Fluid Structure Interaction of a Duckbill Valve

Fluid Structure Interaction of a Duckbill Valve

By

Jing Wang

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2013) McMaster University
(Mechanical Engineering) Hamilton, Ontario

TITLE: Fluid Structure Interaction of a Duckbill Valve

AUTHOR: Jing Wang, P. Eng.
M. A. Sc. McMaster University, Canada
M. Eng. China North Vehicle Research Institute, P. R. of China
B.Eng. Huazhong University of Science and Technology, P. R. of China

SUPERVISOR: Dr. S. Tullis and Dr. D.S. Weaver

NUMBER OF PAGES: xviii, 126, -44-

To my father Jianyi Wang and my mother Yizhen Xie

ABSTRACT

This thesis is concerned with a theoretical and experimental investigation of a duckbill valve (DBV). Duckbill valves are non-return valves made of a composite material, which deforms to open the valve as the upstream pressure increases. The head-discharge behavior is a fluid-structure interaction (FSI) problem since the discharge depends on the valve opening that in turn depends on the pressure distribution along the valve produced by the discharge. To design a duckbill valve, a theoretical model is required, which will predict the head-discharge characteristics as a function of the fluid flow through the valve and the valve material and geometry.

The particular valves of concern in this study, which can be very large, are made from laminated, fiber-reinforced rubber. Thus, the structural problem has strong material as well as geometric nonlinearities due to large deflections. Clearly, a fully coupled FSI analysis using three-dimensional viscous flow would be very challenging and therefore, a simplified approach was sought that treats the essential aspects of the problem in a tractable way. For this purpose, the DBV was modeled using thick shell finite elements, which included the laminates of hyperelastic rubber and orthotropic fabric reinforcement. The finite element method (FEM) was simplified by assuming that the arch side edges of the valve were clamped. The unsteady 1D flow equation was used to model the ideal fluid dynamics that enabled a full FSI analysis. Moreover, verification for the ideal flow was carried out using a transient, Reynolds-averaged Navier-Stokes finite volume solver for the viscous flow corresponding to the deformed valve predicted by the simplified FSI model.

In order to validate the predictions of the FSI simulations, an experimental study was performed at several mass flow rates. Pressure drops along the water tunnel, valve inlet and outlet velocity profiles were measured, as well as valve opening deformations as functions of upstream pressures.

Additionally, the valve deformations under various back pressures were analyzed when the downstream pressure exceeded the upstream pressure using the layered shell model without coupling and with simplified boundary constraints to avoid solving the contact problem for the inward-deformed duckbill valve. Flow-induced vibration (FIV) of the valve at small openings was also examined to improve our understanding of the valve stability behaviour. Some interesting valve oscillation phenomena were observed.

Conclusions are drawn regarding the FSI model on the predictions and comparisons with the experimental results. The transient 1D flow equation has been demonstrated to adequately model the fluid dynamics of a duckbill valve, largely due to the fact that viscous effects are negligible except when the valve is operating at very small openings. Fiber reinforcement of the layered composite rubber was found to play an important role in controlling duckbill valve material stretch, especially at large openings. The model predicts oscillations at small openings but more research is required to better understand this behaviour.

ACKNOWLEDGEMENT

The author would like to sincerely thank his supervisors, Dr. David S. Weaver and Dr. Stephen Tullis, for their considerate advice, guidance, and patient encouragement throughout the entire period of this research work. Thanks also go to my supervisory committee members, Dr. Chan Ching and Dr. Dieter Stolle, for their guidance and advice.

Appreciation also goes to the department technicians, Mr. Ron Lodewyks, Jim Mclaren, Mark MacKenzie, Joe Verhaeghe, and Michael Lee. Special thanks are also expressed to Dr. Honda Wang, Mr. Allen Liu, Mr. Michael Pomeroy, and Dr. Kareem Awny Aly for their kind help and advice in the preparation of the experiments and simulations and PhD thesis writing. The author is also grateful for the FIV group members: Dr. Samir Ziada, Mr. Stephen Murry, Ms. Suzie Brown, and many more, for their questions and suggestions for his presentations and thesis.

The author gratefully acknowledges the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Ontario Centers of Excellence (OCE), Ontario Graduate Scholarship (OGS), and Elasto-Valve Rubber Products Inc. (EVR). EVR also supplied DBV samples for study and some material data and related literature.

Finally, I would like to extend my deepest gratitude to my wife, Hao Zhou, for her patience, understanding, and support. Particularly, I would like to appreciate my two sons, Ze Xi Wang and Dong Le Wang. Their excellent achievements in their schools and happiness in life have definitely been one of my power sources to accomplish my study.

TABLE OF CONTENTS

ABSTRACT.. v

ACKNOWLEDGEMENTS.. vii

TABLE OF FIGURES.. xi

TABLE OF TABLES.. xiv

NOMENCLATURE.. xv

Chapter 1 INTRODUCTION.. 1

1.1 Duckbill valve (DBV) industrial applications.. 1

1.2 Duckbill valve operating procedure.. 1

1.3 Hydrodynamic performance of duckbill valve.. 2

1.4 Objectives of present research.. 3

1.5 General arrangement of present thesis.. 4

Chapter 2 LITERATURE REVIEW.. 7

2.1 Theoretical and experimental studies on duckbill valves............................. 7

2.2 Some similar studies, such as pneumatic tire applications........................... 10

2.3 Mechanical properties of rubber and fiber materials.................................... 12

2.4 Shell and plate theories and their limitations.. 14

2.5 Ideal and viscous flow theories and their limitations.................................. 16

2.6 Fluid structure interaction.. 18

2.7 Flow induced vibration.. 22

Chapter 3 LAYERED SHELL MODEL OF DUCKBILL VALVE.................... 24

3.1 Layered fiber reinforced hyperelastic duckbill valve structure.................... 24

3.2 Mechanical properties of valve materials... 28

3.3 Layered nonlinear shell element……….. 39

3.4 Three point bending……….. 43

3.5 Layered shell model for the valve geometry….. 46

3.6 Back pressure modeling of DBV……………………………………….... 52

Chapter 4 FLOW MODELS OF DUCKBILL VALVE..................................... 55

4.1 Simplification of the fluid flow modeling.. 55

4.2 Unsteady one dimensional ideal flow model of DBV................................. 56

4.3 Viscous flow models of DBV... 62

Chapter 5 FLUID STRUCTURE INTERACTION... 70

5.1 Coupling methodology.. 70

5.2 APDL code implementation.. 73

5.3 Sensitivity analysis.. 74

Chapter 6 WATER TUNNEL EXPERIMENT... 85

6.1 Experimental apparatus.. 85

6.2 Data measurement... 89

6.3 Operation procedure.. 90

Chapter 7 RESULTS, COMPARISON, AND DISCUSSION............................ 92

7.1 Hydraulic characteristics of the DBV... 92

7.2 Pressure and velocity distributions along the DBV tunnel........................... 100

7.3 Valve deformations and layer von Mises stress distributions....................... 104

7.4 Flow induced vibration of DBV... 107

Chapter 8 CONCLUSIONS AND RECOMMENDATIONS............................. 113

8.1 Conclusions…..………………………………………................................ 113

8.2 Recommendations for future work……………………………………….... 120

REFERENCES.. 122

Appendix A: APDL Code for the Simplified FSI Model.................................... -1-

Appendix B: A Solid Valve Model Coupled with Full Viscous Flow.................. -16-
Appendix C: FORTRAN Code for the Internal/External Surfaces of DBV........ -18-
Appendix D: A Workbench Script for the deformed DBV.................................. -23-
Appendix E: A Labview Script for the Data Acquisition System....................... -43-

TABLE OF FIGURES

Figure 1-1 Duckbill valve industrial applications……………......................... 1

Figure 1-2 Duckbill valve operating procedure…………………...................... 2

Figure 1-3 Hydrodynamic performance of duckbill valve.……………….......... 3

Figure 2-1 Simulation approaches of fluid-structure interaction problems......... 19

Figure 2-2 Dynamic meshes…………………………………………………. 20

Figure 2-3 Coupling approach of ANSYS/CFX ……………………................ 22

Figure 3-1 Duckbill valve geometry………………………………................ 25

Figure 3-2 Layered rubber material reinforced with cross plied fiber layers...... 26

Figure 3-3 Thickness measurement of a duckbill valve……………………….. 27

Figure 3-4 Uniaxial tension results of SBR, CR, and EPDM ………….......... 30-31

Figure 3-5 Loading curves extracted from the uniaxial tension tests…………. 31

Figure 3-6 The 2p and 3p Mooney-Rivlin models compared with uniaxial tension tests……………………………………………………………….. 33-34

Figure 3-7 Unidirectional fiber reinforced resins matrix.................................... 35

Figure 3-8 Warp/weft tension curves of the unidirectional fiber/resin matrix…. 36

Figure 3-9 Four-node nonlinear shell element…………………........................ 40

Figure 3-10 An FEA model of three point bending……….............................. 43

Figure 3-11 Layered shell model and solid model for the bending simulation… 44

Figure 3-12 Three-point bending test……………………………………….... 44

Figure 3-13 Flexural stress-strain relationships………………………........... 45

Figure 3-14 Simplification and generation of duckbill valve geometry………... 47

Figure 3-15 Comparison of local mesh quality……………………………..... 48

Figure 3-16 Different constraints of duckbill side edge……………………….. 49
Figure 3-17 Comparison of different duckbill side edge boundary conditions... 49
Figure 3-18 Finite element meshes and boundary conditions of DBV shell model…………………………………………………………….……….. 50
Figure 3-19 Duckbill stopper constraint in the coupled model……………...... 51
Figure 3-20 Back pressure modeling of DBV……………………………..... 52
Figure 3-21 Valve shell deformations under various back pressures………….. 54
Figure 4-1 Transient 1D ideal flow model……………………..…................ 56
Figure 4-2 Schematic diagram of the 1D dynamic meshes..….......................... 58
Figure 4-3 Reconstruction of top and bottom surfaces……............................... 67
Figure 4-4 Three deformed valve geometries for viscous flow modeling......... 68
Figure 4-5 Transient viscous flow model for the 29kPa case…..……………...... 69
Figure 5-1 Coupling procedure of APDL code……………………................. 72
Figure 5-2 APDL code structure... 73
Figure 5-3 Mesh density independence.. 76
Figure 5-4 Time step independence.. 76
Figure 5-5 Influences of Beta damping on the solutions................................... 78
Figure 5-6 Influences of Alpha damping on the solutions……………………..… 79
Figure 5-7 Influence of numerical damping on solution convergence............... 79
Figure 5-8 Influence of stress softening phenomenon (Mullins's effect)……… 81
Figure 5-9 Sensitivity of valve wall thickness to the flow rate……….......... 82
Figure 5-10 Comparison between the unsteady flow model and steady flow model………………………………………………………………….. 83
Figure 6-1 Schematic diagram of 12” diameter water tunnel system................ 85
Figure 6-2 Inlet velocity profiles... 87
Figure 6-3 Photo of test section of DBV.. 88

Figure 6-4 Deformation measurement of the duckbill opening.......................... 90
Figure 7-1 Strain level estimation using the ratio of arc length and chord length……………………………………………………………..………. 93
Figure 7-2 Head-discharge relations of the DBV... 94
Figure 7-3 Influence of fiber orientations on the valve head-discharge performance………………………………………………………………….. 95
Figure 7-4 Pressure drop-discharge relationships……………………………..... 96
Figure 7-5 Area discharge relationships………... 98
Figure 7-6 Outlet velocity profiles of measurements, 1D and 3D predictions.. 99
Figure 7-7 Pressure drop-velocity relations……... 100
Figure 7-8 Velocity fields in the y-z planes of valve.. 102
Figure 7-9 Pressure distributions along the central line of valve........................ 103
Figure 7-10 Deformation shapes of exit area... 104
Figure 7-11 Nodal displacement contour of shell deformation........................... 105
Figure 7-12 Effective stress distributions on each layer of laminate shell structure.. 106
Figure 7-13 Mass flow rates vs. simulation time.. 107
Figure 7-14 Time step independence.. 109
Figure 7-15 Mass flow rate oscillation of duckbill valve................................... 109
Figure 7-16 Centerline node y-displacements of the valve model…................... 110
Figure 7-17 FIV case with mesh and time step independences…........................ 111

Appendix:

Figure B-1 Dynamic meshes of the viscous flow model of the DBV.................. -17-
Figure B-1 A solid valve model coupled with a viscous flow model.................. -17-
Figure E-1 Block diagram of the pressure signal acquisition system................ -43-
Figure E-2 Front panel of the pressure signal acquisition system...................... -44-

TABLE OF TABLES

Table 3-1 Composite materials of duckbill valve.. 25

Table 3-2 Uniaxial tension test of rubber materials………………………………… 29

Table 3-3 Coefficients for the 2p and 3p Mooney-Rivlin models…....................... 33

Table 3-4 Warp and weft tension tests……………... 35

Table 3-5 Orthotropic properties of FIBER ±54.. 38

Table 5-1 Alpha and Beta damping estimations…………………………………... 78

NOMENCLATURE

$\tilde{C}^{ijkl}$ – Fourth-order contravariant constitutive tensor in the converted coordinates r_i.
$^{l}x_i$ – Cartesian coordinates of any point in the element
$\tilde{\varepsilon}_{ij}$ – Covariant components of the infinitesimal strain tensor
$\tilde{\tau}^{ij}$ – Contravariant components of the Cauchy stress tensor
$^{t+\Delta t}_{0}\tilde{S}^{ij}$ – Contravariant components of the second Piola-Kirchhoff stress tensor
$^{t+\Delta t}_{0}\tilde{\varepsilon}_{ij}$ – Covariant components of the Green-Lagrange strain tensor
$^{l}V^{k}_{ni}$ – Components of director vector at node k
$^{l}x^{k}_{i}$ – Cartesian coordinates of nodal point k
$^{t+\Delta t}\mathfrak{R}$ – External virtual work
u^{k}_{i} – Nodal point displacements into the Cartesian coordinate directions
$[^{t}_{0}K_{L}]$ – Linear strain incremental stiffness matrix
$[^{t}_{0}K_{NL}]$ – Non-linear strain incremental stiffness matrix
$[^{t}_{0}F]$ – Nodal point force matrix
$[^{t+\Delta t}R]$ – External virtual work matrix
$[C]$ – Structural damping matrix
$[K]$ – Structural stiffness matrix
$[M]$ – Structural mass matrix
$[\Delta u]$ – Incremental displacement matrix
$\{\ddot{u}\}$ – Nodal acceleration vector
$\{\dot{u}\}$ – Nodal velocity vector
$\{A_e\}$ – Area vector of shell element representing the element's orientation in x, y, z
$\{F\}$ – External load vector
$\{u\}$ – Nodal displacement vector
$A(z)$ – Area function
A_1 – Inlet area or Cross-sectional area of valve outlet
A_{1D} – Cross sectional area of the valve tunnel
A_2 – Outlet area or Cross-sectional area after the expansion
A_{3D} – Projected area of the shell curve in the x-y plane
A_{db} – A_W at the duckbill portion
A_{fg} – A_W at the round flange portion
A_{in} ,A_{out} – Section area of inlet and outlet
a_k – Shell thickness at node k, measured along the vector $^{l}V^{k}_{n}$
A_k , A_{k-1} – Cross sectional areas at node k and k-1

A_W – Half cross section area of the valve wall

A_z – Cross sectional area at any position z

b – Width of test beam

C_{10}, C_{01}, C_{11} – Parameters of Mooney-Rivlin model

d – Depth of tested beam

D – Maximum deflection of the center of the beam

dp – Incremental pressure change at the chosen point

ds – Element of distance along the streamline

E – Young's modulus

F_i – Pressure recovery coefficient between Position 1 and Position i (from2 to 7)

G – Shear modulus

H_{arc} – Arc height

$h_k(r_1,r_2)$ – Two dimensional interpolation functions corresponding to node k

I_1, I_2, I_3 – Strain invariants

k – Turbulent kinetic energy

K – Experiment dependent loss coefficient, a function of expansion ratio of A_1 to A_2

K_i – Ideal pressure recovery coefficient, which is equal to $1-(A_1/A_2)^2$

L – Support span

L_1 – Length of ruler scale

L_2 – Length of the ruler scale outside the pipe

L_3 – Thickness of the fitting where the ruler scale is assembled

L_{arc} – Arc length

L_{chord} – Chord length

M – E, G, or ν of material components

N – Total number of the shell nodes on the curve

P – Load at a given point on the load deflection curve

p – Arbitrary hydrostatic pressure

$P(z)$ – Pressure function along valve tunnel

p_1 – Pressure at valve outlet

p_2 – Pressure after the expansion

P_i – Pressure within tunnel

P_{in}, P_{out} – Pressure at valve inlet and outlet

P_s – Static pressure

P_t – Total pressure

P_{up} – Pressure at upstream inlet of tunnel

P_{vena} – Pressure at vena contracta

Q – Mass flow rate

R – External radius of the test section pipe

r_i – Natural coordinates
S – Offset distance
S_{ij} – Shear strain tensor
S_k – Unsteady term of the 1D flow equation at node k
t, t_i – Time, i=1,2,3,4,5,6
U – Fluid flow velocity at a point on a streamline
U_1 – Inlet velocity
U_2 – Outlet velocity
U_i − Inlet velocity
U_k , U_{k-1} – Velocities at node k and k-1
U_z – Velocity at any position z
V – Velocity
v_1 – Mean flow velocity at valve outlet
V_{in}, V_{out} – Velocity at inlet and outlet
W – Strain energy potential
W_s – Potential of the external loads
x, y, z – x,y,z coordinates
x_{sk} , y_{sk} – x and y coordinates of the skth shell element node on a deformed shell curve
z_{db} – z-position of A_W at the connection of duckbill and saddle
z_{fg} – z-position of A_W at the connection of saddle and flange
z_k – z-position of the kth 1D flow node
z_k , z_{k-1} – z-locations of node k and k-1
z_{sd} – z-position of A_W in the saddle portion which is between z_{db} and z_{fg}
z_{sk} – z-position of the skth shell element node on the kth deformed shell curve
α – Installation angle of ruler scale
α, β – Alpha damping and Beta damping are used to define Rayleigh damping constants
α_k, β_k – Rotations of the director vector ${}^{0}\mathrm{V}_{\mathrm{n}}^{\mathrm{k}}$ about the ${}^{0}\mathrm{V}_{1}^{\mathrm{k}}$ and ${}^{0}\mathrm{V}_{2}^{\mathrm{k}}$ axes
δ_{ij} – Kronecker delta
ΔP – Pressure difference
Δp_k – Static pressure distribution along the valve wall
ε – Turbulent kinetic energy dissipation
ε_f – Strain in the outer surface, (mm/mm)
ε_i – Engineering strains, i=1,2,3
λ_1,λ_2, λ_3 – Principle extension ratios
μ_t – Turbulent viscosity
ν – Poisson ratio
ν_t – Turbulent eddy viscosity
ξ – Adjustable coefficient

ξ_i – Ratio of actual damping to critical damping for a particular mode of vibration, i.
ρ – Fluid density
$\sigma_1, \sigma_2, \sigma_3$ – Principal stresses, i=1,2,3
σ_f – Stress in outer fibers at midspan
τ – Period of cycle
τ_{ij} – Turbulent shear stress tensor
φ – Corresponding constant of the blended k-ω/k-ε model or Volume fraction
φ_1 – Any constant in the original k-ω model in the inner layer
φ_2 – Any constant in the transformed k-ε model in the outer layer
ω – Specific rate of dissipation
Ω – Absolute value of the vorticity.
ω_i – Natural circular frequency of mode i

scribe:

1d, 3d – One and three dimensions
db – Duckbill
e – Element
f , m – Fiber and resins matrix
fg – Flange
i, j, k, l – Indices
in, out – Inlet and outlet
L – Linear
NL – Nonlinear
t – Turbulence
$t\text{-}\Delta t$ – previous time steps
up – Upstream
vena – Vena contracta
W – Wall
x, y, z – x, y, z directions
xy, xz, yz – xy, xz, yz planes

Chapter 1

INTRODUCTION

1.1 Duckbill valve (DBV) industrial applications

There are extensive applications of variable opening duckbill valves in industrial piping systems, such as pumping stations, site drainage, sewer systems and so forth. Fig. 1-1 shows a side view of a row of duckbill valves, through which water is seen to flow into a downstream water body, as well as a fully closed flood control system, viewing from the duckbill ends. These valves can also be applied to submerged opening or inline piping systems. They are intended to prevent flow from occurring in the reverse direction.

Fig. 1-1 Duckbill valve industrial applications

(http://www.redvalve.com)

1.2 Duckbill valve operating procedure

A duckbill valve is a flexible sleeve design made of fabric-reinforced rubber material.

Its typical geometry consists of a short piece of duckbill-like rubber tube at one end, a round flange (or cuff) typically clamped onto a pipe port or pipe flange at the other, and a saddle shape (or taper) at the middle. Fig. 1-2 illustrates the valve operating procedure. When the upstream pressure is not larger than that downstream, the flattened lips remain closed and there is no flow through the valve. When the pressure head upstream increases (larger than a critical value), the valve opens, and the size depending on how much the flow rate goes up. A reverse flow is not allowed through the valve.

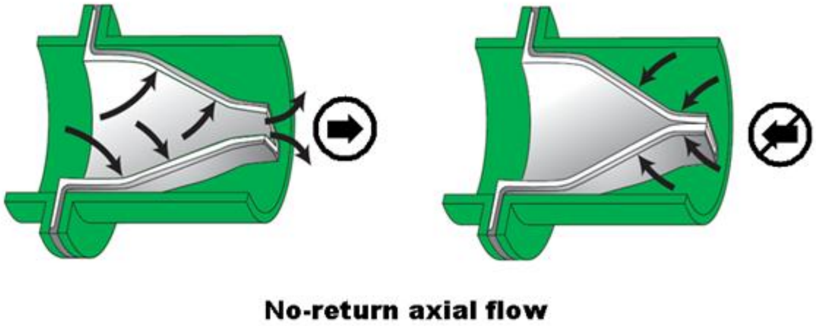


Fig. 1-2 Duckbill valve operating procedure

(EVR Products, CPI Series Specification)

1.3 Hydrodynamic performance of duckbill valve

In general, the duckbill valve is essentially a non-return axial flow valve that is sensitive to the fluid flow. Its operation depends on the variable valve opening. The valve behaviour is a fluid-structure interaction (FSI) problem. Experiments have shown that the valve jet velocity jumps up quickly with increasing flow rate at the initial opening of the duckbill valve. High velocities can be maintained reasonably over a very large range of flow rate while the head-discharge relation is approximately linear (EVR Products, CPI Series Specification, see Fig. 1-3).

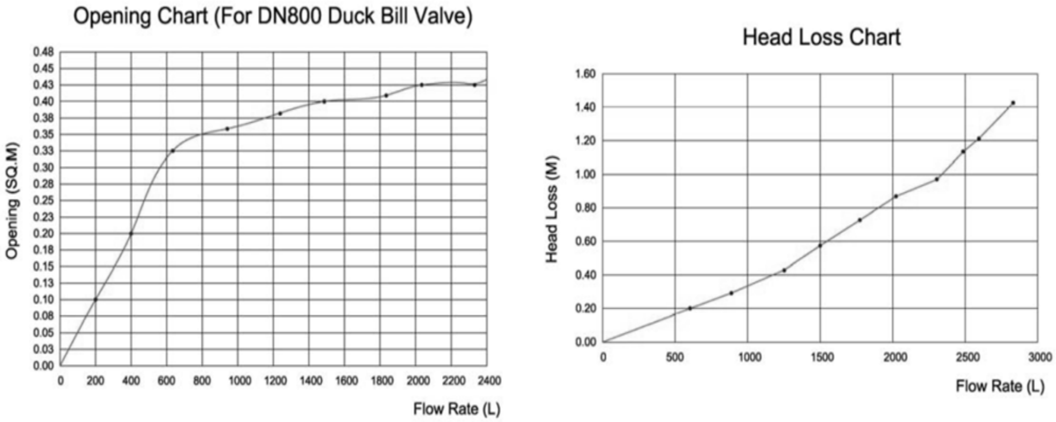


Fig. 1-3 Hydrodynamic performance of duckbill valve

(EVR Products, CPI Series Specification)

1.4 Objectives of present research

Currently, a tractable theoretical approach to predict the hydraulic performance of a duckbill valve does not appear to exist due to technical difficulties involved in solving the fluid-structure interaction problem.

In the first place, fluid-structure interaction can be regarded, not only as a phenomenon occurring between fluid and structure, but also as a technical approach that tries to combine the fluid and structural dynamics together. Therefore, it is often too complex to solve in closed-form. Numerical methods integrating both computational fluid dynamics (CFD) and finite element analysis (FEA) are far from mature and typically cannot solve fluid-structure interaction problems in many engineering applications.

Secondly, there are difficulties associated with practical application skills. For example, expert experience is required to generate high quality CFD and FEA meshes for a flexible duckbill valve with a geometric singularity (fully closed duckbill lips). Additionally, for reconstruction of deformed geometries used for CFD/FEA simulations, a user-defined code of computer aided design (CAD) is frequently needed to generate complex curved surfaces from the known deformed geometries that must satisfy acceptable geometry accuracy for a meaningful comparison of simulations.

Thirdly, the complexity of the duckbill valve material and the strong nonlinearity of geometric deformation have to be solved numerically and experimentally. At present, most studies treat the valve behavior either as linear elastic material coupled with fluid flow or as nonlinear elastic deformation but without any fluid coupling.

Finally, related research studies on duckbill valves are sparse. Manufacturers often rely on past experience to design composite valves. Some small prototypes may be made for water tunnel testing, but the whole procedure often takes several months, that cannot meet the time requirement of customers. Furthermore, it is impractical to test very large size valves and there seems to be a discrepancy between the published testing reports associated with various composite designs.

In order to better understand the FSI mechanism of duckbill valves, the author has developed coupling simulation methods to predict the DBV characteristics and has conducted experiments to validate them. Thus, the objectives of the research has been to

1) Improve understanding of duckbill valve mechanics
2) Develop a tractable theoretical model to predict valve performance and which can be used for valve design
3) Validate the model experimentally

1.5 General arrangement of present thesis

This thesis consists of eight chapters. The first chapter is the introduction, which gives the motivation for the research and provides an overview of the thesis. The second chapter, a literature review, introduces publications investigating duckbill valve behavior and related research results. The third chapter introduces a specific duckbill valve consisting of various rubber layers with fiber reinforcement in between, describes related material models and tests, compares a layered shell model with a layered solid model using some simpler simulations, discusses influences of mesh and boundary issues on the duckbill valve predictions, and finally presents a middle-surface-based-layered-shell (MSBLS) theory for a duckbill valve. Chapter 4 presents a transient one-dimensional ideal flow model for valve flow and a full viscous flow model to determine the effects of neglecting viscosity in the simple ideal flow model. Chapter 5 discusses in detail a fluid structure interaction methodology of a layered hyperelastic shell model and 1D flow model, as well as its implementation using the computer language, ANSYS parametric design language (APDL). Sensitivity analysis of the coupled model, including independence of time step and mesh density, influence of structural and numerical damping, etc. are presented in this chapter. In the sixth chapter, a water tunnel experiment for 12" inline duckbill valves is described that was used to validate the predictions of the coupled model. Chapter 7 presents the theoretical and experimental results on duckbill valve performance, compares the valve hydraulic characteristics predicted by the FSI model with corresponding testing results, discusses pressure and velocity distributions of fluid flow through the valve, as well as measured and predicted valve deformations, and analyzes the influence of back pressure on the valve deformation and flow induced vibration of duckbill valve at small opening. The eighth chapter contains conclusions drawn on the theoretical and experimental results from this research and recommendations for future work.

Appendix A lists the ANSYS parametric design language (APDL) code developed for the middle-surface-based-layered-shell (MSBLS) model that is coupled with 1D potential flow for the present research. In appendix B, a full FSI model coupled with viscous flow is described, where a homogenous solid model is employed. However, this fully coupled model fails to predict the duckbill valve performance for several reasons. Appendix C lists a FORTRAN code for seeking external and internal surfaces from the deformed middle surface of the duckbill valve. A Workbench user defined subroutine is listed in appendix D for viscous flow modeling using a deformed valve shaped predicted by the coupled model. Finally, a Labview code for water tunnel data acquisition system is presented in appendix E.

Chapter 2

LITERATURE REVIEW

In this chapter, a literature review on duckbill valves and related studies is presented. The review includes theoretical and experimental investigations on DBV, corresponding industrial applications, material mechanical properties, related fluid and structure theories, flow induced vibration (FIV) and FSI problems, etc.

2.1 Theoretical and experimental studies on duckbill valves

Although duckbill valves have been widely used in various industries for many years, research on modeling the valves is unexpectedly lacking. Some investigations are in the form of internal reports that are not easily found for academic purposes. The design implications and practical advantages of duckbill valves are discussed by Duer (1998). Essentially, the application of flexible orifice duckbill valves can improve the hydraulic performance of marine diffusers, generate less head-loss at peak flow while keeping sufficient jet velocity at low flows, and optimize wastewater flow to maintain a flatter outflow distribution since the random wastewater flows often vary widely in flow rate range. Gao et al. (2004) published a review to summarize the investigations on duckbill valve jet flows in the past. They argued that the mixing effect of the jet flows of duckbill valves should be evaluated since round jet flow results cannot be directly applied for the non-round case. Gao, et al., (2005) also conducted a series of experiments to study the mixing features of duckbill valve jets in a co-flow. They concluded that the duckbill valve jet has a better effect on dilution compared with a round jet under a same situation. However, these authors focused their interests on the valve flows and their influence on the downstream tailwater. The fluid structure interaction mechanics of the duckbill valves, which play a key role for valve design, is still not well understood.

Lee et al. (2004, 2002, and 2001) published three papers using analytical and experimental methods to investigate the characteristics of a duckbill valve. Lee, et al. (2004) were the first to develop a very simple analytical method to predict the hydraulic performance of a duckbill valve. In their theory, a duckbill valve was considered as a smooth converging nozzle. By modeling the duckbill valve as a linear elastic rubber membrane coupled with a steady 1D potential flow model, the deflection and hydraulic performance of the valve under given pressure drops were calculated. However, the influence of laminated layer structure of the valve materials with fabric reinforcement was excluded in their theory. Therefore, the nonlinearity of the rubber and the orthotropic features of the fiber were not evaluated in their study. Also, Lee, et al. (2002) conducted a CFD simulation as well as jet velocity field measurements to investigate duckbill valve jet flows. The CFD simulation corresponded to a static downstream flow case with a standard k-ε turbulence model. The fluid structural interaction of the duckbill valve was not considered. Furthermore, the k-ε turbulence model generally failed to predict flow near a solid boundary due to a lack of viscous corrections. Lee et al. (2001) carried out an FEA study on the relation of large elastic deformation to the flow variation of a duckbill valve. The valve was modeled by 224 20-node brick elements. The pressure load of steady 1D ideal flow inside the duckbill valve was applied as an inner surface boundary condition of the FEA simulation. The duckbill valve material was assumed to be linearly elastic and no fabric reinforcement was modeled. They argued that the valve deformation depended mainly on the mechanics of the rubber deformation, and only secondarily on the fabric reinforcement and upstream connection. This assumption has yet to be justified. In addition, the influence of fluid viscosity on the pressure and velocity fields was not compared with their predictions of their 1D potential flow model, even though their simulation results were reported to be in good agreement with their experimental data for hydraulic performance.

To better understand the duckbill valve materials, Chouchaoui (2001) conducted a series of tests on the valve composite materials to obtain some necessary parameters for hyperelastic rubber models and orthotropic fiber models. This study included uniaxial tension along with planar tension and volumetric compression on the rubber compounds used to manufacture duckbill valves. Additionally, yarn (tire cord) and resins were tested for stress-strain data of woven fiber reinforcement. This study indicated that the rubber and fiber material properties must be taken into account for modeling the performance of a duckbill valve. A very simple solid model was also developed based on the results from these material tests. The nonlinearities of the valve geometry and material were considered in the model. However, the solid model only worked within a very narrow pressure range (up to only 1.67 kPa) and had no coupling with the fluid flow. Therefore, it was not useful for practical applications.

The author (2012) developed a more realistic coupled model of a duckbill valve as part of the research for this thesis. The valve was considered as a bottom surface based laminated thick shell structure using non-linear 4-node layered shell elements (Dvorkin, 1984 and ANSYS and CFX user's manual, 2008) to accommodate the first order transverse shear effects on the layered shell and to simplify boundary conditions along the duckbill edges. The hyperelastic behavior of the rubber and orthotropy (or anisotropy) of the fiber reinforcement were included, as were the large valve deformations. The flow was modeled as steady 1D idea flow, similar to that used by Lee, et al. (2001). This model was beneficial to determine the deformed valve shape for coupling the 1D flow, since the bottom surface was just the interface between the fluid and solid domains. However, its mesh quality became relatively poor, especially at the connection between the saddle and duckbill portions, which limited the driving pressure range from 5 kPa to 20 kPa. This work however demonstrated the important effects of the fabric layer at higher pressure drops.

2.2 Some similar studies, such as pneumatic tire applications

Outside the direct investigations on duckbill valves, investigations dealing with fiber-rubber composite have been completed for industrial tire applications. Brewer (1973) developed a shell model from composite theory for the prediction of tire stress and deflection. In his model, the pneumatic tire was treated as a laminated, anisotropic, toroidal thin shell of revolution including bending. The classical Kirchhoff-Love's hypothesis was applied, which states that straight lines normal to the undeformed middle surface remain straight and normal to the middle surface after deformation and that the thickness doesn't change during deformation. His theory focused on dealing with the plies consisting of elastic textile cords embedded in an elastic rubber matrix based on the assumption that the basic load was carried by the fiber (cord). The composite properties of each single lamina were empirically expressed in terms of those of the fiber and rubber. The numerical calculation was reported to show good agreement with measurements. Regardless, the Kirchhoff-Love's hypothesis ignores the shear stress through the shell thickness. In the duckbill valve shell model, the shear stress cannot be ignored due to the thick valve wall. Nicholson (1975) employed membrane theory to analyze the contributions of structure and pneumatics to tire behavior when a tire is exposed to vertical external loads. The tire was simplified as a linear elastic inflated ring under compression between rigid, flat, parallel plates. A finite element contact problem between the tire and rigid plates was considered. The membrane theory considers in-plane force only which is even simpler than the Kirchhoff-Love shell model. Therefore, it is not suitable to apply for modeling duckbill valves.

Kaga, et al., (1977) conducted a stress calculation of a tire under vertical loads by using FEA. The tire simulation was treated as a nonlinear axisymmetric problem. The geometric nonlinear behavior was approximated by a sequence of linearized solutions. Detailed components of tire structure, such as tread, textile carcass, inner liner, bead, and side wall, etc., were considered, as well as the orthotropic properties of fiber and the Mooney-Rivlin model for rubber. However, the tire was basically a 2D geometry model (an axisymmetric model).

Tielking (1984) presented a finite element tire model, in which the tire was regarded as a nonlinear four-ply shell of revolution. A contact formulation was used to model the tire deformation under the vertical load of a rigid road surface. Although the nonlinear laminate shell was applied, the shear stress through the tire wall thickness was still ignored to simplify the contact model. Miller, et al., (1985) described a mathematical model to evaluate the mechanical behavior of an inflated tire. The textile mechanics of each ply was considered but the laminated structures were bonded as a whole membrane for stress analysis. The nonlinearities of geometry and material for each ply were taken into account in this model. Cord tension and interplay shear stresses were calculated and discussed in their paper.

In short, the tire models can be grouped into three categories: membrane models; shell models; and 3D structural models. It is noted that fluid structure interaction is not the major interest of tire applications.

2.3 Mechanical properties of rubber and fiber materials

2.3.1 Theoretical studies for rubber and fiber materials

Although FEA is a powerful tool for simulating product performance, this technique relies, in particular for rubber applications, on experimentally determined constitutive laws, among which the Mooney-Rivlin model is one of the most widely applied rubber constitutive relations available in most FEA material libraries. Mooney (1940) firstly developed a theory of large elastic deformation deriving a relationship between strain invariants and a stored energy function for rubber like materials. The predictions followed closely the experimental data on rubber from 400% elongation and 50% compression. Rivlin (1948) derived a set of stress-strain relations for a highly elastic material in terms of a stored energy function and strain invariants, and designed a material experiment to validate his hyperelactic theory. Treloar (1958) discussed the Mooney-Rivlin model in detail and expressed it in a succinct way as follows:

$$W = \sum_{i=0,\, j=0}^{N} C_{ij} (I_1 - 3)^i (I_2 - 3)^j \tag{2.1}$$

where C_{ij} are empirical coefficients, N , i and j are integers, and I_1 , I_2 and I_3 are strain invariants defined by

$$\begin{aligned} I_1 &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\ I_2 &= \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2 \\ I_3 &= \lambda_1^2\lambda_2^2\lambda_3^2 = 1 \end{aligned} \tag{2.2}$$

The principal extension ratios (stretch) are defined as $\lambda_i=1+\varepsilon_i$, ($i=1, 2, 3$), in which ε_i is the principal value of the engineering strain tensor in the ith direction. In terms of principal stress

$$\sigma_i = \lambda_i \left(\frac{\partial W}{\partial I_1} \frac{\partial I_1}{\partial \lambda_i} + \frac{\partial W}{\partial I_2} \frac{\partial I_2}{\partial \lambda_i} \right) - p \tag{2.3}$$

where σ_i is the principal stress and $i=1,2,3$, p is an arbitrary hydrostatic pressure.

Other rubber models include Neo-Hookean, Yeoh, Ogden, and so forth. These models, including the Mooney-Rivlin model, are phenomenological models that use the strain energy functions (different models have different but generally similar stain energy functions) and strain invariants to describe the various rubber deformation behaviors. These models are designed for different rubber materials and are applicable over different strain ranges, such as Neo-Hookean for 0 to 30% strain, Yeoh for 30% to 200%, and Ogden for up to 700%.

Some papers focused on the mechanical property estimation of fiber reinforcement, which is mandatory for material constitutive relations in the rubber material reinforced by fiber. Halpin, et al., (1976) reviewed the Halpin-Tsai equations based on the so called "self-consistent micromechanics method" to obtain the modules of a variety of composites. If the modules of filament and rubber matrix, where filaments are embedded, are known, the effective modulus of the composite material can be calculated using the Halpin-Tsai equations. Hermans (1967) also developed an approach to predict the elastic properties of fiber reinforced materials and compared it with the Halpin-Tsai equations. Tucker, et al., (1998) reviewed and evaluated the stiffness predictions for unidirectional fiber composites. Seven kinds of micromechanical models, including the Halpin-Tsai equations, were discussed. The predictions of these models and related FEM results to a short-fiber composite material were compared in their paper. They concluded that the Mori-Tanaka model was more accurate than the Halpin-Tsai equations only in prediction of the stiffness of short-fiber composites.

2.3.2 Experimental studies for fabric reinforced rubber materials

Clark (1960) developed a theory to describe the stress-strain relation of two cross-ply cord-rubber structures, which made up a tire carcass. A concept of inter-ply shear stresses was introduced to accommodate the rotation of cords with respect to one another between deformed laminates of cord-rubber sheets. To validate his theory, he carried out a set of tests on a tubular specimen, which showed that his coherent theory delivered good predictions in stress modules, Poisson’s ratio, load-circumferential strain relation, and torque-angle relation, etc. except shear modulus. He attributed the discrepancy in the shear modulus to the intricacy of shear modulus measurement.

Posfalvi (1976) employed an interesting simple virtual work method to deduce the effective elastic modulus and Poison’s ratio of cord-rubber composites. The constants for the orthotropic stress-extension ratio and Poisson’s ratio expressions were obtained by uniaxial tension and compression testing. Fiber angles were varied from 0 ° to 90 ° to represent the different orthotropy. Chow, et al., (1987) conducted biaxial tension tests of the mechanical properties of vulcanized rubber to obtain the constants of multi-order Mooney-Rivlin and Valanis-Landel formulations based on the elastic strain energy function expressions.

Additionally, Wadham-Gagnon, et al. (2006) proposed a method to approximate the stress-softening effects (Mullin's effect) of rubber using ANSYS hyperelastic models and conducted a cantilever plate bending test to validate these models at moderate strain (0-25% engineering strain). Axel Product Inc. (2012) provided them with the results of standard tests including uniaxial tension, equal-biaxial extension, and planar tension tests for curve fitting to obtain the constants of these rubber models.

2.4 Shell and plate theories and their limitations

Since a duckbill valve can be regarded as a layered, orthotropic, fabric reinforced, shell/plate undergoing large deformations, a number of theories for plates and shells and their limitations are briefly reviewed here. Timoshenko (1959) discusses his general shell theory including the cases of large deflections of plates and shells. However, the nonlinearity of the materials, laminated structures, and composite orthotropic behavior are not included. Gould (1987) gives an introduction to linearly elastic thin plate and shell theory. The equilibrium, deformation, and constitutive laws of various shells and plates are discussed thoroughly, in a straightforward way. However, this standard text book does not include cases for various complex industrial applications. Whitney and Ashton (1970) describe a theory of laminated plates but do not discuss shells and material complexity (such as, hyperelastic and orthotropic properties). These publications are based on the Kirchhoff-Love's hypothesis, which doesn't take the transverse shear effects throughout the shell thickness into account.

In the investigations on laminate or sandwich composite shells, Reddy (1984) developed analytical solutions of laminated thick shells with typical boundary conditions and under various loads using the Sanders shell theory. In Reddy's theory, the Kirchhoff-Love's hypothesis was extended to consider the transverse shear deformation for laminated thick shells with material anisotropy, to reduce errors in stress and deflection predictions. This paper is cited by ANSYS user’s guide as a benchmark case for various cross-ply laminated shell elements. In addition, Wang et al. (1996) developed a boundary element method to solve the equations for moderately thick laminated shallow shells. The shell orthotropic properties were considered and their numerical solutions were compared with Reddy’s exact solutions. It is noted that the materials used were linearly elastic in both papers. Other studies on composite shells include a constitutive model for fiber-reinforced composite laminates developed by Luccioni (2006), and a geometrically nonlinear shear deformation theory for composite shells presented by Yu and Hodges (2004).

Simmonds (1985) derived an approximate strain energy density function, which can be applied to rubber-like shells with large elastic strains. Similar works included the nonlinear bending theory of rubber-like shells developed by Schieck et al. (1992), and the nonlinear finite element analysis of sandwich shells with rubber and fiber presented by Ferreira et, al. (2003). Most recently, Lu and Zhao et al. (2009) discussed distributive elastic properties in hyperelastic membranes and compared computational and experimental results. However, only in-plane forces are usually considered in a membrane theory.

Perhaps the more significant papers in this section were contributed by Dvorkin (1984) and Bathe (1980). In Bathe's paper, a general plate and shell theory applied for geometrical and material nonlinearities was presented. Based on Bathe's research, a new four-node general quadrilateral shell element for geometric and material non-linear analysis was presented in Dvorkin's paper. The first order transverse shear strains were considered using the so called Dvorkin-Bathe interpolation of nodal point displacements to alleviate the element locking problem. This 4-node element formulation was employed by many commercial FEA codes, including ADINA and ANSYS. The element was further improved for layered shell analysis by ANSYS (ANSYS and CFX user's manual, 2008).

2.5 Ideal and viscous flow theories and their limitations

The fluid flow through a valve can be evaluated using ideal flow theory or fully viscous flow theory, when considering the influence of viscosity. If ideal flow theory is applied for developing a coupled fluid-structural model, the coupled model may be simplified to be built only in an FEA framework for the solver of the ideal flow equation using a user defined subroutine within the FEA solution procedure. On the other hand, if fully viscous flow has to be considered, an existing FSI framework may be employed, for instance, the commercial codes (COMSOL, CFD-ACE+, ADINA, ANSYS+CFX, and ABQAUS+FLUENT), opensource FEA/CFD codes (Deal.II+Openfoam), or even in-house FEA/CFD codes (written using Matlab, C or FORTRAN).

2.5.1 Transient ideal flow theory

Streeter (1985) describes a transient 1D flow model in his fluid mechanics book. For unsteady incompressible ideal flow, the 1D flow equation can be expressed by

$$\frac{\partial u}{\partial t}ds + d\left(\frac{u^2}{2}\right) = -gdz - \frac{dp}{\rho} \tag{2.4}$$

where u is the fluid flow velocity at a point on a streamline, ds is an element of distance along the streamline, g is the acceleration due to gravity, dz is an infinitesimal elevation variation of the point above a reference plane, with the positive z-direction pointing upward, dp is an infinitesimal pressure change at the chosen point, and ρ is the density of the fluid.

The unsteady 1D flow equation is obtained by integrating along a streamline from point 1 to point 2

$$\int_1^2 \frac{\partial u}{\partial t}ds + \frac{p_2 - p_1}{\rho} + \frac{u_2^2 - u_1^2}{2} + g(z_2 - z_1) = 0 \tag{2.5}$$

The 1D flow equation has been widely applied to various industrial low speed pipe flows but limited for the applications with rotational and separated flows or large viscosity effects.

