipe wall, much like marbles dropping on a tin roof. However, internally flowing fluids do not behave this way. The water molecules adjacent to the pipe wall do not move (no-slip condition); in other words, they have no velocity, and consequently no kinetic energy. However, as the molecules approach the wall, they do have kinetic energy. This kinetic energy must be converted to another form of energy as the molecule reaches the pipe wall. According to the first law of thermodynamics, some of the kinetic energy is converted to heat as the turbulent eddies dissipate, but most is converted into potential energy in the form of pressure. This can be verified by integrating the r-momentum equation from the cylindrical form of the Navier-Stokes equations at the wall [8,9]. Turbulent flow is characterized by the chaotic formation and dissipation of eddies, which cause pressure fluctuations [10]. The response of the piping structure to the pressure fluctuations is affected by several factors including the elastic modulus of the material, structural damping, structural mass, etc. It is expected that the pipe will deform in response to the pressure fluctuations; in other words, the pipe will vibrate due to the turbulent flow. This phenomenon can be experienced by placing your hand on a water faucet or hose and feeling the motion increase with flow rate.

### **Numerical**

Several commercial codes have been developed that model the interaction between the fluid and the structure (e.g. FIDAP<sup>®</sup>, ALGOR<sup>®</sup>, ADINA<sup>®</sup>, ANSYS<sup>®</sup>, STRACO<sup>®</sup>, SYSNOISE<sup>®</sup>, and IFSAS<sup>®</sup>). As sophisticated as these programs are, they still have limitations in the resolution of their flow fields. These codes use RANS based turbulent models. Such models do not compute the pressure variations at the fluid-structure interface which have been shown to contribute to the pipe vibration.

### **Analytical**

Most analytical studies use a theoretical wave perspective to analyze this phenomenon by studying the way waves propagate through a pipe when excited by an outside force. However, these studies employ potential flow theory, which does not accurately describe turbulent flow. Three of these studies include those conducted by Chuschieri and Leyrat [11] Brevbart and Fuller [6], and Gorman et al. [12]. Cuschieri and Leyrat

conducted theoretical studies on the vibrational influence of a fluid-loaded pipe flow using potential flow theory and the wave equation. The study developed an equation of motion for an infinitely long pipe influenced by a moving internal fluid. In 1993, Brevbart and Fuller analyzed the effect of internal flow on the wave propagation along an infinite cylinder using potential flow theory and the Flügge model. They showed that flow in a pipe would cause the axial wave number to change. Gorman et al. investigated the effect of annular two-phase flow on the natural frequencies of a pipe using potential flow theory and the Flügge model. They concluded that the phase in contact with the pipe has the greatest effect.

Durant et al. [4] also used an analytical perspective, but refined the previous methods when they characterized the vibroacoustic response of the pipe to random excitation by a cross-power spectral density, given as a two-dimensional integral over the domain occupied by the structure. The high velocity study concluded only a few decibels difference between a numerical prediction based on a Corcos-like model of wall pressure and experimental data. Prior to this study, Durant et al. performed similar experiments on the mass flow rate of a single component, turbulent gas using pipe vibrations [5].

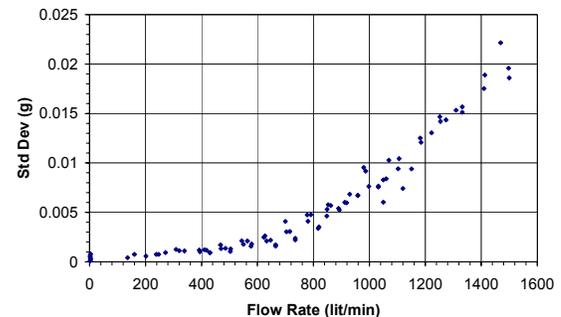
Kim and Kim [7] took another approach by using wave decomposition theory to analyze the pipe vibrations. They estimated flow rates using three accelerometers and an excitation signal on the outside of the pipe wall. Hibiki [13] noted that the flow-induced vibrations due to a two-phase mixture flowing in a loop were proportional to the gas and liquid flow rate.

In all these developments, simplifying assumptions about the flow were made. The development of a model which computes the instantaneous fluctuations has not been presented.

## Experimental

One of the first experimental studies for flow-induced vibration of a pipe due to internal flow was by Saito, et al. in 1990 [2]. They quantified their findings by plotting the RMS pressure and acceleration values against flow velocity. However, measurements were taken immediately after the fluid passed through an orifice, which altered the pipe diameter; hence, the flow was not fully developed. Also, no distinction was made between the vibration caused by the fluid impinging on the orifice and the vibration caused by the turbulence.

In 1999, Evans noted a similar relationship between flow velocity and vibration [3]. In his study, he recorded accelerometer data on the outside of a pipe carrying fully developed flow. He quantified this relationship plotting standard deviation of the time series accelerometer data against the flow rate, as shown in Fig. 1. His studies concluded that there is a strong relationship between the amplitude of the vibrations and the mass flow through the pipe. He also theorized that the vibrations were a direct result of the amplitude of the pressure fluctuations at the pipe wall. Even though Evans made efforts to eliminate all other causes of vibration, his studies are still unclear whether turbulent pressure variations alone caused the vibrations.