2.5.2 Viscous flow theory

Viscous fluid flow can be modeled using the Reynolds-averaged Navier-Stokes (RANS) equations, which are time-averaged equations of motion for fluid flow where an instantaneous flow quantity is Reynolds-decomposed into its time-averaged and fluctuating quantities (White, 1991). For a transient incompressible flow, these equations can be written as:

$$\rho \frac{\partial \bar{u}_i}{\partial t} + \rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \rho \bar{f}_i + \frac{\partial}{\partial x_j}\left[-\bar{p}\delta_{ij} + \mu\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \rho \overline{u_i^{'} u_j^{'}} \right] \qquad (2.6)$$

where ρ is the density of the fluid, $\bar{u}_i$ are the time-averaged velocity components, $\bar{p}$ is the averaged pressure, x_j are the coordinate components, μ is the fluid viscosity, δ_{ij} are the Kronecker delta, $\bar{f}_i$ are the volumetric force components, t is the time, and $-\rho\overline{u_i^{'}}\overline{u_j^{'}}$ are the Reynolds stresses representing the turbulence fluctuations in fluid momentum.

Menter (1994) discussed two-equation eddy-viscosity turbulence models to close the RANS equations, which includes the Shear Stress Transport (SST) *k*-*ω* turbulence model that was successful in pipe confined jet flows. One of the advantages of the *k*-*ω* formulation developed by Wilcox (2008) is the near wall treatment for low-Reynolds number computations. The model does not involve the complex non-linear damping functions required for the *k*-*ε* turbulence model and is therefore more accurate and more robust.

2.6 Fluid structure interaction

2.6.1 Approaches to FSI problems

Fluid structure interaction (FSI), which exists in many scientific and engineering applications, refers to the interaction phenomenon of a movable or deformable structure with internal or surrounding fluid flow. A comprehensive study of FSI problems remains a challenge due to their strong nonlinearity and multidisciplinary nature (Bungartz, et al, 2006). For most FSI problems, analytical solutions to the model equations are impossible to obtain. Thus, to investigate the fundamental physics involved in the complex interaction between fluids and solids, experimental or numerical methods must usually be utilized. Often, the laboratory tests are limited in scope and expensive. Thus, in the last few decades, a large number of numerical methods have been developed for the simulation of FSI. The main driving force for development is the demand from a wide range of scientific and engineering disciplines, where FSI problems play a prominent role. Meanwhile, the considerable improvement of computational power has made large-scale FSI simulations possible and has facilitated many realistic applications of these numerical techniques. Yet research in the fields of computational fluid dynamics and computational structural dynamics is still ongoing. Most numerical simulation approaches or codes (commercially used and opensourced) are developed focusing on some special applications. Mature formulations/models to solve FSI problems for a general purpose of industrial applications are rare. Giannopapa (2004) summarized two main approaches used for the simulation of fluid-structure interaction problems, as shown in Fig. 2-1:

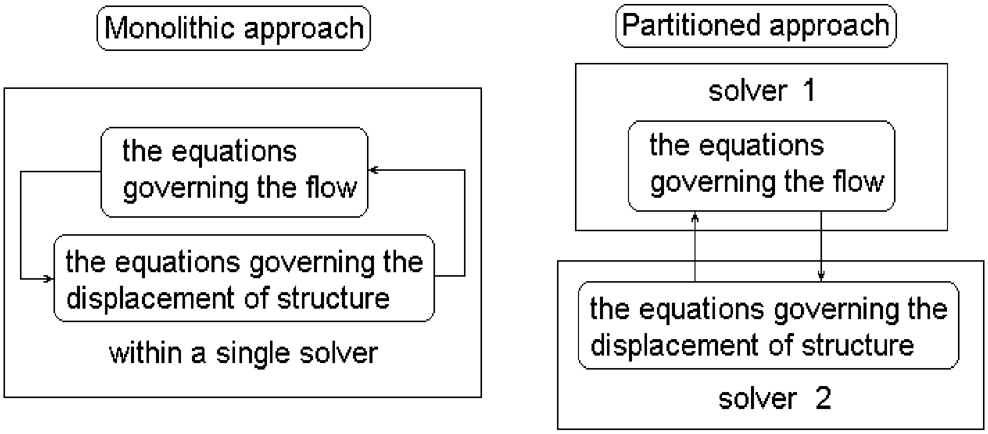


Fig. 2-1 Simulation approaches of fluid-structure interaction problems

The monolithic approach solves the equations governing the fluid and the structure simultaneously within a single solver, while the partitioned approach has the equations governing the fluid and the structure solved separately with two distinct solvers. The monolithic approach requires a code developed for this particular combination of physical problems (in the fluid side or in the structure side), whereas the partitioned approach utilizes an existing fluid flow solver and structural solver and requires the development of a stable and accurate coupling algorithm.

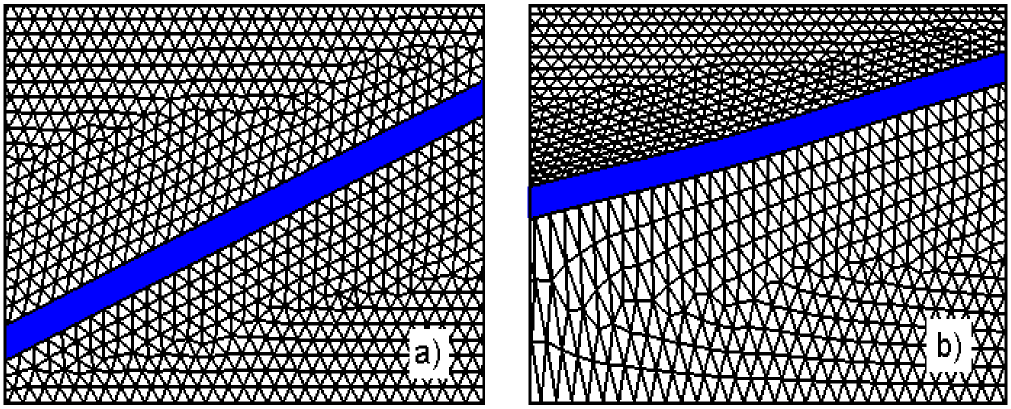


Fig. 2-2 Dynamic meshes: a) $t=t_1$ b) $t=t_2$

Dynamic meshes (as seen in Fig. 2-2) are usually involved in FSI simulation approaches to describe the fluid and solid domains and the related fluid dynamics and structural dynamics. The emphasis of these methods is on the coordination of data transfer and consistency between the existing fluid and structural codes. Most FSI methods use the generalized Gauss-Seidel (GGS) approach for the coupled analysis, in which the fluid and structural computation are performed in a sequential manner to achieve a multidisciplinary solution (Bungartz, et al, 2006). In other words, one may first solve the fluid field at a given time instance with an assumed interface location. The resulting fluid pressure and stress are then applied to the structure as external forces. The structural computation is then conducted to update the position of the structural surface. New fluid mesh is then created to accommodate the new interface location. An iterative process may be required to ensure that the interfacial conditions of both the displacement and the force are satisfied at the given time instance before moving to the next time instance. The challenges that one encounters when computing by means of an iterative coupled procedure are to maintain proper data transfer between the disciplines and to reach the converged solution efficiently. For a duckbill valve, there is a special challenge that a geometrical singularity exists when the valve is fully closed. In other words, meshes are difficult to put in the closed region of a duckbill valve.

2.6.2 FSI modeling of duckbill valve

To develop a fully coupled FSI model of a duckbill valve, the large deformation of the valve geometry, material nonlinearity and orthotropy (or anisotropy), layered design information, three dimensional viscous flow, and bidirectional coupling algorithm should be included. One of the possible ways to achieve such a goal is to utilize an existing FSI simulation framework, i.e. the commercial code ANSYS/CFX (ANSYS and CFX user's manual, 2008).

A complex coupling algorithm has been designed by the author for the FSI framework of ANSYS/CFX (as seen in Fig. 2-3), since both ANSYS and CFX are robust and powerful solvers for either fluid flow or structure dynamics problems individually. This FSI framework has been applied to the duckbill valve (see Appendix B). However, there exist a number of challenges when adapting this framework. For example, dynamic volume meshes are needed for the CFD solutions because the cross section of the duckbill portion varies from a fully closed status to a fully opened status. Thus, the meshes inside the duckbill tunnel need to be remeshed during valve motion or a deforming mesh becomes too distorted during the simulation. The combination of the strong geometric and material nonlinearities of the valve and the computational efforts required to solve 3-D viscous flow problems can make their coupled solution computationally intractable.

To overcome the above problems in the FSI simulation, a simplified FSI model (Wang, Weaver, and Tullis, 2012) has been developed and improved. This model is presented in detail in this thesis. The major simplification of the model was to reduce the 3D viscous flow to 1D ideal flow so that the flow equation can be easily solved, while all other features, such as the large deformation of valve geometry, material nonlinearity and orthotropy (or anisotropy), layered design information, can still be included. The monolithic approach for FSI simulation was taken, requiring the development of a coupled algorithm and fluid flow solver inside the FEA solver. Experimental validation of this code is an essential component of such development.

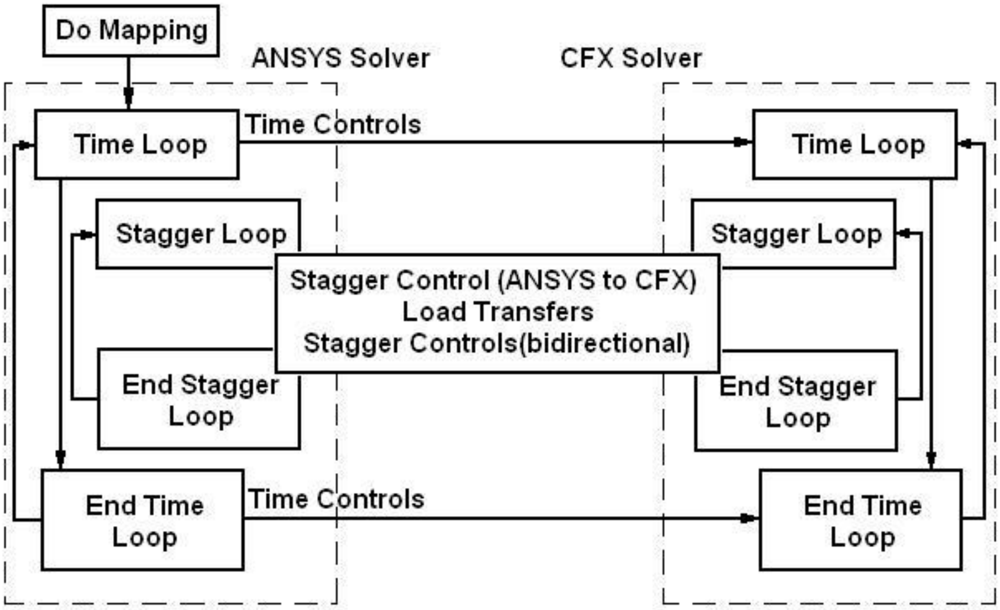


Fig. 2-3 Coupling approach of ANSYS/CFX

2.7 Flow induced vibration

The most difficult phenomenon, which may occur in a duckbill valve, is flow induced vibration. Several monographs on FIV have been published, such as that by Blevins (2001) but none have dealt with duckbill valve dynamics. Adubi (1974) investigated the oscillations of a swing check valve following rapid pump shut-down to discover alleviation methods of such valve vibrations due to self-excitation. Weaver and Ziada (1980) summarized three mechanisms causing such phenomena in certain valves at small openings, i.e. jet flow inertia, turbulence and acoustic resonance. In the case of a duckbill valve, flow-induced vibration is most likely at very small opening in which case the vibration would be similar to those caused by leakage flows. These have proven to be notoriously difficult to analyze (Blevins, 2001).

Chapter 3

LAYERED SHELL MODEL OF DUCKBILL VALVE

This chapter presents a duckbill valve structural model. To begin with, the duckbill valve geometry is described in detail. The experimental results and theoretical models for the hyperelastic materials of valve are discussed. The orthotropic properties of the fiber reinforcement are estimated using warp and weft extension testing data, and the empirical Halpin-Tsai approach. The layered four-node nonlinear shell element used for building the valve model is then presented and compared with a solid element model for a 3-point bending case. Based on these results, a middle-surface-based-layered-shell (MSBLS) model with non-linear geometry and composite is proposed for stable and reliable predictions of valve deformations. Model simplifications and boundary conditions are considered. The back pressure situation when reverse flow occurs is modeled and the corresponding results are discussed.

Since the value model requires empirical data in order to establish the material’s mechanical behavior, a specific 12” duckbill valve is used in model development and testing. The codes developed are generic and are applicable to valves of different size, geometry and materials.

3.1 Layered fiber reinforced hyperelastic duckbill valve structure

Figure 3-1 shows a 12" diameter inline duckbill valve consisting of a flange at one end, a duckbill like tube at the other, and a saddle shape in the middle. The composition of the layered duckbill valve considered consists of the materials listed in Table 3-1.

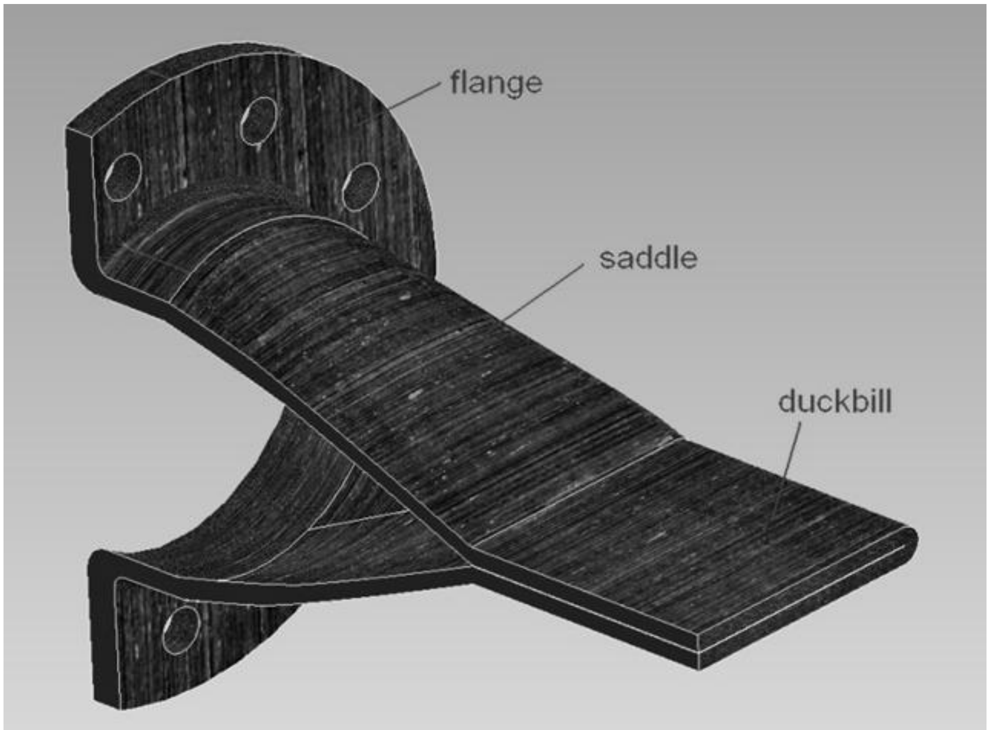


Fig. 3-1 Duckbill valve geometry

Table 3-1 Composite materials of duckbill valve

No.	Materials	Layers	Thickness	Locations
1	Ethylene Propylene Diane Monomer (EPDM)	1~2	3mm/ply	outer tube
2	Fiber reinforcement (FIBER±54)	2	1mm/ply	cross-plied in angles of ±54°, beneath EPDM
3	Styrene-Butadiene Rubber (SBR)	2	3mm/ply	middle tube
4	Chloroprene Rubber (CR)	1	3mm/ply	inner tube

The actual layer arrangement of these valve materials is seen in Fig. 3-2 (this piece is a cut-off part of the saddle portion). The layers are assembled in a duckbill shaped die in the same order, as presented in Table 3-1. There are however extra filler layers at the saddle and flange portion. In addition, the layer sheets are laid on one side of the valve and overlapped with the one on the other side around the side edges of the valve. All the valve layers are baked in an oven to be jointed together according to a specific working procedure.

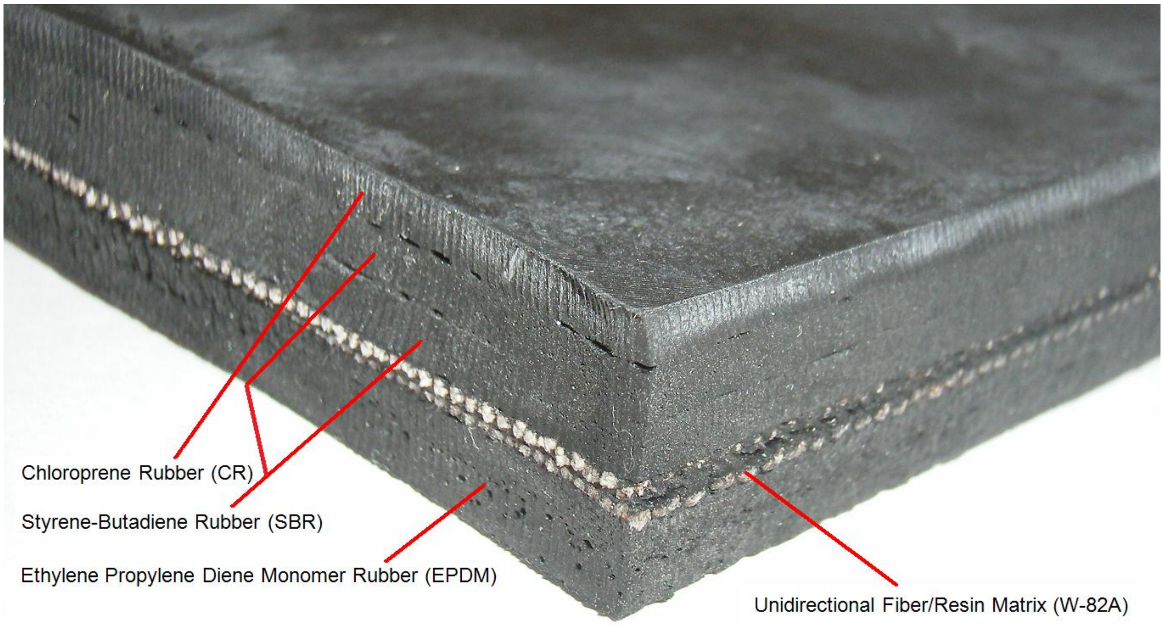


Fig. 3-2 Layered rubber material reinforced with cross plied fiber layers

The total averaged thickness of the composite is approximately 14 mm but within a widely variable range along the valve tunnel. Fig. 3-3 shows the actual measured thickness distribution of a 12" diameter DBV. It is seen that the saddle and flange portions are designed (by adding extra filler layers) to be thicker than the duckbill portion to increase the valve strength and maximum allowable back pressure to accommodate reverse flow. The valve manufacturing tolerances also affect the valve thickness distribution.

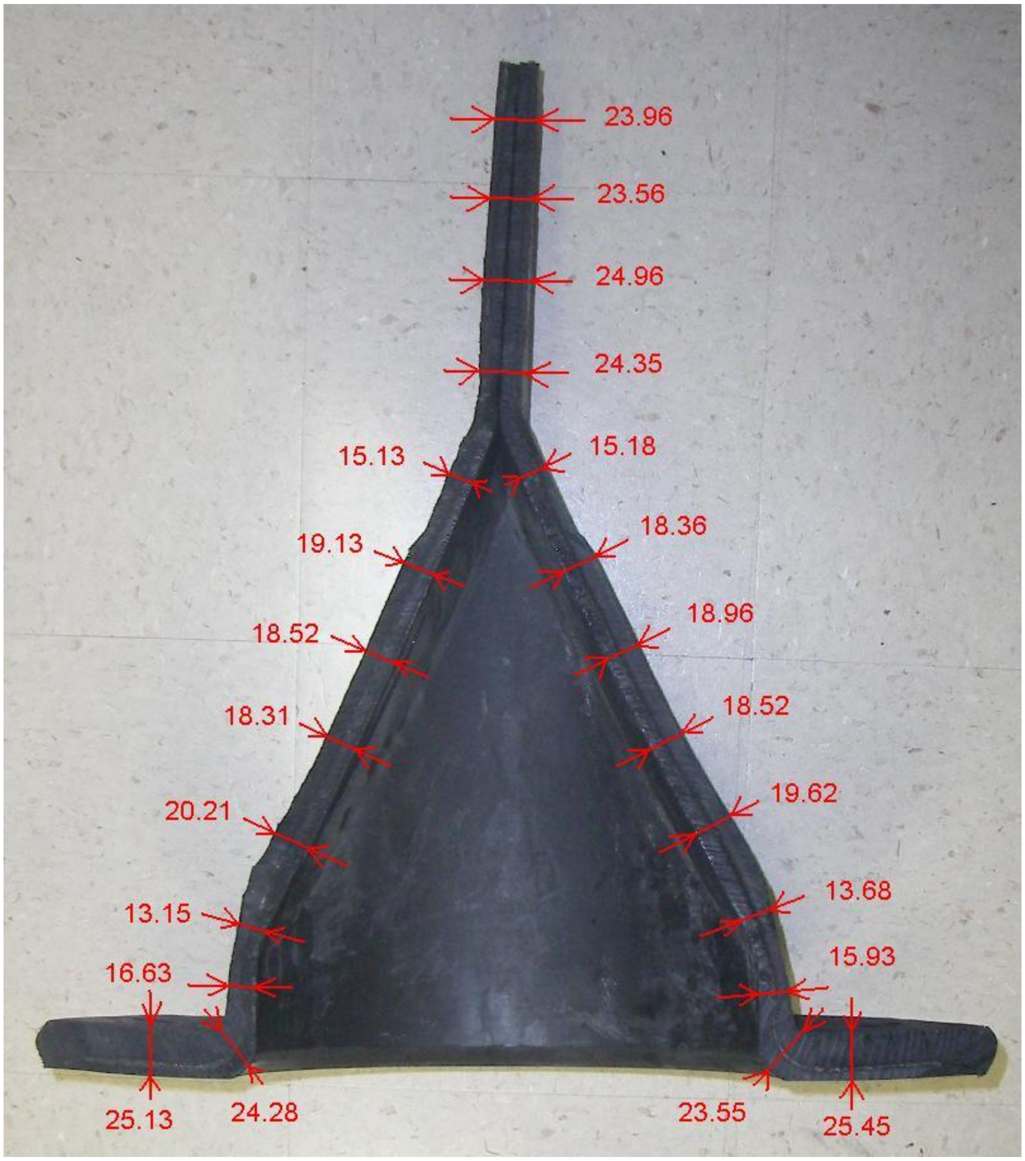


Fig. 3-3 Thickness measurement of a duckbill valve (in mm)

Some simplifications and assumptions are used in the FEA definition of the geometry of the model of the duckbill valve. Firstly, the valve fixed end (flange) and overlap portions of fiber sheet at the duckbill side edges are assumed to have little influence in modeling the duckbill valve and are therefore ignored in the geometry modeling. Secondly, it is assumed that the valve geometric deformation is symmetric along its two central planes during simulation. Thus, only a quarter model is required. Thirdly, the valve wall is assumed with a varied (uneven) thickness throughout the valve model. The measured valve thickness distribution (see in Fig. 3-3) is mapped onto the valve model built with the averaged wall thickness. These simplifications and assumptions will be helpful to automatically build the valve model in an in-house code. In addition, considering the complexity of the layered solid model and the necessity of saving FSI simulation computer time, the valve may be further simplified to be a shell structure since the ratio of valve thickness to duckbill span width is around 5% (Gould, 1987). However, this simplification would need to be validated.

3.2 Mechanical properties of valve materials

As mentioned previously, most rubber and fiber models applied in FEA are determined experimentally. Thus, the material properties of the duckbill valve have to be understood before being used in the simulation. The properties of the valve composite are so complex that only those aspects of the material properties of specific interest are considered here. Specifically, for the rubber properties, we mainly focus on the typical stable loading process of rubber, whereas the viscoelastic unloading process and material failure are not within the scope of this thesis. The significance of the Mullin’s effect will be evaluated in the section of sensitivity analysis (in Chapter 5) but is not of concern in the present research. For the fabric reinforcement properties, only warp and weft extension tests are conducted for the orthotropic elastic moduli while the orthotropic shear moduli and Poisson ratios of this composite are estimated using an empirical approach due to the intricacy of shear modulus and Poisson ratio measurement.

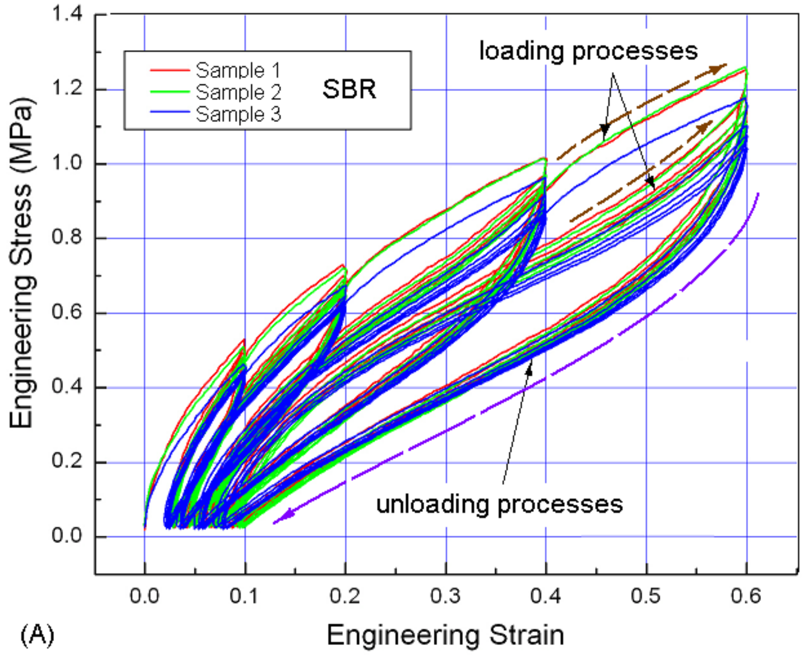
3.2.1 Uniaxial tension tests for rubbers

The three kinds of rubber, SBR, CR, and EPDM, were tested using a uniaxial tension method. Table 3-2 lists the information of the simple tension tests.

Table 3-2 Uniaxial tension test of rubber materials

Testing items	Materials	Nominal strain levels	Loading strain rate
Simple tension of a dense elastomer at 23 ℃	SBR, CR, and EPDM	10%, 20%, 40%, 60%	$0.01s^{-1}$, 5 times per level
Note: Test standard and procedure refers to ASTM D412-06a (2006) Rubbers are stamped out of dumbbell samples using a die cutting tool (Die C, the testing part of the dumbbell samples is 33mm×6mm×2mm in size) Loading/unloading processes repeated five times per strain level in an order from low to high levels Tensile test machine of AXEL Inc.			

Figure 3-4 A-C show the uniaxial tension results of SBR, CR, and EPDM (three samples tested for each material). It should be noted that there exists permanent strain deformation (ε_c), stress softening (Mullin's effect) and elastic hysteresis phenomena in the results. It is seen from Fig. 3-4 that there are a variety of curves for potential use in the simulations. The data curve used for an FEA simulation depends on the specific application purpose. For instance, if the focus of an analysis is to examine the first time straining of the rubber, then the first time stress-strain curves from the material tests should be used. If the interest is to understand the typical structural condition of a part in service, the stress strain curves should be derived by cycling a material until it is stable, and the stabilized increasing strain curve should be extracted for simulations (in Fig.3-4, the fifth cycles are expected to be stable). Moreover, the permanent strain deformations are usually not considered in the geometry modeling. Hence, zero stress is always zero strain for the hyperelastic models as a principle (Axel Products, Inc., 2012).



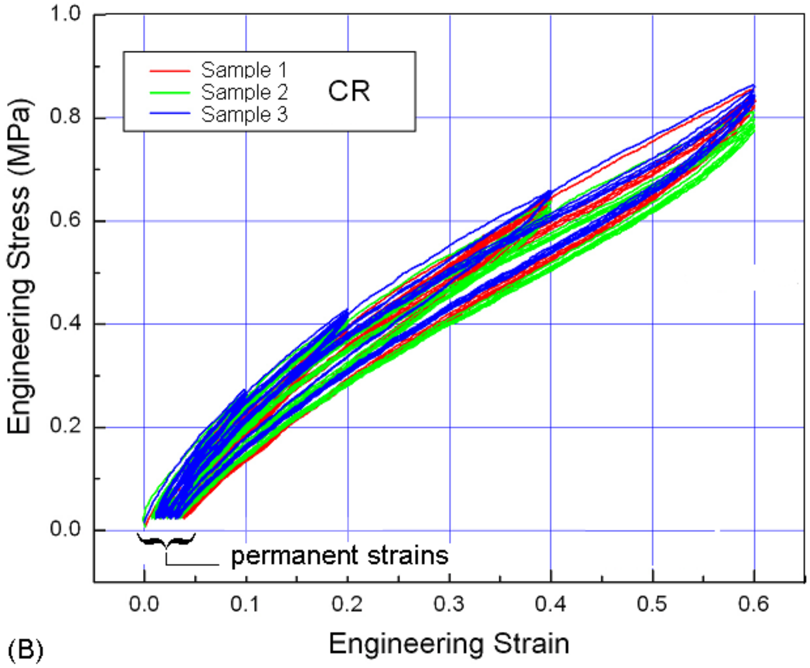


Fig. 3-4 Uniaxial tension results of (A) SBR, (B) CR, and (C) EPDM

(continued on next page)

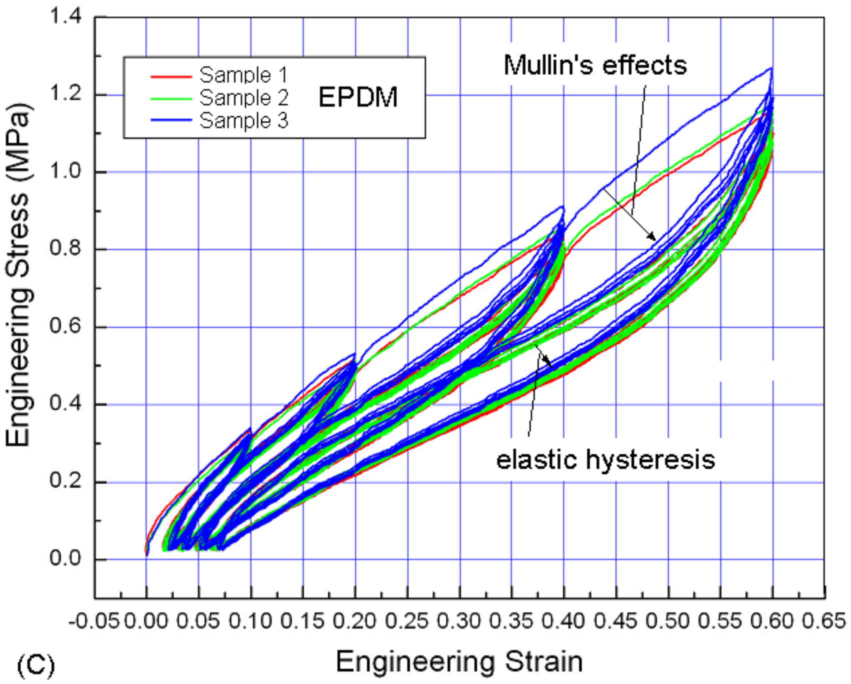


Fig. 3-4 Uniaxial tension results of (A) SBR, (B) CR, and (C) EPDM

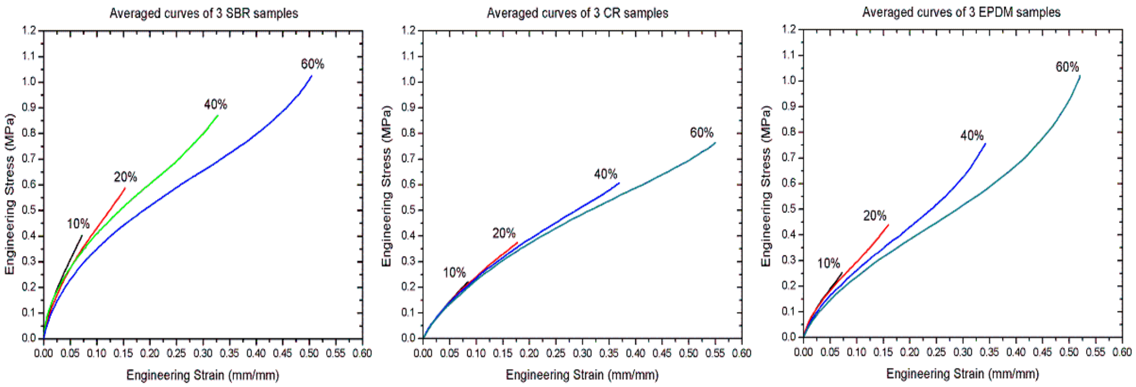


Fig. 3-5 Loading curves extracted from the uniaxial tension tests

According to the above operation principles, the fifth loading curve is extracted from the averaged result of three samples of each rubber (CR, EPDM, and SBR) for the material fitting procedure. The permanent strains are ignored as shown in Fig. 3-5.

3.2.2 Mooney-Rivlin model

To deal with the parameters of the rubber model from the experimental data, a material curve fitting tool of ANSYS is applied. The data is fed to the ANSYS software in the form of tab delimited stress-strain text files of the experimental data for uniaxial tension. The curve fitting tool adjusts these coefficients of the rubber model to obtain the best mathematical fit to the experimental data. The coefficients in the strain energy density functions may be viewed as material constants.

Two and three-parameter Mooney-Rivlin models are applied for the constitutive relations of the rubber materials. The forms of strain energy density function for the 2p and 3p Mooney-Rivlin models are derived from Equation (2.1) where *N* is equal to 1 and 2 respectively, and thus:

2p M-R model:

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) \tag{3.1}$$

3p M-R model:

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) + C_{11}(I_1 - 3)(I_2 - 3) \tag{3.2}$$

Table 3-3 lists the coefficients of the 2p and 3p Mooney-Rivlin models for SBR, CR and EPDM at per stain level.

A 3D bar model with 300 hyperelastic brick elements was simulated using the two rubber models. A bar that is exactly the same size of the uniaxial tension test specimen is stretched from 0 to 60% during the simulation. It is seen from Fig. 3-6 that the 3p Mooney-Rivlin model could better match the test results than the 2p Mooney-Rivlin model within the strain range of interest. Thus, the 3p Mooney-Rivlin model was selected for the valve model.

Table 3-3 Coefficients for the 2p and 3p Mooney-Rivlin models

Strain Level	SBR	2p	3p	CR	2p	3p	EPDM	2p	3p
10%	C10	-10.57	-36.10	C10	-1.87	-6.09	C10	-5.33	-16.24
	C01	12.28	38.22	C01	2.51	6.80	C01	6.31	17.39
	C11		53.52	C11		8.10	C11		23.35
20%	C10	-3.21	-12.01	C10	-0.77	-3.11	C10	-1.80	-8.94
	C01	4.44	13.53	C01	1.36	3.78	C01	2.62	9.99
	C11		9.25	C11		2.23	C11		7.25
40%	C10	-1.14	-6.55	C10	-0.18	-1.50	C10	-0.15	-2.91
	C01	2.14	7.92	C01	0.70	2.11	C01	0.72	3.66
	C11		2.82	C11		0.66	C11		1.47
60%	C10	-0.30	-2.92	C10	-0.03	-0.95	C10	0.18	-1.83
	C01	1.04	3.91	C01	0.51	1.51	C01	0.30	2.48
	C11		0.97	C11		0.34	C11		0.74

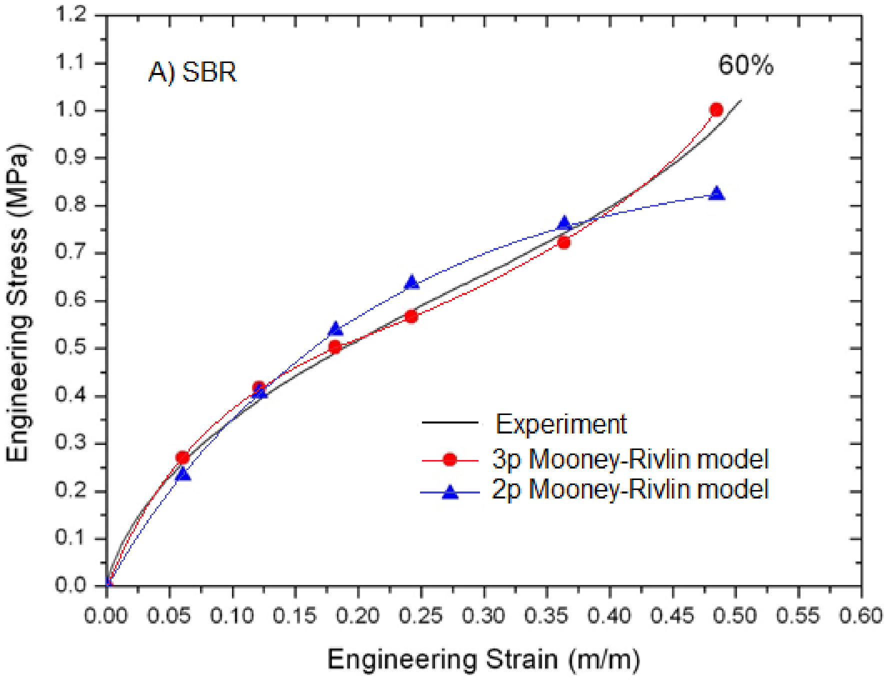


Fig. 3-6 The 2p and 3p Mooney-Rivlin models compared with uniaxial tension tests

(continued on next page)

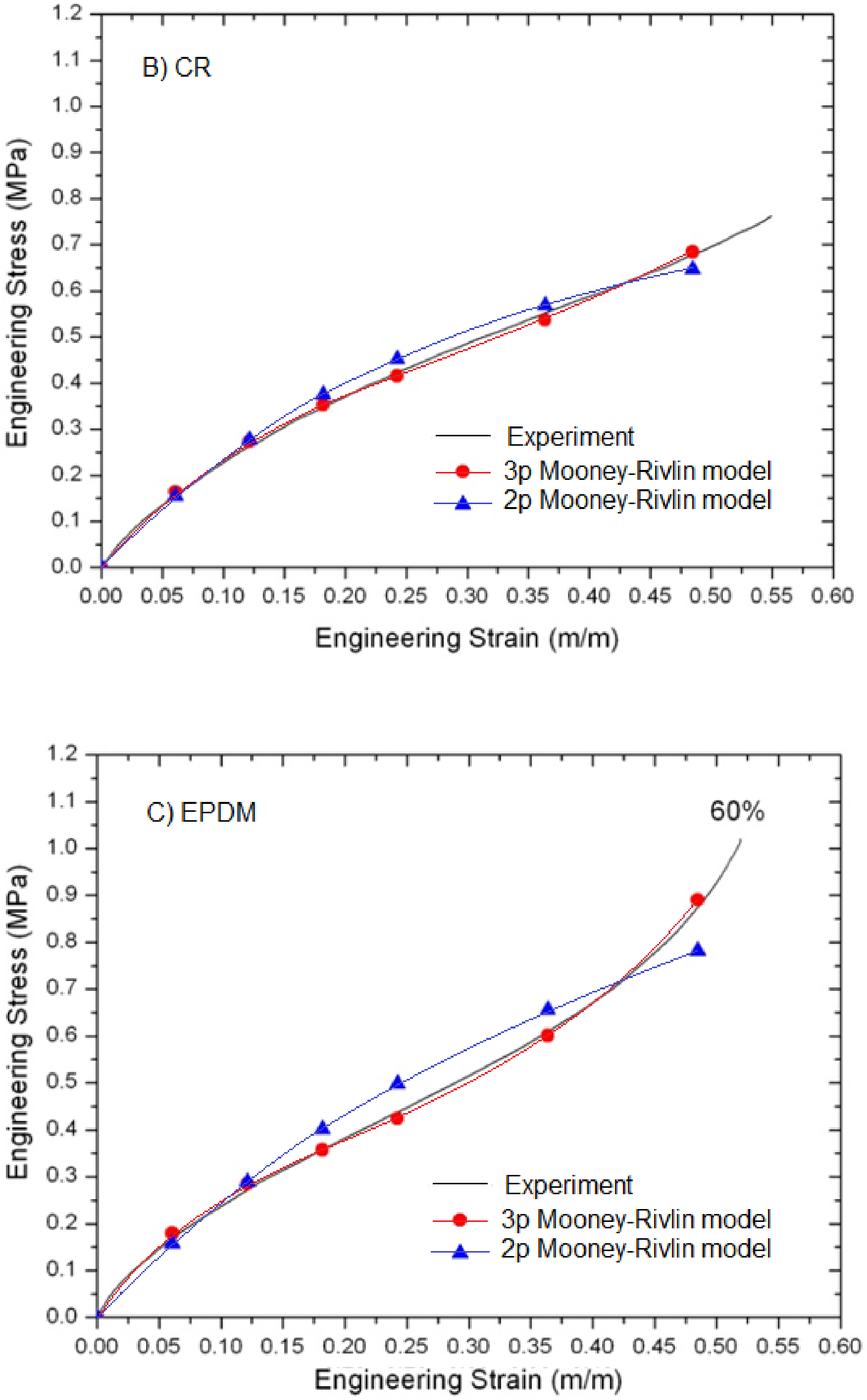


Fig. 3-6 The 2p and 3p Mooney-Rivlin models compared with uniaxial tension tests

3.2.3 Warp and weft extension tests for fiber/resins matrix

For the fiber reinforcement, three samples were tested in the warp direction (along the fiber orientation) and weft direction (perpendicular to the fiber orientation) respectively. Fig. 3-7 shows a schematic diagram for the unidirectional fiber/resin matrix. Table 3-4 shows the test parameters for the warp and weft tension tests.

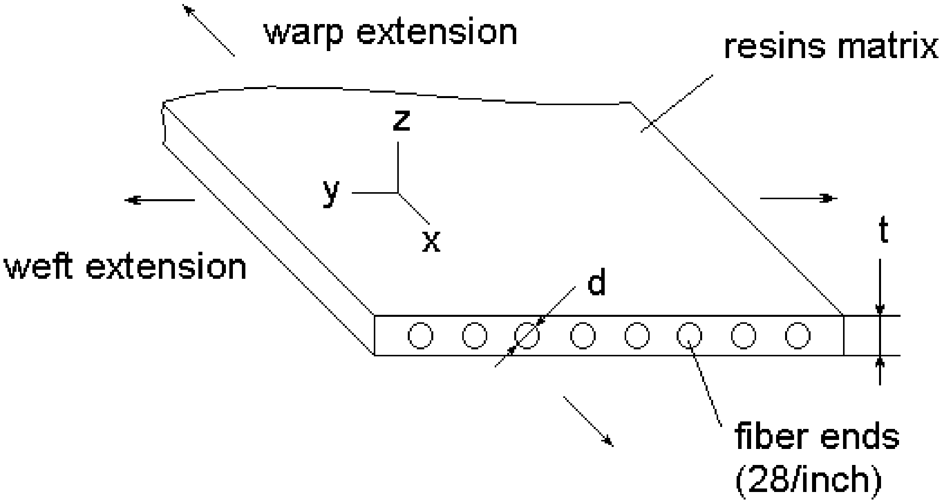


Fig. 3-7 Unidirectional fiber reinforced resins matrix

Table 3-4 Warp and weft tension tests

Testing items	Materials	Loading strain rate
Tension experiment on a fabric at 23 ℃ in the warp direction	Dupont Kevlar Aramid	$0.001s^{-1}$
Tension experiment on a fabric at 23 ℃ in the weft direction	Dupont Kevlar Aramid	$0.001s^{-1}$
Note: Kevlar Aramid is a kind of fiber material of Dupont, Inc. (Dupont, Inc, 2012), which is adhered in a resin matrix to become a sheet of fiber/resin reinforcement as seen in Fig. 3-7.		

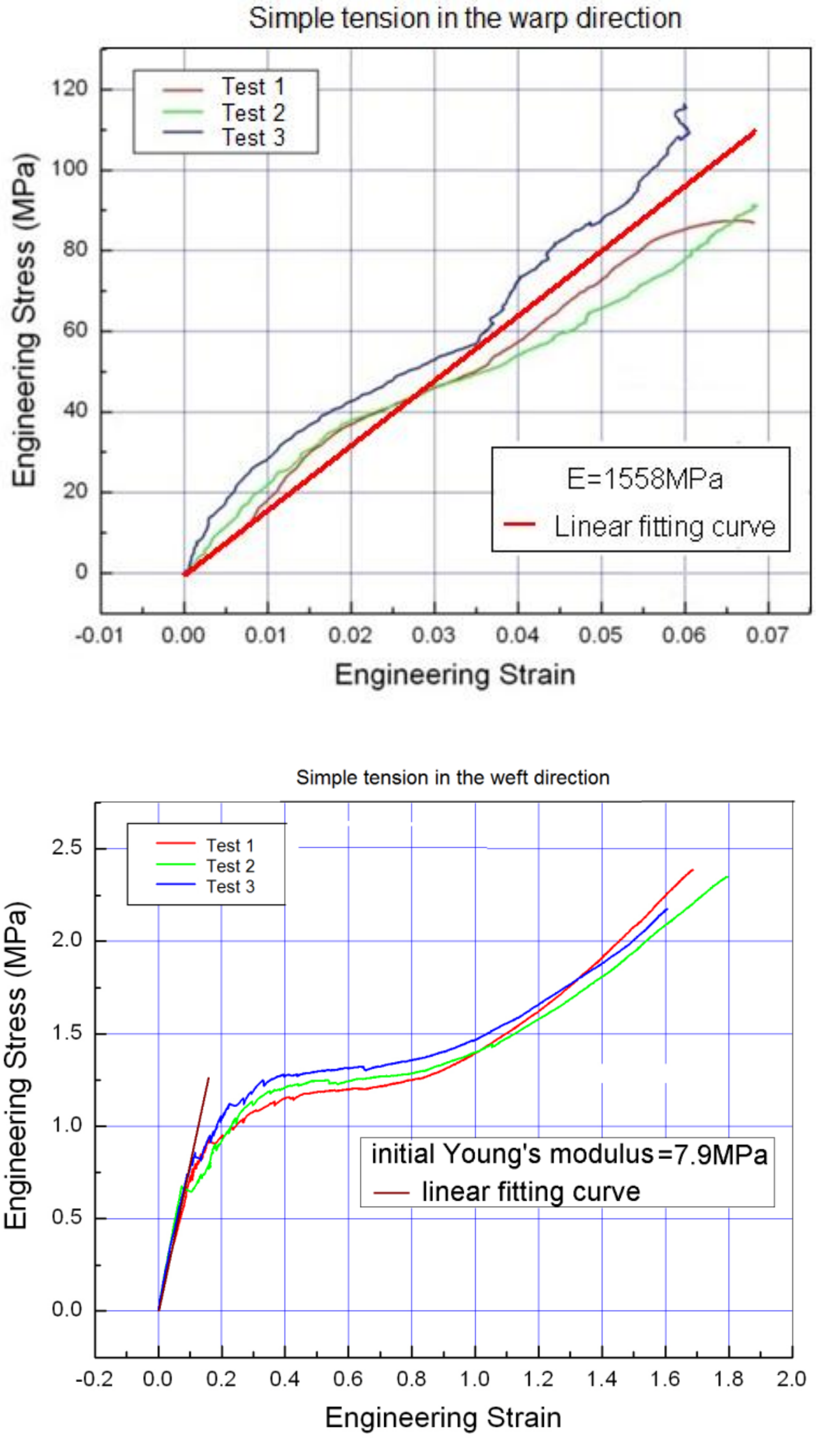


Fig. 3-8 Warp/weft tension curves of the unidirectional fiber/resin matrix

Figure 3-8 shows the warp and weft tension testing results for the three fiber samples. Although the behavior of a group of fiber filaments is actually complicated as seen in the warp direction plot of Fig. 3-8, the fiber sheet can be approximated as linearly elastic in the warp direction. So, by linearly fitting and averaging the three curves in the warp direction, we obtain the elastic modulus of the fiber reinforcement in the warp direction. On the other hand, the behavior of the fiber sheet in the weft direction is mainly determined by the properties of resin matrix, which is nonlinear over a large strain range. However, the strain range of the duckbill valve is usually within 10% (Lee, et al, 2004). The initial elastic modulus (maximum value) in the weft direction adequately represents the magnitude within this strain range, and compared to the warp direction, the initial elastic modulus is small (only 0.5% of the elastic modulus in the warp direction). Thus, it is not expected to influence the simulation results; although, it cannot be ignored as one of the necessary items for the orthotropic material model.

3.2.4 Halpin-Tsai estimates

Although the warp and weft tension tests of the fiber/resin matrix have been carried out, the orthotropic shear moduli and Poisson ratios of the material, which are necessary for the fiber model, were not obtained experimentally. The main reason is that the related experiment is difficult and expensive to carry out and the accuracy of corresponding results cannot be guaranteed. Fortunately, these properties are the secondary properties in predictions of valve deformation. Therefore, an empirical approach to estimate these properties of the fabric reinforcement is more practical and effective. Based on these, Halpin-Tsai equation is given as (Halpin, 1976):

$$M = \frac{M_m\{M_f + \xi[\varphi_f M_f + (1-\varphi_f)M_m]\}}{\varphi_f M_m + (1-\varphi_f)M_f + \xi M_m} \tag{3.3}$$

where M can be the elastic modulus, E, shear modulus, G, and Poisson ratio, v, of the material components; φ is the volume fraction of each material; ξ is an adjustable coefficient (when ξ=0, the weft properties can be estimated; when ξ=2, the shear properties can be estimated; when ξ goes to infinity, the warp properties can be estimated); and subscripts *f* and *m* refer to the fiber and resin matrix

Before dealing with the properties of fiber sheet, the Poisson ratios of fiber and resin are assumed to be known (test data from Chouchaoui, 2001), and the elastic modulus of fiber and resin can be calculated from the warp and weft tension tests. Thus, the shear modulus of fiber and resin can be estimated via

$$G = \frac{E}{2(1+\nu)} \tag{3.4}$$

Consequently, the orthotropic properties of a unidirectional fabric reinforced resin matrix are obtained using the Halpin-Tsai equation. In the first step, the elastic modulus's of fiber and resin are calculated using the elastic moduli of the fiber/resin sheet obtained from the extension tests and the Halpin-Tsai equation. In the second step, the shear moduli of fiber and resin can be calculated using Equation (3.4) and then the shear moduli of the fiber/resin matrix in each plane can be obtained using the Halpin-Tsai equation and the shear moduli obtained in this step. In the third step, the Poisson ratios of the fiber/resin matrix in each direction are calculated using the Halpin-Tsai equation and the Poisson ratios of fiber and resin. Table 3-5 lists the orthotropic properties of the fiber/resin matrix.

Table 3-5 Orthotropic properties of FIBER±54

E_x	1558MPa	G_{xy}	2.94MPa	ν_{xy}	0.27
E_y	7.94MPa	G_{xz}	2.94MPa	ν_{xz}	0.27
E_z	7.94MPa	G_{yz}	5.57MPa	ν_{yz}	0.28

where E_x, E_y, E_z are elastic moduli in *x, y, z* directions respectively, G_{xy}, G_{xz}, G_{yz} are the shear moduli in each plane, and v_{xy}, v_{xz}, v_{yz} are the Poisson ratio for each direction. The linear stress-strain relations are determined using Hooke's law.

It is empirically estimated that the strain range of the duckbill valve is not larger than 10%. Thus, the fitting of weft data is applied within this range. The elastic modulus in the warp direction is also applicable in this range, though the experimental strain range was limited to 7% due to measurement difficulty.

3.3 Layered nonlinear shell element

Although high-order shell elements with 9 or 16 nodes can produce more accurate results using fewer elements in some analyses, these shell elements are quite expensive in terms of computational time and in some analyses a large number of elements are still needed (Dvorkin, 1984 and Bathe, 1980). For general shell analyses, a simple, reliable and cost-effective 4-node shell element is presented by Dvorkin (1984). This element is formulated using three-dimensional continuum mechanics theory. It is applicable to the nonlinear analysis of thick shells.

Strain-displacement relationships

Figure 3-9 shows the geometry of the 4-node shell element, which is described using (Bathe, 1980 and Dvorkin, 1984):

$$^{l}x_i = \sum_{k=1}^{4} h_k \, {}^{l}x_i^k + \frac{r_3}{2} \sum_{k=1}^{4} a_k h_k \, {}^{l}V_{ni}^k \tag{3.5}$$

where the $h_k(r_1,r_2)$ are the two-dimensional interpolation functions corresponding to node k; the r_i are the natural coordinates; and $^{l}x_i$ are Cartesian coordinates of any point in the element; ${}^{l}x_i^k$ are Cartesian coordinates of nodal point *k*; ${}^{l}V_{ni}^k$ are components of director vector at node *k* (which is not necessarily normal to the middle surface of the element); and a_k is the shell thickness at node k, measured along the vector ${}^{l}V_n^k$. The left superscript is zero for the initial geometry of the element and is equal to 1 for the deformed element geometry. The thickness of the element varies and the element is in general non-flat.

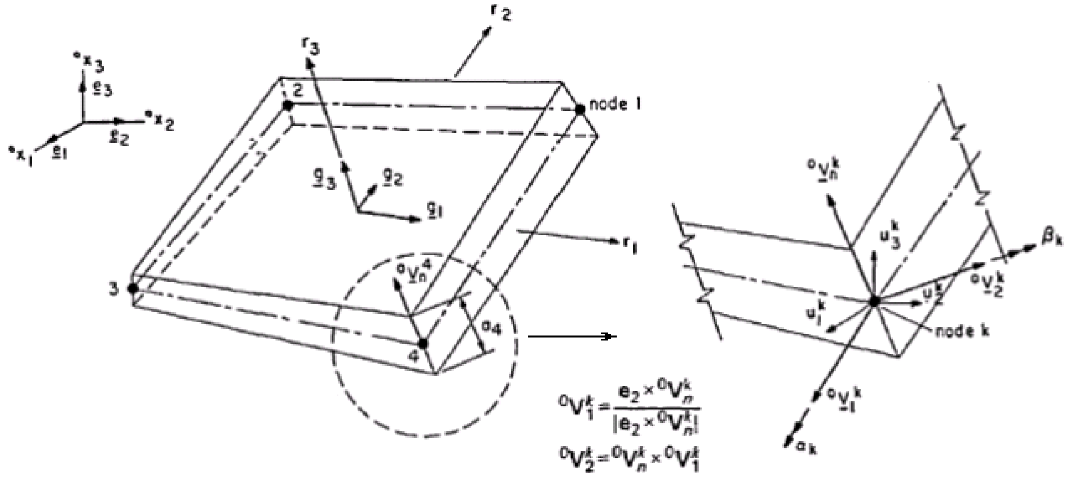


Fig. 3-9 Four-node nonlinear shell element

(Dvorkin, 1984)

The displacements of any particle with natural coordinates r_i of the shell element in stationary Cartesian coordinate system are given by (Bathe, 1980 and Dvorkin, 1984):

$$u_i = \sum_{k=1}^{4} h_k u_i^k + \frac{r_3}{2} \sum_{k=1}^{4} a_k h_k \left(-{}^{0}V_{2i}^k \alpha_k + {}^{0}V_{1i}^k \beta_k \right) \qquad (3.6)$$

where the u_i^k are nodal point displacements in the Cartesian coordinate directions, and the α_k and β_k are the rotations of the director vector ${}^{0}V_n^k$ about the ${}^{0}V_1^k$ and ${}^{0}V_2^k$ axes (see Fig. 3-9). An initial displacement effect is considered using Equation (3.5) at time t and time 0 (Bathe, 1980).

To mitigate the so called “lock” problem associated with the interpolation of the displacements and the derivation of the strain-displacement matrices when the shell element is thin, separate interpolations (not derived from the displacements) for the transverse shear strains are used here (see Dvorkin and Bathe’s papers for more details). Since separate interpolations are performed more effectively in convected coordinates than in global stationary coordinates which are used for the shell element formulation, a tilde (~) over the stress and strain quantities is used to emphasize that all the measures are in the convected coordinate system (Dvorkin, 1984 and Bathe, 1980). Hence, the strain components $\tilde{\varepsilon}_{11}$, $\tilde{\varepsilon}_{22}$ and $\tilde{\varepsilon}_{12}$ are evaluated using Equation (3.5) and (3.6). The transverse shear strain components $\tilde{\varepsilon}_{13}$, $\tilde{\varepsilon}_{23}$ are evaluated using the separate interpolation equations. The complete element stiffness matrix is calculated using the total potential energy equation (Dvorkin, 1984):

$$\Pi^* = \frac{1}{2}\int_V \tilde{\tau}^{ij}\tilde{\varepsilon}_{ij}dV - W_s \tag{3.7}$$

where the $\tilde{\tau}^{ij}$ are the contravariant components of the Cauchy stress tensor (Fung, 1965), the $\tilde{\varepsilon}_{ij}$ are the covariant components of the infinitesimal strain tensor, and W_s is the potential of the external loads.

Constitutive relations

To close Equation (3.7), the appropriate constitutive relations are required:

$$\tilde{\tau}^{ij} = \tilde{C}^{ijkl}\tilde{\varepsilon}_{kl} \tag{3.8}$$

where $\tilde{C}^{ijkl}$ is the fourth-order constitutive tensor in the converted coordinates r_i.

Equilibrium relation using total Lagrangian formulation

The nonlinearity of material has been described for the Mooney-Rivlin model in the previous section. The large deformation (geometrical nonlinearity) of the shell element is presented using the total Lagrangian formulation and the principle of virtual work. The principle of virtual work applied to the configuration at time $t+\Delta t$ is (Dvorkin, 1984):

$$\int_{^0V} {}^{t+\Delta t}_{\ 0}\tilde{S}^{ij}\,\delta\,{}^{t+\Delta t}_{\ 0}\tilde{\varepsilon}_{ij}\,{}^{0}dV = {}^{t+\Delta t}\Re \tag{3.9}$$

where the ${}^{t+\Delta t}_{\ 0}\tilde{S}^{ij}$ are the contravariant components of the second Piola-Kirchhoff stress tensor (Bathe, 1980) at time $t+\Delta t$ and referred to the configuration at time 0, and the ${}^{t+\Delta t}_{\ 0}\tilde{\varepsilon}_{ij}$ are covariant components of the Green-Lagrange strain tensor at time $t+\Delta t$ and referred to time 0. Both sets of tensor components are measured in the convected coordinate system $r_i, i = 1,2,3$. The external virtual work, which is given by ${}^{t+\Delta t}\Re$, includes the work due to the applied surface tractions and body forces.

The basic equilibrium relation employed to develop the governing finite element matrices can be derived using Equation (3.9), known stress and strain quantities, and unknown stress and strain increments (Dvorkin, 1984 and Bathe, 1980). By the equilibrium equation linearization and finite element discretization, the finite element incremental equilibrium relations are given as:

$$\left(\left[{}^{t}_{0}K_{L}\right]+\left[{}^{t}_{0}K_{NL}\right]\right)\cdot\left[\Delta u\right]=\left[{}^{t+\Delta t}R\right]-\left[{}^{t}_{0}F\right] \tag{3.10}$$

where $[\Delta u]$ is the incremental displacement matrix, $[{}^{t}_{0}K_{L}]$ is the linear strain incremental stiffness matrix, $[{}^{t}_{0}K_{NL}]$ is the non-linear strain incremental stiffness matrix, $[{}^{t}_{0}F]$ is the nodal point force matrix, and $[{}^{t+\Delta t}R]$ is the external virtual work matrix.

This 4-node quadrilateral shell element with six degrees of freedom at each node (translation in the x, y, and z directions and rotations about the x, y, and z axes), has been developed to apply for layered structures in the ANSYS program; see e.g., Reddy, (1984) and ANSYS and CFX user’s manual, (2008).

There are some application limitations for the ANSYS shell element (ANSYS and CFX user's manual, 2008): Solid submodeling should be used to precisely evaluate interlaminar shear stresses if necessary. The through-thickness stress, (S_z), is always zero. The applicability of the shell model, in which the hyperelastic and elastic materials are defined in lamina, must be validated using a comparable solid model. If there are curved thick shell structures in a simulation, the shell element must also be validated via a full 3D model in a simpler representative case (ANSYS and CFX user's manual, 2008).

3.4 Three Point Bending

The layered 4-noded nonlinear shell element is validated via full 3D modeling with a solid element for the 3-point bending case, as well as via 3-point bending experiments. Fig. 3-10 shows an FEA model of three-point bending. The size of the 3D model is 160×50×17 mm, which is exactly the same as the testing specimen. Its layer arrangement is: 1 layer of CR with 3mm thickness, 2 layers of SBR with 3mm thickness each, 2 layers of Fiber±54 with 1mm thickness each, and 2 layers of EPDM with 3mm thickness from top to bottom. The model is simply supported along both sides 50mm from the sample center on which the load/displacement acts. Both layered shell elements and solid elements are applied to this geometric model, with the 3-parameter Mooney-Rivlin model and orthotropic linearly elastic model being applied for the corresponding layers. Fig. 3-11 shows the meshes of the layered shell model and solid model.

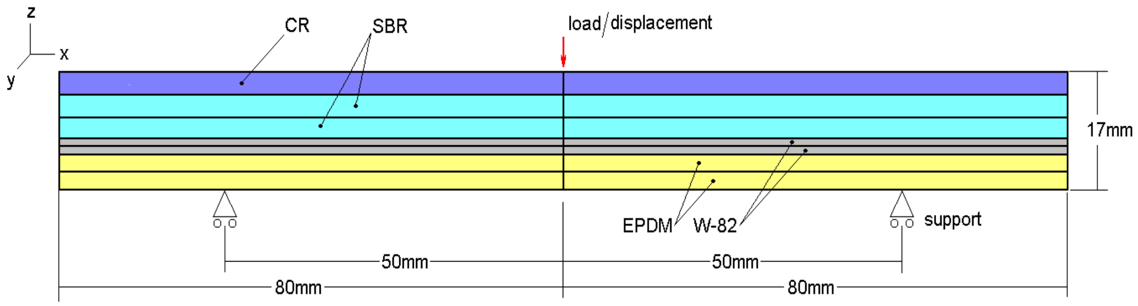


Fig. 3-10 An FEA model of three point bending

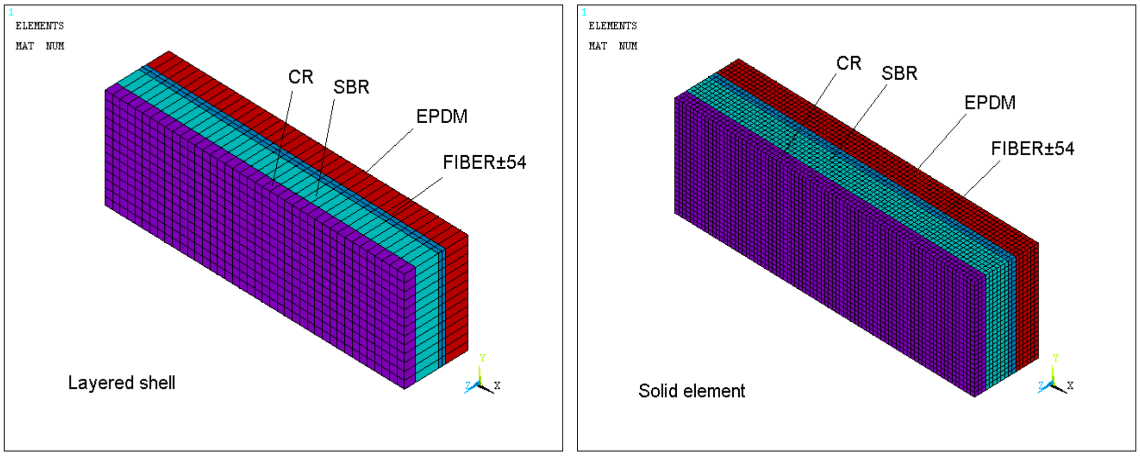


Fig. 3-11 Layered shell model and solid model for the bending simulation (half models)

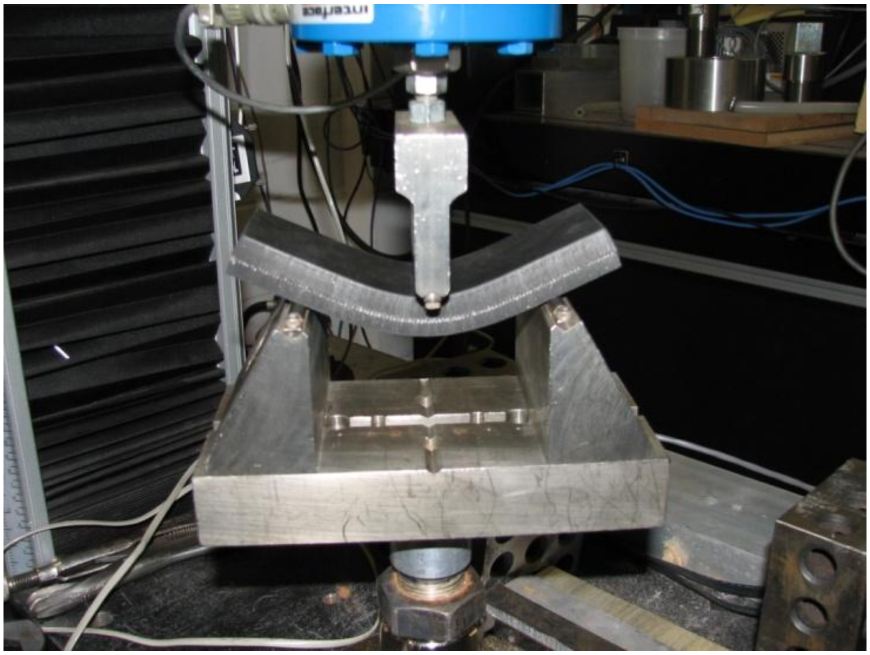


Fig. 3-12 Three-point bending test

Figure 3-12 presents a photograph of the bending test. The principal advantage of the 3-point bending test is the ease of the sample preparation and testing. The 3-point bending flexural test provides values for the flexural stress σ_f, flexural strain ε_f , and the flexural stress-strain relationship of the tested material.

Calculations of the flexural stress σ_f and strain ε_f for a rectangular cross section are given as:

$$\sigma_f = \frac{3PL}{2bd^2}, \quad \varepsilon_f = \frac{6Dd}{L^2}. \tag{3.25}$$

In these formulas, the following parameters are used: σ_f = stress in outer fibers at midspan, (MPa), ε_f = strain in the outer surface, (mm/mm), P = load at a given point on the load deflection curve, (N), L = support span, (mm), b = width of test beam, (mm), d = depth of tested beam, (mm), D = maximum deflection of the center of the beam, (mm).

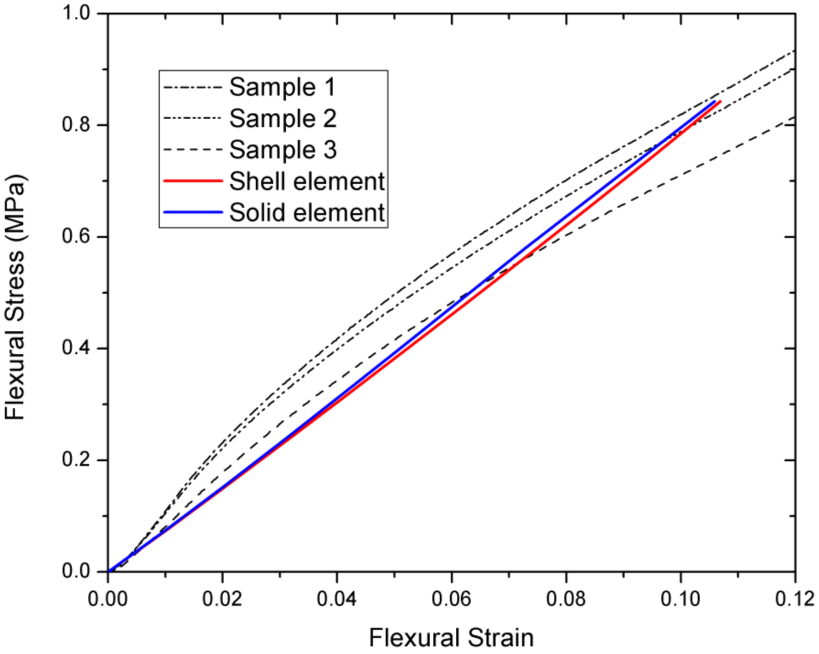


Fig. 3-13 Flexural stress-strain relationships

(Both test results and predictions of two models)

Figure 3-13 compares the simulations and test results. It can be seen that the relation between flexural stress and strain predicted by the layered shell model and the solid model are basically linear, with an average relative difference of 1%. This indicates that the layered shell element gives essentially the same results as the solid element for the 3-point bending case. The curves of the test results are however seen to be nonlinear. Error analysis shows that the average relative difference between the two models (shell and solid) and the averaged data of the three sets of test results are 13.6% and 12.4%, respectively. It is worth noting that the average relative difference between the 3-point bending measurements is within±9.6%, which means that the difference between the experimental specimens is of the same order as that between the predicted and averaged measured data, as also observed in the figure.

3.5 Layered shell model for the valve geometry

In this section, the layered shell model for the duckbill valve geometry is described in detail. Mesh controls and boundary conditions, as well as the solution controls and limitations of the shell model are discussed.

3.5.1 Generation of duckbill valve geometry

Based on the simplifications and assumptions discussed in the section 3.1, the middle surface is determined from the 3D duckbill valve geometry to represent the whole geometry. Fig. 3-14 shows the key point profiles and shell structure of the duckbill valve created based on these profiles. As seen in the figure, each profile curve consists of 15 key points. There are total 135 key points for 9 curves from which 8 strips can be created to form the valve shell. Based on the data of the 135 key points coming from the valve designers or valve measurement, an in-house code was developed to create the middle surface valve shell geometry automatically. This code is for future use as a valve design and analysis tool - an industrial requirement for a fast and simple way for the definition of valve geometry, mesh generation, FSI simulation, and post-processing.

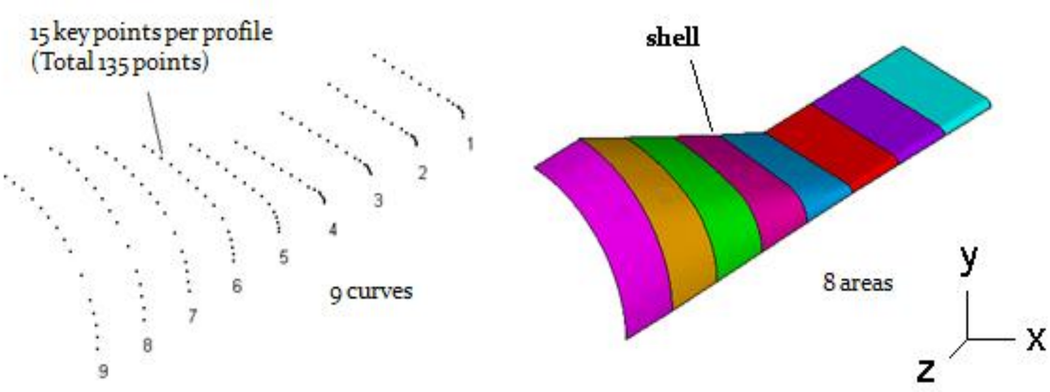


Fig. 3-14 Simplification and generation of duckbill valve geometry

3.5.2 Middle surface based shell verse bottom surface based shell

In the author's previous work (Wang, et al. 2012), an inside surface based shell model was developed, which was capable of directly producing the cross section area of flow tunnel based on the deformed valve shape. This was useful for coupling with the 1D flow, since the inside surface was the interface between the fluid and structure domains. However, the mesh quality of the model was found to be locally poor due to the unsmooth geometric surface (especially at the connection between the saddle and duckbill portions). Some quadrilateral meshes degenerated to triangular meshes observed in Fig. 3-15 (a), which resulted in 'bad shapes' of shell elements at large valve deformation. These 'bad elements' affect the numerical stability of the FSI simulations. In comparison, the middle surface based valve geometry, especially at the connection between the duckbill and saddle portions, is smooth, which is helpful in generating meshes with higher quality as seen in Fig. 3-15 (b). The quadrilateral meshes can keep 'good shapes' with large geometric deformations. Since the coupling algorithm (see Chapter 5) must deal with the element node movements during FSI simulation, good quality of the structural meshes is helpful in promoting stable and more rapidly convergent analysis of the coupled FSI problem.

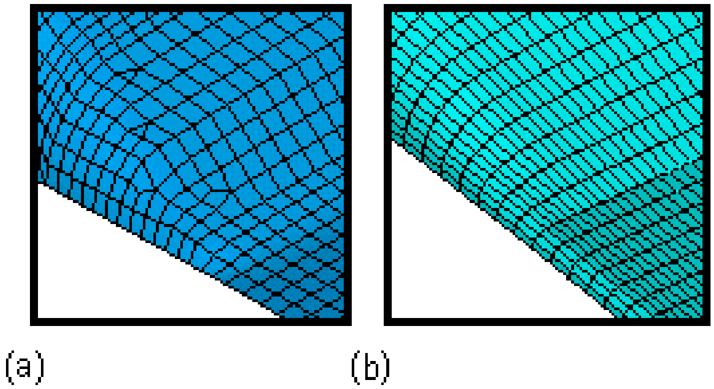


Fig. 3-15 Comparison of local mesh quality between (a) bottom surface based shell model and (b) middle surface based model

3.5.3 Duckbill side edge boundary

In an earlier model (Wang, et al. 2012), it was difficult to define the boundary conditions of the duckbill side edges, because the actual physical rotation along these edges was neither fixed nor free. Idealized fixed or free constraints either overpredict or underpredict the actual centerline valve displacement as seen in Fig. 3-16 (a). Actually, some rotation in the x-y plane occurs along the side of the duckbill portion as the joint has finite stiffness, i.e. $\partial y / \partial x \neq 0$. This rotation is a function of valve opening and is largest near the valve outlet. Thus, some error associated with this constraint simplification is expected. In other words, the inside surface based shell model, so constrained, cannot predict the rotation behaviour along the valve sides accurately.

This issue has been resolved using the middle surface based shell model. Fig. 3-16 (b) shows that some rotation of the inside surface is now permitted along the duckbill side edge since the fixed rotation constraint is moved to the small arch structure of the middle surface, which deforms as a function of valve opening.

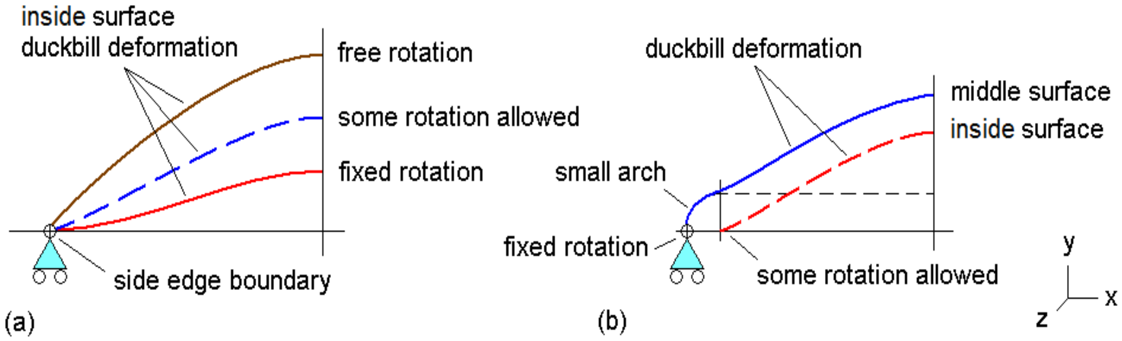


Fig. 3-16 Different constraints of duckbill side edge

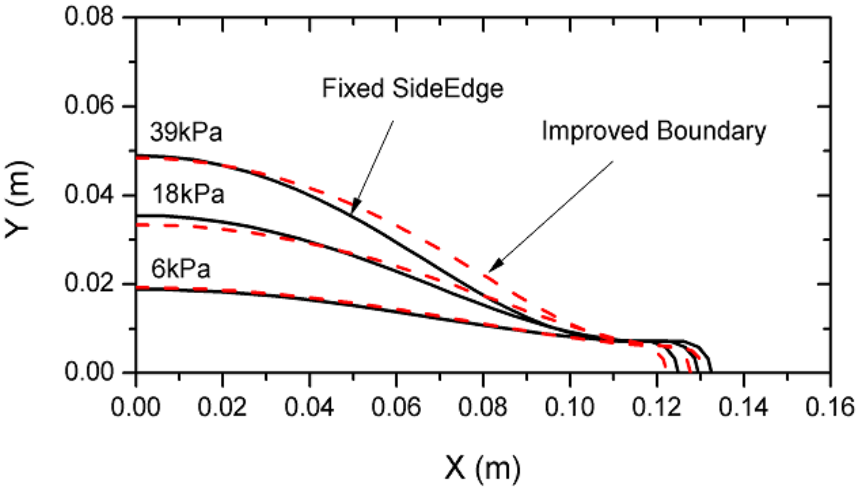


Fig. 3-17 Comparison of different duckbill side edge boundary conditions

Figure 3-17 shows computed results for the deformations of the duckbill exit opening predicted by the inside surface based shell model with fixed side edge rotation and the middle surface based shell model with some rotation allowed. The driving pressure drops used in the calculations are 6 kPa, 18 kPa, and 39 kPa, respectively. The solid black curves represent the deformations predicted by the former model while the dashed red curves represent the deformations predicted by the current MSBLS model. The y-coordinate represents the direction of valve opening measured from the closed valve centerline. At lower pressure, the edge restraint appears to have little effect on the valve opening. However, at larger pressure, finite edge stiffness permits larger valve openings accompanied by a slight reduction in the valve width (x-coordinate).

3.5.4 ANSYS layered shell model

Figure 3-18 shows the quadrilateral meshes and boundary conditions for the DBV shell model. All quadrilateral meshes are of good quality as seen in the figure. Symmetries about the x-z plane and y-z plane are assumed so that only one quarter of the valve about the planes of symmetry is modeled. The side edge of the DBV is a symmetric boundary in the y-direction so that the degrees of freedom (DOF) in x-rotation, y-displacement and z-rotation are set equal to zero. The central edge of the DBV is also symmetric in the x-direction; similarly, the constraints in x-displacement, y-rotation and z-rotation are all set equal to zero. The inlet boundary is fully clamped (no displacement or rotation in any direction). The fluid load, which depends on the fluid flow through the valve, is applied normal to the valve surface.

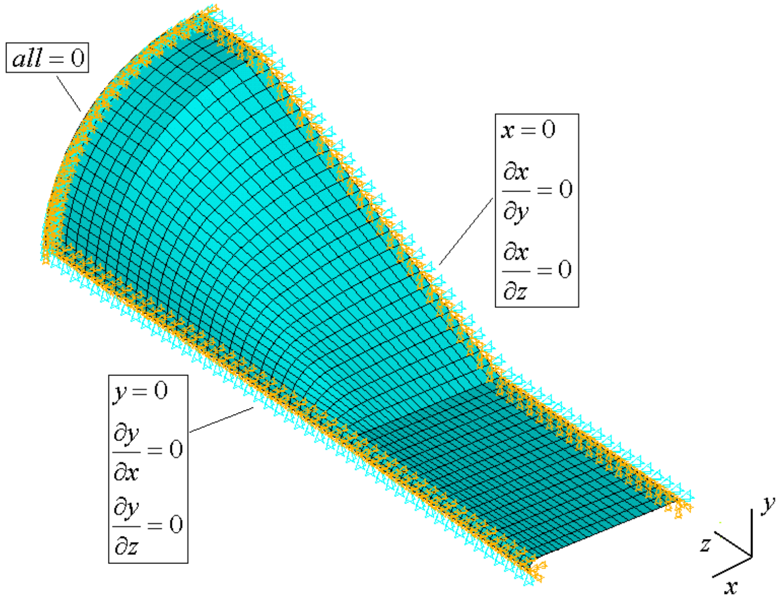


Fig. 3-18 Finite element meshes and boundary conditions of DBV shell model

There are a total of 1344 layered hyperelastic shell elements (48×28) and 1421 nodes with 6 DOFs per node (the independence of mesh density is validated in the sensitivity analysis section of Chapter 5). The layer arrangement model is defined as EPDM / FIBER+54 / FIBER-54 / SBR / SBR / CR from the external layer to the internal layer.

3.5.5 Duckbill closure constraint

In addition to the above boundary conditions, the duckbill portion has to be further constrained. Since a quarter symmetry model is used, the deflection of the duckbill portion of the valve cannot become negative, that is, the y-direction stiffness is governed by the valve material and geometry but the negative y-direction stiffness is infinite. To model this behavior, the elements in the duckbill region of the valve have 'compression only' elements linking them to the x-z plane at y=0. These elements have zero stiffness in tension, thereby permitting unrestrained motion in the positive y-direction. On the other hand, they have very high stiffness in compression, thereby permitting no deflection in the negative y-direction. These compression only link elements are shown schematically in Fig. 3-19.

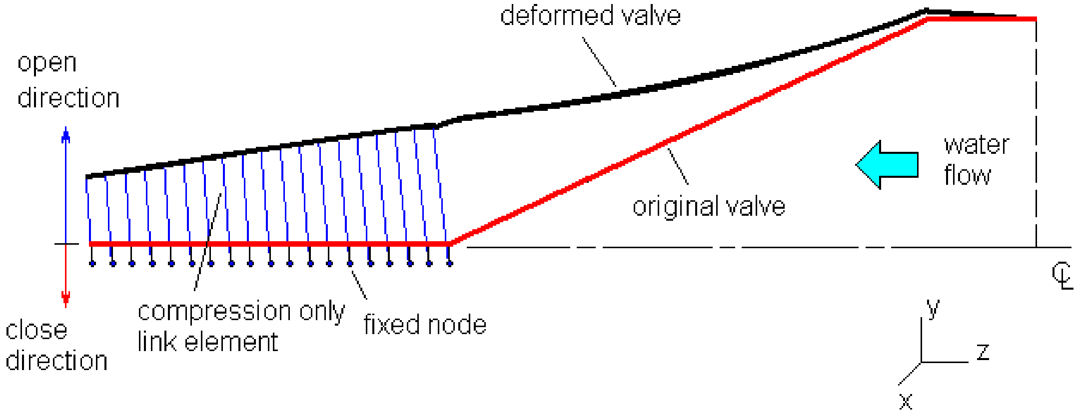


Fig. 3-19 Duckbill stopper constraint in the coupled model

3.5.6 Solution controls of the layered shell model

A transient sparse matrix solver was selected with large deflection, along with auto time stepping ranging from a minimum of 20 sub-steps up to a maximum of 100,000 sub-steps. Equilibrium iteration steps were set to 100, with ramp-up load steps of 100. Time integration for the full transient analysis used the Hilber-Hughes-Taylor algorithm. The L2 norm criteria were used as a convergence check, in which the Square Root of the Sum of the Squares (SRSS) of the force imbalances for all degrees of freedom (ANSYS and CFX user's manual, 2008) was used and the force convergence tolerance value was selected as 5% of the force. The structural damping coefficient (β) can be selected over a range from 0.002 to 0.2 (further discussed in Chapter 5).

3.6 Back pressure modeling of DBV

The duckbill valves are non-return valves that are intended to prevent flow from occurring in the reverse direction. To investigate such a valve deformation behavior under various back pressures, a back pressure model of the DBV was built using the layered shell model but without FSI for there is no flow through the closed valve.

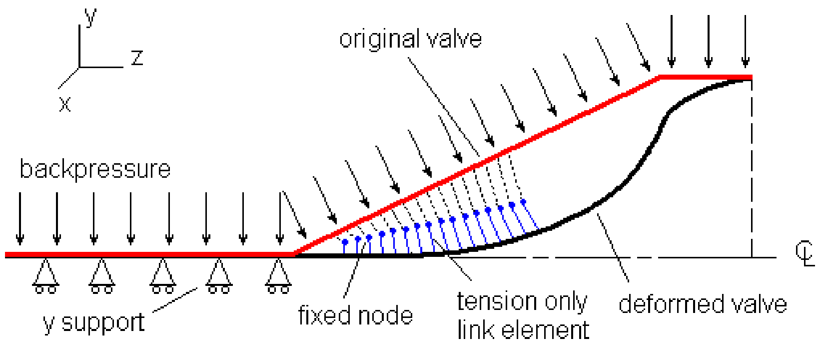


Fig. 3-20 Back pressure modeling of DBV

A simplified approach to avoid the contact issue as the sides of the valve come together at the duckbill and saddle portions was applied as shown in Fig. 3-20. When a back-pressure load is applied to the valve external surface, the duckbill portion is closed and thus its boundary condition is set to zero in the y-displacement (contact issue between the duckbill lips is simplified here). As the back pressure increases, the saddle portion deforms and finally contacts. To simulate such a situation, a tension-only link element (opposite to the compression-only link element) was used to prevent the nodes of the saddle portion from passing through the symmetric x-z plane. The other boundary conditions were the same as those of the coupled model. In addition, the back-pressure loads were assumed to be symmetric in the y-z plane and x-z plane and buckling is not considered in the model.

Valve deformations under various back pressures

Figure 3-21 shows the valve deformations under various back pressures of 7, 14, 21, and 28kPa respectively. The side views of half valve geometry with smaller corresponding isometric views at the corners are shown in the figure. The contour colors represent the total displacements. When the downstream reverse pressure occurs, the duckbill portion closes first as the increasing downstream pressure deforms the valve inward and prevents back flow from developing, as seen in Fig. 3-21 (a) and (b). As the back-pressure load increases, the valve deforms inward further and the internal surfaces of the valve saddle portion begin to touch together as seen in Fig. 3-21 (c). The contact region of the saddle portion continues to increase as the back-pressure rises, as observed in Fig. 3-21 (d). Buckling, asymmetric pressure loading, and contact issues are not considered in the present model. Ideally, the present model can predict the effects of the maximum back pressure upto 28kPa (4 psi or 2.9 mH_2O).

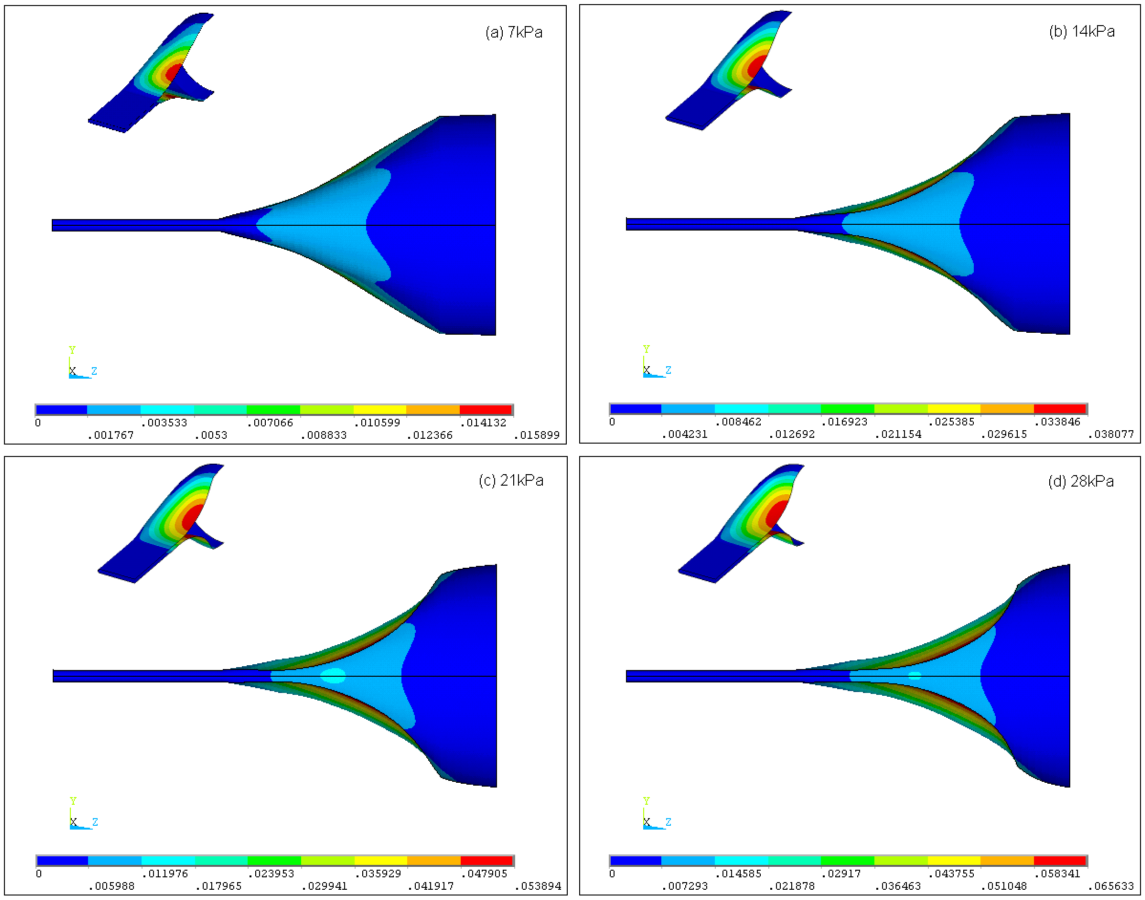


Fig. 3-21 Valve shell deformations under various back pressures

(a) 7 kPa, (b) 14 kPa, (c) 21 kPa, and (d) 28 kPa

Summary

A middle-surface-based-layered-shell (MSBLS) model was developed to predict the valve deformation for the coupling simulation. The utilization of layered nonlinear shell elements, materials, meshes, and boundary conditions were presented and discussed. The model was extended to consider the case of closure end deformation due to increased downstream pressure.