**Figure 1, Evans pipe experiment results (by permission)**

Every experimental case studied inherently has difficulties isolating the vibrations due to pressure fluctuations alone. Uncontrollable factors such as pump noise, clamps, bends in the pipe and irregularities in the cylindrical geometry contribute to the overall vibration sensed by the accelerometer. It is also difficult to eliminate variation in the flow rate at high velocities—which affects the fully developed nature of the flow. Consequently, it is difficult to experimentally determine the effect the turbulent eddies have on the vibrations of the pipe.

## RANS VS. LES

The general equations governing Newtonian fluid motion were independently derived by L. M. H. Navier (French engineer, 1785-1836) and G. G. Stokes (English mathematician, 1819-1903) almost 150 years ago. These equations, known as the Navier-Stokes equations, coupled with the continuity equation explicitly define the motion of a Newtonian fluid. Techniques used to model turbulent flow are

typically based on these equations. These equations can be discretized and used to numerically solve for the flow field. Techniques which use this approach are commonly referred to as Direct Numerical Simulation (DNS) methods. A major drawback with DNS is that the magnitude of the discretizations is so small that it makes large Reynolds number flows practically impossible to model and solve. For DNS models it is recommended that the number of discretizations should go as  $N \sim (10 \text{ Re}_i^{0.75})^3$  [14]. Because DNS is so computationally intensive, alternative methods have been developed. This section continues by discussing two these alternatives.

### RANS

The RANS method separates the velocity and pressure terms in the Navier-Stokes equations into time-averaged mean and fluctuating components. The governing RANS equations for incompressible flow are shown in Eqs. (1-2) [14].

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (1)$$

$$\frac{\partial}{\partial x_j} (\bar{u}_i \bar{u}_j) = -\frac{1}{\rho} \frac{\partial \bar{P}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} + \frac{\partial R_{ij}}{\partial x_j} \quad (2)$$

These equations resemble the Navier-Stokes equations except for the  $R_{ij}$  term which is known as the Reynolds stress. The Reynolds stress is a product of two fluctuating components of the flow field and is an unknown, which must be modeled [15]. A variety of empirical modeling techniques have been developed for this closure problem. However, they still only result in average quantities ( $\bar{u}_i$ ). This approach is not adequate for pipe flow since the velocities perpendicular to the streamwise direction, (i.e.  $v_r$  and  $v_\theta$ ), which are the modes of energy transfer, are zero. Even the unsteady RANS equations compute the average velocity at an instant in time and will not model the fluctuations. Since the interest is to obtain the instantaneous properties of the flow field, RANS based techniques will not provide the needed results.

### LES

In contrast to a time averaged approach, LES provides a model which computes the instantaneous velocity and pressure fields. LES solves the same Navier-Stokes equations as DNS but the equations are "spatially filtered" to the size of the grid. Filtering the Navier-Stokes equations means that the flow is resolved to a characteristic

scale, usually taken to be the size of the grid, and then modeled on the smaller scales. The motivation for this comes from the fact that large eddies possess an anisotropic behavior and need to be resolved. The smaller eddies possess a more universally isotropic behavior and like the RANS models can be treated from a statistical standpoint. Scales the size of the grid or larger are known as the grid scale (GS) and reference to scales smaller than that are referred to as subgrid scales (SGS). Typically, the grid spacing is such that most of the total turbulent kinetic energy contained in the large eddies is directly calculated [16]. The remaining fraction of the kinetic energy that is not resolved to the GS must be modeled for the flow to be physically realistic.

In LES, the instantaneous quantities are resolved to the size of the grid. Instead of the time-averaged quantities, spatially or locally averaged values,  $\tilde{u}_i$ , are obtained. The governing equations for LES flow are shown in Eqs. (3-4) [14].

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0 \quad (3)$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial}{\partial x_j} (\tilde{u}_i \tilde{u}_j) = -\frac{1}{\rho} \frac{\partial \tilde{P}}{\partial x_i} + \nu \frac{\partial^2 \tilde{u}_i}{\partial x_j^2} + \frac{\partial \tau_{ij}}{\partial x_j} \quad (4)$$

$\tau_{ij}$  is the stress tensor which represents the SGS contributions to the overall GS velocity. It is a term similar to the  $R_{ij}$  stress in RANS and is defined as the difference of the local average of the product of the instantaneous velocities and the product of the local averages as shown in Eq. (5).

$$\tau_{ij} = \widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j \quad (5)$$

$\tau_{ij}$  is modeled on the SGS and the accuracy of the model falls on the assumption that velocities smaller than the size of the grid are indeed homogeneous and accurately modeled. This results in restrictions on the grid size. A finer grid will produce a flow with minimal modeling as compared to a coarse grid. The accuracy of LES is largely a function of the resolution of the large eddies. When flows increase in Reynolds number, so does the spectrum of eddies which lends itself to finer meshes to capture all the large scale kinetic energy. When the Reynolds number increases, the amount of modeling increases. The goal of LES is to resolve most of the flow and model very little of it. Therefore, with LES there

is a trade off between computation cost and model accuracy. However, if various constraints are followed a good balance can be obtained.

As with the Reynolds stress, the SGS stress,  $\tau_{ij}$ , is modeled since there are no governing equations to compute the local average of the velocity products. It is mathematically computed by relating the subgrid stress with the turbulent viscosity and strain rate as shown in Eq. (6).

$$\tau_{ij} - \frac{1}{3} \delta_{ij} \tau_{kk} = -2\mu_t S_{ij} \quad (6)$$

In Eq. (6),  $\delta$  is the Kronecker delta,  $S_{ij}$  represents the rate of strain tensor and  $\mu_t$  is the SGS eddy viscosity.