Chapter 4

FLOW MODELS OF DUCKBILL VALVE

An unsteady (transient) 1D ideal flow model, which is used for the simplified FSI model, is presented in this chapter. A transient viscous flow model, based on deformed valve shapes predicted by the simplified FSI model, is also developed for comparison with the transient 1D ideal flow model so as to evaluate the effects of fluid viscosity and downstream pressure recovery.

4.1 Simplification of the fluid flow modeling

The transient 1D ideal flow model is a reasonable approach for flow simplification since the flow through the deformable valve shape is expected to be essentially one dimensional; i.e. there is basically no flow rotation or separation throughout the valve tunnel. It is also expected that the fluid viscosity in the open valve is dominant only in a very thin boundary layer near the valve wall and the thickness of boundary layer can be neglected comparing to the dimensions of valve tunnel. Thus, it is reasonable to assume that the flow through the duckbill valve tunnel can be regarded as an unsteady one dimensional ideal flow.

Such a flow simplification is potentially very beneficial in the present study. First of all, the 3D dynamic volume mesh issue of the viscous flow model is shifted to determine the much simpler 1D dynamic mesh issue. Secondly, the transient 1D ideal flow solver is more compact than the CFD viscous flow solver. It can be developed using an in-house code and can be easily inserted in the FEA solution procedure. Therefore, a monolithic

FSI approach, which solves the equations governing the 1D ideal flow and the layered hyperelastic and orthotropic structure simultaneously within a single FEM solver, seems to be a more time-effective and reliable solution approach than coupling with a 3D viscous flow model.

4.2 Unsteady one dimensional ideal flow model of DBV

Figure 4-1 shows the schematic diagram of the unsteady 1D ideal flow model. The 1D approach is to treat all the flow as just one streamline (i.e. from the inlet (1) to outlet (2) as seen in Fig. 4-1).

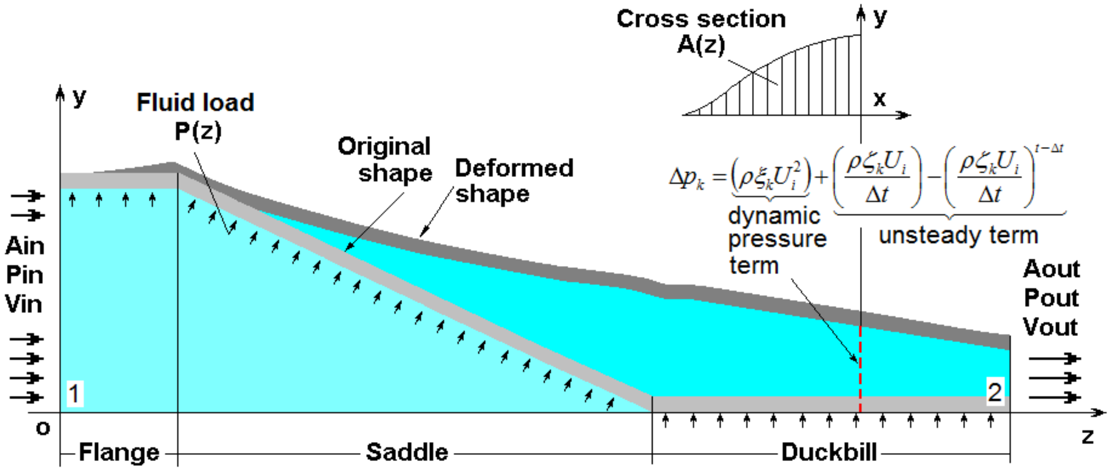


Fig. 4-1 Transient 1D ideal flow model

To formulate the unsteady 1D ideal flow through the valve, it is assumed that the flow is incompressible and that the effects of gravity are negligible. Since the cross-sectional area of valve tunnel is rapidly but smoothly decreasing from the inlet (1) to outlet (2), the flow can be modeled using the unsteady 1D ideal fluid flow equation (Streeter, 1985).

$$\frac{\partial U}{\partial t}ds + d\left(\frac{U^2}{2}\right) + \frac{dp}{\rho} = 0 \tag{4.1}$$

where *U* is the fluid flow velocity at a point on a streamline, *ds* is an element of distance along the streamline, *dp* is the incremental pressure change at the chosen point, ρ is the fluid density, *t* is time.

Integration along the streamline from the inlet (1) to any downstream location z in the valve tunnel leads to the unsteady 1D flow equation for incompressible flows

$$\int_1^z \frac{\partial U}{\partial t} ds + \frac{U_z^2 - U_1^2}{2} - \frac{p_1 - p_z}{\rho} = 0 \tag{4.2}$$

Noting that if the integration is from the inlet to outlet of the valve, the 1D flow equation is written as

$$\int_1^2 \frac{\partial U}{\partial t} ds + \frac{U_2^2 - U_1^2}{2} - \frac{p_1 - p_2}{\rho} = 0 \tag{4.3}$$

Mass continuity equation is given by

$$Q = \rho U_1 A_1 = \rho U_2 A_2 = \rho U_z A_z \tag{4.4}$$

where *Q* is a mass flow rate through the local area *A(z)*, U_1 is the inlet velocity, A_1 is the inlet area, U_2 is the outlet velocity, A_2 is the outlet area, U_z is the velocity at any position z, A_z is the cross section area at any position z.

The local pressure, *p(z)*, along the valve deflects the valve surface as determined by the valve's material properties. This deflection establishes the flow area, *A(z)*, and by mass continuity, the velocity variation through the valve. This velocity determines the pressure which in turn controls the valve deflection. Thus, the problem of determining the mass flux through a DBV for a given pressure drop requires a coupled fluid-structure analysis. For a submerged discharge, it is assumed that there is no pressure recovery downstream of the valve outlet, so that the total pressure at any point in the valve can be determined in terms of the total pressure at the valve outlet. On the other hand, a numerical singularity issue exists when the duckbill lips become fully closed, i.e. when the flow changes from a finite value to a value of zero. To avoid the numerical singularity, it is assumed that there is a very small initial opening (0.0001 m) at the duckbill portion.

Discretization of the transient 1D flow equation

Figure 4-2 shows a schematic of the 1D dynamic mesh (or nodes) distributed along the streamline (i.e. the center line of the valve tunnel). For convenience, nodes are located in terms of coordinates $(0,0,z_k)$, where z_k represents the z-position of the k^{th} node counted from the inlet. The nodes at the inlet and outlet are indicated using subscripts, i and o, respectively.

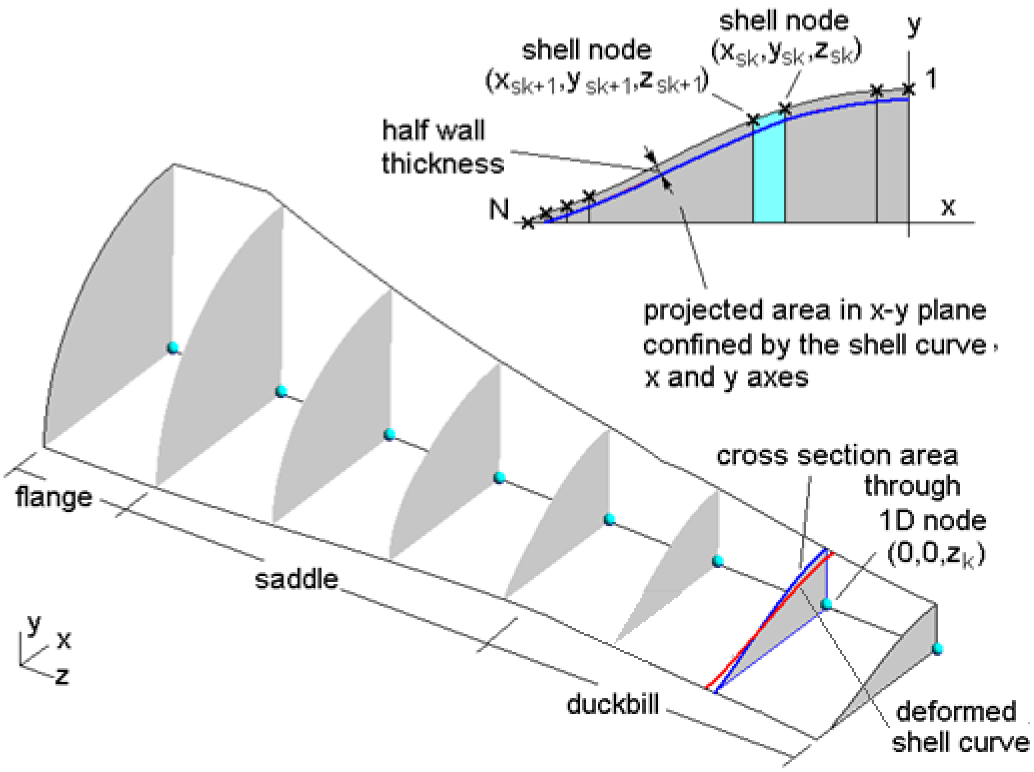


Fig. 4-2 Schematic diagram of the 1D dynamic meshes

Integration of the unsteady term of 1D flow equation is performed using the trapezoidal integration formula and mass continuity equation (4.4).

$$\begin{aligned} S_k &= \int_i^k \frac{\partial U}{\partial t} ds = \frac{\partial}{\partial t}\int_i^k U ds = \frac{\partial}{\partial t}\sum_{z=i}^{z} \frac{z_k - z_{k-1}}{2}\left(U_k + U_{k-1}\right) \\ &= \frac{\partial}{\partial t}\sum_{k=i}^{k} \frac{z_k - z_{k-1}}{2}\left(\frac{A_i}{A_k} + \frac{A_i}{A_{k-1}}\right)U_i \end{aligned} \tag{4.5}$$

where S_k is the unsteady term of the 1D flow equation at node k, U_k and U_{k-1} are the velocities at node k and $k-1$, A_k and A_{k-1} are the cross section areas at node k and $k-1$, z_k and z_{k-1} are the z-locations of node k and $k-1$.

Defining the parameters

$$\xi_k = \frac{1}{2}\left[\left(\frac{A_i}{A_k}\right)^2 - 1\right] \tag{4.6}$$

and

$$\zeta_k = \sum_{k=i}^{k} \frac{z_k - z_{k-1}}{2}\left(\frac{A_i}{A_k} + \frac{A_i}{A_{k-1}}\right) \tag{4.7}$$

The transient 1D flow equation (4.2) can be written in a discrete form as

$$\underbrace{\frac{\left(\zeta_k U_i\right) - \left(\zeta_k U_i\right)^{t-\Delta t}}{\Delta t}}_{\substack{unsteady \\ term}} + \underbrace{\left(\xi_k U_i^2\right)}_{\substack{dynamic \\ pressure \\ term}} - \underbrace{\left(\frac{\Delta p_k}{\rho}\right)}_{\substack{static \\ pressure \\ term}} = 0 \tag{4.8}$$

where superscripts, $t\text{-}\Delta t$ indicates the previous time steps.

Since the inlet velocity, U_i , can be regarded as an independent variable of Equation (4.8) and is larger than zero, it can be solved by rearranging Equation (4.8)

$$U_i = \frac{1}{2\xi_k}\left\{-\frac{\zeta_k}{\Delta t} + \sqrt{\left(\frac{\zeta_k}{\Delta t}\right)^2 + 4\xi_k\left[\left(\frac{\zeta_k U_i}{\Delta t}\right)^{t-\Delta t} + \left(\frac{\Delta p_k}{\rho}\right)\right]}\right\} \tag{4.9}$$

Once the inlet velocity is obtained, the flow rate and outlet velocity are then determined by the mass continuity equation (4.4). The static pressure distribution is then updated by

$$\Delta p_k = \underbrace{\left(\rho \xi_k U_i^2\right)}_{\substack{dynamic \\ pressure \\ term}} + \underbrace{\left(\frac{\rho \zeta_k U_i}{\Delta t}\right) - \left(\frac{\rho \zeta_k U_i}{\Delta t}\right)^{t-\Delta t}}_{unsteady\ \ term} \tag{4.10}$$

To verify that this has all been done correctly, an easy way is to compare the unsteady 1D flow model with the steady 1D flow model. The difference between the two models is that the unsteady term of the former influences the static pressure distribution (4.10). If the coupled FSI model transient has reached a steady state, both of the 1D flow models should produce the same results because the unsteady term is zero in the case of steady flow. The verification is discussed in the sensitivity analysis section of Chapter 5.

1D dynamic meshes

To record the valve deformation and the pressure and velocity fields in the valve tunnel for the current and previous time steps, 1D dynamic meshes or nodes are required to save information of the corresponding node coordinates, cross section area of valve tunnel through the specific node, and local variables (unsteady term values, dynamic pressure (or velocity), and static pressure) of the discrete 1D flow equation (4.8) at the node.

The z-positions of 1D flow nodes ($0,0,z_k$) are determined using the z-positions of valve shell nodes. Since the valve is dynamically stretched due to the variable fluid force acting on the valve wall, the z-positions of 1D flow nodes are changeable in the z-direction. Since the duckbill valve is meshed using a number of four node shell elements, the valve geometry is represented by a number of rows of shell nodes from the valve inlet to outlet after meshing and thus the valve deformation is described by the movement of these shell nodes during the simulation. In Fig. 4-2, the deformed shell curve (the red solid curve) consists of a row of shell nodes. Thus, the z-position of the corresponding 1D flow node is defined by the averaged z-position of the shell nodes on the shell curve, which is expressed as:

$$z_k = \sum_{sk=1}^{N} \frac{z_{sk}}{N} \tag{4.11}$$

where z_k is the z-position of the k^{th} 1D flow node, z_{sk} is the z-position of the sk^{th} shell element node on the k^{th} deformed shell curve, N is the total number of the nodes on the curve.

To determine the pressure and velocity fields in the valve tunnel, the cross-sectional area of the valve tunnel, A_{1D}, through a specific 1D flow node has to be calculated. It is assumed that the projected area, A_{3D}, of the shell curve in the x-y plane with a deduction of a half cross section area, A_W, of the valve wall (see Fig. 4-2) is an excellent approximation to A_{1D}. Noting that A_W is different at the duckbill portion, saddle portion, and round flange portion, A_W is equal to A_{db} for each 1D flow node at the duckbill portion; A_W is equal to A_{fg} for each 1D flow node at the round flange portion; and A_W is equal to an interpolation between A_{db} and A_{fg} for each 1D flow node at the saddle portion. In addition, the projected area, A_{3D}, can be calculated using the trapezoidal integration of the coordinates of the neighbor shell nodes on the shell curve (see Fig. 4-2). Consequently, the cross section area through the 1D flow node is formulated:

for the duckbill portion:

$$A_{1D} \approx A_{3D} - A_W = \sum_{sk=1}^{N-1} \left(\frac{1}{2} \times \frac{y_{sk+1} + y_{sk}}{x_{sk+1} - x_{sk}} \right) - A_{db} \tag{4.12}$$

for the saddle portion:

$$A_{1D} \approx A_{3D} - A_W = \sum_{sk=1}^{N-1}\left(\frac{1}{2}\times\frac{y_{sk+1}+y_{sk}}{x_{sk+1}-x_{sk}}\right) - \left(A_{db} + \frac{z_{sd}-z_{db}}{z_{fg}-z_{db}}\times(A_{fg}-A_{db})\right) \quad (4.13)$$

for the flange portion:

$$A_{1D} \approx A_{3D} - A_W = \sum_{sk=1}^{N-1}\left(\frac{1}{2}\times\frac{y_{sk+1}+y_{sk}}{x_{sk+1}-x_{sk}}\right) - A_{fg} \quad (4.14)$$

In these equations, z_{db} is the z-position of A_W at the connection of duckbill and saddle, z_{fg} is the z-position of A_W at the connection of saddle and flange, z_{sd} is the z-position of A_W in the saddle portion which is between z_{db} and z_{fg}, x_{sk} and y_{sk} are the x and y coordinates of the sk^{th} shell element node on a deformed shell curve, N is the total number of the shell nodes on the curve.

4.3 Viscous flow models of DBV

It is noted that the 1D flow model cannot predict the effects of fluid viscosity and downstream pressure recovery for submerged flows. To validate the 1D flow assumption, a viscous flow model was built using commercial code, CFX, to calculate the 3D transient flow through the same duckbill valve used in the 1D flow model.

4.3.1 SST *k-ω* turbulence model

The Reynolds Averaged Navier-Stokes (RANS) equations (see Chapter 2) were solved using a CFD solver for viscous flow. To close the RANS equations, the *k-ω*/*k-ε* blended shear stress transport (SST) turbulence model (Menter, 1994) was used here.

The *k-ε* model (Jones and Launder, 1972) is robust, computationally inexpensive, and has been widely used in a large variety of different flow situations (Wilcox, 2008). However, it performs poorly for complex flows involving severe pressure gradients (e.g. wall-bounded flow), separation, strong streamline curvature, and lacks sensitivity to adverse pressure gradients (Mentor, 1994 and Wilcox, 2008).

The *k*-*ω* model (Wilcox, 2008) significantly overcomes the shortcomings of the *k*-*ε* model under the flow situations of near wall and adverse pressure gradients by solving one equation for the turbulence kinetic energy *k* and a second equation for the specific turbulence dissipation rate *ω*. However, it is also strongly sensitive to the free-stream values, ω_f, that are specified outside the shear layer (or the inner part of boundary layer) (Mentor, 1994).

The *k*-*ω*/*k*-*ε* blended shear stress transport (SST) turbulence model transitions smoothly switch using blending functions between the *k*-*ω* model (Wilcox, 2008) near the wall and the *k*-*ε* model (Jones and Launder, 1972) away from the wall, avoiding the free-stream sensitivity of the *k*-*ω* model, as well as near wall poor performance of the *k*-*ε* model. Therefore, the SST *k*-*ω* model takes the advantage of the *k*-*ε* model in the outer part of the boundary layer. It improves prediction of flow near the wall, as well as that of adverse pressure gradient and separated flows. Such adverse pressure gradients and flow separation may occur at the duckbill valve outlet towards the downstream water body or piping system.

Original *k*-*ω* model

The transport equations for the turbulent kinetic energy *k* and the specific rate of dissipation *ω* (which is proportional to the turbulent kinetic energy dissipation *ε* over the turbulent kinetic energy *k*) are typically written as (Wilcox, 2008):

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial(\rho u_j k)}{\partial x_j}=\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta^{*}\rho\omega k+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_{k1}\mu_t)\frac{\partial k}{\partial x_j}\right] \qquad (4.15)$$

$$\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho u_j\omega)}{\partial x_j}=\frac{\gamma_1}{\upsilon_t}\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta_1\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_{\omega 1}\mu_t)\frac{\partial\omega}{\partial x_j}\right] \tag{4.16}$$

Transformed *k-ε* model

In order to combine or blend the two turbulence models together, the *k-ε* model is first transformed into a *k-ω* formulation. The transformed form of the *k-ε* turbulence model is given below (Mentor, 1994):

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial(\rho u_j k)}{\partial x_j}=\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta^*\rho\omega k+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_{k2}\mu_t)\frac{\partial k}{\partial x_j}\right] \tag{4.17}$$

$$\begin{aligned}\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho u_j\omega)}{\partial x_j}&=\frac{\gamma_2}{\upsilon_t}\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta_2\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_{\omega 2}\mu_t)\frac{\partial\omega}{\partial x_j}\right]\\&+2\frac{\rho\sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j}\end{aligned} \tag{4.18}$$

Blended *k-ω*/*k-ε* model

The original *k-ω* model is multiplied by a function F_1 and the transformed *k-ε* model is multiplied by a function $(1-F_1)$ and both are added together. The function F_1 will be designed to be one in the near wall region (inner layer) and zero away from the surface (outer layer) (Mentor, 1994). The blended *k-ω*/*k-ε* model is thus given as (Mentor, 1994):

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial(\rho u_j k)}{\partial x_j}=\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta^*\rho\omega k+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_k\mu_t)\frac{\partial k}{\partial x_j}\right] \tag{4.19}$$

$$\begin{aligned}\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho u_j\omega)}{\partial x_j}&=\frac{\gamma}{\upsilon_t}\tau_{ij}\frac{\partial u_i}{\partial x_j}-\beta\rho\omega^2+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_\omega\mu_t)\frac{\partial\omega}{\partial x_j}\right]\\&+2(1-F_1)\frac{\rho\sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j}\end{aligned} \tag{4.20}$$

Let φ_1 represent any constant in the original *k-ω* model in the inner layer and φ_2 represent any constant in the transformed *k-ε* model in the outer layer. Then, φ represents the corresponding constant of the blended *k*-*ω*/*k*-*ε* model, which is given by (Mentor, 1994):

$$\varphi = F_1\varphi_1 + (1-F_1)\varphi_2 \tag{4.21}$$

where the following two sets of constants will be used:

Set 1 φ_1 for the original *k*-*ω* model (inner layer):

$$\begin{aligned} &\sigma_{k1} = 0.5, \quad \sigma_{\omega 1} = 0.5, \quad \beta_1 = 0.075, \\ &\beta^* = 0.09, \quad \kappa = 0.41, \quad \gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}} \end{aligned} \tag{4.22}$$

Set 2 φ_2 for the transformed *k*-*ε* model (outer layer):

$$\begin{aligned} &\sigma_{k2} = 1.0, \quad \sigma_{\omega 2} = 0.856, \quad \beta_2 = 0.0828, \\ &\beta^* = 0.09, \quad \kappa = 0.41, \quad \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2}\kappa^2}{\sqrt{\beta^*}} \end{aligned} \tag{4.23}$$

Blending function

The F_1 blending function is defined by [40]:

$$F_1 = tanh[(arg_1)^4] \tag{4.24}$$

With $$\arg_1 = \min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right), \frac{4\rho\sigma_{\varpi 2}k}{CD_{k\omega}d^2}\right] \tag{4.25}$$

where $$CD_{k\omega} = \max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 10^{-20}\right) \tag{4.26}$$

and ρ is the density; d is the distance from the field point to the nearest wall.

In addition, the turbulent shear stress tensor is given by:

$$\tau_{ij} = \mu_t\left(2S_{ij} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho k\delta_{ij} \tag{4.27}$$

where the shear strain tensor is defined by:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \qquad (4.28)$$

The Shear Stress Transport (SST) model

To significantly improve the calculation for adverse pressure gradient flows, a modified turbulent eddy viscosity is suggested using the SST model (Mentor, 1994):

$$\nu_t = \frac{a_1 k}{\max(a_1\omega, \Omega F_1)} \qquad (4.29)$$

where Ω is the absolute value of the vorticity.

Similarly, the blended function for the SST model, F_2 is given by:

$$F_2 = tanh[(arg_2)^2] \qquad (4.30)$$

where $\arg_2 = \max\left(2\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right)$ (4.31)

Noting that the shear stress is limited to the near wall region, Set 1 φ_1 for the SST model (inner) is slightly changed:

$$\begin{aligned} &\sigma_{k1} = 0.85, \quad \sigma_{\omega 1} = 0.5, \quad \beta_1 = 0.075, \quad a_1 = 0.31 \\ &\beta^* = 0.09, \quad \kappa = 0.41, \quad \gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1}\kappa^2}{\sqrt{\beta^*}} \end{aligned} \qquad (4.32)$$

The original formulation of the eddy-viscosity ($\nu_t = k/\omega$) for free shear layers should be recovered, thus set 2 for the SST model (outer) remains unchanged as (4.23)

4.3.2 Construction of deformed valve geometry

To build the deformed valve geometries for the viscous flow model, inner and outer valve surfaces were created from the deformed middle surfaces predicted by the simplified FSI model under various driving pressures. Some in-house codes were developed for the valve geometric reconstruction.

Figure 4-3 schematically illustrates how to determine the inner and outer surfaces from the deformed middle surface predicted by the simplified FSI model. After a successful FSI simulation, the coupling model (see Chapter 5 for details) can output a deformed valve geometry file, which records the nodal coordinates of the deformed valve middle surface (see the black solid curve in Fig. 4-3). To determine the inner and outer curves (see the red and blue solid curves in Fig. 4-3), each node on the middle surface curve is as the center of a circle (in the x-y plane) with a diameter equal to the valve wall thickness (see the circles in Fig. 4-3). Thus the inner and outer curves consist of the curves tangential to these circles on their upper and lower sides respectively. A small area at the duckbill side edge (see the green area in Fig. 4-3) is filled because it is difficult to generate meshes in such an area. The filled triangular area is 5~6 mm wide, less than 3 mm high, and is less than 0.4% of the entire cross-sectional area. The small area and the very low flow rate, dominated by viscous effects, suggest that the error caused by this filling approximation is very small. A FORTRAN code (see Appendix C) was developed to determine the coordinates of these points of tangency.

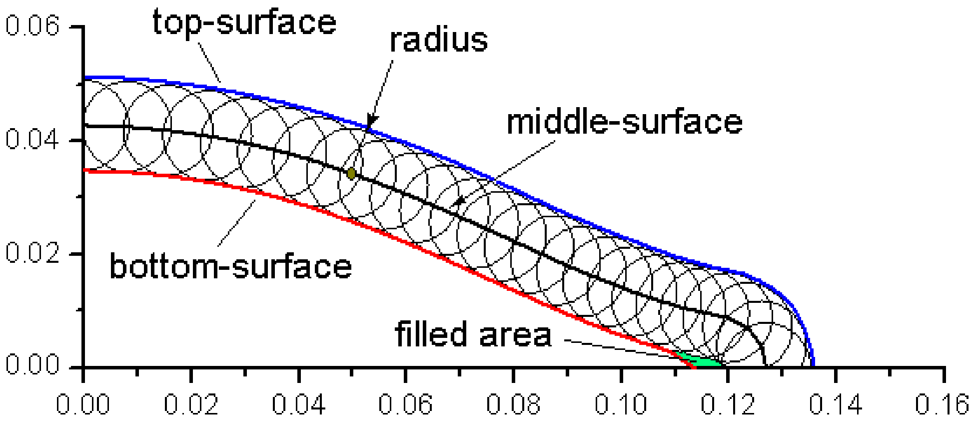


Fig. 4-3 Reconstruction of top (outer) and bottom (inner) surfaces from middle surface with some modifications at the duckbill side edge corner

When these tangent point cloud coordinates representing the deformed valve geometry are obtained using the FORTRAN code, the 3D model for viscous flow (mesh and boundary conditions) can be created using a user-defined mesh generating script, Workbench (see Appendix D). Fig. 4-4 shows the three deformed valve geometries with surface meshes produced using the Workbench user-defined script for the three different driving pressures analyzed.

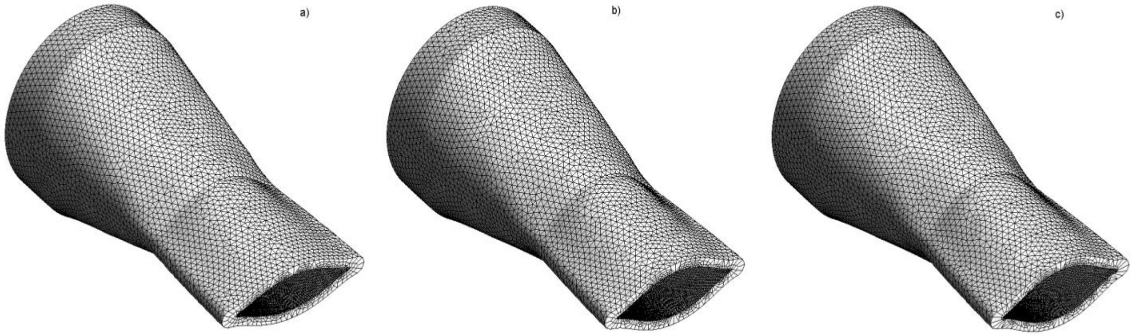


Fig. 4-4 Three deformed valve geometries for viscous flow modeling under increasing driving pressures: a) 18kPa, b) 29kPa, c) 36kPa

4.3.3 Transient viscous flow models

A viscous flow model can provide detailed information to validate the 1D flow model. Therefore, three transient viscous flow models were developed based on the deformed valve geometries predicted by the simplified FSI model, corresponding to the driving pressure drops of 18kPa, 29kPa, and 36kPa, respectively. Fig. 4-5 shows the deformed valve geometry inserted into a larger computational domain including the pipe sections upstream (5-diameters) and downstream (12-diameters) of the valve. The SST *k*-*ω* turbulence model was used due to its success in pipe confined jet flows (Mentor, 1994). Combined unstructured and structured (hybrid) meshes were applied to fill the valve tunnel, as well as the upstream and downstream pipe domains. The solutions were tested for grid independence, which were to be found to be 2,000,000 hybrid cells for the 18kPa geometry, 2,080,000 hybrid cells for the 29kPa geometry, and 2,160,000 hybrid cells for the 38kPa geometry. 5 layers of near wall boundary cells are fine enough to satisfy the requirement of the SST *k*-*ω* turbulence model on the wall boundary (Mentor, 1994). A mass flow rates matching that from the coupled model were specified at the upstream inlet (47.88, 75.52 and 90.82 kg/s for the 18, 29 and 38kPa cases respectively). The downstream boundary was a uniform pressure outlet. The commercial CFD code, CFX, was used for these RANS simulations.

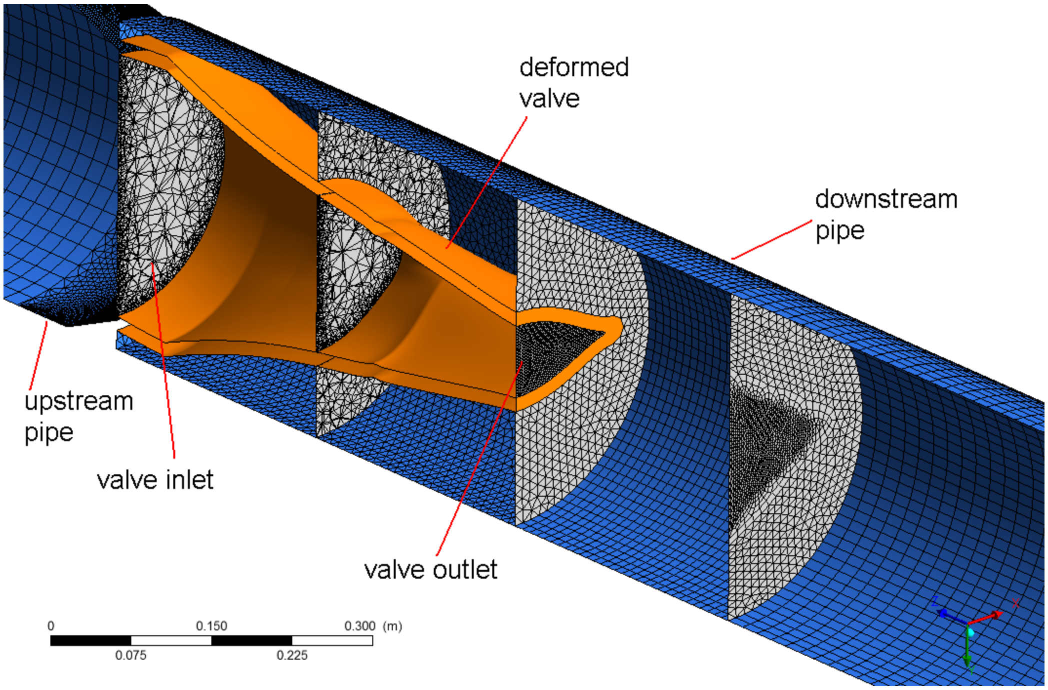


Fig. 4-5 Transient viscous flow model for the 29kPa case (cutoff view)

Summary

A 1D ideal flow model was formulated and a 1D dynamic mesh technique was developed for the simplified coupled algorithm (which will be further discussed in Chapter 5). A geometric reconstruction approach was presented and three transient viscous flow models were developed to validate the 1D simple flow model.

Chapter 5

FLUID STRUCTURE INTERACTION

A coupling approach of the shell model and the transient 1D ideal flow model is presented in this chapter. A user defined code using ANSYS Parametric Design Language (APDL) for coupling algorithm, transient 1D flow solution, and pre/post processing procedures of the simulation is described. Sensitivities of the coupled model to the damping ratio, mesh density, numerical damping, time step, Mullin's effect, valve wall thickness, and the unsteady term of the 1D flow model are analyzed and discussed.

5.1 Coupling methodology

On the solid side of the coupled model, the transient dynamic equilibrium equation [37] of a valve shell node is expressed as:

$$[M]\{\ddot{u}\}+[C]\{\dot{u}\}+[K]\{u\}=\{F\} \tag{5.1}$$

where [M] is structural mass matrix, [C] is structural damping matrix, [K] is structural stiffness matrix, $\{\ddot{u}\}$ is nodal acceleration vector, $\{\dot{u}\}$ is nodal velocity vector, {u} is nodal displacement vector, and {F} is external load vector.

The external load vector, {F}, provides the local fluid force normal to the corresponding element produced by the fluid flow through the valve. Therefore, {F} is a function of shell element size and orientation, local flow velocity (which is a function of nodal z position for 1D ideal flow), cross-sectional area of valve, and time. According to the assumption of inviscid flow, the friction force tangential to the shell elements is neglected here.

On the fluid side of the coupled model, the governing equations of the fluid force presented in Chapter 4 can be rewritten as:

$$\{F\} = \Delta p_k \{A_e\} \tag{5.2}$$

$$Q = \rho U_1 A_1 = \rho U_2 A_2 = \rho U_z A_z \tag{5.3}$$

where Q is a mass flow rate through the local area $A(z)$, U_1 is the inlet velocity, A_1 is the inlet area, U_2 is the outlet velocity, A_2 is the outlet area, U_z is the velocity at any position z, A_z is the cross-sectional area at any position z, ρ is the fluid density, {F} is the external load vector and its direction is always normal to a shell element, $\{A_e\}$ is the area vector of shell element representing the element's orientation in x, y, z, and Δp_k is the static pressure distribution (see Chapter 4).

Given Equations (5.1) ~ (5.3), the cross-sectional area, that is a function of the shell nodal positions and time, can be shown to depend on the external loading; i.e. the local flow pressure and velocity. On the other hand, the local flow pressure and velocity depend on the nodal positions, which determine the size of the cross-sectional area (see the section 4.2 of Chapter 4). Therefore, the transient structural equations (5.1) and the 1D flow equations (5.2) and (5.3) must be coupled and solved simultaneously. In other words, the ideal flow equations and shell model solution must be solved for the same time step.

The coupling procedure of the APDL code that was developed to solve the simplified FSI model is seen in Fig.5-1: Firstly, an upstream pressure is imposed at the outset, the valve deformation is computed solving Equation (5.1) and the initial status is determined by updating the 1D dynamic meshes; i.e. calculating the z-positions and the cross-sectional areas of the 1D dynamic meshes (including the minimum open area of the duckbill portion). Secondly, the inlet and outlet flow velocities are determined solving

Equation (5.2) and (5.3) given that p_2 at the outlet is assumed to be zero (the pressure recovery will be discussed in Chapter 7), p_1 at the inlet is given, and the cross-sectional areas at the inlet and outlet are known. Thirdly, the static pressure distribution along the valve tunnel is updated using Equation (5.2) and (5.3). Thereafter, the new static pressure distribution is imposed onto the valve shell elements (or nodes) for the next iteration run of the valve deformation by solving Equation (5.1). After each time step loop, the newly revised valve shape is returned to the user defined function to calculate the z-positions and cross-sectional areas of the 1D dynamic meshes again. This loop run continues until a steady state is achieved; i.e., for an imposed inlet pressure, the newly computed mass flow rate agrees with that from the last iteration step within an acceptable error.

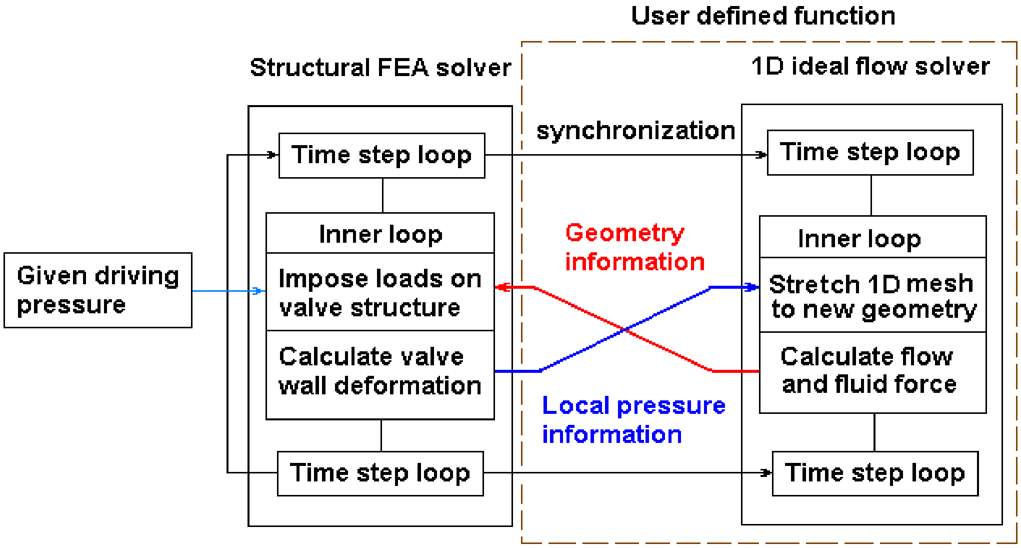


Fig. 5-1 Coupling procedure of APDL code

For some instances, a stable equilibrium could not be found when valve oscillation occurred. In such cases, structure damping physically exists but additional damping is required to force convergence of the simulation. By adjusting the damping, the dynamic response of the valve to the flow (i.e. FIV) could be observed.

5.2 APDL code implementation

An existing FEA solver, ANSYS was used to solve the shell model. The 1D flow solver and coupling algorithm were developed and inserted into the iteration procedure of ANSYS. To implement these user-defined subroutines, ANSYS parametric design language (APDL) was used to develop an in-house code for coupling the shell model and the fluid flow model and for the 1D flow solver. Fig. 5-2 shows the APDL code structure (see Appendix A for detailed information of the code).

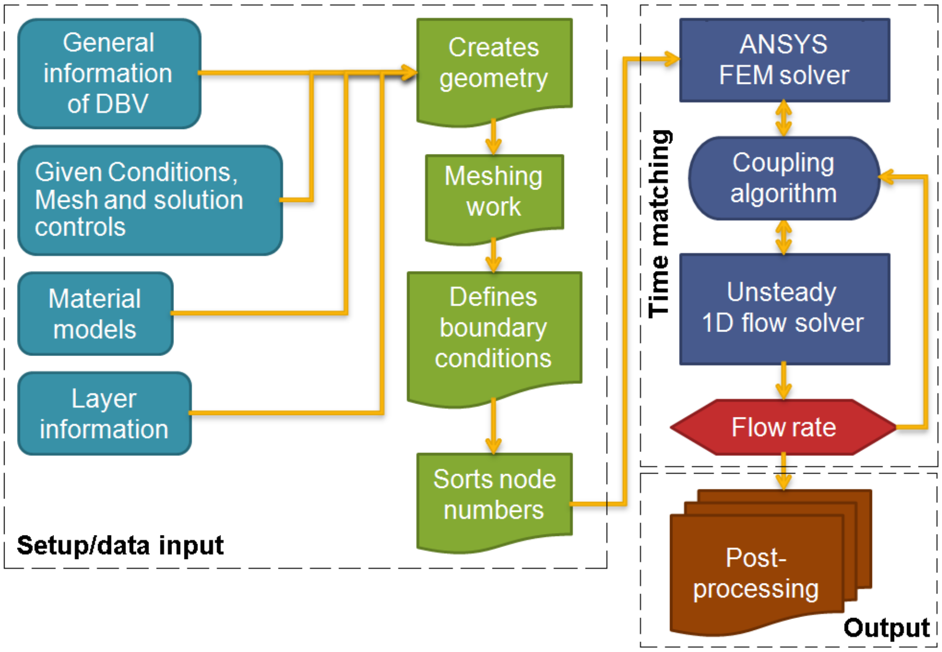


Fig. 5-2 APDL code structure

Since the code was developed for the general application purpose of valve design, the valve design information was provided to the code, including initial driving pressure, general information of duckbill valve geometry (valve length, width, wall thickness, diameter, initial duckbill opening), material layer design information (rubber layer arrangements, fiber angle orientations, layer thicknesses), parameters for the material models (Mooney-Rivlin model and orthotropically linear elastic model), path and directory of the valve geometry data file, mesh generation and solution controls. Hereafter, the in-house code automatically creates the valve geometry, generates meshes, defines boundary conditions, and conducts some preprocessing (node sorting and wall thickness mapping) before starting the FSI simulation (see the setup/data input section of Fig. 5-2).

The coupling algorithm runs the coupling procedure as described in Fig. 5-1 until the outflow rate asymptotically approaches steady value or achieves a stable dynamic equilibrium status if FIV occurs (We tracked the simulation time history to determine if the convergence is acceptable).

In the postprocessing stage, a nodal coordinate file of the deformed valve geometry was saved as output for further investigations; e.g. see the viscous flow simulations (see Chapter 4). A solution history, file recording the relationship between solution variables (mass flow rate, exit area, and velocity) and simulation time, is created, along with a document of the converged pressure, velocity, and cross section area distributions along the valve tunnel.

5.3 Sensitivity analysis

In this section, the sensitivity of the mass flow rate of the coupled model to the valve damping ratios, mesh densities, time steps, and Mullin's effects is analyzed. The influences of the unsteady term of the 1D flow model, composite angle orientations, and valve wall thicknesses on the simulation results are discussed. Hence, the reasonable parameters for the simplified FSI model were determined to obtain reliable and time-efficient simulation results. For the cases of damping, mesh size, and time step, the simulation histories are provided to illustrate the influences of the structural and numerical damping value, mesh size, and time step on the valve mass flow rates. For the case of Mullins's effects, the first loading process and the fifth loading process are compared to demonstrate the effect of the stress softening phenomenon of the valve material on the simulation results. For the case of wall thickness, an averaged wall thickness is compared with a mapped wall thickness based on the measurement. Additionally, for the case of the unsteady term of 1D flow model, the unsteady 1D flow model prediction is compared with that of the steady 1D flow model to verify that the unsteady solution converges to that found assuming steady flow.

Mesh and time step independence

The mesh and time step independences are analyzed here. Fig. 5-3 shows three cases of various mesh density (where 5×8 (coarse), 10×24 (medium), and 36×64 (fine) indicate the mesh numbers in the transverse and longitudinal directions of the duckbill valve). As the meshes become finer the curves are seen to converge. When the mesh density is finer than the medium case, error due to mesh size is considered negligible.

Figure 5-4 shows cases with different time step (which are 1ms, 0.5 ms, and 0.2 ms, respectively), all converging to the same solution. It is noted that there are some irregular or even negative variations in the discharge at the stage of initial valve opening (between 0 and 0.025 s). Recalling the one-way supporting constraint at the duckbill portion discussed in Chapter 3, the valve deformation begins at the flange and saddle portions first due to the initial given pressure loading, while the duckbill portion remains closed. This is seen during the stage up to 0.02 s, as observed in Fig. 5-4. The small negative discharge in the time period of 0.01~0.02 s is an artifact of the initial opening stage of the duckbill being dominated by the artificial small gap and compression only link elements. During this time period, the time step affects the valve opening. When the valve is fully opened, all the mass flow rates converge steadily to the same solution.