The most common SGS eddy viscosity model is the Smagorinsky-Lilly model [17]. In this model, the eddy viscosity is proportional to a sub-grid mixing length ( $L$ ) and the strain rate as defined in Eq. (7).

$$\mu_t = \rho L^2 \sqrt{2S_{ij}S_{ij}} \quad (7)$$

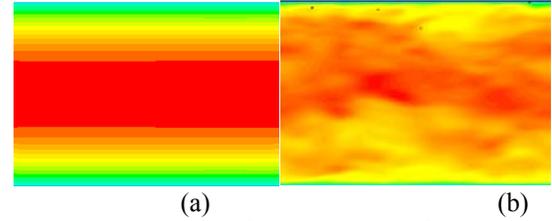
Overall, Smagorinsky's model is good for isotropic flows but usually breaks down near boundaries unless near wall treatment is employed since the contribution of turbulent viscosity at the wall should be zero. Therefore, accurately accounting for the wall boundary condition requires modifications to the mixing length. The method used in FLUENT® [18] is shown in Eq. (8).

$$L = \min \left( \kappa d, C_s V^{\frac{1}{3}} \right) \quad (8)$$

In Eq. 8,  $\kappa$  is the von Karmen constant ( $\kappa=0.42$ ),  $d$  is the distance to the closest wall,  $C_s$  is the Smagorinsky constant and  $V$  is the volume of the computational cell. In general,  $C_s=0.1$  yields the best results for a wide range of flows and will be used in this research. Recent advances in LES have focused on ways to model sub-grid scales and account for turbulent energy transport between the modeled turbulence and the calculated turbulence. These developments have mainly improved LES modeling for low Reynolds flows. Therefore, in this research, the Smagorinsky-Lilly model will be the SGS model of choice.

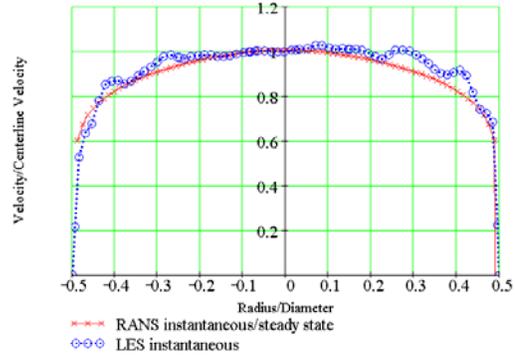
To illustrate the differences between RANS and LES, the flow field for turbulent flow in a pipe was modeled using both approaches. The velocity field results for the RANS and LES based models are shown in Figs. 2. Figure 2a shows the average nature of a RANS model while Fig. 2b

illustrates the instantaneous (non-steady) result obtained by a LES approach.

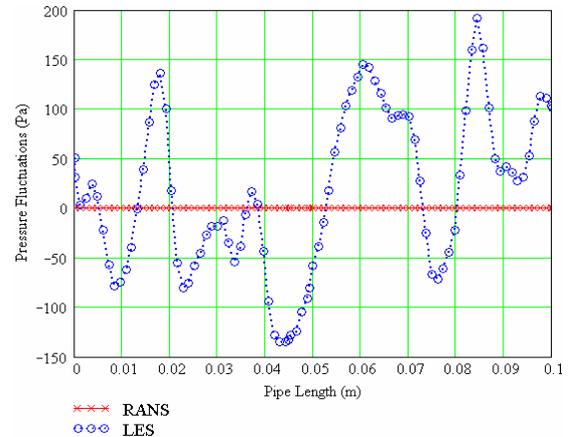


**Figure 2, Velocity field models of turbulent flow in a pipe, a) RANS based model, b) LES based model**

To further illustrate the differences between RANS and LES, a velocity profile plot illustrating the flow along a cross section of the pipe is shown in Fig. 3. A plot of the pressure fluctuations along the length of the pipe as computed by the models is shown in Fig. 4. Figures 3 and 4 distinguish between the average values computed in RANS models and the fluctuations computed in LES models.



**Figure 3, Velocity profile comparison of RANS and LES based models**



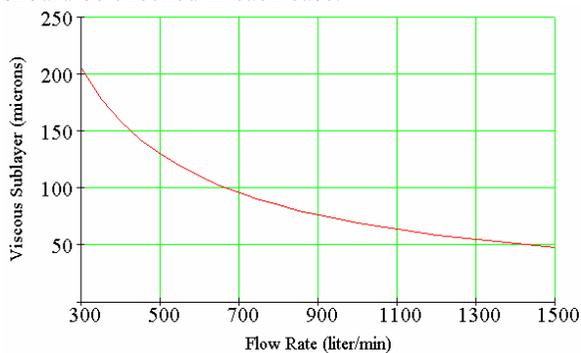
**Figure 4, Pressure fluctuations along the length of the pipe as computed by RANS and LES based models**

## SOLUTION PROCEDURE

The procedure for determining the relationship between flow rate and pipe vibration consisted of first solving the flow problem. So that this process could be repeated by others, commercial software was used when possible. The LES model of the flow field is obtained through the commercial software package FLUENT.

Once the solution to the flow field has been obtained, the pressure field on the pipe wall is periodically copied and exported to a commercially available structural finite element package known as ANSYS. ANSYS uses the pressure field solved for in FLUENT to calculate the pipe response. This is a computationally intensive process in that tens of thousands of pressure points for each time step are exported from FLUENT and mapped to the pipe model in ANSYS.

Initially it was assumed that the deflections in the pipe would change the flow field and that an update of the flow model geometry would be required between time steps in the flow solution. To verify this assumption, the displacements caused by the pressure fluctuations at the pipe wall were computed and compared to the viscous sublayer for the pipe used in the development. A plot of the viscous sublayer as a function of flow rate is shown in Fig. 5. The displacement due purely to the turbulent flow calculated using ANSYS was shown to be on the order of nanometers ( $1 \times 10^{-9}$  m). Since the viscous sublayer is much larger than the displacements caused by the turbulent flow it was assumed that a structural update of the deformed pipe geometry was not needed between time steps. Therefore, the deformed geometry is not included in the flow solution. This result could change for various pipe configurations and flow rates and should be checked in each case.



**Figure 5, Viscous sublayer thickness versus flow rate**

## RESULTS

One goal of this research is to develop a numerical based approach to determine the impact the turbulent eddies have on the vibration of a pipe with internal flow. The details and results of the flow and structural models are presented in the remainder of this section.

### Flow Model

The final LES model of fully developed turbulent flow used in the analysis includes the following fundamental characteristics:

*Pipe length (flow model) = 10cm, Dia. = 7.62cm*

*Element Number  $\approx 3 \times 10^5$ ,  $y^+ \approx [20, 250]$*

*Periodic Boundary Conditions in the streamwise direction and no slip at the wall.*

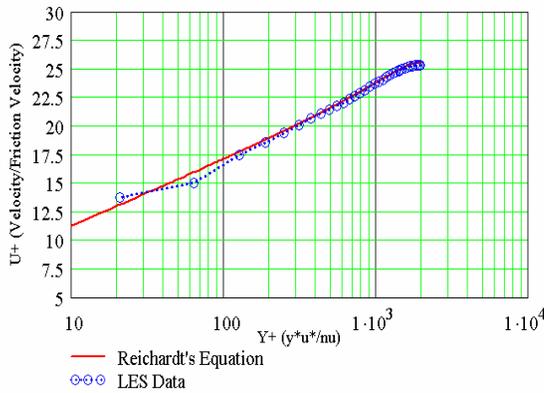
According to Egges [14], the pipe domain was suggested to be 5D with a resolution of  $y^+ < 1$  near the wall. However, the ranges of Reynolds numbers based on pipe diameter for the investigation were between 83,000 and 415,000, which would require an extremely fine grid resolution. Grid independence studies of the velocity profiles showed that using a  $y^+$  value in the lower half of the log-law layer produced adequate results and did not significantly change the fluctuating pressure fields. A 32 GB RAM, 64 processor, 400 MHz super computer was used to solve the numerical models. A  $y^+$  value of  $\approx 20$ -250 was used and required approximately 100 hours of compute time. Because the  $y^+$  value is significantly greater than one, a law of the wall profile was used to fit the data and obtain the pressure field results. Slight correlation errors were noted in the pressure or velocity fields when a 4/3D length was used as opposed to a 5D length, however, the differences were not significant enough to change the end result. The  $y^+$  and pipe domain values used are modest compared to suggested values but provide a feasible compute time and a reasonable flow model.

To provide some validation to the LES model, the time-averaged velocity profile of the LES model was compared against the well established law-of-the-wall based Reichardt equation shown in Eq. (9) where  $u$  is the velocity,  $u_\tau$  is the friction velocity,  $y^+$  is the inner coordinate,  $r$  is the radial position, and  $R$  is the

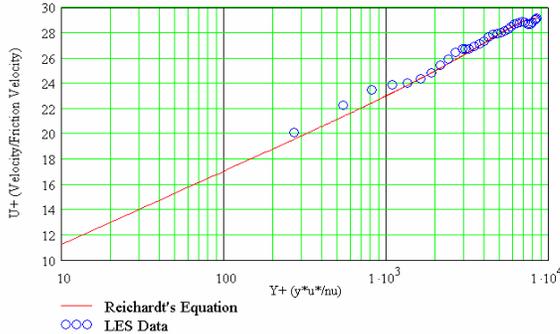
pipe radius [15]. The velocity profile comparison of the Reichardt equation and the resulting LES based model was very good as shown in Figs. 6 and 7 for the lower and upper Reynolds number ranges respectively.

$$\frac{u}{u_\tau} = 2.5 \left[ y^+ \left( \frac{1.5(1 + \frac{r}{R})}{1 + 2(\frac{r}{R})^2} \right) \right] + 5.5 \quad (9)$$

$$\text{where } u_\tau = \sqrt{\frac{\tau_{wall}}{\rho}} ; y^+ = \frac{yu_\tau}{\nu} \text{ and } y = R - r$$



**Figure 6, Comparison of the Reichardt Equation and the LES model for lower Reynolds number**



**Figure 7, Comparison of the Reichardt Equation and the LES model for upper Reynolds number**

To determine the effects of the turbulent flow on the vibration of the pipe, six discrete flow rates were investigated. These flow rates were 300, 500, 750, 1000, 1250, and 1500 lit/min. Another verification of the flow model is to compare the pressure gradients obtained numerically against the Colebrook equation and theoretical pressure gradient Eqs. (10-11). These results are summarized in Table 1.