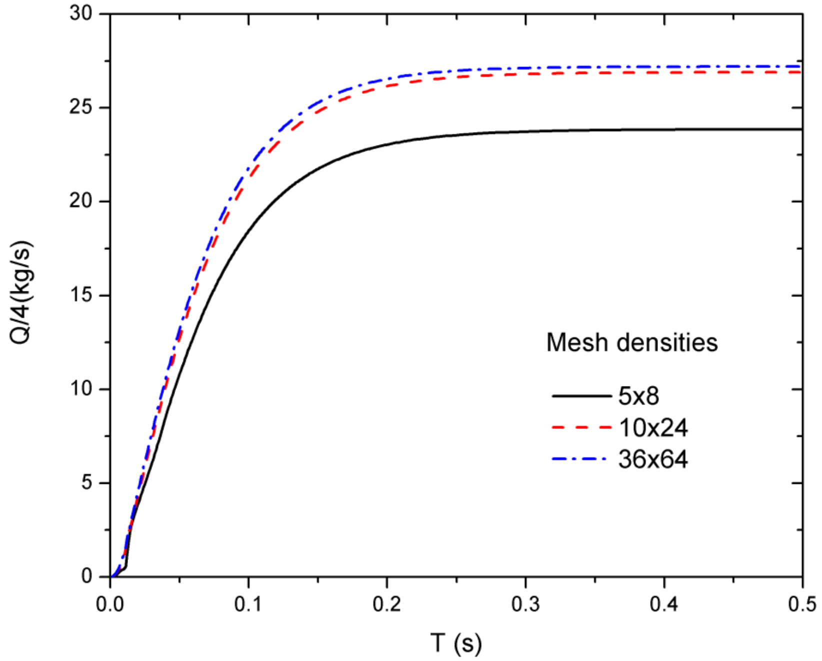


Fig. 5-3 Mesh density independence: with α=3, β=0.003, numerical damping=0.0, time step=0.001s, and driving pressure = 36 kPa

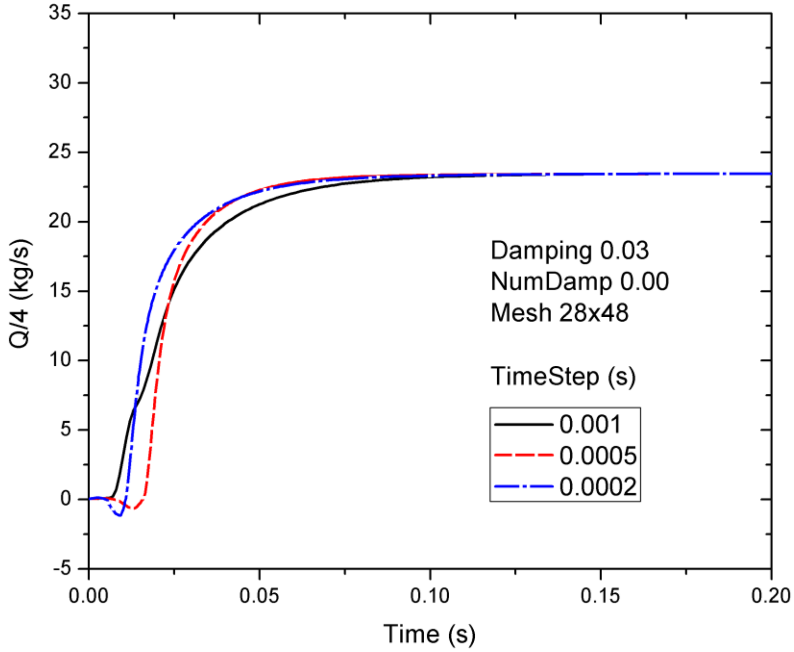


Fig. 5-4 Time step independence: with driving pressure of 29 kPa with α=3, β=0.003, numerical damping=0.0, mesh density=28×48

Influence of structural damping and numerical damping

Various forms of damping are available in the ANSYS program, among which structural damping and numerical damping are the most frequently employed (ANSYS and CFX user's manual, 2008).

To estimate the structural damping matrix, *[C]*, alpha α and beta β constants are specified to define Rayleigh damping, via the damping matrix, *[C]*, is defined as:

$$[C] = \alpha[M] + \beta[K] \tag{5.4}$$

The values of α and β are not generally known directly, but can be estimated from modal damping ratio, ξ_i, which is the ratio of actual damping to critical damping for a particular mode of vibration, *i*. If ω_i is the natural circular frequency of mode *i*, α and β satisfy the following relation

$$\xi_i = \frac{\alpha}{2\omega_i} + \frac{\beta\omega_i}{2} \tag{5.5}$$

To specify both α and β for a given damping ratio ξ_i, it is commonly assumed that the sum of the α and β terms is nearly constant over a range of frequencies (ANSYS and CFX user's manual, 2008). Therefore, the given ξ_i and a frequency range between ω_{i1} to ω_{i2} create two simultaneous equations, which can be solved to determine α and β.

As described by Snowdon (1965), rubber damping ratios depend on the frequency and temperature. However, at lower frequencies range (e.g. from 1 to 20 Hz), the damping ratio is basically frequency and temperature independent (Snowdon, 1965). For example, the averaged damping ratio of SBR is around 0.15 within the frequency range from 1 to 10 Hz. Here, we empirically estimated the valve damping ratio to be in the range from 0.05 to 0.5, based on the loss factor (which is the double of the damping ratio) of rubber materials (Beards, 1996) and its circular frequency range from 2π to 20π (rad/s) (corresponding to the frequency from 1 to 10Hz). Equation 5.5 was used to estimate the damping coefficients presented in Table 5-1.

Table 5-1 Alpha and Beta damping estimations

	Damping ratio	Circular frequency	α	β
Min	0.05	2π~20π	0.57	0.002
Max	0.5	2π~20π	5.7	0.02

Figure 5-5 shows that the flow rate convergent processes under increasing β of 0.003, 0.01, and 0.02. It is seen that the convergence processes are slightly different over the first 0.2s, but the flow rate curve asymptotes to the same solution after that.

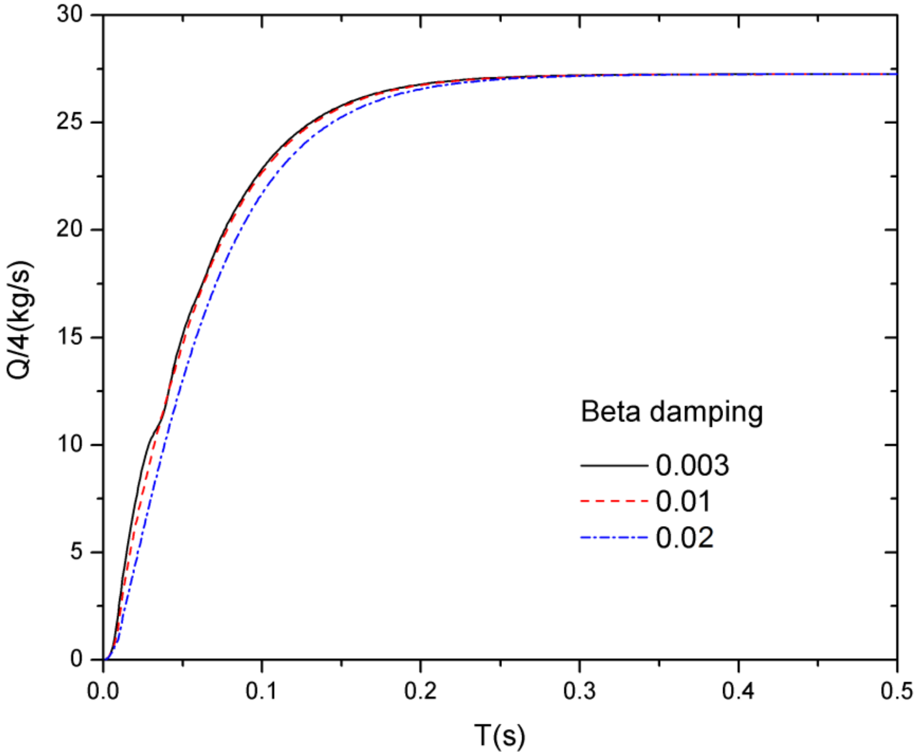


Fig. 5-5 Influences of β on the solutions:

with α=3, numerical damping=0.0, and time step=0.001s

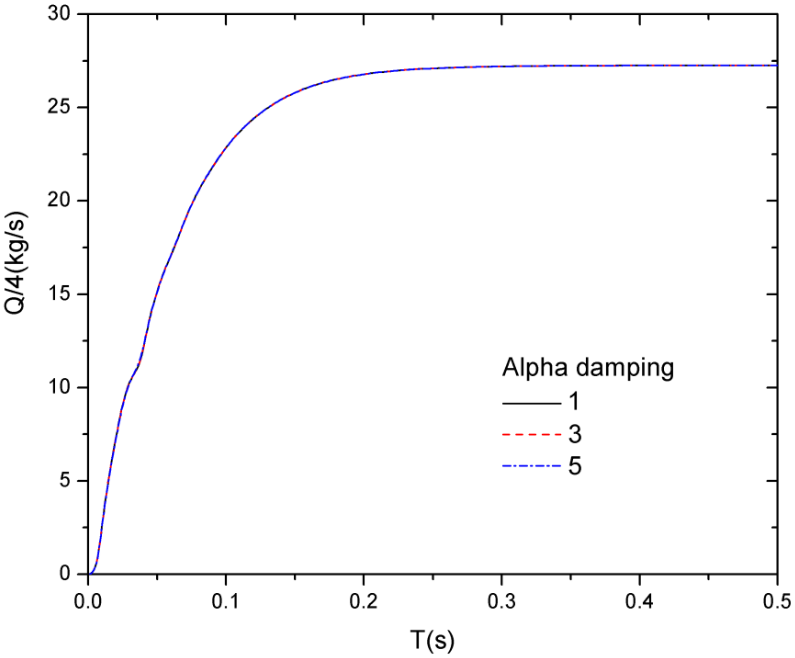


Fig. 5-6 Influences of α on the solutions:

with β=0.003, numerical damping=0.0, and time step=0.001s

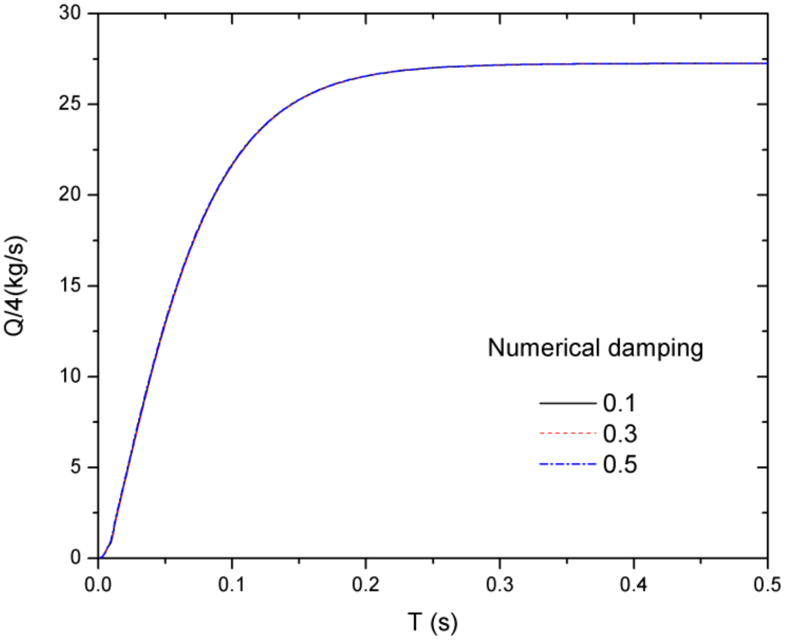


Fig. 5-7 Influence of numerical damping on solution convergence:

with α=3, β=0.003, and time step=0.001s

Fig. 5-6 shows the convergence process with increasing α of 1, 3, and 5. No difference is observed between the predicted values, implying the solution is not sensitive to α.

Numerical damping can be used to stabilize the numerical integration scheme by damping out the high frequency modes (ANSYS and CFX user’s manual, 2008). In other words, numerical damping can only affect the oscillating convergence process of the simulations, and it has little influence on the monotonic convergence curves. It is seen from Fig. 5-7 that the numerical damping seems to have little influence on the convergence of the solutions, even though the numerical damping value varies within the recommended range from 0.1 to 0.5 (ANSYS and CFX user’s manual, 2008).

Mullin's effect

The effect of the stress softening phenomenon, i.e. Mullins's effect (see Fig.3-4 C in section 3.2.1 of Chapter 3), on the valve flow results is analyzed here. As shown in Fig. 5-8, the mass flow rate in the case of the first loading curve employed to obtain the parameters of Mooney-Rivlin model is clearly lower than that when the fifth loading curve is used. The former possesses stiffer rubber properties than the latter.

The Mullins’s effect is seen to be significant over the first 5 loading cycles. Fig.5-8 shows that material softening produces a more flexible valve resulting in an increase in discharge of about 15%. In industrial applications, multiple loadings are expected and the long term material behavior is the one of primary interest. Therefore, the fifth loading curve of each rubber material was used for the material behavior of the present research.

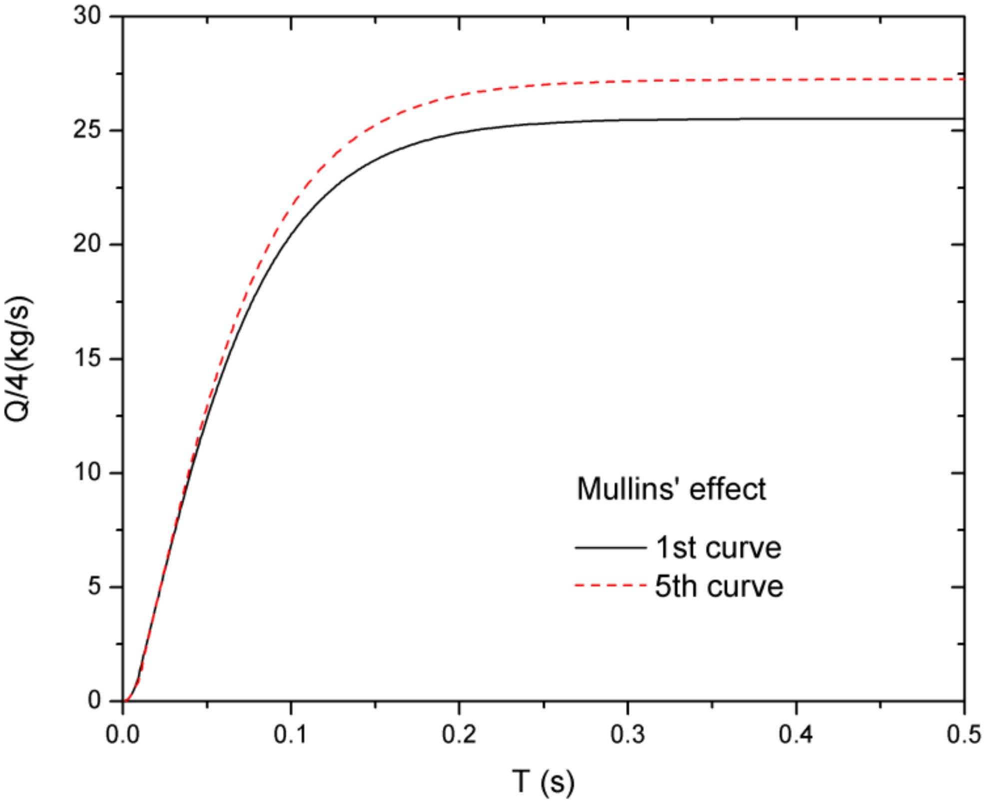


Fig. 5-8 Influence of stress softening phenomenon (Mullins's effect): with α=3, β=0.003, numerical damping=0.0, time step=0.001 s, and driving pressure=36 kPa

Valve thickness effect

As noted previously, the valve wall thickness is actually not uniform (see Fig. 3-3 in Section 3.1 of Chapter 3). The effect of the wall thickness distribution on the valve characteristics is evaluated here by comparing the averaged wall thickness and the wall thickness based on measurements.

Fig. 5-9 shows the flow rate convergence behavior of these two models. It is clear that the valve model with the averaged wall thickness in much stiffer than that with the measured wall thickness. This is attributed to the stiffening of the valve duckbill which reduces the flow rate by about 20% below that of the actual duckbill. Therefore, the measured wall thickness must be used in the coupled model in order to obtain more reliable predictions of valve performance.

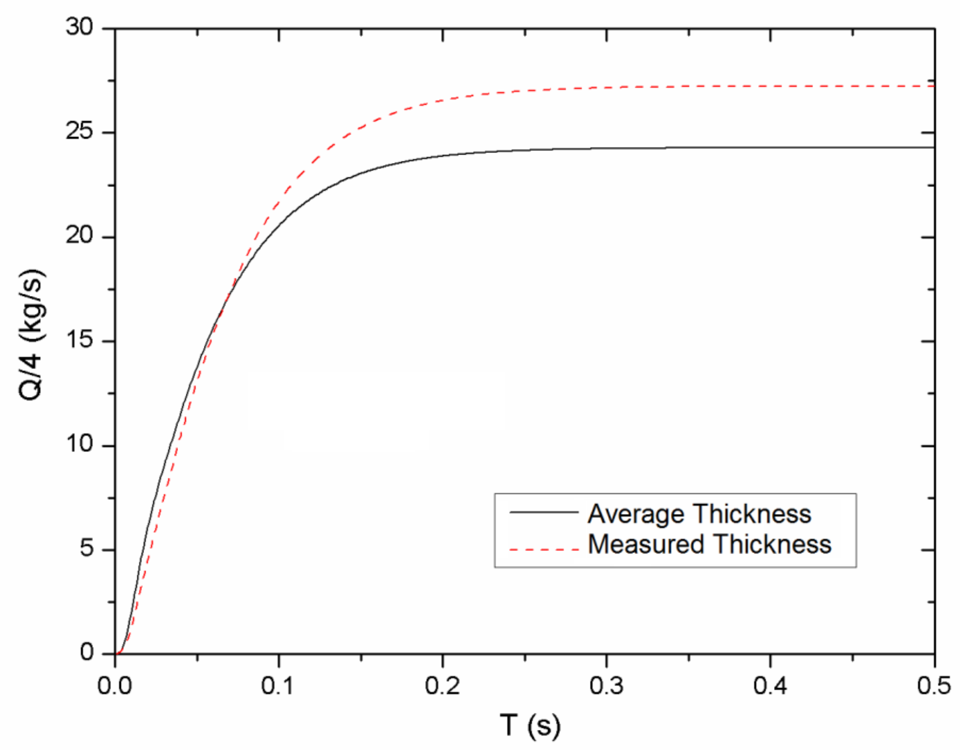


Fig. 5-9 Sensitivity of valve wall thickness to the flow rate: with α=3, β=0.003, numerical damping=0.0, time step=0.001s, and driving pressure=36 kPa

Influences of the unsteady term of the 1D flow model

It was mentioned in Chapter 4 that it would be interesting to determine the effects of the unsteady term in the 1D model of the duckbill FSI. The difference between the two models (with and without the unsteady term) is that the unsteady term of the former (with the unsteady term) influences the static pressure distribution (Equation 4.10 in Chapter 4). If the coupled FSI model transient reaches a stable steady state, both of the 1D flow models should produce the same result because the unsteady term is zero in the case of steady flow. Fig. 5-10 shows the significant difference between the convergence processes of the unsteady and steady flow models under a driving pressure of 36kPa. Fig. 5-10 shows for the steady 1D flow case that the valve opens quickly, overshoots, and goes through decaying oscillations before the flow asymptotes to the steady flow condition for the assumed upstream driving pressure. Interestingly for the unsteady 1D flow case, the initial large oscillations in discharge do not occur, undoubtedly because the inertia of the fluid prevent the rapid acceleration and deceleration of the fluid flowing through the valve associated with large valve vibrations. The steady case asymptotes to the same discharge rate predicted by the unsteady 1D flow model, which indicates that the unsteady term serves to stabilize convergence of the transient solution but does not influence the final flow rate, as one would expect.

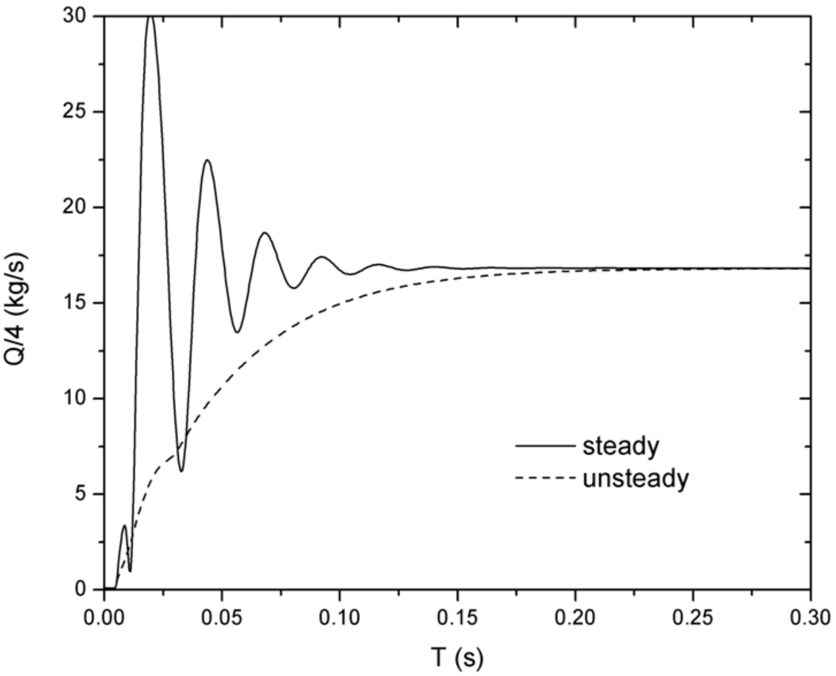


Fig. 5-10 Comparison between the unsteady flow model and steady flow model: uder driving pressure of 36 kPa: Steady 1D flow model (α=0, β=0.003) vs. Unsteady 1D flow model (α=0, β=0.003)

Summary

A coupling approach of the shell model and the transient 1D ideal flow model has been presented. The user defined code using ANSYS Parametric Design Language (APDL) for the entire FSI simulation implementation is described. Sensitivity of the coupled model to the mesh density, structure damping α and β, numerical damping, time step, and Mullin's effect were evaluated. Furthermore, the influence of the valve wall thickness and the unsteady term of the 1D flow model were analyzed and discussed. The sensitivity analysis has confirmed that mesh and solution control parameters can be reasonably determined to carry out more robust, stable, and time-efficient simulations.

Chapter 6

WATER TUNNEL EXPERIMENT

In this chapter, a water tunnel experiment is presented to validate the predictions of the simplified FSI model. The experimental setup is introduced firstly, then a Labview code is developed for data acquisition, and finally data measurement and test operation procedure are described.

6.1 Experimental apparatus

A water tunnel experiment was carried out to verify the predictions of the coupled model and its viscous flow corrections. Fig. 6-1 shows the schematic diagram of 12" diameter water tunnel system.

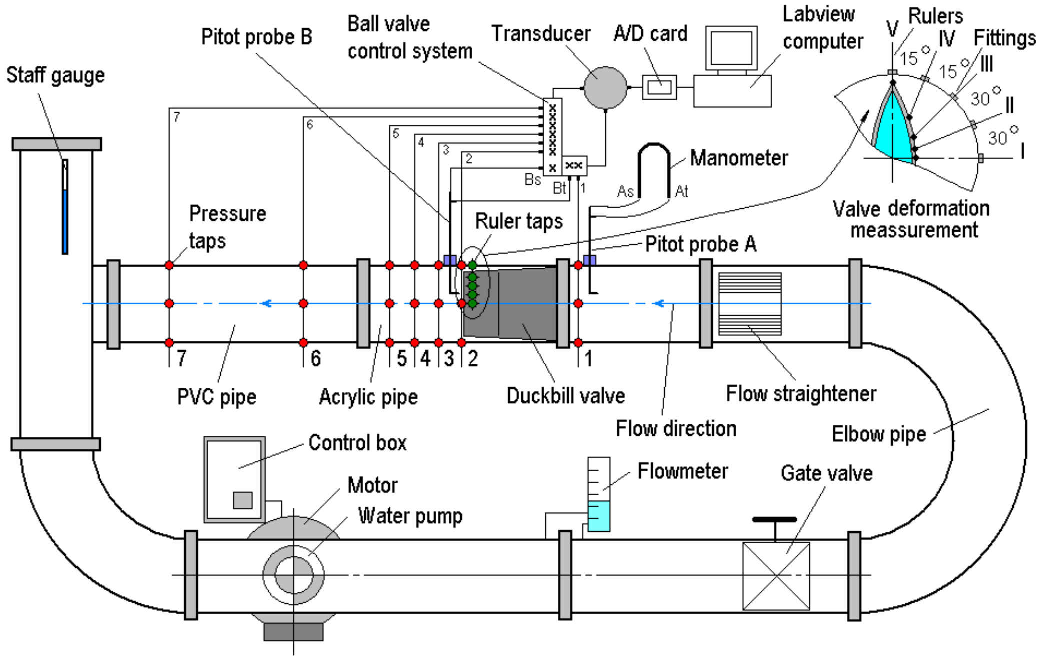


Fig. 6-1 Schematic diagram of 12" diameter water tunnel system

A differential pressure transducer and seven sets of pressure taps were used to measure the static pressure drops between Position 1 and Positions 2 to 7. Each position had 4 taps equally spaced around the test section pipe, circumferentially connected by plastic hoses to obtain the averaged local static pressure. Position 1 is located 0.165 m upstream the valve flange. Position 2 is exactly on a cross-section plane of the test section pipe where the duckbill exit is located. The distance between Positions 2 and 3 is 0.165 m, while Positions 3, 4 and 5 are equally spaced with a distance of 0.15 m. Position 6 is positioned 1.274 m downstream of the valve exit and Position 7 is 2.477 m away from the exit. Two Pitot probes were set close to the inlet and outlet of the duckbill valve. They could be moved vertically to measure the velocity profiles at the valve inlet and outlet. These taps, including the Pitot probes, were all connected by 1/4" or 1/8" plastic hoses to a ball valve system, which was linked to a Validyne DP103 pressure transducer. By switching on/off the ball valves, the required pressure measurement points could be chosen for the experiment. Two diaphragms (Range dash No. 26 and 40 respectively) were selected to cover different pressure ranges and maintain sufficient measurement accuracy. The pressure data acquisition system was calibrated using a Ralston DPPV-0000 pump and HP pressure calibrator before the experiment.

The output signals from the pressure transducer were amplified by a Validyne CD101 signal conditioner and transferred to a NI PCI-6024E A/D card, which was used with a HP desktop computer. An object-oriented code (see Appendix E) was developed using the commercial code, Labview, to analyze the digital signals from the transducer. A real time signal curve, including root mean square (RMS), maximum, and minimum magnitudes, were monitored by the computer and 500 signal samples per time could be saved as an output file.

The water flow in the tunnel was driven by a water pump driven connected to a Cage motor (200 Hp, Max.1780 rpm, 200 F.L.A). The motor could be adjusted by a Toshiba motor control box (600 V, 200 A, 3-Phase, 1~80 Hz). A flow conditioner/straightener was used to produce a uniform approach flow and minimum turbulence intensity. The straightener consisted of a circular birdcage frame with two woven wire screens at the frame ends and with 110 pieces of round aluminum tube (each with 1" diameter and 12" length). Fig. 6-2 shows three velocity profiles near the valve inlet and downstream the flow straightener. It is seen that they are reasonably flat outside the pipe boundary layers, although a trace of the deflection of the flow to the outside of the upstream bend remains.

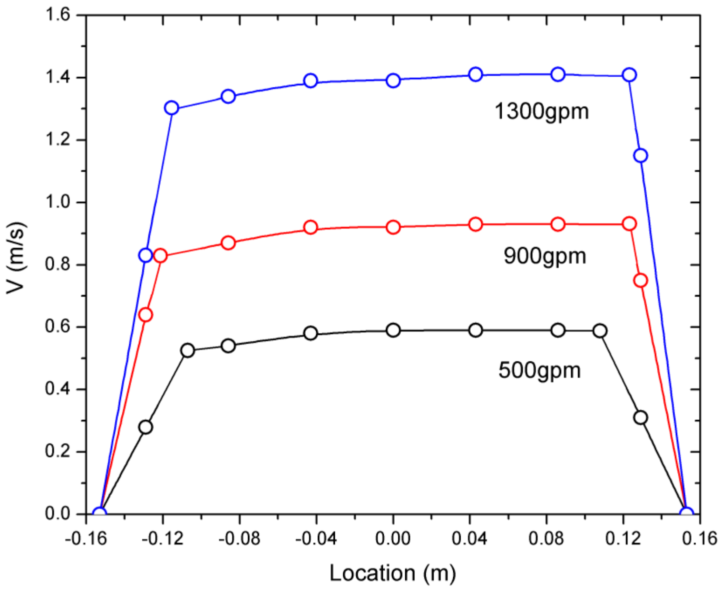


Fig. 6-2 Inlet velocity profiles: corresponding to three flow rates of 500 gpm (31.5 kg/s), 900 gpm (56.8 kg/s), and 1300 gpm (82.0 kg/s)

The overall flow rate in the water tunnel was measured using a Brooks flowmeter, which is a variable area flowmeter that measures volume flow by allowing the flow stream to change the opening within the flowmeter by moving an internal part (buoy) with a flowrate scale. When flow increases, the fluid generates more force and moves the buoy farther. The flowmeter measurement ranges from 100 to 1700 gpm (or from 6.3 to 110 kg/s) with a relative error of 5%.

Since the deformation of the duckbill valve is an essential component of the model, it was measured using five angled ruler scales with sliding probes installed near the valve outlet to determine the valve deformation shape. The sliding probes were located on a cross section plane of the test section pipe, which was 0.045m upstream the valve exit. Based on the assumption of symmetric deformation, only a quarter part of the valve opening was measured. The sliding probes I to V were arranged at angles of 0°, 30°,60°,75° and 90° (see Fig. 6-1) around the circumference of the test section pipe. Fig. 6-3 is a photo of the test section containing the duckbill valve. The five angled ruler scales and static pressure taps with the connecting hoses are seen around the test section.

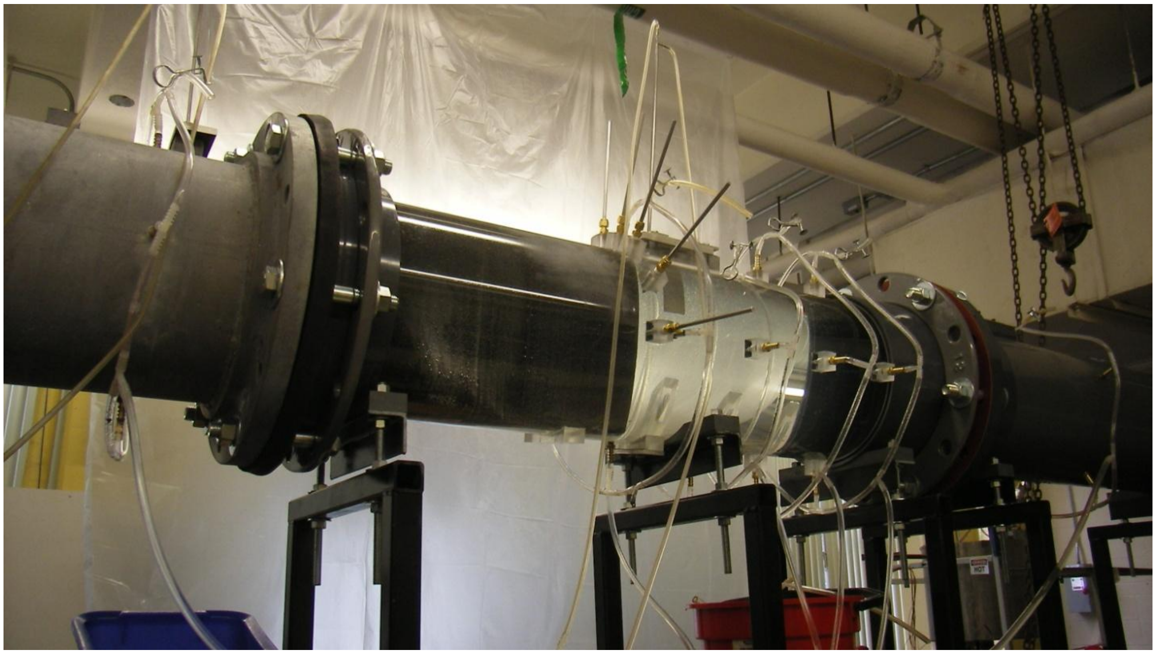


Fig. 6-3 Photo of test section of DBV

6.2 Data measurement

After the pressure drops, *ΔP*, were measured using the pressure transducer, the pressure recovery coefficient could be calculated by:

$$F_i = \sqrt{\frac{P_1 - P_i}{P_1 - P_2}} \tag{6.1}$$

where *i*=2 to 7, F_i represents the pressure recovery coefficient between Position 1 and Position *i*.

The outlet velocity profiles were measured using the Pitot probe and pressure transducer system, while the inlet velocities were measured using a U-tube manometer due to the relative lower flow velocities and therefore lower pressure drops at the valve inlet. The velocity measured by the Pitot probes is defined by:

$$V = \sqrt{\frac{2}{\rho}\left(P_t - P_s\right)} \tag{6.2}$$

where *V* is the velocity, ρ is the fluid density, P_t is the total pressure and P_s is the static pressure.

Figure 6-4 schematically shows how the measured point was determined by the angled ruler scale. Owing to a valve installation deviation or geometric error produced by the flexible rubber, there is usually a translational offset of the duckbill nozzle centerline, *S*, between the valve and the water tunnel pipe centerline. The valve rotational offset in the cylindrical direction was expected to be small due to the location pin on the valve flange. Since the offset exists during the valve installation, the actual measurement point, *p1*, was different from the ideal measurement point, *p2*, on the valve opening. Considering this difference, the coordinates of the point, *p2*, are given by:

$$x = (R - L_1 + L_2 + L_3) \times cos\alpha - S \tag{6.3}$$

$$y = (R - L_1 + L_2 + L_3) \times sin\alpha \tag{6.4}$$

where x and y are the coordinates of $p2$; R is the external radius of the test section pipe; L_1 is the length of ruler scale; L_2 is the length of the ruler scale outside the pipe; L_3 is the thickness of the fitting where the ruler scale is assembled; α is the installation angle of ruler scale; S is the offset distance, which can be estimated by measuring the distance of the test section center and the duckbill exit center after the tested valve is assembled into the test section.

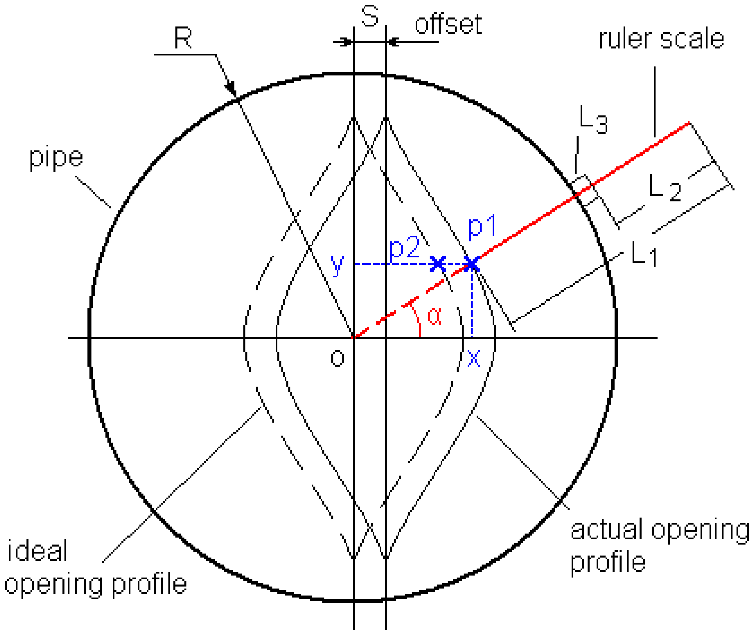


Fig. 6-4 Deformation measurement of the duckbill opening

6.3 Operation procedure

The water tunnel operation procedure refers to the mechanical engineering water tunnel laboratory standard operating procedure (SOP) of McMaster University. Here, the important remarks on the experiment are briefly described.

Firstly, the air inside the loop had to be bled out by driving the water flow (running the water pump) and loosening the clampers of soft hose (see Fig. 6-2) to ensure that these hoses were also filled with water. Several preliminary runs for the water flow in the tunnel were necessary to load and unload a new valve in order to eliminate the Mullins' effect of the rubber.

Secondly, since the flowrate could be easily adjusted by the motor controller, the pressure, velocity, and valve deformation measurements were conducted based on several fixed flowrates. The relationship of the rpm of the motor and controller frequency is given by: rpm = freq (Hz) $\times 1785 \times 0.75/60$.

Thirdly, two different valves were tested in the water tunnel and the full set of measurements for each valve was repeated three times.

Summary

A water tunnel experiment for validation of the simplified coupled model is presented in this chapter. The valve inlet and outlet velocity profiles, pressure drops along the water tunnel, and valve outlet deformation are designed to be measured by a data acquisition system. System deviations of the data measurement are analyzed. The test operating procedure is described.

Chapter 7

RESULTS, COMPARISON, AND DISCUSSION

In this chapter, the theoretical and experimental results are presented and compared. The head-discharge, area-discharge, and velocity-head relationships representing the overall hydraulic characteristics of the DBV are discussed first. The pressure and velocity distributions along the valve tunnel predicted by the 1D flow model and the 3D viscous flow model are compared with the water tunnel experiments. The valve deformations, stress/strain distributions, fluid viscosity effects, and downstream pressure recovery are also compared and discussed. Additionally, fluid flow induced oscillations of the DBV at small openings for a specific situation are analyzed and discussed.

7.1 Hydraulic characteristics of the DBV

Strain level estimation

Four strain level curves for the rubber properties were obtained from the uniaxial tension tests (see Chapter 3). The most appropriate curve which should be employed for the coupled model needs to be estimated before the simulation. The strain level was estimated using the ratio of arc length and chord length (as seen in Fig. 7-1) since the potential maximum strain was expected to exist at the duckbill portion. In Fig. 7-1, the duckbill nozzle opening was simply approximated by the arc height, H_{arc}. Similarly, the original duckbill span was approximated by the chord length, L_{chord}, and the deformed duckbill length was represented by the arc length, L_{arc}. Thus, the strain level was defined by the ratio of arc length and chord length minus one as follows:

$$Strain = \frac{L_{arc}}{L_{chord}} - 1 \tag{7.1}$$

where $L_{arc} = 2R \times arcsin\left(\frac{L_{chord}}{2R}\right)$ and $2R = H_{arc} + \frac{L_{chord}^2}{4H_{arc}}$

For the case with an arc height of 50mm (we assumed that the duckbill opening was less than 50mm) and a chord length of 250mm, the stain is equal to about 10%. Thus, the 10% strain level curve was considered to be the most appropriate for use in the coupled model (Lee et al. (2004) also estimated that his rubber valve had strains up to about 10%).

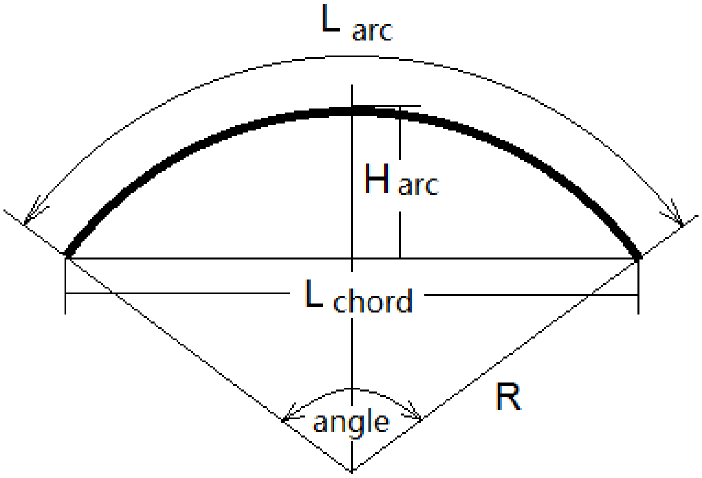


Fig.7-1 Strain level estimation using the ratio of arc length and chord length

If the estimated strain level is 15%, for instance, a curve interpolation can be used between the 10% and 20% strain levels (available from the uniaxial tension test results). Fig. 7-2 shows the head-discharge relationships predicted by the coupled model with the two rubber strain levels (i.e. 10% and 20%). The curve based on the 15% strain level can simply be obtained by interpolation. It is seen that, for a given driving pressure, doubling the estimated strain level from 10% to 20% causes about a 12% increase in the predicted flow rate. This is because the rubber is nonlinear softening so its effective stiffness reduces at larger strain levels

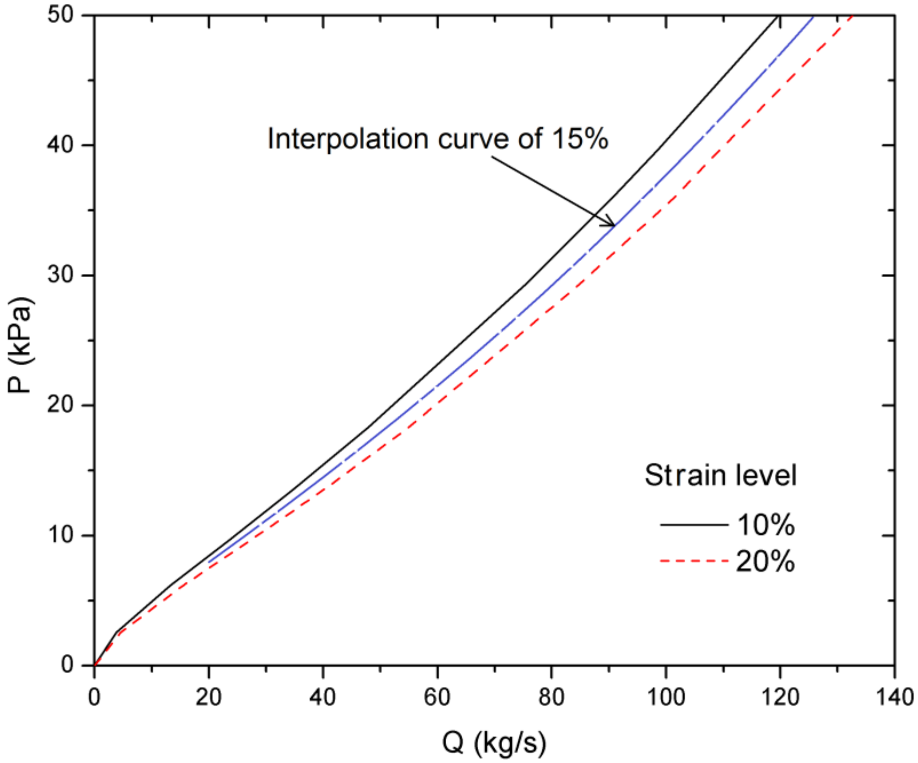


Fig.7-2 Head-discharge relations of the DBV: predicted based on the 10% and 20% strain levels vs. the interpolation curve of 15%

Influence of fiber layer angle orientation

The fiber layer angle orientation can affect the simplified FSI simulation results. The head-discharge relations of the valve with different fiber orientations and within a pressure range from 0 to 50 kPa are shown in Fig. 7-3, which demonstrates the sensitivity of fiber angle arrangement on the hydraulic performance of the DBV. The unidirectional fiber/resin plies are layered in $\pm 36°, \pm 45°, \pm 50°,$ and $\pm 54°$ (designed angle) along the longitudinal direction of the valve. The case without fabric reinforcement is also included as a reference. Fig. 7-3 shows that the head-discharge curves become steeper as the fiber layer angle increases. The effect of the fiber layer is to stiffen the valve and therefore reduce the discharge for a given driving pressure. This effect increases with fiber orientation angle.

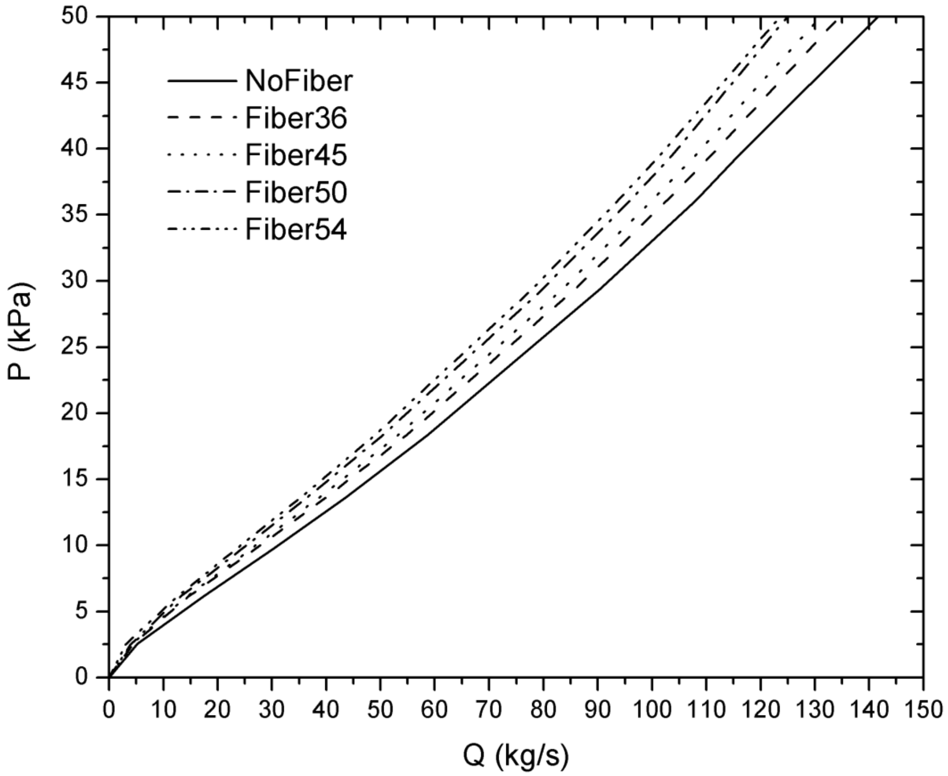


Fig. 7-3 Influence of fiber orientations on the valve head-discharge performance

7.1.1 Head-discharge relationships

For convenience of comparing the simulations with the experiments, the driving pressure drops across the valve used in the simplified FSI model predictions are the same as those measured from the water tunnel experiments. Fig. 7-4 shows the relationship between the pressure drop and the mass flow rate for the duckbill valve, including both the theoretical predictions and the experimental results. The data points for each of the two valves represent the average of three repeated experiments for each valve. The two different valves show good agreement with each other, the results being within 5% of each other.