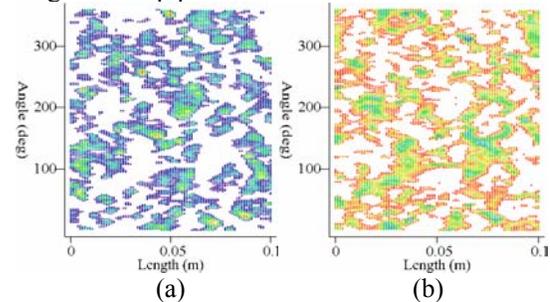
$$\frac{1}{\sqrt{f}} = -0.869 \ln \left( \frac{\varepsilon}{3.7D} + \frac{2.523}{\text{Re}\sqrt{f}} \right) \quad (10)$$

$$\frac{\Delta P}{L_{pipe}} = f \frac{\rho \cdot V^2}{2D} \quad (11)$$

**Table 1—Pressure gradient comparison**

Flow Rate (liters/min)	Theory (Pa/m)	FLUENT (Pa/m)	% Error
300	-147	-136	-7.48%
500	-368	-350	-4.89%
750	-764	-694	-9.16%
1000	-1284	-1195	-6.93%
1250	-1924	-1764	-8.32%
1500	-2679	-2460	-8.17%
Average			-7.49%

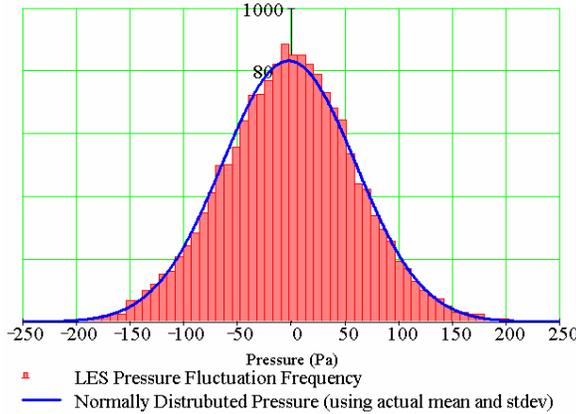
At each time step in the flow solution, the pressure fluctuations were exported for use in the structural model. The positive and negative pressure fluctuations on the surface of the pipe for one time step are illustrated in Figs. 8. This provides insight to the spatial distributions along the pipe wall. The vertical axis is the angle measurement of the pipe (i.e. the circumferential length) and the horizontal axis represents the length of the pipe.



**Figure 8, Pressure field on the pipe surface, a) positive pressure field, b) negative pressure field**

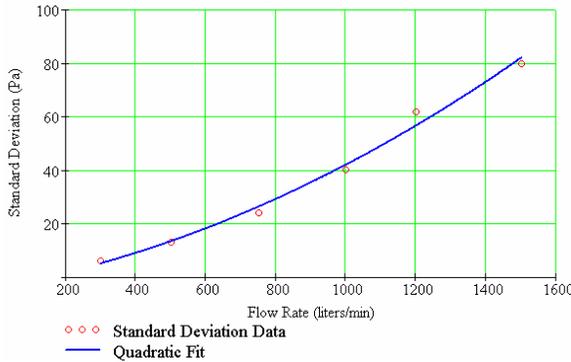
Although the pressure fields shown in Figs. 8 may appear random it has been well documented that turbulence is not a random phenomena [15,16,18]. By plotting the pressure fluctuation data on the pipe wall (Figs. 8), a near normal distribution was obtained as shown in Fig. 9. Since the main goal of the CFD simulation is to obtain the pressure fluctuations, avenues to obtain these pressures need not be limited to the CFD approach used here. Knowing that the pressure fluctuations always behave Gaussian concedes the possibility of statistically charactering the

pressure fields. Such a method could then circumvent the expensive CFD techniques.



**Figure 9, Distribution of the pressure field**

As the flow rate increases, the standard deviation of the pressure field also increases. The standard deviation of the pressure field for the six flow rates was computed and the results are plotted in Fig. 10. These points were fit with a second order polynomial. The fit equation is given by Eq. (12) where  $P_\sigma$  is the standard deviation of the pressure field and  $Q$  is the flow rate in lit/min. The  $R^2$  value for the fit was  $R^2 = 0.998$ . Since pressure is the source of energy transfer between the fluid and the structure, it is assumed that a similar relationship between flow rate and pipe acceleration exists.



**Figure 10, Standard deviation of the pressure fluctuations at a point on the pipe surface versus flow rate**

$$P_\sigma(Q) = 2.3762e-05 \cdot Q^2 + 2.01198e-02 \cdot Q - 2.477 \quad (12)$$

### Structural Model

In the flow model, fully developed flow conditions were initiated and periodic boundary conditions were used to reduce the length of the pipe and the number of elements. In the structural solution however, the entire pipe was modeled such

that the appropriate boundary conditions could be applied. Since the LES flow model was only a 10cm domain, wall pressures extracted from the model were periodically copied 11 times to cover the structural domain. Due to computational and software constraints, fifty time-steps were taken from the LES model to determine the structural response.