The experimental results may be considered to vary roughly linearly over a wide pressure range (noting that the head-discharge behaviour of the valve was purely linear in the previous work by Lee et al. (2004)), where an increase in *ΔP* results in an increase in Q not just due to driving pressure but also due to valve area opening (Recall that this head-discharge characteristic contrasts sharply with rigid valves which typically have a pressure drop which increases with the square of the flow rate, $\Delta P \propto Q^2$). The experimental results also show that the slope of the mass flow rate curve increases slightly at higher flow rates implying that the valve opening area does not continue to increase with increasing pressure so quickly due to its nonlinear stiffening behaviour.

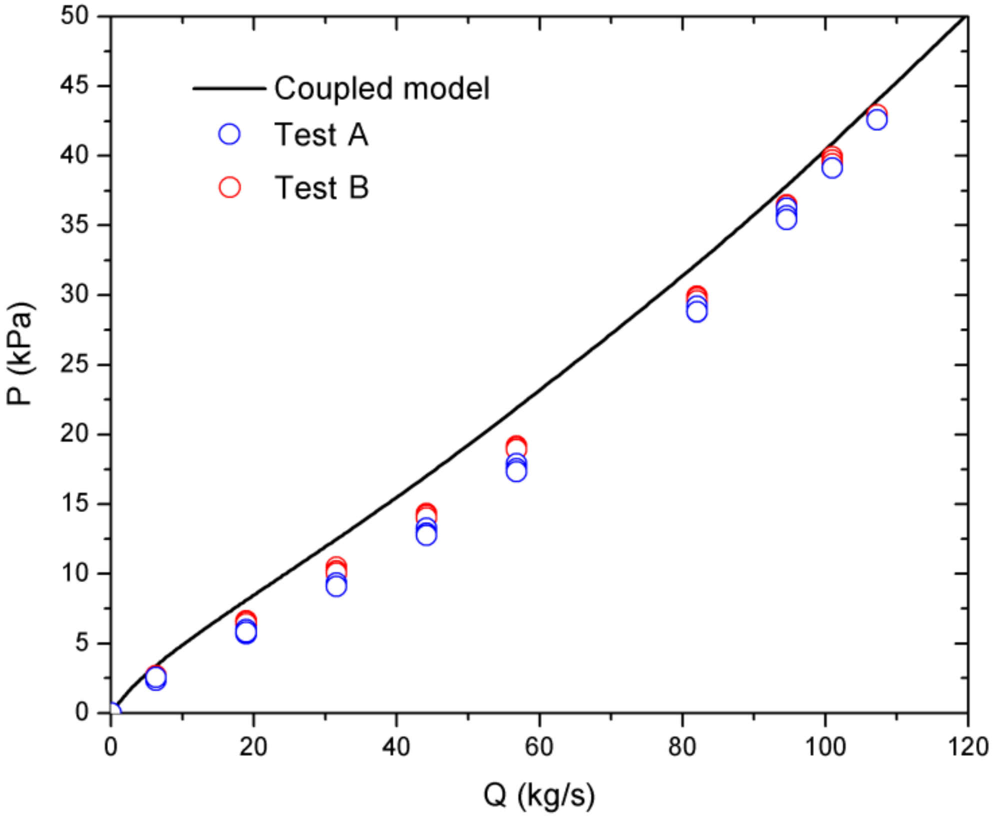


Fig. 7-4 Pressure drop-discharge relationships

the coupled model predictions vs. the experimental results

Interestingly, the valve model with the 10% strain level also predicts roughly linear results with the average deviation being about 10% (minimum 3% but maximum 20%) larger than the water tunnel results. This implies that the estimated strain level is too low and the 15% strain level curve may be a better approximation. It is also noted that the valve model predictions approach the experimental results at larger flow rates, clearly better than at smaller flow rates. Comparing Fig. 7-3 and Fig. 7-4, it is found that the experimental results approach the valve model without fabric reinforcement at smaller flow rates (lower driving pressure drops), but approach the valve model with a fiber layer at higher flow rates. This suggests physically that the valve performance depends primarily on the mechanics of the rubber bending deformation under lower pressure loadings, while the fabric extension reinforcement dominates under higher pressure loadings. This makes sense because the relatively stiffer fiber layer is located near the neutral axis of bending and therefore will have little influence on bending stiffness but dominates in-plane stretching. It follows that neglecting the fiber layer in modelling valve deformation due to significant pressure drops can lead to serious errors, contrary to the assumption of Lee, et al. (2001). On the other hand, the present model with the 10% stain level slightly overpredicts valve stiffness when the pressures are low, leading to under-prediction of valve discharge.

7.1.2 Area-discharge and head-velocity relationships

Two approaches were employed to evaluate the valve opening areas. The first was from the direct deformation measurements of the valve openings (see test A and test B in Fig. 7-5), while the second was to estimate the opening areas using the measured outlet velocities and flow rates (see esti A and esti B in Fig. 7-5). The solid curve seen in Fig. 7-5 is the prediction of the coupled model with the 10% strain level and fiber layer angles of $\pm 54°$. It is observed that the simulation results match reasonably well those of the test valve A within an averaged relative difference of 6.6%, while noting that there are larger differences in the results for the test valve B within an averaged relative difference of 16.1%. This may be attributable to the imperfectly symmetric valve openings, with the displacement in a quarter of the valve being measured (see Chapter 6). On the other hand, the estimated areas based on the measured outlet velocities and flow rates are seen to approach the solid curve predicted by the valve model within averaged relative differences of 6.5% and 4% for valves A and B, respectively. It appears that the direct area measurement methodology introduces significant measurement error.

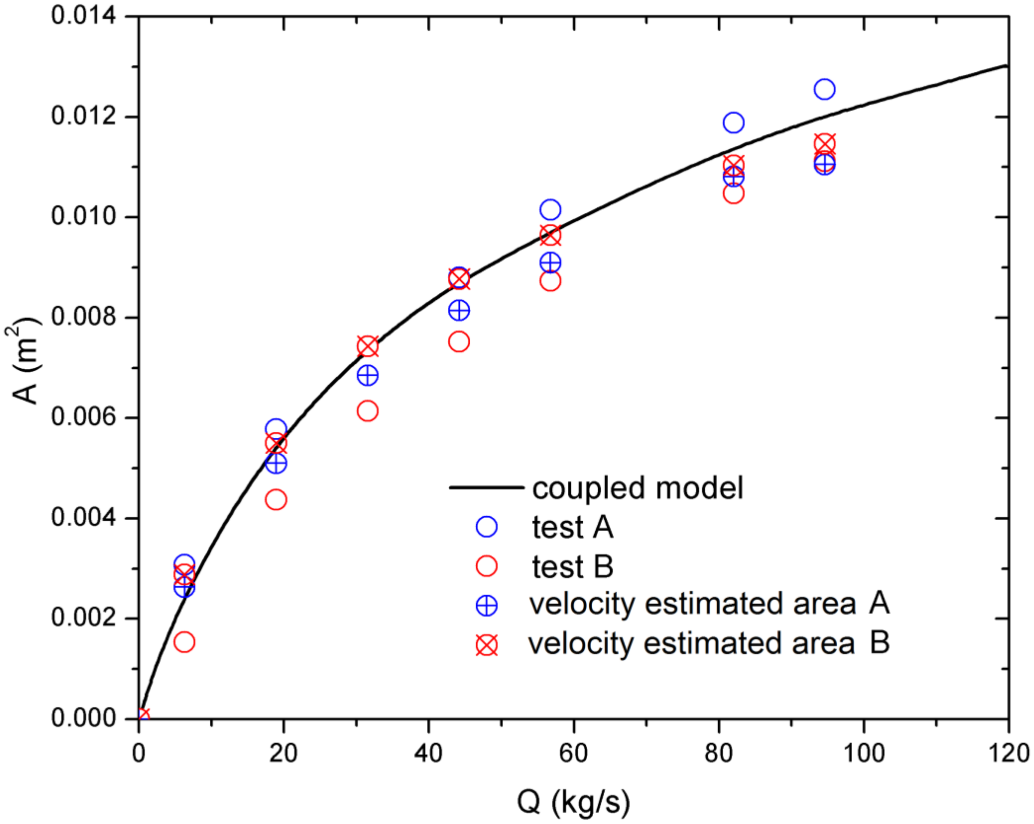


Fig. 7-5 Area-discharge relationships

The coupled model predictions vs. the direct measurements of duckbill openings and the estimations based on the measured outlet velocities and mass flow rates

Figure 7-6 shows the measured valve outlet velocity profiles compared with the predictions of the 1D and 3D flow models. The geometrical models were built based on the predictions of the simplified FSI model with the 10% strain level and fiber layer angles of ±54°. Three pressure drop cases for each of two valves were used in the comparisons, where driving pressure drops of 18, 29, and 36 kPa are considered. Three sets of data are shown in Fig. 7-6 to demonstrate repeatability. It is seen that the predictions of the 1D flow model matches well with the experimental results for each case, while the 3D flow model predicts slightly higher flow velocities. This is attributed to the influence of the viscous boundary layer near the valve wall.

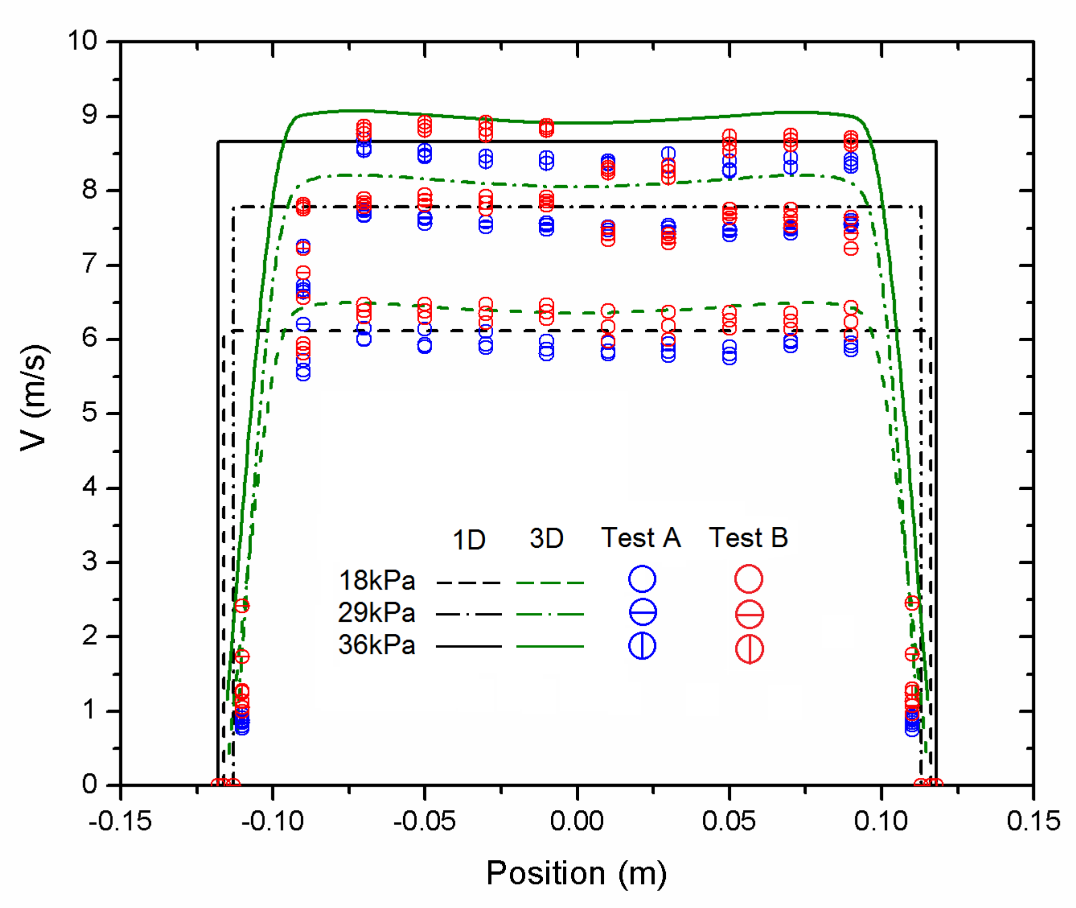


Fig. 7-6 Outlet velocity profiles of measurements, 1D and 3D predictions

Figure 7-7 shows the pressure drop-velocity relationship of the DBV. Since the weight-averaged (area-averaged) method is impractical for calculating the average velocity at the valve outlet, all the measured velocities of each valve outlet area are included here but the data near the valve wall is excluded. There are total of seven sets of velocity data for each valve. It is seen that the 1D flow predictions agree very well with the experimental results.

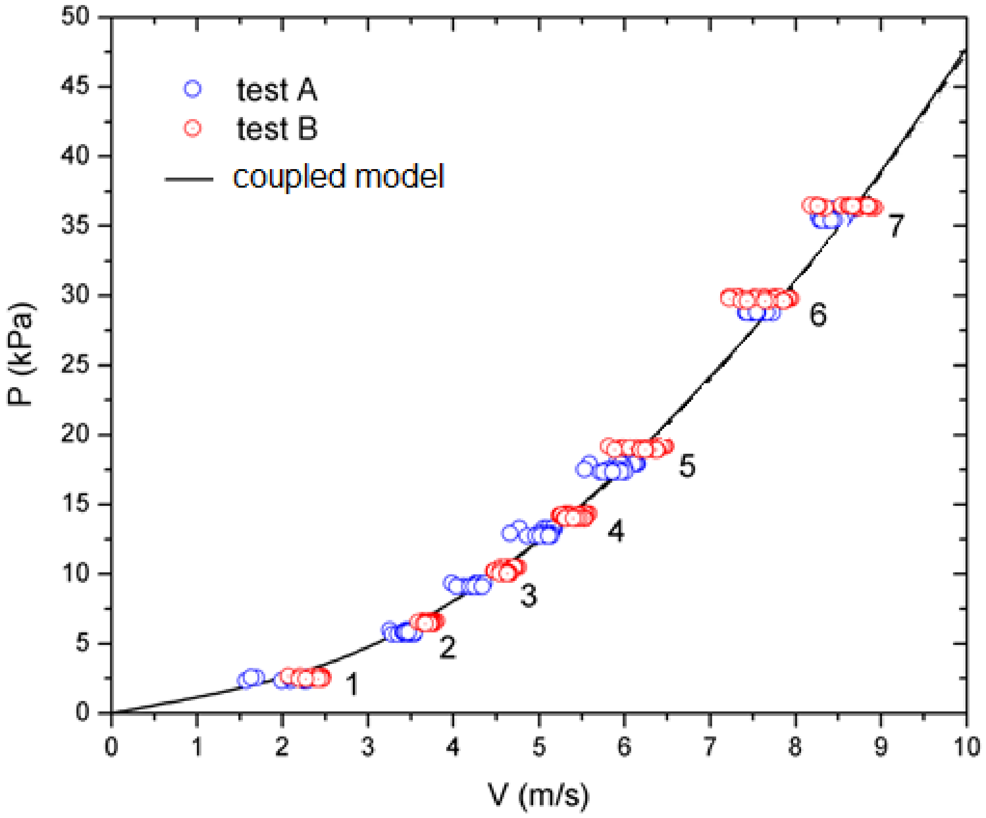


Fig. 7-7 Pressure drop-velocity relations

7.2 Pressure and velocity distributions along the DBV tunnel

The flow velocity and pressure distribution along the valve tunnel were calculated based on the non-recovery assumption of the velocity head at the valve exit. It should be noted that pressure recovery only occurs if the outlet jet expands from a submerged valve exit. In the present case, the valve exit was centered in the water filled downstream pipe. For a comparison with the 3D viscous flow predictions, the 1D flow through a sudden expansion of a horizontal pipe can be accounted for by including a pressure recovery term based on the valve open area, which is a standard *K* factor for a flow sudden expansion to a non-infinite area (Streeter, 1985). The equation for the recovery term is given as:

$$p_2 - p_1 = \left[K_i - K\right]\frac{\rho v_1^2}{2} \tag{7.2}$$

where K_i is the ideal pressure recovery coefficient, which is equal to $1-(A_1/A_2)^2$, K is an experiment dependent loss coefficient, which is a function of expansion ratio of A_1 to A_2 (Streeter, 1985), p_1 is the pressure at valve outlet, v_1 is the mean flow velocity at valve outlet, A_1 is the cross-sectional area of valve outlet, ρ is the water density, and p_2 and A_2 are the corresponding flow quantities after the expansion, i.e. in the downstream pipe.

Velocity distributions along the valve tunnel

Figure 7-8 compares the velocity fields from the 3D viscous CFD results and the 1D ideal flow model with increasing driving pressures of 18 kPa, 29 kPa and 36 kPa. The velocity contours of the 3D model are in the central y-z planes of the DBV and the downstream velocity contours are also included. It can be clearly seen that the flow is essentially one dimensional through the valve, with the obvious exception of a thin boundary layer associated with the effect of fluid viscosity. Note that the velocity boundary layers are insignificant (as expected) and the velocity distribution predicted by the 1D flow model match well that of the 3D flow model indicating that the 1D assumption is a reasonable model for predicting the flow through the valve tunnel.

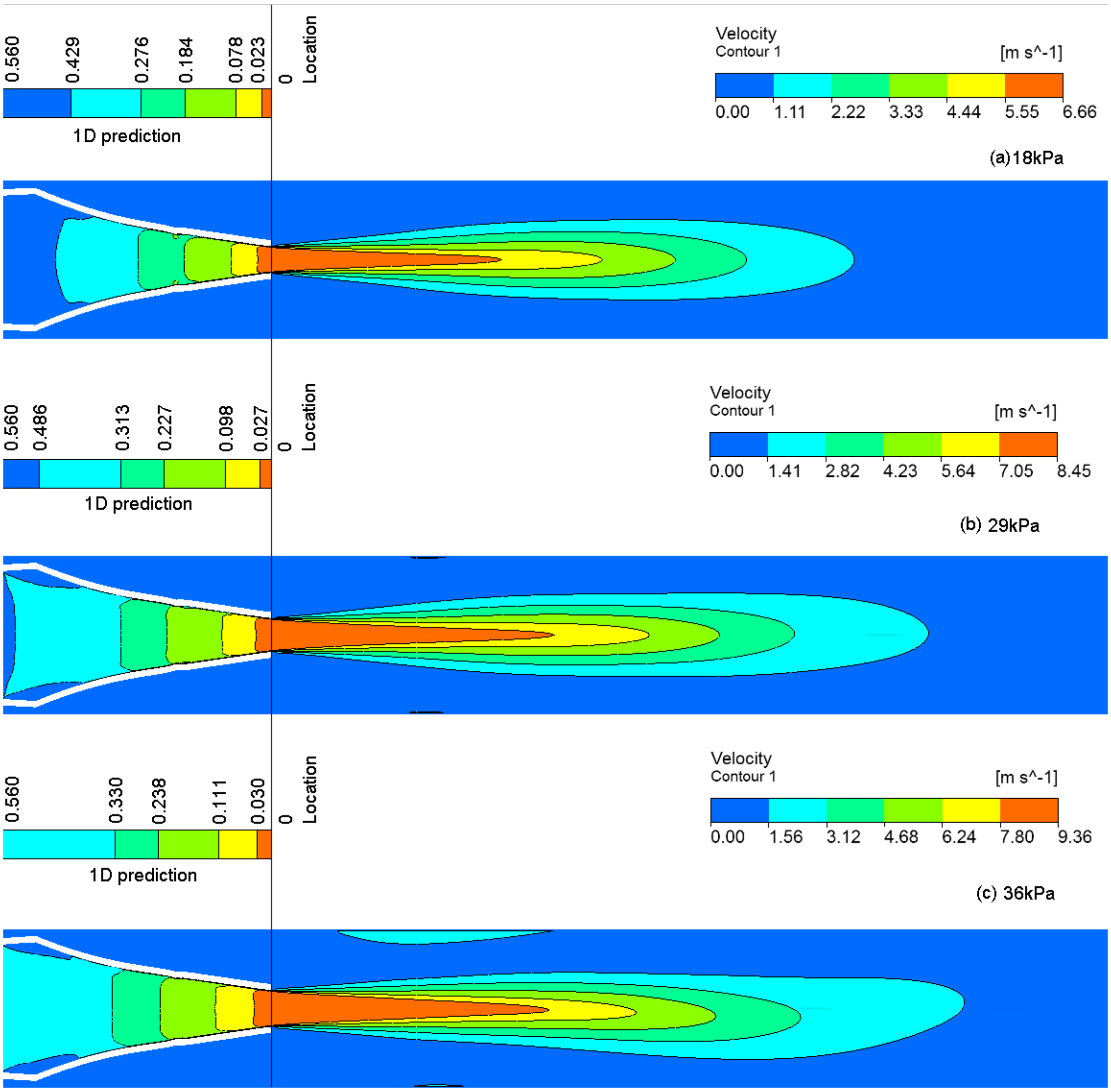


Fig. 7-8 Velocity fields in the y-z planes of valve: predicted by the viscous flow model for the cases with increasing driving pressures of 18, 29, and 36 kPa and compared with corresponding 1D flow predictions

Pressure distributions along the valve tunnel

Figure 7-9 shows the averaged pressure distributions along the valve as well as the upstream and downstream tunnels of the viscous flow model under driving pressure drops of 18 kPa, 29 kPa, and 36 kPa (solid lines). The corresponding 1D flow corrections using Equation (7.2) and the experimental results are also shown. The 3D CFD model covers the whole valve tunnel including upstream and downstream pipes, while the 1D model and water tunnel tests cover only the valve tunnel and downstream pipe. The 3D flow also shows the pressure recovery of the jet in the downstream pipe. It can be seen that the vena contracta predicted by the 3D model agrees well with the experimental measurements. The vena contracta is the point in a contracting flow where the area of the jet is smallest and therefore, the velocity is the highest. The static pressure at the vena contracta is a minimum as seen in the figure. The 1D flow corrections and 3D flow predictions also show excellent agreement, indicating that the pressure drop is primarily due to the increase in dynamic head, with little viscous losses.

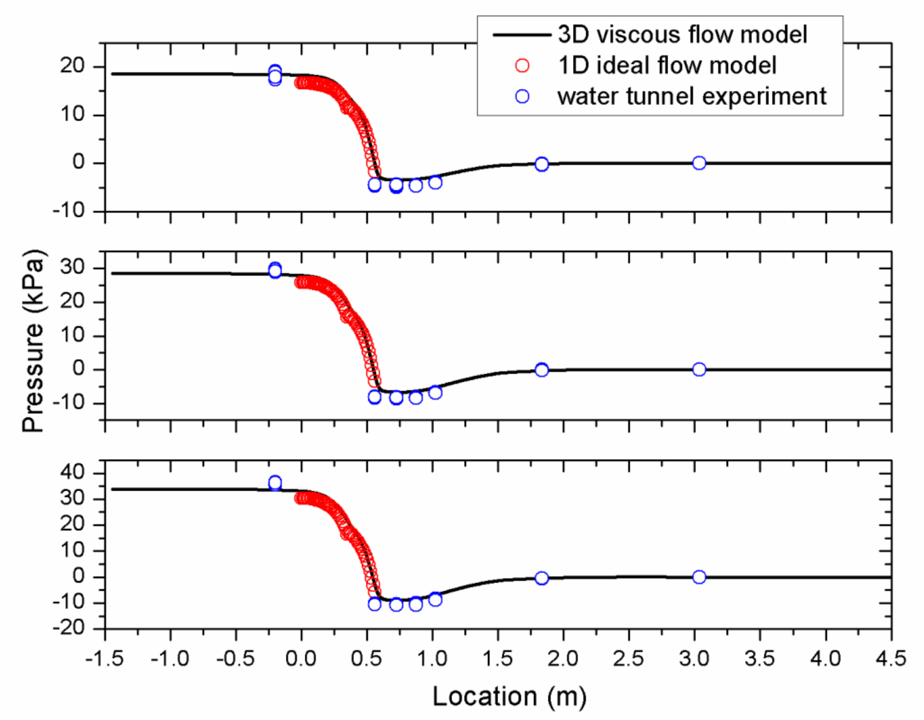


Fig. 7-9 Pressure distributions along the central line of valve: under increasing driving pressure values of 18, 29, and 36 kPa (the 1D flow predictions were corrected by the pressure recovery, i.e. Equation (7.2), where *K*=0.86, 0.83, and 0.79)

7.3 Valve deformations and layer von Mises stress distributions

The duckbill opening deformations predicted by the FSI model under the various pressure drops were compared with the measured ones. Generally, the results of test A match well with the predictions of the simplified FSI model, while the test B results have larger deviations compared with the predictions (especially the points located around X=0.09 m as seen in Fig.7-10). These measurement differences may be attributed to valve structure (permanent deformations, asymmetric geometry and material distribution, etc.), valve installation, and ruler scale errors.

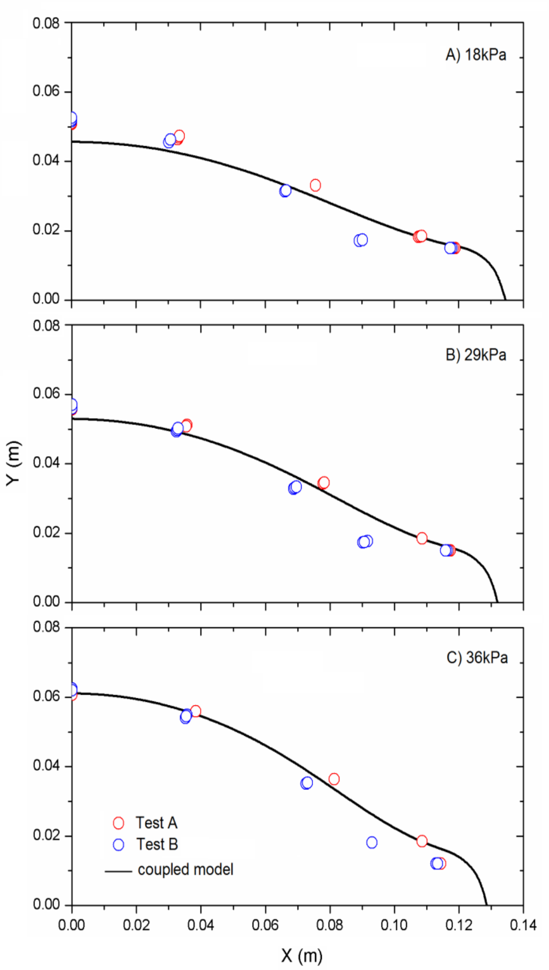


Fig. 7-10 Deformation shapes of exit area: coupled model vs. measurements under increasing driving pressures of A)18 kPa, B) 29 kPa, and C) 36 kPa

Figure 7-11 shows the valve deformation predicted by the coupled model with a driving pressure of 36kPa. The maximum deformation, which is at approximately 0.069m, occurs at the middle of the joint edge where the duckbill and saddle portions connect together. The shell geometry is shown in a 3D solid view so that the valve material layers can be observed clearly.

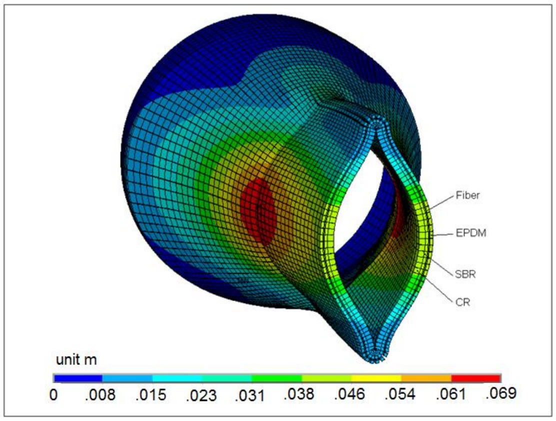


Fig. 7-11 Nodal displacement contour of shell deformation under a pressure drop of 36 kPa (in a 3D solid view)

Figure 7-12 shows the von Mises stress distributions on each layer of the sandwich structure of the DBV for the case of the 36kPa driving pressure. In Fig. 7-12 (b) and (c), the stresses in the fabric layer are seen at least 5 times greater than those for the inside and outside rubber layers. This shows that the pressure load is carried primarily by tension in the fabric reinforcement and demonstrates that the neglect of this layer in Lee, et al. (2001) is not justified. On the other hand, since the fiber reinforcement is near the middle of the cross-section, the bending load is carried by the outside rubber layer. This arrangement of relatively low stiffness rubber layers on the outside and inside, and relatively high stiffness fibre reinforcement layer near the center is responsible for the unique behaviour of the DBV. High internal pressures are contained by the fabric layers while the large bending deformations permit large valve openings for high flow rates with minimum pressure drop.

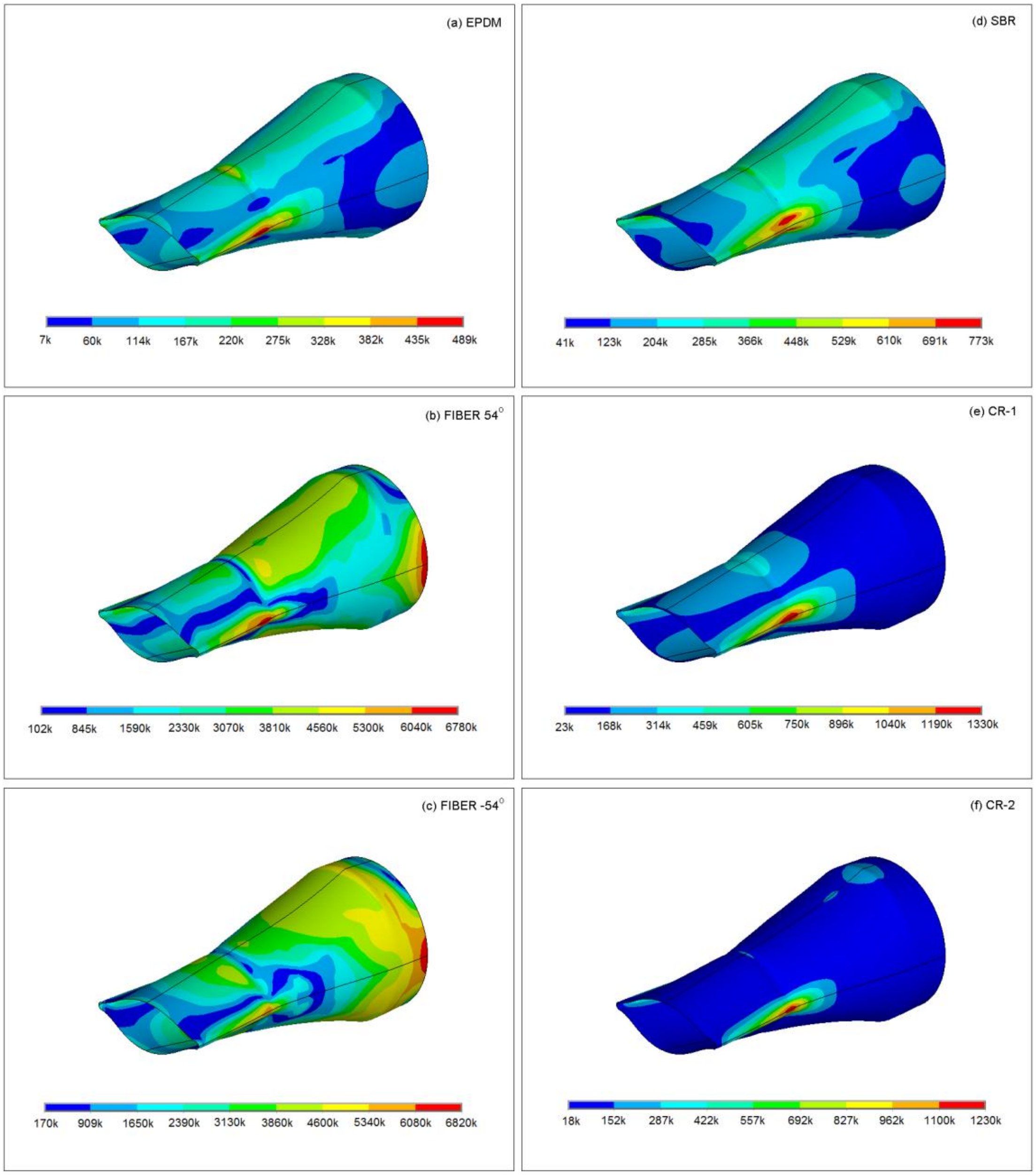


Fig. 7-12 Effective stress distributions on each layer of laminate shell structure under a driving pressure drop of 36 kPa: (a) is for the top layer or external EPDM layer of the duckbill valve; (b) and (c) are for the second and third layers of fabric reinforcement with fiber angles of +54 ° and -54 ° respectively; (d) is for the fourth SBR layer; and (e) and (f) are for the two bottom layers or internal CR layers.

7.4 Flow induced vibration of DBV

7.4.1 Stable convergences of duckbill valve solutions

In the sensitivity analysis section of Chapter 5, we discussed the effects of the unsteady term of the 1D flow model on the valve mass flow rate convergence. It was found that the fluid inertia (i.e. the unsteady term) has a great influence on the valve transient oscillation (see Fig. 5-10 A)). In other words, a fluid with small density will behave quite differently compared to one with large density.

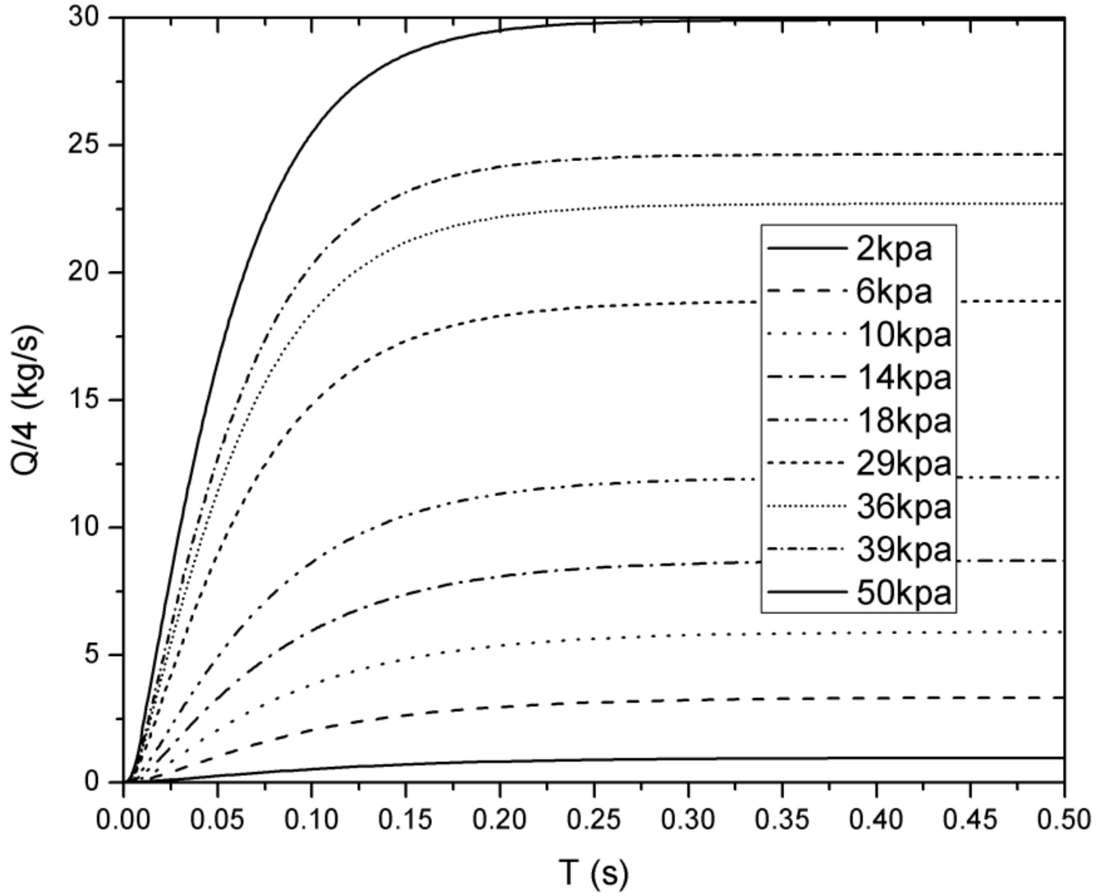


Fig. 7-13 Mass flow rates vs. simulation time

Transient 1D flow model under a driving pressure range from 2 kPa to 50 kPa

Rayleigh damping α and β are 5 and 0.003, respectively

Fig. 7-13 shows the relationship of mass flow rate to simulation time for various suddenly imposed upstream pressures ranging from 2 kPa to 50 kPa. A small damping ratio with a magnitude of 0.005 is applied for each case to accelerate the convergence of the iterations. All cases are convergent within the solution time of 0.5 s. The flows monotonically asymptote to the steady flow condition for the assumed upstream pressures, demonstrating that there are no oscillations predicted, presumably due to the large density of water. Actually, duckbill oscillations were not observed in the water tunnel experiments even when the valve was at small openings. From the comparisons between the experimental results and the simulations in the previous sections, it is established that the simplified coupled model can handle the stable convergent FSI simulations within the wide pressure range studied.

7.4.2 Flow induced vibrations of the duckbill valve at small openings

To further investigate the simulation capability of the coupled model, we examine some cases with different conditions to understand if it can handle cases of flow induced vibration (FIV). Adubi (1974) and Weaver, et al. (1980) has concluded from their studies that some types of flow induced vibration phenomena may exist in certain valves when these valves are at small openings and under specific conditions. To predict such behaviour at small openings of the DBV, we can decrease the fluid density to be small enough so that the fluid inertia (i.e. the unsteady term) can be ignored in the FSI simulation (e.g. using air instead of water as the fluid through the valve). We first study the time step independence to ensure there are no numerical artifacts in the transient valve oscillation analysis. Fig. 7-14 shows that the flow rate amplitude asymptotes to fixed values as the time step reduces, indicating that the valve dynamic equilibrium status is time step independent provided that the time step is sufficiently small.

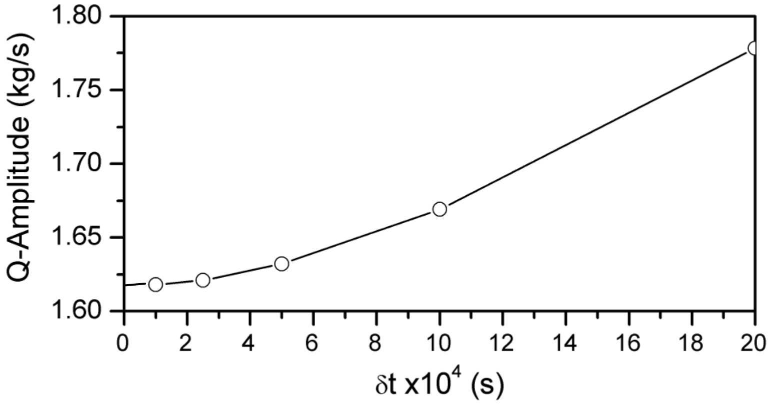


Fig. 7-14 Time step independence

with decreasing time steps of 2, 1, 0.5, 0.25, and 0.1 ms

Figure 7-15 shows that the valve oscillates at constant amplitude when the fluid density is small, the time step is set to 0.1ms, the pressure drop is equal to 6 kPa, and β is 0.003 (α is set close to zero). The FIV phenomenon is successfully observed under the above conditions. Thus, the FSI model is capable of predicting stable limit cycle oscillation of the valve operating at small openings

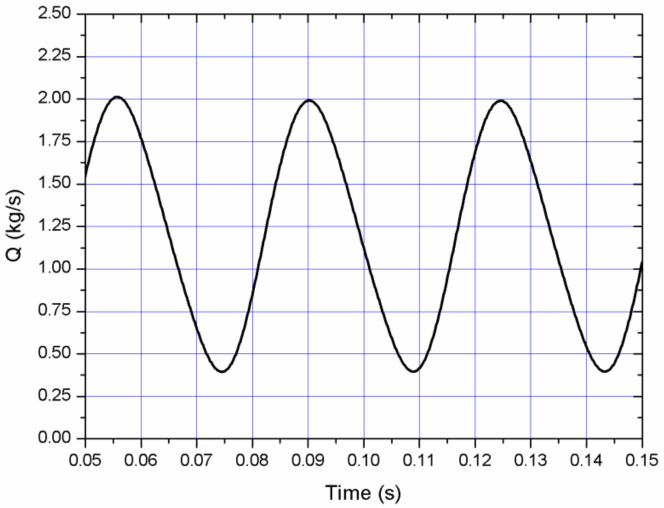


Fig. 7-15 Mass flow rate oscillation of duckbill valve

with the low pressure drop=6 kPa, time step=0.1ms, and β=0.003

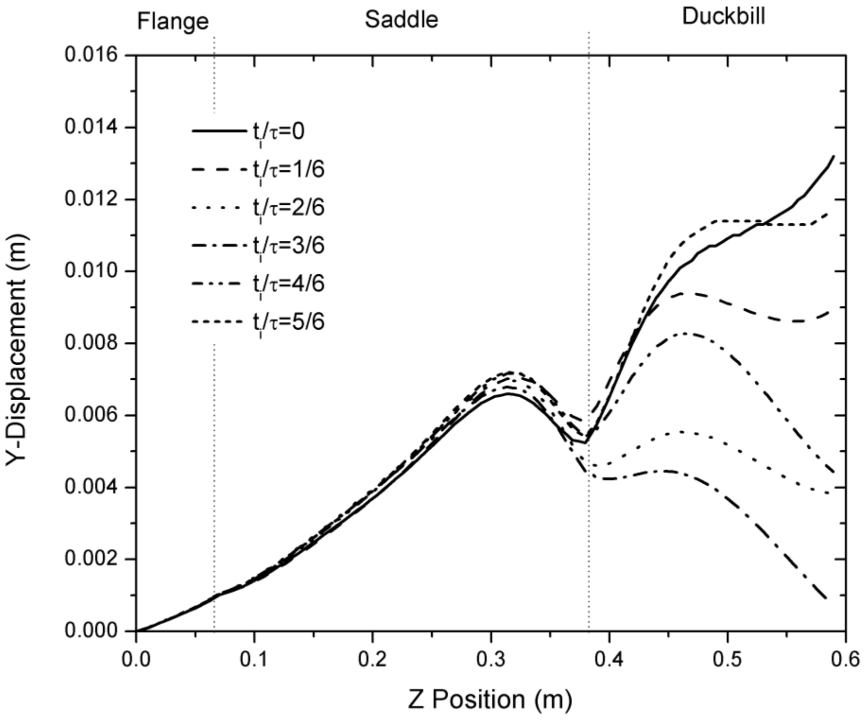


Fig. 7-16 Centerline node y-displacements of the valve model at various times in an oscillation cycle

Figure 7-16 shows the node y-displacements of the path line at the central symmetric y-z plane of the valve model for the 6 kPa case. There are six sets of node displacement curves corresponding to six different times over an oscillation cycle, illustrating that the vibration occurs mainly in the duckbill portion. The vibration is seen to have its largest amplitude at the valve exit with a streamwise motion resembling a travelling wave.

Figure 7-17 shows the further investigation for the case of 2 kPa driving pressure. The simulation time history result shows that valve oscillation occurs but with smaller amplitude when compared to the previous case. The oscillatory behavior of the valve opening during the transient opening process is shown for a suddenly imposed upstream pressure of 2 kPa. It is seen that the valve opens quickly, overshoots, and then goes through stable oscillation. The magnitude of structural damping used here is within a realistic structural damping range as estimated previously. No numerical damping is added. The magnitude of fluid mass acceleration damping (α) is insignificant due to the negligible fluid inertia in these cases. These results suggest that realistic values of structural damping may not be sufficient to suppress the oscillatory behaviour under certain conditions. However, experimental verification is required to validate these code predictions.