The final structural model consisted of the following characteristics:

$$L_{structure} = 1.1m, \quad Dia. = 7.62cm$$

$$Element \ Type = Shell \ (8 \ nodes \times \ 6 \ DOF/node)$$

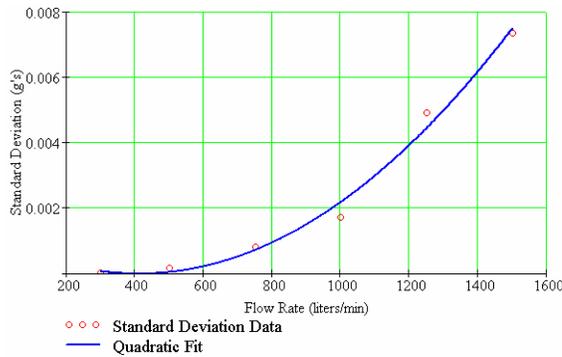
$$Element \ number \ \approx \ 11,000, \quad Node \ Number \ \approx \ 33,000$$

$$Material \ Type = AISI \ 304 \ Steel, \quad Wall \ thickness = 5.49mm$$

$$Boundary \ Conditions = Simply \ supported \ (UX, \ UY, \ UZ) \ on \ one \ end \ and \ (UY, \ UZ) \ on \ other$$

The maximum deflection of the pipe at the flow rate of 1500 lit/min was approximately 100 nanometers. This is approximately 500 times smaller than the size of the viscous sublayer and supports the idea of not updating the flow model with the deformed pipe geometry.

After the deflections for the various flow rates were determined, the acceleration of the pipe at a point was determined by computing the second derivative of the position with respect to time. The standard deviation of these accelerations were plotted against flow rate as shown in Fig. 11. This data was also fit with a quadratic expression shown in Eq. (13) ( $R^2 = 0.974$ ) where  $A_\sigma$  is the standard deviation of the acceleration in gravitational units and  $Q$  is the flow rate in lit/min. It should be noted that these acceleration values are within the measurement range and resolution of many piezoelectric accelerometers. This plot is similar to the experimental work presented by Evans [3]. However, by comparing Fig. 11 with the work presented by Evans, it should be noted that the vibration due to turbulent flow alone, becomes a more significant factor as the flow rate is increased.



**Figure 11, Standard deviation of the acceleration on the pipe surface at a point versus flow rate**

$$A_o(Q) = 6.323e-9 \cdot Q^2 - 5.171e-6 \cdot Q + 1.041e-3 \quad (13)$$

## CONCLUSIONS

A method for investigating FSI problems based on LES flow models has been presented. A model of fully developed turbulent flow in a pipe was developed to investigate the contribution of only the turbulent flow to the pipe vibration. It was determined that the turbulent flow contributes to the pipe vibration to an extent that can be measured with an accelerometer. By comparison with other experimental results, the contribution of the turbulent flow to the pipe vibration at low flow rates is a rather small component of the overall pipe vibration. However, as the flow rate is increased the turbulent flow induced vibration becomes a more significant component of the total response. From this research it was also concluded that the pressure fluctuations on the pipe wall have a quadratic relationship with the flow rate. Furthermore, it was concluded that there is a definite relationship between the acceleration of the pipe (pipe vibration) and the flow rate. These last two concepts open possible avenues for the development of a non-intrusive mass flow sensor.

## ACKNOWLEDGMENTS

Funding for this research was provided by DOE through the Idaho National Engineering and Environmental Laboratory.

## REFERENCES

[1] Weaver, D., Ziada, S., Au-Yang, M., Chen, S., Paidoussis, M., Pettigrew, M., 2000, "Flow-induced vibrations in power and process plant components - progress and prospects," *Journal of Pressure Vessel Technology*, Transactions of the ASME, **123**, pp. 339-348.

- [2] Saito, N., Miyano, H., Furukawa, S., 1990, "Study on vibration response of pipes induced by internal flow," *Pressure Vessels and Piping Conference*, ASME Pressure Vessels Piping Div Publ PVP, pp. 233-238.
- [3] Evans, R., 1999, "Two-phase mass flow measurement using noise analysis," *45<sup>th</sup> International Instrumentation Symposium, Albuquerque, NM, ISA - International Society for Measurement and Controls*.
- [4] Durant, C., Robert, G., Filippi, P., Mattei, P., 2000, "Vibroacoustic response of a thin cylindrical shell excited by a turbulent internal flow: comparison between numerical prediction," *Journal of Sound and Vibration*, **229**, pp. 1115-1155.
- [5] Durant, C. and Robert, G., 1998, "Vibroacoustic response of a pipe excited by a turbulent internal flow," *Flow, Turbulence and Combustion*, **61**, pp. 55-69.
- [6] Brevart B., Fuller, C., 1993, "Effect of an internal flow on the distribution of vibration energy in an infinite fluid filled thin cylindrical elastic shell," *Journal of the Sound and Vibration*, **167**, pp. 149-163.
- [7] Kim, Y.K. and Kim, Y.H., 1996, "Three accelerometer method for the measurement of flow rate in pipe," *Journal of the Acoustical Society of America*, **100**, pp. 717-726.
- [8] Munson, B., Young, D., Okiishi, T., 1998, *Fundamentals of Fluid Mechanics 3<sup>rd</sup> Edition*, John Wiley & Sons, Inc. New York, NY, pp. 460-496.
- [9] Panton, R., *Fundamentals of Fluid Mechanics 2<sup>nd</sup> Edition*, John Wiley & Sons, Inc. New York, NY, 1996 pp 91-125, 800-805.
- [10] Tennekes, H., Lumley, J., 1972, *A First Course in Turbulence*, The MIT Press, Cambridge, MA.
- [11] Cuschieri, J.M. and Leyrat, G.F., 1990, "Vibration transmission through a fluid loaded pipe shell with internal non zero flow velocity," *Winter Annual Meeting of the American Society of Mechanical Engineers*, ASME Pressure Vessels Piping Division Publication PVP, **202**, pp. 31-35.
- [12] Gorman, D., Liu, M., Horachek, J. "The effect of annular two-phase flow upon the natural frequencies of a pipe," *Engineering Structures*, 20 (n 8), August 1998, p 726-731.
- [13] Hibiki, T. and Ishii, M., 1998, "Effect of flow-induced vibration on local flow parameters of two-phase flow," *Nuclear Engineering and Design*, **185**, pp. 113-125.
- [14] Eggels, J., 1994, "Direct and Large Eddy Simulation of Turbulent Flow in a Cylindrical Pipe Geometry," Ph.D. thesis, Delft University, Delft, Netherlands.
- [15] Kays, W. and Crawford, M. 1993, *Convective Heat and Mass Transfer 3<sup>rd</sup> edition*, McGraw Hill, New York, New York, pp. 192-253.
- [16] Pope, S., 2000, *Turbulent Flows*, Cambridge University Press, New York, NY.
- [17] Lesieur, M., Metais, O., 1996, "New Trends in Large-Eddy Simulations of Turbulence." *Annual Review of Fluid Mechanics*, **28**, pp. 45-82.
- [18] FLUENT v5.5 *Users Manual*, Ch 9, pp. 40-60.