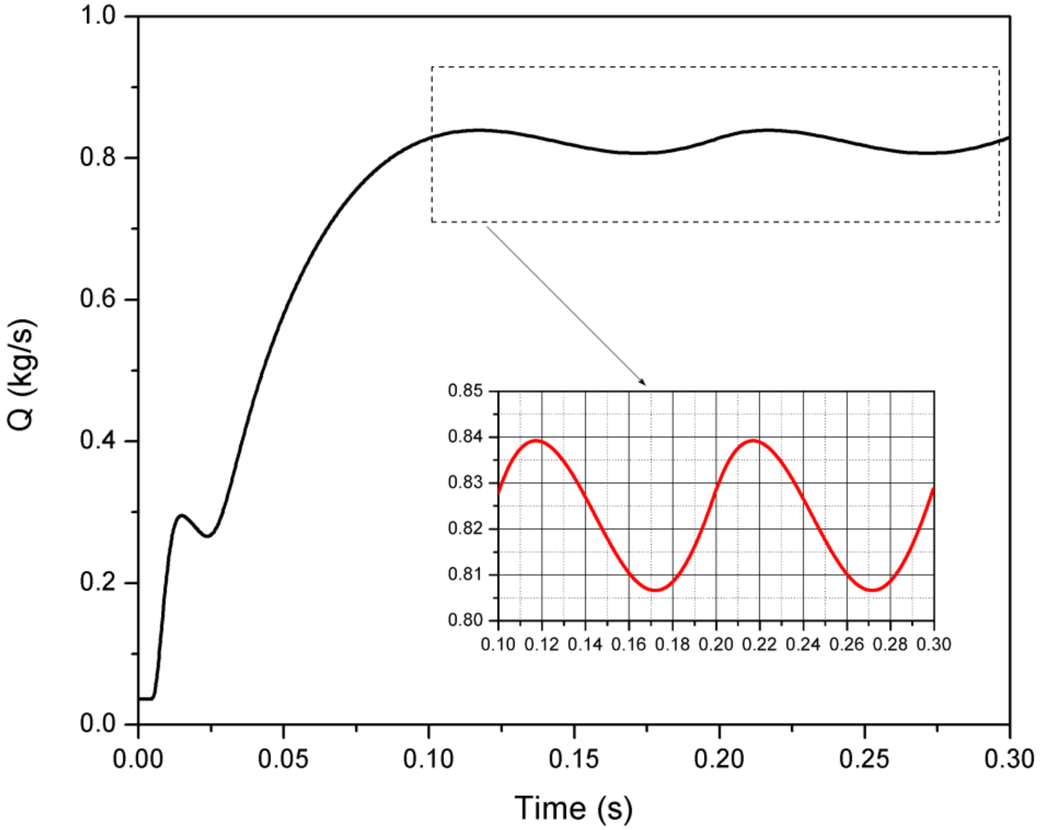


Fig. 7-17 FIV case with mesh and time step independences: under the driving pressure=2 kPa, β=0.003, and time step=0.2ms

Summary

The DBV hydraulic and deformation behavior predicted by the simplified coupled model were validated by comparison with the 3D viscous flow model and the water tunnel experimental measurements. The comparison with the 3D viscous flow model demonstrated that the assumption that the effects of fluid viscosity are negligible is reasonable and the downstream pressure recovery can be corrected based on the valve nozzle opening after the FSI simulations. The head discharge relationship of the duckbill valve predicted by the theoretical model agrees reasonably well with the experimental results. In addition, the FIV phenomenon was successfully predicted at small valve openings when the inertia term is small enough and the damping is within a realistic range, but this result must be validated experimentally.

Chapter 8

CONCLUSIONS AND RECOMMENDATIONS

To improve our fundamental understanding of fluid-structure interaction of a duckbill valve, a theoretical/numerical and experimental study was conducted to provide insight into the physics of the FSI behavior of duckbill valves, including the effects of fluid viscosity, downstream flow, and the influence of material properties on the valve deformation. For the purpose of simplification, a layered shell model of the duckbill valve structure was developed to consider the nonlinearities of the fabric reinforced rubber composite material and the large valve geometric deformation. Samples of the valve material were commercially tested to determine their physical behavior and empirical constants for use in the theoretical model. An unsteady 1D ideal flow model was built based on the assumptions that the effects of fluid viscosity could be negligible and that the effects of pressure recovery could be accounted for using a correction factor. An in-house APDL code was developed to couple the layered shell model with the 1D flow model for FSI simulations. To validate the assumptions and predictions of the simplified coupled theoretical model, a 3D viscous flow model was built, based on the deformed valve geometries predicted by the coupled model. The results were compared with water tunnel experiments carried out using full scale valves. The primary conclusions and novel contributions to knowledge are summarized below.

8.1 Conclusions

The hydraulic performance of a duckbill valve is difficult to precisely evaluate due to the complexity of the duckbill valve composite material, the strong geometric nonlinearity associated with large deformations, the variability of the layered rubber mechanical properties, and the challenging issues of FSI numerical approaches (as mentioned in Chapter 1). Previous studies (see Chapter 2) typically treated the valve behavior either as a linear elastic homogenous material coupled with a 1D ideal flow, or as a nonlinear elastic material without any fluid coupling. For instance, Lee et al. (2004, 2001) applied the former method to study duckbill valve behavior. In their research, the duckbill valve material was assumed to be linearly elastic and the fluid flow was assumed as a steady 1D ideal flow. Furthermore, they assumed that the valve deformation was depended mainly on the mechanics of the rubber deformation, and only secondarily on the fabric reinforcement. Hence the valve material layer design (including the rubber/fiber layer arrangement and fiber orientation) was simply ignored in their theoretical model. Although this simplified model was claimed to predict the valve performance within a good range of deflection, it obviously could not satisfy the requirements of valve design because the valve material design parameters were not considered and the assumption of a linear elastic material appears to be too simple to describe the complex rubber behavior especially at large deformations. In addition, the physical effect of fluid inertia property was neglected by the steady 1D flow model. The influence of fluid viscosity on the pressure and velocity fields was not compared with the predictions of their model for validation.

On the other hand, Chouchaoui (2001) argued that the rubber and fiber material properties must be taken into account for modeling the performance of a duckbill valve. A series of material tests on the valve composite materials were conduced to obtain necessary parameters for the hyperelastic rubber and orthotropic fiber models. A very simple solid model was developed based on these material tests. The nonlinearities of the valve geometry and material were considered. However, this solid model only worked within a very narrow pressure load range (up to only 1.67 kPa) and had no coupling with the fluid flow. Therefore, it is considered to be inadequate for practical applications.

The first primary contribution of the thesis is the development of a tractable theoretical approach to predict the hydraulic performance of a duckbill valve and, therefore, is considered to be a practical design tool. To overcome the above technical and practical shortcomings or difficulties involved in solving the FSI problem, a layered shell model which considered the nonlinearities of valve material and geometric deformation, as well as the valve material layer design parameters, was developed (see Chapter 3). An unsteady 1D flow model was built to consider the effect of fluid inertia term (see Chapter 4) and was coupled with the layered shell model using a monolithic FSI approach (see Chapter 5).

The actual layer arrangement of the valve materials (SBR, CR, EPDM, and fabric reinforcement) was defined using the shell layer section function. The three-parameter Mooney-Rivlin model was applied for the constitutive relations of the rubber materials. The parameters of the rubber model were obtained from the experimental data for uniaxial tension. The orthotropic linear elastic fiber layer was modeled using Hooke's law and the material properties were obtained based on the warp and weft tension tests of the fiber/resin matrix and the empirical Halpin-Tsai estimation. The middle surface was determined from the 3D duckbill valve geometry to represent the whole geometry and the quadrilateral meshes and boundary conditions were setup for the DBV shell model. ANSYS was used to solve the shell model.

On the fluid side, a transient 1D flow approach was developed to treat all the flow as one streamline from the inlet to the outlet of the valve. The 1D dynamic meshes were created in order to record the valve deformation and the pressure and velocity fields in the valve tunnel. The 1D flow solver and coupling algorithm were developed and inserted into the iteration procedure of ANSYS.

ANSYS parametric design language (APDL) was used to develop an in-house code for coupling the shell model and the fluid flow model and for the transient 1D flow solver. The valve design information was provided to the code, including initial driving pressure, general information of the duckbill valve geometry (valve length, width, wall thickness, diameter, initial duckbill opening), material layer design information (rubber layer arrangements, fiber angle orientations, layer thicknesses), parameters for the material models (Mooney-Rivlin rubber model and orthotropically linear elastic fiber model). Hence, this fully coupled FSI model can be applied to predict hydraulic performance and deformation behavior of a duckbill valve with arbitrary size and material layers. The mesh density is adjustable based on the numerical convergence, solution accuracy, and simulation time constraints. Taking the 12" diameter DBV as an example, the pressure load can be varied from 0 kPa to over 50 kPa, which completely covers the industrial application range of this valve.

The second major contribution of this research was the validation of the simplified coupled model with the 3D viscous flow model (see Chapter 4) and the water tunnel experiment (see Chapter 6).

The comparisons between the velocity fields from the 3D viscous CFD results and the 1D ideal flow model demonstrated that the flow is essentially one dimensional through the valve, with the obvious exception of the thin boundary layers due to the effect of fluid viscosity. However, the influence of the boundary layers were found to be insignificant and the velocity distribution predicted by the 1D flow model matched that of the 3D flow model well, indicating that the 1D assumption is a reasonable model for predicting the flow through the valve. Thus, it can be concluded from the comparisons that the transient 1D ideal flow equation can adequately model the fluid dynamics of a duckbill valve since viscous effects are negligible except when the valve is operating at very small openings.

To validate the theoretical model experimentally, a water tunnel experiment was carried out for investigating the DBV hydraulic performance and its deformations. At several mass flow rates, the pressure drop along the water tunnel, valve inlet and outlet velocity profiles, and valve opening deformations were all measured as functions of upstream pressure. The valve model with the 10% strain level roughly predicted the linear results of the head-discharge relationship of the DBV with an average deviation of about 10% (minimum 3% but maximum 20%) larger than the water tunnel results. Compared to the predictions of the valve model with the 15% and 20% strain level curves, it is demonstrated that the 15% strain level curve is the best approximation. This implies that the estimated strain level affects the selection of the rubber material curve and, therefore, affects the accuracy of the valve model.

From the comparisons of the head-discharging relation, it was also found that the experimental results approach the valve model without fabric reinforcement at smaller flow rates (lower driving pressure drops), but approach the valve model with a fiber layer at higher flow rates. This suggests physically that the valve performance depends primarily on the mechanics of the rubber bending deformation under lower pressure loadings, while the fabric extension reinforcement becomes dominant under higher pressure loadings. Neglecting the fiber layer in modelling valve deformation due to significant pressure drop could lead to serious errors, contrary to the assumption of Lee, et al. (2001). Therefore, fiber reinforcement of the layered composite rubber plays an important role in controlling duckbill valve material stretch, especially at large openings.

Comparing the averaged pressure distributions along the valve as well as the upstream and downstream tunnels, the 1D flow pressure recovery corrections and 3D flow predictions show excellent agreement with each other, indicating that the pressure drop is primarily due to the increase in dynamic head, with little viscous losses. The 3D viscous flow model also demonstrated good agreement with the experimental results for the pressure recovery of the jet in the downstream pipe. Thus, the downstream flow assumption of the 1D flow model was found to be reasonable. Based on the findings, the downstream pressure recovery can be corrected based on the valve nozzle opening after FSI simulations if the valve flow is submerged.

The duckbill opening deformations predicted by the valve model under the various pressure drops were also compared with the ones measured based on experimental results on two 12” diameter valves. The results of one of the valves match the predictions of the valve model well, while the other results had larger deviations compared with the predictions. These measurement differences may be attributed to valve manufacture tolerances and valve installation errors (see Chapter 6). Generally, the comparison shows that the duckbill opening deformations are predicted reasonably well by the valve model.

The third contribution of the present theoretical study was the extended application of the coupled model for the FIV simulation of a duckbill valve at small openings and the case of closure end deformation due to increased downstream pressure.

To improve the understanding of valve stability behaviour, flow induced vibration of the valve at small openings was examined using the simplified FSI model. The current valve design was found to be generally stable due to large fluid inertia and material damping as seen in both the water tunnel experiment and coupled model results. However, the simplified FSI model may predict duckbill oscillations at small openings if the fluid inertia is sufficiently small. The oscillatory behavior of duckbill portion was purposely demonstrated for small valve openings by setting the inertial term small enough and keeping the structural damping within a realistic range. Thus, the FSI model is capable of predicting stable limit cycle oscillation of the valve operating at small openings.

Further investigation of the coupled model with reduced fluid inertia showed that the vibration occurs mainly in the duckbill portion. The vibration was seen to have its largest amplitude at the valve exit with a streamwise motion resembling a travelling wave. The simulation time history also illustrated the oscillatory behavior of the valve opening during the transient opening process for a suddenly imposed upstream pressure (2~6 kPa). It was observed that the valve opens quickly, overshoots, and then goes through stable oscillation. The magnitude of structural damping used was within a realistic structural damping range. These results suggest that realistic values of structural damping may not be sufficient to suppress the oscillatory behaviour under certain conditions. We conclude that the FIV phenomenon predicted by the theoretical model is physically realistic.

Duckbill valves are intended to prevent flow from occurring in the reverse direction. To investigate such a valve deformation behavior under various back pressures and to provide comparison of back pressure resistance in future designs, a back pressure model of the DBV was built using the layered shell model but without FSI since there is no flow through the closed valve. Since the model is not intended to model details of the back pressure failure, it is set with a simple modification of boundary constraint that does not include buckling, asymmetric pressure loading, and contact stress issues. The back pressure layered shell model can be employed to predict the effects of the maximum back pressure up to 28 kPa (see Chapter 3)

8.2 Future work

Although the current simplified model is adequate for current industrial design use, a fully coupled FSI model, which considers both the nonlinear solid model and the fully viscous flow model, is recommended for the future work. There are many challenges in the available FSI simulation frameworks at present, such as the issues of dynamic mesh generation, re-orientation of the layered solid elements, and coupling with viscous flow. However, it is believed that the above challenges can be solved effectively in the future. The benefits of the recommended model are discussed below.

First of all, prediction accuracy can be further improved by use of solid element based structural FEA and fully viscous flow model. It is expected that a solid FEA model of a duckbill valve would provide more accurate prediction of the bending simulation than the shell model, especially near the side edges of the duckbill portion. In the previous discussion, it was found that the layered shell model predicts the head-discharge relationship with a larger deviation under lower pressure loadings, while the valve performance depends primarily on the mechanics of the rubber bending deformation under these loadings. This implies that the shell model is less accurate in modeling the flexual deformations. This can be attributed to the valve wall being relatively thick, especially near the side edges of the duckbill portion where the shell element behavior is not physically true. The solid element may effectively overcome this shortcoming of the shell element. On the other hand, a full viscous flow model is more reasonable than the

1D flow model when the duckbill nozzle is initially opened because the effect of fluid viscosity cannot be ignored and the downstream flow may be naturally included without any empirical correction. It is also expected that the viscous flow model will provide more detail turbulent flow information since the 1D flow model can only provide the information of the pressure and velocity fields.

Secondly, alternative valve structure designs could require complex geometry modifications, such as more corrugated duckbill cross sections and valves with any asymmetry. The solid element FEA and fully viscous CFD can provide higher flexibility in geometric modeling and mesh generation and, therefore, more accurate predictions of the valve deformation and fluid flow. For instance, out-of-plane bending of curved duckbill would require viscous CFD to handle the flow curvature and solid element FEA for the tight radii curvature.

More theoretical and experimental research on the flow induced vibration of the valve at small openings is also recommended to better understand the valve oscillation behavior. The effects of fluid inertia and viscosity, as well as valve stiffness and damping, on the FIV of duckbill valve at small openings should be further studied by the fully coupled FSI model recommended above and corresponding experimental approaches should be carried out to validate the theoretical FIV research.

REFERENCES

Adubi, F.A.A., The Hydroelastic Vibration of a Hydraulic Swing Check Valve, PhD Thesis, McMaster University, 1974.

ANSYS and CFX user's manual, ANSYS Inc., 2008.

Ashton, J. E. and Whitney, J. M., Theory of Laminated Plates, Published by Technomic Publishing Company, Inc. USA, 1970.

Axel Products, Inc., Using Slow Cyclic Loadings to Create Stress Strain Curves for Input into Hyperelastic Curve Fitting Routines, www.axelproducts.com, 2012.

Bath, K. J. and Bolourchi, S., A Geometrical and Material Nonlinear Plate and Shell Element, Computers & Structures, Vol. 11, pp.23-48, 1980.

Beards, C. F., Structural Vibration: Analysis and Damping, published by Elsevier, ISBN: 978-0-340-64580-2, page 173, 1996.

Blevins, R. D., Flow-Induced Vibration, 2nd Edition, Krieger Publishing Company, 2001.

Brewer, H. K., Prediction of Tire Stress and Deformation from Composite Theory, Tire Science and Technology, TSTCA, Vol.1, No.1, pp.47-76, 1973.

Bungartz, Hans-Joachim, Schäfer, Michael, eds. Fluid-structure Interaction: Modelling, Simulation, Optimization. Springer-Verlag. ISBN 3-540-34595-7, 2006.

Chouchaoui, B., Simulating Flow through 8" Duckbill Valve, Final Report, Windsor Industrial Development Laboratory, Inc., 2001.

Chow, C. L. and Cudiff, C. H., On the Characterization of Mechanical Properties of Rubber Vulcanizates, Tire Science and Technology, TSTCA, Vol. 15, No. 2, pp.73-96, 1987.

Clark, S. K., Plane Elastic Characteristics of Cord-Rubber Laminates, Technical Report No.2, UMRI Project 02957, University of Michigan, 1960.

Duer, M. J., Use of variable orifice duckbill valves for hydraulic and dilution optimization of multiport diffusers, Water Science and Technology, Vol.38, No.10, pp. 277-284, 1998.

Dupont, Inc., Kevlar Aramid Fiber Technical Guide, 2012.

Durbin, P.A. Separated flow computations with the k-epsilon-v^2 model, AIAA Journal, Vol. 33, No. 4, pp. 659-664, 1995.

Dvorkin, E. N. and Bathe, K. J., A Continuum Mechanics Based Four-Node Shell Element for General Non-linear Analysis, Engineering Computations, Vol.1, pp77~88, 1984.

EVR Products, CPI Series Specification (www.evrproducts.com).

Ferreira, A.J.M., Sá, J.M.A.C., and Marques, A.T., Nonlinear Finite Element Analysis of Rubber Composite Shells, Strength of Materials, Vol. 35, No.3, pp. 225~235, 2003.

Fung, Y.C., Foundations of Solid Mechanics, Prentice-Hall, Englewood Cliffs, New Jersey, 1965.

Gao, Z., Tang, H.W., and Lee, J.H.W., Mass Entrainment in a Duckbill Valve Jet, Journal of Hohai University, Vol.33 No.6, 2005 (in Chinese).

Gao, Z., Tang, H.W., and Gu, Z.H, Review of Duckbill Check Valve Jets, Water Science and Engineering, p120~123, Vol. 24 S., 2004 (in Chinese).

Giannopapa, C.G., Fluid Structure Interaction in Flexible Vessels, PhD Thesis, the University of London, King’s College London, 2004.

Gould, P. L., Analysis of Shells and Plates, Springer-Verlag, USA, 1987.

Halpin, J.C. and Kardos, J.K., The Halpin-Tsai Equations: A Review, Polymer Engineering and Science, Vol. 16, No.5, pp344~352, 1976.

Hermans, J.J., The Elastic Properties of Fiber Reinforced Materials When The Fibers Are Aligned, Koninklijke Nederlandsche Akademic Van Wetenschappen, Proceedings, Series B, Vol.70, pp1~9 1967.

Jones, W.P., and Launder, B.E., The Prediction of Laminarization with a Two-Equation Model of Turbulence, International Journal of Heat and Mass Transfer, Vol. 15, pp. 301-314,1972.

Kaga, H., Okamoto, K., and Tozawa, Y., Stress Analysis of a Tire Under Vertical Load by a Finite Element Method, Tire Science and Technology, TSTCA, Vol.5, No.2, pp.102-118, 1977.

Lee, J.H.W., Kuang, C.P. and Chan, H.C., The Velocity Field of a Duckbill Valve (DBV) Jet in Coflow, 15th ASCE Engineering Mechanics Conference, Columbia University, New York, NY, Jun 2-5, 2002.

Lee, J.H.W., Lo, S.H. and Lee, K.L., Flow Variation of Duckbill Valve Jet in Relation to Large Elastic Deformation, Journal of Engineering Mechanics, ASCE, p971~981, Aug, 2004.

Lee, J.H.W., Wilkinson, D.L., and Wood, Ian R., On the Head-Discharge Relation of a "Duckbill" Elastomer Check Valve, Journal of Hydraulic Research, Vol.39, No.6, pp.619~628, 2001.

Lu, J. and Zhao, X., Pointwise Identification of Elastic Properties in Nonlinear Hyperelastic Membranes—Part I: Theoretical and Computational Developments, Journal of Applied Mechanics, Vol. 76, 061013-1~10, Nov. 2009.

Luccioni, B. M., Constitutive Model for Fiber-Reinforced Composite Laminates, Journal of Applied Mechanics, Vol. 73, pp901~910, Nov. 2006.

Menter, F.R., Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications, AIAA Journal, vol. 32, pp. 269-289, 1994.

Miller, C., Popper P., Gilmour, P.W, and Schaffers, W.J., Textile Mechanics Model of a Pneumatic Tire, Tire Science and Technology, TSTCA Vol.13, No.4, pp.187-226, 1985.

Mooney, M., A Theory of Large Elastic Deformation, Journal of Applied Physics, p582~592, 1940.

Nicholson, D.W., A Model Analysis of the Structural and Pneumatic Contributions to Tire Behavior under Vertical loads, Tire Science and Technology, TSTCA, Vol.3, No.1, pp.29-42, 1975.

Pósfalvi, Ödön, On the Mechanical Behavior of the Orthotrpic Cord-Rubber Composite, Tire Science Technology, TSTCA, Vol. 4, No. 4, pp.219-232, 1976.

Reddy, J. N., Exact Solutions of Moderately Thick Laminated Shells, Journal of Engineering Mechanics, Vol. 110, No. 5, 1984.

Rivlin, R.S., Large Elastic Deformations of Isotropic Materials, Philosophical

Transactions of The Royal Society, Vol.241, p379-397, 1948.

Schieck, B., Pietraszkiewicz, W., and Stumpf, H., Theory and Numerical Analysis of Shells Undergoing Large Elastic Strains, Int. J. Solids Structures, Vol. 29, No. 6, pp. 689~709, 1992.

Simmonds, J. G., The Strain Energy Density of Rubber-like Shells, Int. J. Solids Structures, Vol. 21, No. 1, pp. 67~77, 1985.

Snowdon, J.C., Rubberlike Materials, Their Internal Damping and Role in Vibration Isolation, Journal of Sound Vibrations, 2(2), pp 175-193, 1965.

Standard Test Methods for Vulcanized Rubber and Thermoplastic Elastomers-Tension, ASTM D412-06a, 2006.

Streeter, V.L., and Wylie, E.B., Fluid Mechanics, 8th edition, New York, McGraw Hill, 1985.

Tielking, J. T., A Finite Element Tire Model, Tire Science and Technology. Vol. 11 Nos. 1-4, pp.50-63, 1984.

Timoshenko, S. P. and Woinowsky-Krieger, S., Theory of Plates and Shells, 2nd Edition, McGraw-Hill Book Company, pp580, 1959.

Treloar, L. R. G., The Physics of Rubber Elasticity, 2nd Edition, Oxford University Press, pp342, 1958.

Tucker, III, C. L., and Liang, E., Stiffness Predictions for Unidirectional Short-Fiber Composites: Review and Evaluation, Internal Report, University of Illinois and GE, 1998.

Wadham-Gagnon, M., et al., Hyperelastic Modeling of Rubber in Commercial Finite Element Software (ANSYSTM), Website report, 2006.

Wang, J., Weaver, D.S., and Tullis, S., Simplified Fluid-Structural Model for Duckbill Valve Flow, Journal of Pressure Vessel Technology, ASME Transactions, Vol. 134, Issue 4, 041301 (8 pages), 2012.

Wang, J.G. and Schweiezerhof, K., Boundary Integral Equation Formulation for Moderately thick Laminated Orthotropic Shallow Shells, Computer & Structures, Vol. 58. No. 2. pp.277-287, 1996.

Weaver, D.S. and Ziada, S., A Theoretical Model for Self-Excited Vibrations in

Hydraulic Gates, Valves and Seals, Journal of Fluids Engineering, Transactions of the ASME, Vol. 100, p239-245, 1980.

White, F. M., Viscous Fluid Flow, 2nd Edition, McGraw-Hill, Inc, 1991.

Wilcox, D.C., Formulation of the k-omega Turbulence Model Revisited, AIAA Journal, Vol. 46, No. 11, pp. 2823-2838, 2008.

Yu, W. and Hodges, D. H., A Geometrically Nonlinear Shear Deformation Theory for Composite Shells, Journal of Applied Mechanics, Vol. 71, pp1~ 9, 2004.

APPENDIX A

APDL Code for the Simplified FSI Model

This is an APDL subroutine developed to create the layered shell model of DBV and the transient 1D flow model and couple them.

```
/CLEAR
/TITLE SIMPLE DBV MODEL COUPLED WITH 1D FLOW AND SMALL OPENING
/COM, ! Assuming the environment of ANSYS has already been set down
/COM, PREFERENCES FOR GUI FILTERING HAVE BEEN SET TO DISPLAY:
/COM, STRUCTURAL ! Structural option
/CONFIG,nres,30000 ! limit result file to 3G
/CONFIG,fsplit,750 ! split result file (max 99)
!* =======================================================================
/PREP7 ! Preprocessing
!* =======================================================================
!* %AWP_ROOT121%\ansys\hpmpi\sethpmpipassword
!*%%%%%%%%%%%%Given conditions %%%%%%%%%%%%%%%%%%%%%
PI=3.1415926 ! Constant
R1=0.00619 ! Duckbill side edge radius
Rin=0.13919 ! Inlet radius
TH1=0.00619 ! Half duckbill thickness
TH2=0.00819 ! Half saddle thickness
W1=0.125 ! Opening width
Opening=0.0001 ! Small opening
!*
A1=0.25*PI*R1*R1+W1*(TH1-Opening) ! Key area 1
A2=0.25*PI*(Rin*Rin-(Rin-TH2)*(Rin-TH2)) ! Key area 2
Ain=0.25*PI*Rin*Rin-A2 ! Inlet area of 1D flow, m^2
!*
Mu1=14.38/14.38 ! Layer scale
CRThick=Mu1*0.25*25.4/1000 ! CR thickness
SBRThick=Mu1*0.125*25.4/1000 ! SBR thickness
EPDMThick=Mu1*0.125*25.4/1000 ! EPDM thickness
FIBThick=Mu1*0.033*25.4/1000 ! FIBER thickness
!* Halpin-Tsai Estimates of unidirectional fabric reinforced resins matrix
```

```
!* Fiber scale: Mu2=2(a/b) where a is fiber length and b is fiber thickness
Mu2=120000
VF=0.45 ! Volume fraction of fiber
EF=3460E6 ! Fiber modulus
EM=4.37E6 ! Resin modulus
PrF=0.21 ! Fiber possion ratio
PrM=0.35 ! Resin possion ratio
GF=0.5*EF/(1+PrF) ! Fiber shear modulus
GM=0.5*EM/(1+PrM) ! Resin shear modulus
!* EX1: Mu2-> infinity, longitudinal direction
EX1=EM*(EF+Mu2*(VF*EF+(1-VF)*EM))/(VF*EM+(1-VF)*EF+Mu2*EM)
EX2=EF*EM/(VF*EM+(1-VF)*EF) ! Mu2=0, transverse direction
EX3=EX2
G1=GM*(GF+2*(VF*GF+(1-VF)*GM))/(VF*GM+(1-VF)*GF+2*GM) !Mu2=2, shear modulus
G2=GF*GM/(VF*GM+(1-VF)*GF) !Mu2=0, shear modulus
G3=G2
!* Mu2=2, possion ratio
PR1=PrM*(PrF+2*(VF*PrF+(1-VF)*PrM))/(VF*PrM+(1-VF)*PrF+2*PrM)
PR2=PrF*PrM/(VF*PrM+(1-VF)*PrF) !Mu2=0, possion ratio
PR3=PR2
!*
ZDBV=0.56 ! DBV total length in z direction
ZBILL=0.21 ! Duckbill length in z direction
ZSADDLE=0.28 ! Saddle length in z direction
!*
MXNUM=28 ! Defines mesh number in x direction
MZNUM=48 ! Defines mesh number in z direction
MSIZE=ZDBV/MZNUM ! Mesh size in z direction
!*
DPmax=36100.0 ! Given driving pressure,Pa
DPmin=0.0 ! Initial driving pressure,Pa
DenW=1000.0 ! Water density,kg/m^3
DTIME=0.001! Time step
LOOPS=500 ! Sets solution loops
NRAMP=1 ! Pressure loads ramped up in NRAMP loops
!*
MatDamping=0.000 ! Material damping 0.000~0.05
AlphaDamping=5 ! Alpha damping 0.5~10
BetaDamping=0.003 ! Beta damping 0.0002~0.0045
```

```
NumDamping=0.0 ! Numerical damping 0.0~0.5
!* Defines filename path and name
*DIM, PROFILE, STRING, 80 ! String nomore than 80 characters
!* Valve profile data
PROFILE(1)='C:\Temp\alphadumping\alpha-1\DBVKP-S14'
!* %%%%%%%%%%%%    END    %%%%%%%%%%%%%%%%%%%%%%%
!*
!* +++++++++++++++++++ Defines elements ++++++++++++++++++++++++++++++++
!*
ETCON,SET ! Allows ANSYS to choose best KEYOP's for 180x elements
!* 4-node shell181 for layered hyperelastic structure
ET,1,SHELL181
KEYOPT,1,1,0 ! Bending and membrane stiffness
KEYOPT,1,3,2 ! Full integration with incompatible modes
KEYOPT,1,4,1 ! Incompressible algorithm
KEYOPT,1,8,2 ! Store data for TOP and BOTTOM, for all layers
KEYOPT,1,9,0 ! No user subroutine to provide initial thickness
KEYOPT,1,10,0 ! No user subroutine to provide initial stress
!* Compression-only support
ET,2,LINK10
KEYOPT,2,2,0
KEYOPT,2,3,1 ! Compression-only
R,2,0.0001,0, ! Cross section area 0.0001m^2
!*
!* ++++++++++++++++++++++ Defines materials +++++++++++++++++++++++++++++
!*
!* Rubber properties #1-#3, 3-parameter Mooney-Rivlin model
!* SBR parameters for 10%, 20%, 40%, and 60% engineering strains
SBRC10=-6549060 ! 10%    -36101640    20%    -12013900 40%    -6549060    60%    -2917216
SBRC01= 7920202 ! 10%    38216458    20%    13526835 40%    7920202    60%    3907476
SBRC11= 2823821 ! 10%    53521835    20%     9253176 40%    2823821    60%     970057
SBRdd=0 ! Imcompressible material
!* #1 SBR
MP,DENS,1,1300,   ! Rubber density kgm^-3
MP,DAMP,1,MatDamping, ! Material damping
!3-parameter Mooney-Rivlin model
TB,HYPE,1,1,3,MOON
TBDATA,,SBRC10,SBRC01,SBRC11,SBRdd,,,
MPTEMP,,,,,,,,
```

```
MPTEMP,1,0
!* CR parameters for 10%, 20%, 40%, and 60% engineering strains
CRC10=-1500010 ! 10%    -6086450    20%    -3111330 40%    -1500010    60%    -953413
CRC01= 2105740 ! 10%    6797653    20%    3775827 40%    2105740    60%    1511713
CRC11=    655776 ! 10%    8099948    20%    2226240 40%      655776    60%      336156
CRdd=0 ! Imcompressible material
!* #2 CR
MP,DENS,2,1300,   ! Rubber density kgm^-3
MP,DAMP,2,MatDamping, ! Material damping
!3-parameter Mooney-Rivlin model
TB,HYPE,2,1,3,MOON
TBDATA,,CRC10,CRC01,CRC11,CRdd,,,
MPTEMP,,,,,,,,
MPTEMP,1,0
!* EPDM parameters for 10%, 20%, 40%, and 60% engineering strains
EPDMC10=-2906950   !   10%     -16237700     20%     -8941970     40%     -2906950     60%     -1827190
EPDMC01= 3660460 ! 10%    17391260    20%    9992474    40%    3660460    60%    2476305
EPDMC11= 1469574 ! 10%    23354930    20%    7253447    40%    1469574    60%      743825
EPDMdd=0 ! Imcompressible material
!* #3 EPDM
MP,DENS,3,1300,   ! Rubber density kgm^-3
MP,DAMP,3,MatDamping, ! Material damping
!3-parameter Mooney-Rivlin model
TB,HYPE,3,1,3,MOON
TBDATA,,EPDMC10,EPDMC01,EPDMC11,EPDMdd,,,
MPTEMP,,,,,,,,
MPTEMP,1,0
!* Orthotropic fabric material #4
MP,DENS,4,1270,   ! Density kgm^-3
MP,DAMP,4,MatDamping, ! Material damping
MPDATA,EX,4,,EX1
MPDATA,EY,4,,EX2
MPDATA,EZ,4,,EX3
MPDATA,PRXY,4,,PR1
MPDATA,PRYZ,4,,PR2
MPDATA,PRXZ,4,,PR3
MPDATA,GXY,4,,G1
MPDATA,GYZ,4,,G2
```

```
MPDATA,GXZ,4,,G3
MPTEMP,,,,,,,,
MPTEMP,1,0
!* Compression-only spar material #5
MPDATA,EX,5,,1E10 ! Define a big stiffness
MPDATA,PRXY,5,,0.3
MPTEMP,,,,,,,,
MPTEMP,1,0
!*
!* +++++++++ Defines properties of each material layer +++++++++++++++++++++++++
!*
*DEL,_FNCNAME
*DEL,_FNCMTID
*DEL,_FNCCSYS
*SET,_FNCNAME,'DBVThk'
*SET,_FNCCSYS,0
! /INPUT,DBVThickness.func,,,1
*DIM,%_FNCNAME%,TABLE,6,3,5,,,,%_FNCCSYS%
!
! Begin of equation: {Z}
*SET,%_FNCNAME%(0,0,1), 0.0, -999
*SET,%_FNCNAME%(2,0,1), 0.0
*SET,%_FNCNAME%(3,0,1), 0.0
*SET,%_FNCNAME%(4,0,1), 0.0
*SET,%_FNCNAME%(5,0,1), 0.0
*SET,%_FNCNAME%(6,0,1), 0.0
*SET,%_FNCNAME%(0,1,1), 1.0, 99, 0, 1, 4, 0, 0
*SET,%_FNCNAME%(0,2,1),      0
*SET,%_FNCNAME%(0,3,1),      0
! End of equation: {Z}
!
! Begin of equation: 0.012
*SET,%_FNCNAME%(0,0,2), 0.21, -999
*SET,%_FNCNAME%(2,0,2), 0.0
*SET,%_FNCNAME%(3,0,2), 0.0
*SET,%_FNCNAME%(4,0,2), 0.0
*SET,%_FNCNAME%(5,0,2), 0.0
*SET,%_FNCNAME%(6,0,2), 0.0
*SET,%_FNCNAME%(0,1,2), 1.0, 99, 0, 0.012, 0, 0, 0
```

```
*SET,%_FNCNAME%(0,2,2),     0
*SET,%_FNCNAME%(0,3,2),     0
! End of equation: 0.012
!
! Begin of equation: 0.015
*SET,%_FNCNAME%(0,0,3), 0.3, -999
*SET,%_FNCNAME%(2,0,3), 0.0
*SET,%_FNCNAME%(3,0,3), 0.0
*SET,%_FNCNAME%(4,0,3), 0.0
*SET,%_FNCNAME%(5,0,3), 0.0
*SET,%_FNCNAME%(6,0,3), 0.0
*SET,%_FNCNAME%(0,1,3), 1.0, 99, 0, 0.015, 0, 0, 0
*SET,%_FNCNAME%(0,2,3),     0
*SET,%_FNCNAME%(0,3,3),     0
! End of equation: 0.015
!
! Begin of equation: 0.0185
*SET,%_FNCNAME%(0,0,4), 0.49, -999
*SET,%_FNCNAME%(2,0,4), 0.0
*SET,%_FNCNAME%(3,0,4), 0.0
*SET,%_FNCNAME%(4,0,4), 0.0
*SET,%_FNCNAME%(5,0,4), 0.0
*SET,%_FNCNAME%(6,0,4), 0.0
*SET,%_FNCNAME%(0,1,4), 1.0, 99, 0, 0.019, 0, 0, 0
*SET,%_FNCNAME%(0,2,4),     0
*SET,%_FNCNAME%(0,3,4),     0
! End of equation: 0.0185
!
! Begin of equation: 0.016
*SET,%_FNCNAME%(0,0,5), 0.56, -999
*SET,%_FNCNAME%(2,0,5), 0.0
*SET,%_FNCNAME%(3,0,5), 0.0
*SET,%_FNCNAME%(4,0,5), 0.0
*SET,%_FNCNAME%(5,0,5), 0.0
*SET,%_FNCNAME%(6,0,5), 0.0
*SET,%_FNCNAME%(0,1,5), 1.0, 99, 0, 0.016, 0, 0, 0
*SET,%_FNCNAME%(0,2,5),     0
*SET,%_FNCNAME%(0,3,5),     0
! End of equation: 0.016
```

```
!-->
SECT,1,SHELL,,
SECDATA,CRThick,2,0.0,3 ! CR layer
SECDATA,SBRThick,1,0.0,3 ! SBR layer
SECDATA,FIBThick,4,54,3 ! FIB layer
SECDATA,FIBThick,4,-54,3 ! FIB layer
SECDATA,EPDMThick,3,0.0,3 ! EPDM layer
SECOFFSET,MID ! Middle surface set as shell surface
secf,%DBVTHK%,0
SECCONTROL,0,0,0,0,1,1,1
!*
!* ++++++++++++++++++ Creates geometry ++++++++++++++++++++++++++++++
!*
!* Creates keypoints
*DIM,KPXYZ,,3,135 ! Defines 3x9x15 keypoint array
*VREAD,KPXYZ(1,1),PROFILE(1),DAT,,IJK,3,135
(3F9.5)
NUM_KP=1 ! First keypoint number
*DO,JK,1,135,1
K,NUM_KP,KPXYZ(1,JK),KPXYZ(2,JK),KPXYZ(3,JK) ! Creates keypoints
NUM_KP=NUM_KP+1
*ENDDO
!* Draws 9 B-spline curves
*DO,IKP,1,9
FLST,3,15,3
*DO,JKP,(IKP-1)*15+1,(IKP-1)*15+15
FITEM,3,JKP
*ENDDO
BSPLINE,,P51X ! Draws splines
*ENDDO
!* Skins 8 areas
*DO,IA,1,8
FLST,2,2,4
FITEM,2,IA
FITEM,2,IA+1
ASKIN,P51X ! Skins surfaces
*ENDDO
AREVERSE,ALL ! Reverse surface direction
ALLS
```

```
!*
!* +++++++++++++++++ Meshing work ++++++++++++++++++++++++++++++++++++
!*
*DO,I,1,9
LESIZE,I,,,MXNUM,0.5,1,,, ! 0.5 Spacing ratio
*ENDDO
!*
*DO,I,10,25
LESIZE,I,,,NINT(MZNUM/8),,1,,,
*ENDDO
!*
MSHKEY,1
AMESH,ALL
MSHKEY,0
!*
!* ++++++++++++++++ Define boundary conditions ++++++++++++++++++++++++
!*
LSEL,S,LOC,Z,ZDBV
DL,ALL,,ALL,0 ! Fixed flange
ALLS
!*
LSEL,S,LOC,X,0 ! Symmetric centre edge
DL,ALL,,UX,0
DL,ALL,,ROTY,0
DL,ALL,,ROTZ,0
ALLS
!*
LSEL,S,LOC,Y,0 ! Symmetric side edge
DL,ALL,,UY,0
DL,ALL,,ROTX,0
DL,ALL,,ROTZ,0
ALLS
!*
!* +++++++++++++++++ Sorts node numbers ++++++++++++++++++++++++++++++++
!*
! Sets a 2D array to record all node numbers
*DIM,NDNUM,ARRAY,MXNUM+1,MZNUM+1,
! Do-loops of each row for sorting the node numbers according to node positions
! in the x coordinate
```

```
*DO,IXF,1,MZNUM+1
NX1=0
NX2=0
NXTEMP=0
ICOUNT=0
MINZ=(IXF-1)*MSIZE
NSEL,S,LOC,Z,MINZ, ! Selects nodes along z direction
*GET,NDMIN,NODE,,NUM,MIN ! Gets the minimum node number
NDNUM(1,IXF)=NDMIN
*DO,IND,1,MXNUM,1
NDNUM(IND+1,IXF)= NDNEXT(NDNUM(IND,IXF)) ! Sorts node numbers
*ENDDO
!* Bubble sorting
*DO,I,MXNUM,1,-1
*DO,J,1,I
NX1=NX(NDNUM(J,IXF)) ! X coordintes of two neighbor nodes
NX2=NX(NDNUM(J+1,IXF))
*IF,NX1,GT,NX2,THEN ! Exchanges node numbers
NXTEMP=NDNUM(J,IXF)
NDNUM(J,IXF)=NDNUM(J+1,IXF)
NDNUM(J+1,IXF)=NXTEMP
*ENDIF
*ENDDO
*ENDDO
*ENDDO
ALLS
!*
!* +++++++++++++++++ Generates link10 elements ++++++++++++++++++++++++
!*
TYPE,2 ! Select LINK10 attribution
MAT,5 ! Select material #5
REAL,2
TSHAP,LINE
*DO,I,1,NINT(ZBILL/MSIZE)+1
*DO,J,1,NINT(MXNUM-MXNUM/2)
! Copies node, node number increment 100000, transposition y=-0.01m
NGEN,2,100000,NDNUM(J,I),,,,-0.01,,1,
E,100000+NDNUM(J,I),NDNUM(J,I) ! Creates LINK10
D,100000+NDNUM(J,I),ALL,0,,,,,,,, ! Fixes the ends
```

```
*ENDDO
*ENDDO
ALLSEL, ALL
!* ==============================================================
/SOLU ! solution
!* ==============================================================
ANTYPE,4 ! Transient solver
TRNOPT,FULL ! Transient option, full method
LUMPM,0 ! Closes the lumped mass matrix
NLGEOM,1 ! Large deflection geom
AUTOTS,1 ! Switches on auto time step
NSUBST,20,100000,20 ! Defines substep,MIN 20,MAX 100000
NEQIT,100 ! Defines equilibrium iteration steps
KBC,0 ! RAMPED STEP
PRED,OFF ! closes predictor, not recommended with rotational DOF
EQSLV,SPARSE,, ! Select a sparse solver
RESC,,NONE ! Do not keep any restart files
TRNOPT,FULL,,,,,HHT ! HHT time integration method
TIMINT,ON ! Turn on time integration effects
ALPHAD,AlphaDamping ! Defines the mass matrix multiplier for damping
BETAD,BetaDamping ! Defines the stiffness matrix multiplier for damping
CNVTOL,F,,0.05,,0.01 !Sets convergence values for force
!/PSF,PRES,NORM,2,0,1 ! Uses arrows to show pressure loads
!*
!* ############################## starts solution #############################
!*
! Sets a 2D array to record pressure, velocity, and cross section area of
! potential flow grids, the 1D grids are identical to MZNUM
*DIM,PVA,ARRAY,MZNUM+1,4,
*DIM,ZVTK,ARRAY,MZNUM+1
*VFILL,ZVTK(1),RAMP,0,0
! Sets a 2D array to record outlet variable history,including mass flow rate,
! kg/s, velocity, m/s, and area, m^2
*DIM,QVA,ARRAY,LOOPS,5
*VFILL,QVA(1,1),RAMP,0,0
*VFILL,QVA(1,2),RAMP,0,0
*VFILL,QVA(1,3),RAMP,0,0
*VFILL,QVA(1,4),RAMP,0,0
*VFILL,QVA(1,5),RAMP,0,0
```

```
ZVT0=0.0 ! previous zeta*V/DTIME
DDP=(DPmax-DPmin)/NRAMP ! Pressure load increment
DP=DPmin ! Initializes DP value
!*
*DO,ISOL,1,LOOPS ! Do-loops of solution
*IF,ISOL,LE,NRAMP,THEN
DP=DP+DDP ! Rump up DP
*ENDIF
ALLS ! Selects all entities
SFEDELE,ALL,ALL,ALL ! Deletes all surface loads
*VFILL,PVA(1,1),RAMP,0,0 ! Array reset, P
*VFILL,PVA(1,2),RAMP,0,0 ! V
*VFILL,PVA(1,3),RAMP,0,0 ! A
*VFILL,PVA(1,4),RAMP,0,0 ! Location
zeta=0
ksi=0
!* =================== Determines crosssection areas =======================
*DO,IXF,1,MZNUM+1
NDX=0
NDY=0
NDX1=0
NDY1=0
!* ----------------------------------------------------------------------------
*DO,IND,1,MXNUM
NDX=NX(NDNUM(IND,IXF))+UX(NDNUM(IND,IXF)) ! New node x coordinates
NDY=NY(NDNUM(IND,IXF))+UY(NDNUM(IND,IXF)) ! New node y coordinates
NDX1=NX(NDNUM(IND+1,IXF))+UX(NDNUM(IND+1,IXF)) ! New next node x coordinates
NDY1=NY(NDNUM(IND+1,IXF))+UY(NDNUM(IND+1,IXF)) ! New next node y coordinates
! Crosssection area calculation, sum of trapezoid area, actually the trapezoid
! area is a projected area on xy plane, assuming that the deformation in z
! direction is small
PVA(IXF,3)=PVA(IXF,3)+0.5*ABS(NDY1+NDY)*ABS(NDX1-NDX)
*ENDDO
*IF,IXF,LE,NINT(ZBILL/MSIZE)+1,THEN
PVA(IXF,3)=PVA(IXF,3)-A1
*ELSEIF,IXF,LE,NINT((ZSADDLE+ZBILL)/MSIZE)+1
B1=(A2-A1)*(IXF-NINT(ZBILL/MSIZE)-1)/(NINT((ZSADDLE)/MSIZE))
PVA(IXF,3)=PVA(IXF,3)-A1-B1
*ELSE
```

```
PVA(IXF,3)=PVA(IXF,3)-A2
*ENDIF
*ENDDO
ALLS
!* ====================== Evaluates QVA =============================
Aout=PVA(1,3) ! PVA(1,3) is outlet area
ksi=0.5*((Ain/Aout)**2-1) !ksi
*DO,kk,2,MZNUM+1
zeta=zeta+0.5*MSIZE*(Ain/PVA(kk,3)+Ain/PVA(kk-1,3)) !zeta
*ENDDO
! inlet velocity
Vin=0.5*(sqrt(4*ksi*(ZVT0+DP/DenW)+(zeta/DTIME)**2)-zeta/DTIME)/ksi
Q=DenW*Vin*Ain ! Updates Q
Vout=Q/(DenW*Aout)
!* -------------- Determines pressure and velocity distributions ---------------
PVA(1,2)=Vout
PVA(1,1)=0
PVA(1,4)=0
!
QVA(ISOL,1)=Q
QVA(ISOL,2)=Vout
QVA(ISOL,3)=Aout
QVA(ISOL,4)=ISOL*DTIME
QVA(ISOL,5)=zeta*Vin/DTIME-ZVT0
ZVT0=zeta*Vin/DTIME ! Updates ZVT0
!
*DO,IXF,2,MZNUM+1
zetak=0
PVA(IXF,2)=Q/ABS(DenW*PVA(IXF,3)) ! Local velocity
PVA(IXF,4)=(IXF-1)*MSIZE ! Location
ksik=0.5*((Ain/PVA(IXF,3))**2-1) !ksik
*DO,kk1,IXF,MZNUM+1
zetak=zetak+0.5*MSIZE*(Ain/PVA(kk1,3)+Ain/PVA(kk1-1,3)) !zetak
*ENDDO
! Local static pressure
Unsteady=zetak*Vin/DTIME-ZVTK(IXF)
PVA(IXF,1)=DP-DenW*(ksik*Vin*Vin+Unsteady)
ZVTK(IXF)=zetak*Vin/DTIME
*ENDDO
```

```
!*
!* +++++++++++++++++++ Imposes pressure loads +++++++++++++++++++++++++
!*
!* Duckbill portion
*DO,IXF,1,NINT(ZBILL/MSIZE)
*DO,IND,1,MXNUM-3 ! Excludes side edge
PP=0.5*(PVA(IXF,1)+PVA(IXF+1,1))
FLST,2,4,1,ORDE,4
FITEM,2,NDNUM(IND,IXF)
FITEM,2,NDNUM(IND+1,IXF)
FITEM,2,NDNUM(IND,IXF+1)
FITEM,2,NDNUM(IND+1,IXF+1)
SF,P51X,PRES,PP
*ENDDO
*ENDDO
!* Other portions of DBV
*DO,IXF,NINT(ZBILL/MSIZE)+1,MZNUM
*DO,IND,1,MXNUM
PP=0.5*(PVA(IXF,1)+PVA(IXF+1,1))
FLST,2,4,1,ORDE,4
FITEM,2,NDNUM(IND,IXF)
FITEM,2,NDNUM(IND+1,IXF)
FITEM,2,NDNUM(IND,IXF+1)
FITEM,2,NDNUM(IND+1,IXF+1)
SF,P51X,PRES,PP
*ENDDO
*ENDDO
!* ++++++++++++++++++++ Loadstep solving ++++++++++++++++++++++++++++++
ALLSEL, ALL ! Selects all entities
TIME,ISOL*DTIME ! Loadstep time, increases 1 unit per loop
*IF,ISOL,LE,NRAMP,THEN
TINTP,0.5-ISOL*(0.5-NumDamping)/NRAMP ! Numerical damping, GAMMA
*ENDIF
LSWRITE,ISOL ! Writes loadstep file
LSSOLVE,ISOL,ISOL ! Solution of loadstep file
*ENDDO
!*
!* ############################# Solution loop end ##############################
!*
```

```
SAVE,'SH1DModel','DB', ! Saved in a default path
FINISH
!* =======================================================
/POST1 ! Postprocessing
!* =======================================================
!* outputs deformed DBV geometry
*CFOPEN,DEFORMGEOM,txt,,APPEND ! writes node coordinate file
*DIM,NDXYZ,ARRAY,3
*DO,IXF,1,MZNUM+1
*DO,IND,1,MXNUM+1
NDXYZ(1)=NX(NDNUM(IND,IXF))+UX(NDNUM(IND,IXF))
NDXYZ(2)=NY(NDNUM(IND,IXF))+UY(NDNUM(IND,IXF))
NDXYZ(3)=NZ(NDNUM(IND,IXF))+UZ(NDNUM(IND,IXF))
*VWRITE,NDXYZ(1),NDXYZ(2),NDXYZ(3)
(3F12.6)
*ENDDO
*ENDDO
*CFCLOS
!*
*CFOPEN,PVA,txt,, ! Writes pressure, velocity and cross section area
*VWRITE
(/'# Location(m)---P(Pa)---V(m/s)---A(m^2) #'/)
*VWRITE,PVA(1,4),PVA(1,1),PVA(1,2),PVA(1,3)
(F8.4,3F15.6)
*CFCLOS
!* ---------------------------------------------------------------------------
*CFOPEN,QVA,txt,, ! Writes outlet variable time history
*VWRITE
(/'# Time(s)---Q(kg/s---V(m/s)---A(m^2)---Unsteady Item #'/)
*VWRITE,QVA(1,4),QVA(1,1),QVA(1,2),QVA(1,3),QVA(1,5)
(F8.4,4F15.6)
*CFCLOS
!*
!* ++++++++++++++++ Creates BMP pictures ++++++++++++++++++++++++++++++
!*
!ESEL,U,TYPE,,2 ! Excludes link10
!PLNSOL,U,Y,0,1.0 ! Node solution, y displacement
!* User defirned expension of symmetry
!/ESHAPE,0.9 ! Views in 3D
```

```
!/EXPAND,1,RECT,HALF,0.00001,0,0,1,RECT,HALF,0,0.00001,0,,RECT,FULL,,,,
!/REPLOT
!/VIEW,1,0,0,-1 ! Z view
!/ANG,1
!/REP,FAST
!FILENAME='PT'
!*DO,I,1,LOOPS,NINT(LOOPS/100)
!SET,I ! Reads result data according to file No
!/REPLOT
!* Saves BMP files
!/IMAGE,SAVE,'H:\DBVSH1D-1\10kpa\BMP\%FILENAME%%I%','BMP',
!*ENDDO
!/VIEW,1,0.45,0.45,-0.72 ! Isometric view
!/ANG,1
!/REP,FAST
!FILENAME='PTT'
!*DO,I,1,LOOPS,NINT(LOOPS/100)
!SET,I
!/REPLOT
!/IMAGE,SAVE,'H:\DBVSH1D-1\10kpa\BMP1\%FILENAME%%I%','BMP',
!*ENDDO
FINISH
```