## References

---

1. <sup>1</sup> Koobus, B., Farhat, C., Tran, H., "Computation of unsteady viscous flows around moving bodies using the k- $\epsilon$  turbulence model on unstructured dynamic grids," *Computer Methods in Applied Mechanics and Engineering*, 190 (n11-12), December 2000, p 1441-1466.
2. <sup>2</sup> Weaver, D., Ziada, S., Au-Yang, M., Chen, S., Paidoussis, M., Pettigrew, M., "Flow-induced vibrations in power and process plant components - progress and prospects," *Journal of Pressure Vessel Technology, Transactions of the ASME*, 123 (n3), August 2000, p 339-348.
3. <sup>3</sup> Saito, N., Miyano, H., Furukawa, S., "Study on vibration response of pipes induced by internal flow," *1990 Pressure Vessels and Piping Conference*, Nashville, TN, ASME Pressure Vessels Piping Div Publ PVP, June 17-21 1990, p 233-238.
4. <sup>4</sup> Evans, R., "Two-phase mass flow measurement using noise analysis," *45<sup>th</sup> International Instrumentation Symposium*, Albuquerque, NM, ISA - International Society for Measurement and Controls, May 1999.
5. <sup>5</sup> Durant, C., Robert, G., Filippi, P., Mattei, P., "Vibroacoustic response of a thin cylindrical shell excited by a turbulent internal flow: comparison between numerical prediction," *Journal of Sound and Vibration*, 229 (n5), 2000, p 1115-1155.
6. <sup>6</sup> Durant, C., Robert, G. "Vibroacoustic response of a pipe excited by a turbulent internal flow," *Flow, Turbulence and Combustion*, 61 (n1), 1998, p 55-69.
7. <sup>7</sup> Brevart B., Fuller, C., "Effect of an internal flow on the distribution of vibration energy in an infinite fluid filled thin cylindrical elastic shell," *Journal of the Sound and Vibration*, 167, October 8, 1993, p 149-163.
8. <sup>8</sup> Kim, Y.K., Kim, Y.H., "Three accelerometer method for the measurement of flow rate in pipe," *Journal of the Acoustical Society of America*, 100 (n2 pt 1), August 1996, p 717-726.
9. <sup>9</sup> Murakami, S., Mochida, A., Sakamoto, S., "CFD analysis of wind-structure interaction for oscillating square cylinders," *Journal of Wind Engineering and Industrial Aerodynamics*, 72 (n 1-3), Nov-Dec, 1997, p 33-46.
10. <sup>10</sup> Munson, B., Young, D., Okiishi, T., *Fundamentals of Fluid Mechanics 3<sup>rd</sup> Edition*, John Wiley & Sons, Inc. New York, NY, 1998.

- 
11. <sup>11</sup> Panton, R., *Fundamentals of Fluid Mechanics 2<sup>nd</sup> Edition*, John Wiley & Sons, Inc. New York, NY, 1996.
  12. <sup>12</sup> Tennekes, H., Lumley, J., *A First Course in Turbulence*, The MIT Press, Cambridge, MA, 1972.
  13. <sup>13</sup> Cuschieri, J., Leyrat, G., "Vibration transmission through a fluid loaded pipe shell with internal non zero flow velocity," *Winter Annual Meeting of the American Society of Mechanical Engineers*, Dallas, TX, ASME Pressure Vessels Piping Division Publication PVP, Nov 25-30 1990, 202, p 31-35.
  14. <sup>14</sup> Gorman, D., Liu, M., Horachek, J. "The effect of annular two-phase flow upon the natural frequencies of a pipe," *Engineering Structures*, 20 (n 8), August 1998, p 726-731.
  15. <sup>15</sup> Hibiki, T., Ishii, M., "Effect of flow-induced vibration on local flow parameters of two-phase flow," *Nuclear Engineering and Design*, 185, 1998, p 113-125.
  16. <sup>16</sup> Evans, R., Wilkins, C., Goodrich, L., Blotter, J, *Method and apparatus for measuring the mass flow rate of a fluid*, United States Patent 6,412,352, July 2, 2002.
  17. <sup>17</sup> Ortega, H., "Technical report. Computer simulation helps predict cerebral aneurysms," *Journal of Medical Engineering & Technology*, 22 (n 4), Jul-Aug, 1998, p 179-181.
  18. <sup>18</sup> Koobus, B., Farhat, C., Tran, H., "Computation of unsteady viscous flows around moving bodies using the k- $\epsilon$  turbulence model on unstructured dynamic grids," *Computer Methods in Applied Mechanics and Engineering*, 190 (n 11-12), December 2000, p 1441-1466.
  19. <sup>19</sup> Xiong, G., Huang, P., Saunders, C., Heink, P., "CFD simulation of laser printhead - a case study on noise reduction," *Transactions of the Aeronautical and Astronautical Society of the Republic of China*, 33, March 2001, p 1-6.
  20. <sup>20</sup> Eggels, J. *Direct and Large Eddy Simulation of Turbulent Flow in a Cylindrical Pipe Geometry*, Delft University Press, Delft, Netherlands, 1994.
  21. <sup>21</sup> Davis, F., Hassan, Y., "Two-dimensional finite element method large eddy simulation for application to turbulent steam generator flow," *Nuclear Technology*, 106, April 1994, p 83-99.

- 
22. <sup>22</sup> Patankar, S., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Pub. Corp, New York, NY, 1980.
  23. <sup>23</sup> Galperin, B., Orszag, S. *Large Eddy Simulation of Complex Engineering and Geophysical Flows*, Cambridge University Press, New York, NY, 1993.
  24. <sup>24</sup> Wilcox, D., *Turbulence modeling for CFD*, DCW Industries, La Cañada, CA, 2000.
  25. <sup>25</sup> FLUENT instructional video, *Advanced Turbulence Modeling*, Corporate Training, Lebanon, NH, 2000.
  26. <sup>26</sup> Pope, S., *Turbulent Flows*, Cambridge University Press, New York, NY, 2000.
  27. <sup>27</sup> FLUENT v5.5 *Users Manual*, Ch 9.
  28. <sup>28</sup> Rumsey, C., Vatsa, V., "A Comparison of the Predictive Capabilities of Several Turbulence Models Using Upwind and Central-difference Computer Codes." *AIAA-Fluid Mechanics Division*, 1993, p. 1-16.
  29. <sup>29</sup> Rogallo, R., Moin, P., "Numerical Simulation of Turbulent Flows." *Annual Review of Fluid Mechanics* 16, 1984, p 99-137.
  30. <sup>30</sup> Lesieur, M., Metais, O., "New Trends in Large-Eddy Simulations of Turbulence." *Annual Review of Fluid Mechanics*, 28, 1996, p 45-82.
  31. <sup>31</sup> FLUENT v5.5 *Users Manual*, Ch 9, pp. 40-60.
  32. <sup>32</sup> Kays, W., Crawford, M. *Convective Heat and Mass Transfer 3<sup>rd</sup> edition*, McGraw Hill, New York, New York, 1993.
  33. <sup>33</sup> Goldin, G, FLUENT, June 2002.
  34. <sup>34</sup> Ligrani, P. M. "Structure of turbulent boundary layers." *Aerodynamics and Compressible Flows, Encyclopedia of Fluid Mechanics* (Editor: N. Chermisinoff), Vol. 8, pp. 111-189, Gulf Publishing, October 1988.
  35. <sup>35</sup> Chang, P. A. "Relationships between wall pressure and velocity field sources." University of Maryland, 1998.

- 
36. <sup>36</sup> Balling, R. *Fundamental Methods of Structural Analysis*, BYU professor's book to be published soon.
  37. <sup>37</sup> Hibbeler, R. C. *Mechanics of Materials 3<sup>rd</sup> edition*, Prentice Hall, Upper Saddle River, New Jersey, 1997.
  38. <sup>38</sup> Cook, R.D., Malkus, D.S., Plesha, M.E., *Concepts and Applications of Finite Element Analysis 3<sup>rd</sup> edition*, Wiley, New York, New York, 1989.
  39. <sup>39</sup> ANSYS website, [www.ansys.com](http://www.ansys.com), 06/03.
  40. <sup>40</sup> ANSYS v6.1. *Users Manual*.
  41. <sup>41</sup> Roark, R., Young, W., *Formulas for Stress and Strain 5<sup>th</sup> edition*, McGraw-Hill, New York, New York, 1975.
  42. <sup>42</sup> Mathcad v11, *Users Manual*.
  43. <sup>43</sup> Kuehl, R. O. *Design of experiments : statistical principles of research design and analysis*, Duxbury, Pacific Grove, CA, 2000.
  44. <sup>44</sup> Snarski, S.R. "Relation Between the Fluctuating Wall Pressure and the Turbulent Structure of a Boundary Layer on a Cylinder in Axial Flow", NUWC-NPT Technical Report 10,223, Naval Undersea Warfare Center Detachment, New London, CT, August 1993.
  45. <sup>45</sup> Pittard, M., Blotter, J. "Fluid structure interaction using LES models developed in FLUENT," *2003 FLUENT USERS GROUP MEETING*, Manchester, NH, May 2003.
  46. <sup>46</sup> Pittard, M., Blotter, J., "Numerical Modeling of LES based turbulent flow induced vibration," *2003 ASME IMECE*, Washington, D.C., November 2003.