APPENDIX B

A Solid Valve Model Coupled with Full Viscous Flow

For a full FSI model of duckbill valve, a number of challenging issues exists on both the structure side and the fluid flow side. For instance, dynamic mesh becomes the most challenging problem for the viscous flow model. When the valve was fully closed, the fluid domain was separated into two sub-domains, which were not connected with each other by meshes. In this situation, special data transportation issue between the two sub-domains had to be overcome using user defined subroutine so that the governing equations of viscous flow can be solved correctly. After the valve was initially opened with a small opening, a thin fluid sub-domain was built at the duckbill portion to connect the upstream and downstream fluid domains as shown in Fig. B-1 (a). However, this initial valve opening was so narrow that meshes in this slot were not enough and a highly skillful mesh operation was required. Fig. B-1 (b) shows eight layers of tetrahedral meshes in the originally 5mm thick slot are stretched as the valve is fully open. Because of the limitations of the spring dynamic mesh technology of CFX the eight layers of meshes cannot be further refined to model the turbulent flow through the valve tunnel. In addition, there is a bug in the FSI framework (ANSYS+CFX), which results in numerical divergence if the fluid flow is incompressible. To overcome this bug, a compressible water flow model had to be applied. On the solid side, since the reorientation of the layer solid elements is a challenging operation during the mesh generation, the layered material could not be applied to the solid model. Thus, only a homogenous solid valve model was built. Fig. B-2 shows the full FSI model of duckbill valve with the velocity contour of x-z center plane under a driving pressure of 40kPa. Generally, although this full FSI model was numerically stable, its simulation results did not meet expectations. We wish these challenging issues could be effectively solved in the future.

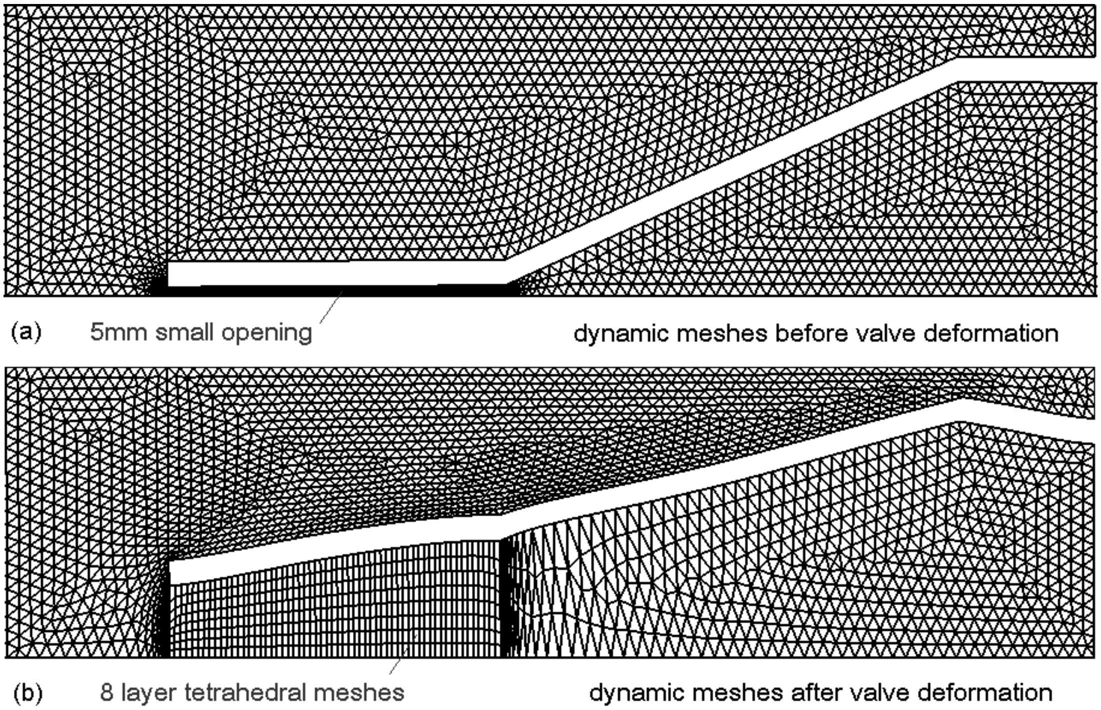
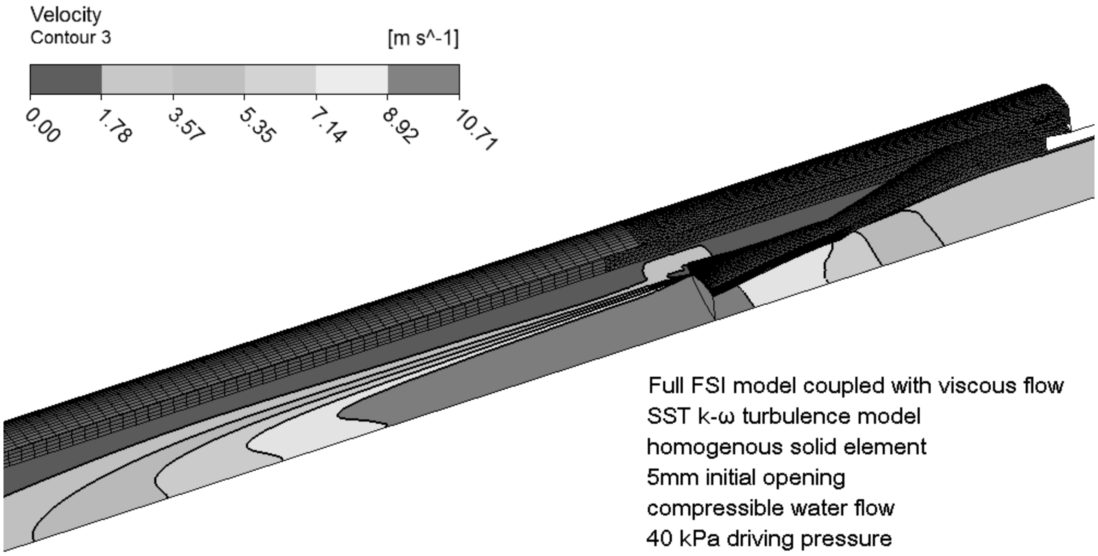


Fig. B-1 Dynamic meshes for the viscous flow model



Full FSI model coupled with viscous flow
SST k-ω turbulence model
homogenous solid element
5mm initial opening
compressible water flow
40 kPa driving pressure

Fig. B-2 A solid valve model coupled with viscous flow

APPENDIX C

FORTRAN Codes for Seeking the Internal and External Surfaces of DBV

The FORTRAN Code (1) is applied to find the internal surface data based on the deformed middle surface predicted by the simplified coupled model, while the Code (2) is used to find the external surface.

Code (1)

```
PROGRAM DBV_INTERNAL_SURFACE_DESIGN
IMPLICIT NONE
INTEGER::I,J,K,M=1
! DEFINE KEYPOINT ARRAY DIMENSIONS:
! NODAL POINTS OF DEFORMED MIDDLE-SURFACE
! 49=KEYPOINT GROUP ID, 29=KEYPOINT ID, 3=KEYPOINT XYZ COORDINATES
REAL::KPS1(3,29,49)
! KEYPOINTS OF INTERNAL-SURFACE
! 49=KEYPOINT GROUP ID, 29=KEYPOINT ID, 3=KEYPOINT XYZ COORDINATES
REAL::KPS2(3,29,49)
! OTHER VARIABLES
INTEGER:: COUNT(49)
REAL::L1,L2=0.00719,L0,X,Y,PI=3.1415926,EPS,EPS1
! ---------------------------------------------------------------------------
! OPEN KEYPOINT DATA FILES
   OPEN(UNIT=10,FILE='H:\DBVReconstruction\InternalSurface\10kpa-54\10kpa54.txt')
   OPEN(UNIT=20,FILE='H:\DBVReconstruction\InternalSurface\10kpa-54\10kpa54in.txt')
! READ FROM MID-SURFACE KEYPOINT FILE
   READ(10,*)KPS1
!
   DO I=1,49
      KPS2(1,1,I)=KPS1(1,1,I)
        KPS2(2,1,I)=KPS1(2,1,I)-L2
          KPS2(3,1,I)=KPS1(3,1,I)
!
     DO J=2,29
      EPS1=10.
      L0=SQRT((KPS1(1,J,I)-KPS2(1,J-1,I))**2+(KPS1(2,J,I)-KPS2(2,J-1,I))**2 )
        IF(KPS1(2,J,I).GE.KPS2(2,J-1,I))THEN
            DO K=1,101
```

```
        X=KPS1(1,J,I)-L2*COS(0.25*PI+(K-1)*PI/400)
         Y=KPS1(2,J,I)-L2*SIN(0.25*PI+(K-1)*PI/400)
          L1=SQRT((X-KPS2(1,J-1,I))**2+(Y-KPS2(2,J-1,I))**2)
           EPS=ABS(L1*L1+L2*L2-L0*L0)
         IF(EPS.LT.EPS1)THEN
           EPS1=EPS
            M=K
         ENDIF
        ENDDO
     ELSE
       DO K=1,101
        X=KPS1(1,J,I)-L2*COS((K-1)*PI/400)
         Y=KPS1(2,J,I)-L2*SIN((K-1)*PI/400)
          L1=SQRT((X-KPS2(1,J-1,I))**2+(Y-KPS2(2,J-1,I))**2)
           EPS=ABS(L1*L1+L2*L2-L0*L0)
         IF(EPS.LT.EPS1)THEN
           EPS1=EPS
            M=K
         ENDIF
       ENDDO
     ENDIF
!
    IF(KPS1(2,J,I).GE.KPS2(2,J-1,I))THEN
     KPS2(1,J,I)=KPS1(1,J,I)-L2*COS(0.25*PI+(M-1)*PI/400)
      KPS2(2,J,I)=KPS1(2,J,I)-L2*SIN(0.25*PI+(M-1)*PI/400)
       KPS2(3,J,I)=KPS1(3,J,I)
    ELSE
     KPS2(1,J,I)=KPS1(1,J,I)-L2*COS((M-1)*PI/400)
      KPS2(2,J,I)=KPS1(2,J,I)-L2*SIN((M-1)*PI/400)
       KPS2(3,J,I)=KPS1(3,J,I)
    ENDIF
     IF((KPS1(1,29,I)-KPS2(1,J,I)-L2).LE.0)THEN
          KPS2(1,J,I)=KPS1(1,29,I)-L2
     ENDIF
     IF(KPS2(2,J,I).LE.0.002)THEN
          KPS2(2,J,I)=0
     ENDIF
   ENDDO
  ENDDO
```

```
! WRITE TO BOT-SURFACE KEYPOINT FILE
   DO J=1,29
      KPS2(3,J,1)=0
   ENDDO
!
   DO I=1,49
    DO J=1,29
         WRITE(20,"(3F12.6)")KPS2(1:3,J,I)
         IF(KPS2(2,J,I).LE.0)THEN
            COUNT(I)=J
            GOTO 111
         ENDIF
        ENDDO
111 ENDDO
!
   CLOSE(UNIT=10)
   CLOSE(UNIT=20)
! COUNTING THE KEYPOINTS OF EACH CURVE
   write (*,"(10I4)") COUNT(1:49)
STOP
END
```

Code (2)

```
PROGRAM DBV_SEEK_EXTERNAL_SURFACE
IMPLICIT NONE
INTEGER::I,J,K,M=1
! DEFINE KEYPOINT ARRAY DIMENSIONS:
! NODAL POINTS OF DEFORMED MIDDLE-SURFACE
! 49=KEYPOINT GROUP ID, 29=KEYPOINT ID, 3=KEYPOINT XYZ COORDINATES
REAL::KPS1(3,29,49)
! KEYPOINTS OF EXTERNAL-SURFACE
! 9=KEYPOINT GROUP ID, 15=KEYPOINT ID, 3=KEYPOINT XYZ COORDINATES
REAL::KPS2(3,29,49)
! OTHER VARIABLES
REAL::L1,L2=0.0035,L0,X,Y,PI=3.1415926,EPS,EPS1
! ---------------------------------------------------------------------------
! OPEN KEYPOINT DATA FILES
   OPEN(UNIT=10,FILE='H:\DBVReconstruction\ExternalSurface\10kpa-54\10kpa54.txt')
   OPEN(UNIT=20,FILE='H:\DBVReconstruction\ExternalSurface\10kpa-54\10kpa54ex.txt')
```

```
! READ KEYPOINT DATA FROM THE ORIGINAL SURFACE FILE
   READ(10,*)KPS1
!
   DO I=1,49
      KPS2(1,1,I)=KPS1(1,1,I)
        KPS2(2,1,I)=KPS1(2,1,I)+L2
          KPS2(3,1,I)=KPS1(3,1,I)
      KPS2(1,29,I)=KPS1(1,29,I)+L2
        KPS2(2,29,I)=KPS1(2,29,I)
          KPS2(3,29,I)=KPS1(3,29,I)
!
     DO J=2,28
      EPS1=10.
      L0=SQRT((KPS1(1,J,I)-KPS2(1,J-1,I))**2+(KPS1(2,J,I)-KPS2(2,J-1,I))**2 )
        IF(KPS1(2,J,I).LE.KPS2(2,J-1,I))THEN
            DO K=1,201
              X=KPS1(1,J,I)+L2*COS(0.0*PI/12+(K-1)*PI/400)
                Y=KPS1(2,J,I)+L2*SIN(0.0*PI/12+(K-1)*PI/400)
                  L1=SQRT((X-KPS2(1,J-1,I))**2+(Y-KPS2(2,J-1,I))**2)
                    EPS=ABS(L1*L1+L2*L2-L0*L0)
                IF(EPS < EPS1 .AND. EPS > 0.00)THEN
                    EPS1=EPS
                      M=K
                ENDIF
              ENDDO
        ELSE
            DO K=1,201
              X=KPS1(1,J,I)+L2*COS((K-1)*PI/400)
                Y=KPS1(2,J,I)+L2*SIN((K-1)*PI/400)
                  L1=SQRT((X-KPS2(1,J-1,I))**2+(Y-KPS2(2,J-1,I))**2)
                    EPS=ABS(L1*L1+L2*L2-L0*L0)
                IF(EPS < EPS1 .AND. EPS > 0.00)THEN
                    EPS1=EPS
                      M=K
                ENDIF
            ENDDO
        ENDIF
!     IF SEEKING EXTERNAL POINTS OUTSIDE THE ORIGINAL SURFACE!
      IF(KPS1(2,J,I).LE.KPS2(2,J-1,I))THEN
```

```
     KPS2(1,J,I)=KPS1(1,J,I)+L2*COS(0.0*PI/12+(M-1)*PI/400)
       KPS2(2,J,I)=KPS1(2,J,I)+L2*SIN(0.0*PI/12+(M-1)*PI/400)
         KPS2(3,J,I)=KPS1(3,J,I)
    ELSE
      KPS2(1,J,I)=KPS1(1,J,I)+L2*COS((M-1)*PI/400)
        KPS2(2,J,I)=KPS1(2,J,I)+L2*SIN((M-1)*PI/400)
          KPS2(3,J,I)=KPS1(3,J,I)
    ENDIF
  ENDDO
 ENDDO
! WRITE TO BOT-SURFACE KEYPOINT FILE
 DO J=1,29
   KPS2(3,J,1)=0
 ENDDO
!
 DO I=1,49
  DO J=1,29
      WRITE(20,"(3F9.6)")KPS2(1:3,J,I)
  ENDDO
 ENDDO
!
 CLOSE(UNIT=10)
 CLOSE(UNIT=20)
STOP
END
```

APPENDIX D

A Workbench Script for the Deformed DBV

This Workbench user defined script is developed to generate the valve geometry based on the two deformed internal and external surface data computed by the FORTRAN codes in Appendix C, and set up the meshes and boundary conditions of the fluid domain automatically.

```
/ Journal File for WORKBENCH, Database 2.4.4, ntx86 SP2007051421
/ Identifier "default_id2720"
/ File opened for write Mon Jun 25 19:30:55 2012.
import vertexdata "C:\\Temp\\36kpa54ex.txt"
import vertexdata "C:\\Temp\\36kpa54in.txt"
edge create nurbs "vertex.1" "vertex.2" "vertex.3" "vertex.4" "vertex.5" \
  "vertex.6" "vertex.7" "vertex.8" "vertex.9" "vertex.10" "vertex.11" \
  "vertex.12" "vertex.13" "vertex.14" "vertex.15" "vertex.16" "vertex.17" \
  "vertex.18" "vertex.19" "vertex.20" "vertex.21" "vertex.22" "vertex.23" \
  "vertex.24" "vertex.25" "vertex.26" "vertex.27" "vertex.28" "vertex.29" \
  interpolate
edge create nurbs "vertex.30" "vertex.31" "vertex.32" "vertex.33" "vertex.34" \
  "vertex.35" "vertex.36" "vertex.37" "vertex.38" "vertex.39" "vertex.40" \
  "vertex.41" "vertex.42" "vertex.43" "vertex.44" "vertex.45" "vertex.46" \
  "vertex.47" "vertex.48" "vertex.49" "vertex.50" "vertex.51" "vertex.52" \
  "vertex.53" "vertex.54" "vertex.55" "vertex.56" "vertex.57" "vertex.58" \
  interpolate
edge create nurbs "vertex.59" "vertex.60" "vertex.61" "vertex.62" "vertex.63" \
  "vertex.64" "vertex.65" "vertex.66" "vertex.67" "vertex.68" "vertex.69" \
  "vertex.70" "vertex.71" "vertex.72" "vertex.73" "vertex.74" "vertex.75" \
  "vertex.76" "vertex.77" "vertex.78" "vertex.79" "vertex.80" "vertex.81" \
  "vertex.82" "vertex.83" "vertex.84" "vertex.85" "vertex.86" "vertex.87" \
  interpolate
edge create nurbs "vertex.88" "vertex.89" "vertex.90" "vertex.91" "vertex.92" \
  "vertex.93" "vertex.94" "vertex.95" "vertex.96" "vertex.97" "vertex.98" \
  "vertex.99" "vertex.100" "vertex.101" "vertex.102" "vertex.103" \
  "vertex.104" "vertex.105" "vertex.106" "vertex.107" "vertex.108" \
  "vertex.109" "vertex.110" "vertex.111" "vertex.112" "vertex.113" \
```

```
   "vertex.114" "vertex.115" "vertex.116" interpolate
edge create nurbs "vertex.117" "vertex.118" "vertex.119" "vertex.120" \
   "vertex.121" "vertex.122" "vertex.123" "vertex.124" "vertex.125" \
   "vertex.126" "vertex.127" "vertex.128" "vertex.129" "vertex.130" \
   "vertex.131" "vertex.132" "vertex.133" "vertex.134" "vertex.135" \
   "vertex.136" "vertex.137" "vertex.138" "vertex.139" "vertex.140" \
   "vertex.141" "vertex.142" "vertex.143" "vertex.144" "vertex.145" \
   interpolate
edge create nurbs "vertex.146" "vertex.147" "vertex.148" "vertex.149" \
   "vertex.150" "vertex.151" "vertex.152" "vertex.153" "vertex.154" \
   "vertex.155" "vertex.156" "vertex.157" "vertex.158" "vertex.159" \
   "vertex.160" "vertex.161" "vertex.162" "vertex.163" "vertex.164" \
   "vertex.165" "vertex.166" "vertex.167" "vertex.168" "vertex.169" \
   "vertex.170" "vertex.171" "vertex.172" "vertex.173" "vertex.174" \
   interpolate
edge create nurbs "vertex.175" "vertex.176" "vertex.177" "vertex.178" \
   "vertex.179" "vertex.180" "vertex.181" "vertex.182" "vertex.183" \
   "vertex.184" "vertex.185" "vertex.186" "vertex.187" "vertex.188" \
   "vertex.189" "vertex.190" "vertex.191" "vertex.192" "vertex.193" \
   "vertex.194" "vertex.195" "vertex.196" "vertex.197" "vertex.198" \
   "vertex.199" "vertex.200" "vertex.201" "vertex.202" "vertex.203" \
   interpolate
edge create nurbs "vertex.204" "vertex.205" "vertex.206" "vertex.207" \
   "vertex.208" "vertex.209" "vertex.210" "vertex.211" "vertex.212" \
   "vertex.213" "vertex.214" "vertex.215" "vertex.216" "vertex.217" \
   "vertex.218" "vertex.219" "vertex.220" "vertex.221" "vertex.222" \
   "vertex.223" "vertex.224" "vertex.225" "vertex.226" "vertex.227" \
   "vertex.228" "vertex.229" "vertex.230" "vertex.231" "vertex.232" \
   interpolate
edge create nurbs "vertex.233" "vertex.234" "vertex.235" "vertex.236" \
   "vertex.237" "vertex.238" "vertex.239" "vertex.240" "vertex.241" \
   "vertex.242" "vertex.243" "vertex.244" "vertex.245" "vertex.246" \
   "vertex.247" "vertex.248" "vertex.249" "vertex.250" "vertex.251" \
   "vertex.252" "vertex.253" "vertex.254" "vertex.255" "vertex.256" \
   "vertex.257" "vertex.258" "vertex.259" "vertex.260" "vertex.261" \
   interpolate
edge create nurbs "vertex.262" "vertex.263" "vertex.264" "vertex.265" \
   "vertex.266" "vertex.267" "vertex.268" "vertex.269" "vertex.270" \
   "vertex.271" "vertex.272" "vertex.273" "vertex.274" "vertex.275" \
```

```
    "vertex.276" "vertex.277" "vertex.278" "vertex.279" "vertex.280" \
    "vertex.281" "vertex.282" "vertex.283" "vertex.284" "vertex.285" \
    "vertex.286" "vertex.287" "vertex.288" "vertex.289" "vertex.290" \
    interpolate
edge create nurbs "vertex.291" "vertex.292" "vertex.293" "vertex.294" \
    "vertex.295" "vertex.296" "vertex.297" "vertex.298" "vertex.299" \
    "vertex.300" "vertex.301" "vertex.302" "vertex.303" "vertex.304" \
    "vertex.305" "vertex.306" "vertex.307" "vertex.308" "vertex.309" \
    "vertex.310" "vertex.311" "vertex.312" "vertex.313" "vertex.314" \
    "vertex.315" "vertex.316" "vertex.317" "vertex.318" "vertex.319" \
    interpolate
edge create nurbs "vertex.320" "vertex.321" "vertex.322" "vertex.323" \
    "vertex.324" "vertex.325" "vertex.326" "vertex.327" "vertex.328" \
    "vertex.329" "vertex.330" "vertex.331" "vertex.332" "vertex.333" \
    "vertex.334" "vertex.335" "vertex.336" "vertex.337" "vertex.338" \
    "vertex.339" "vertex.340" "vertex.341" "vertex.342" "vertex.343" \
    "vertex.344" "vertex.345" "vertex.346" "vertex.347" "vertex.348" \
    interpolate
edge create nurbs "vertex.349" "vertex.350" "vertex.351" "vertex.352" \
    "vertex.353" "vertex.354" "vertex.355" "vertex.356" "vertex.357" \
    "vertex.358" "vertex.359" "vertex.360" "vertex.361" "vertex.362" \
    "vertex.363" "vertex.364" "vertex.365" "vertex.366" "vertex.367" \
    "vertex.368" "vertex.369" "vertex.370" "vertex.371" "vertex.372" \
    "vertex.373" "vertex.374" "vertex.375" "vertex.376" "vertex.377" \
    interpolate
edge create nurbs "vertex.378" "vertex.379" "vertex.380" "vertex.381" \
    "vertex.382" "vertex.383" "vertex.384" "vertex.385" "vertex.386" \
    "vertex.387" "vertex.388" "vertex.389" "vertex.390" "vertex.391" \
    "vertex.392" "vertex.393" "vertex.394" "vertex.395" "vertex.396" \
    "vertex.397" "vertex.398" "vertex.399" "vertex.400" "vertex.401" \
    "vertex.402" "vertex.403" "vertex.404" "vertex.405" "vertex.406" \
    interpolate
edge create nurbs "vertex.407" "vertex.408" "vertex.409" "vertex.410" \
    "vertex.411" "vertex.412" "vertex.413" "vertex.414" "vertex.415" \
    "vertex.416" "vertex.417" "vertex.418" "vertex.419" "vertex.420" \
    "vertex.421" "vertex.422" "vertex.423" "vertex.424" "vertex.425" \
    "vertex.426" "vertex.427" "vertex.428" "vertex.429" "vertex.430" \
    "vertex.431" "vertex.432" "vertex.433" "vertex.434" "vertex.435" \
    interpolate
```

```
edge create nurbs "vertex.436" "vertex.437" "vertex.438" "vertex.439" \
   "vertex.440" "vertex.441" "vertex.442" "vertex.443" "vertex.444" \
   "vertex.445" "vertex.446" "vertex.447" "vertex.448" "vertex.449" \
   "vertex.450" "vertex.451" "vertex.452" "vertex.453" "vertex.454" \
   "vertex.455" "vertex.456" "vertex.457" "vertex.458" "vertex.459" \
   "vertex.460" "vertex.461" "vertex.462" "vertex.463" "vertex.464" \
   interpolate
edge create nurbs "vertex.465" "vertex.466" "vertex.467" "vertex.468" \
   "vertex.469" "vertex.470" "vertex.471" "vertex.472" "vertex.473" \
   "vertex.474" "vertex.475" "vertex.476" "vertex.477" "vertex.478" \
   "vertex.479" "vertex.480" "vertex.481" "vertex.482" "vertex.483" \
   "vertex.484" "vertex.485" "vertex.486" "vertex.487" "vertex.488" \
   "vertex.489" "vertex.490" "vertex.491" "vertex.492" "vertex.493" \
   interpolate
edge create nurbs "vertex.494" "vertex.495" "vertex.496" "vertex.497" \
   "vertex.498" "vertex.499" "vertex.500" "vertex.501" "vertex.502" \
   "vertex.503" "vertex.504" "vertex.505" "vertex.506" "vertex.507" \
   "vertex.508" "vertex.509" "vertex.510" "vertex.511" "vertex.512" \
   "vertex.513" "vertex.514" "vertex.515" "vertex.516" "vertex.517" \
   "vertex.518" "vertex.519" "vertex.520" "vertex.521" "vertex.522" \
   interpolate
edge create nurbs "vertex.523" "vertex.524" "vertex.525" "vertex.526" \
   "vertex.527" "vertex.528" "vertex.529" "vertex.530" "vertex.531" \
   "vertex.532" "vertex.533" "vertex.534" "vertex.535" "vertex.536" \
   "vertex.537" "vertex.538" "vertex.539" "vertex.540" "vertex.541" \
   "vertex.542" "vertex.543" "vertex.544" "vertex.545" "vertex.546" \
   "vertex.547" "vertex.548" "vertex.549" "vertex.550" "vertex.551" \
   interpolate
edge create nurbs "vertex.552" "vertex.553" "vertex.554" "vertex.555" \
   "vertex.556" "vertex.557" "vertex.558" "vertex.559" "vertex.560" \
   "vertex.561" "vertex.562" "vertex.563" "vertex.564" "vertex.565" \
   "vertex.566" "vertex.567" "vertex.568" "vertex.569" "vertex.570" \
   "vertex.571" "vertex.572" "vertex.573" "vertex.574" "vertex.575" \
   "vertex.576" "vertex.577" "vertex.578" "vertex.579" "vertex.580" \
   interpolate
edge create nurbs "vertex.581" "vertex.582" "vertex.583" "vertex.584" \
   "vertex.585" "vertex.586" "vertex.587" "vertex.588" "vertex.589" \
   "vertex.590" "vertex.591" "vertex.592" "vertex.593" "vertex.594" \
   "vertex.595" "vertex.596" "vertex.597" "vertex.598" "vertex.599" \
```

```
  "vertex.600" "vertex.601" "vertex.602" "vertex.603" "vertex.604" \
  "vertex.605" "vertex.606" "vertex.607" "vertex.608" "vertex.609" \
  interpolate
edge create nurbs "vertex.610" "vertex.611" "vertex.612" "vertex.613" \
  "vertex.614" "vertex.615" "vertex.616" "vertex.617" "vertex.618" \
  "vertex.619" "vertex.620" "vertex.621" "vertex.622" "vertex.623" \
  "vertex.624" "vertex.625" "vertex.626" "vertex.627" "vertex.628" \
  "vertex.629" "vertex.630" "vertex.631" "vertex.632" "vertex.633" \
  "vertex.634" "vertex.635" "vertex.636" "vertex.637" "vertex.638" \
  interpolate
edge create nurbs "vertex.639" "vertex.640" "vertex.641" "vertex.642" \
  "vertex.643" "vertex.644" "vertex.645" "vertex.646" "vertex.647" \
  "vertex.648" "vertex.649" "vertex.650" "vertex.651" "vertex.652" \
  "vertex.653" "vertex.654" "vertex.655" "vertex.656" "vertex.657" \
  "vertex.658" "vertex.659" "vertex.660" "vertex.661" "vertex.662" \
  "vertex.663" "vertex.664" "vertex.665" "vertex.666" "vertex.667" \
  interpolate
edge create nurbs "vertex.668" "vertex.669" "vertex.670" "vertex.671" \
  "vertex.672" "vertex.673" "vertex.674" "vertex.675" "vertex.676" \
  "vertex.677" "vertex.678" "vertex.679" "vertex.680" "vertex.681" \
  "vertex.682" "vertex.683" "vertex.684" "vertex.685" "vertex.686" \
  "vertex.687" "vertex.688" "vertex.689" "vertex.690" "vertex.691" \
  "vertex.692" "vertex.693" "vertex.694" "vertex.695" "vertex.696" \
  interpolate
edge create nurbs "vertex.697" "vertex.698" "vertex.699" "vertex.700" \
  "vertex.701" "vertex.702" "vertex.703" "vertex.704" "vertex.705" \
  "vertex.706" "vertex.707" "vertex.708" "vertex.709" "vertex.710" \
  "vertex.711" "vertex.712" "vertex.713" "vertex.714" "vertex.715" \
  "vertex.716" "vertex.717" "vertex.718" "vertex.719" "vertex.720" \
  "vertex.721" "vertex.722" "vertex.723" "vertex.724" "vertex.725" \
  interpolate
edge create nurbs "vertex.726" "vertex.727" "vertex.728" "vertex.729" \
  "vertex.730" "vertex.731" "vertex.732" "vertex.733" "vertex.734" \
  "vertex.735" "vertex.736" "vertex.737" "vertex.738" "vertex.739" \
  "vertex.740" "vertex.741" "vertex.742" "vertex.743" "vertex.744" \
  "vertex.745" "vertex.746" "vertex.747" "vertex.748" "vertex.749" \
  "vertex.750" "vertex.751" "vertex.752" "vertex.753" "vertex.754" \
  interpolate
edge create nurbs "vertex.755" "vertex.756" "vertex.757" "vertex.758" \
```

```
    "vertex.759" "vertex.760" "vertex.761" "vertex.762" "vertex.763" \
    "vertex.764" "vertex.765" "vertex.766" "vertex.767" "vertex.768" \
    "vertex.769" "vertex.770" "vertex.771" "vertex.772" "vertex.773" \
    "vertex.774" "vertex.775" "vertex.776" "vertex.777" "vertex.778" \
    "vertex.779" "vertex.780" "vertex.781" "vertex.782" "vertex.783" \
    interpolate
edge create nurbs "vertex.784" "vertex.785" "vertex.786" "vertex.787" \
    "vertex.788" "vertex.789" "vertex.790" "vertex.791" "vertex.792" \
    "vertex.793" "vertex.794" "vertex.795" "vertex.796" "vertex.797" \
    "vertex.798" "vertex.799" "vertex.800" "vertex.801" "vertex.802" \
    "vertex.803" "vertex.804" "vertex.805" "vertex.806" "vertex.807" \
    "vertex.808" "vertex.809" "vertex.810" "vertex.811" "vertex.812" \
    interpolate
edge create nurbs "vertex.813" "vertex.814" "vertex.815" "vertex.816" \
    "vertex.817" "vertex.818" "vertex.819" "vertex.820" "vertex.821" \
    "vertex.822" "vertex.823" "vertex.824" "vertex.825" "vertex.826" \
    "vertex.827" "vertex.828" "vertex.829" "vertex.830" "vertex.831" \
    "vertex.832" "vertex.833" "vertex.834" "vertex.835" "vertex.836" \
    "vertex.837" "vertex.838" "vertex.839" "vertex.840" "vertex.841" \
    interpolate
edge create nurbs "vertex.842" "vertex.843" "vertex.844" "vertex.845" \
    "vertex.846" "vertex.847" "vertex.848" "vertex.849" "vertex.850" \
    "vertex.851" "vertex.852" "vertex.853" "vertex.854" "vertex.855" \
    "vertex.856" "vertex.857" "vertex.858" "vertex.859" "vertex.860" \
    "vertex.861" "vertex.862" "vertex.863" "vertex.864" "vertex.865" \
    "vertex.866" "vertex.867" "vertex.868" "vertex.869" "vertex.870" \
    interpolate
edge create nurbs "vertex.871" "vertex.872" "vertex.873" "vertex.874" \
    "vertex.875" "vertex.876" "vertex.877" "vertex.878" "vertex.879" \
    "vertex.880" "vertex.881" "vertex.882" "vertex.883" "vertex.884" \
    "vertex.885" "vertex.886" "vertex.887" "vertex.888" "vertex.889" \
    "vertex.890" "vertex.891" "vertex.892" "vertex.893" "vertex.894" \
    "vertex.895" "vertex.896" "vertex.897" "vertex.898" "vertex.899" \
    interpolate
edge create nurbs "vertex.900" "vertex.901" "vertex.902" "vertex.903" \
    "vertex.904" "vertex.905" "vertex.906" "vertex.907" "vertex.908" \
    "vertex.909" "vertex.910" "vertex.911" "vertex.912" "vertex.913" \
    "vertex.914" "vertex.915" "vertex.916" "vertex.917" "vertex.918" \
    "vertex.919" "vertex.920" "vertex.921" "vertex.922" "vertex.923" \
```

```
   "vertex.924" "vertex.925" "vertex.926" "vertex.927" "vertex.928" \
   interpolate
edge create nurbs "vertex.929" "vertex.930" "vertex.931" "vertex.932" \
   "vertex.933" "vertex.934" "vertex.935" "vertex.936" "vertex.937" \
   "vertex.938" "vertex.939" "vertex.940" "vertex.941" "vertex.942" \
   "vertex.943" "vertex.944" "vertex.945" "vertex.946" "vertex.947" \
   "vertex.948" "vertex.949" "vertex.950" "vertex.951" "vertex.952" \
   "vertex.953" "vertex.954" "vertex.955" "vertex.956" "vertex.957" \
   interpolate
edge create nurbs "vertex.958" "vertex.959" "vertex.960" "vertex.961" \
   "vertex.962" "vertex.963" "vertex.964" "vertex.965" "vertex.966" \
   "vertex.967" "vertex.968" "vertex.969" "vertex.970" "vertex.971" \
   "vertex.972" "vertex.973" "vertex.974" "vertex.975" "vertex.976" \
   "vertex.977" "vertex.978" "vertex.979" "vertex.980" "vertex.981" \
   "vertex.982" "vertex.983" "vertex.984" "vertex.985" "vertex.986" \
   interpolate
edge create nurbs "vertex.987" "vertex.988" "vertex.989" "vertex.990" \
   "vertex.991" "vertex.992" "vertex.993" "vertex.994" "vertex.995" \
   "vertex.996" "vertex.997" "vertex.998" "vertex.999" "vertex.1000" \
   "vertex.1001" "vertex.1002" "vertex.1003" "vertex.1004" "vertex.1005" \
   "vertex.1006" "vertex.1007" "vertex.1008" "vertex.1009" "vertex.1010" \
   "vertex.1011" "vertex.1012" "vertex.1013" "vertex.1014" "vertex.1015" \
   interpolate
edge create nurbs "vertex.1016" "vertex.1017" "vertex.1018" "vertex.1019" \
   "vertex.1020" "vertex.1021" "vertex.1022" "vertex.1023" "vertex.1024" \
   "vertex.1025" "vertex.1026" "vertex.1027" "vertex.1028" "vertex.1029" \
   "vertex.1030" "vertex.1031" "vertex.1032" "vertex.1033" "vertex.1034" \
   "vertex.1035" "vertex.1036" "vertex.1037" "vertex.1038" "vertex.1039" \
   "vertex.1040" "vertex.1041" "vertex.1042" "vertex.1043" "vertex.1044" \
   interpolate
edge create nurbs "vertex.1045" "vertex.1046" "vertex.1047" "vertex.1048" \
   "vertex.1049" "vertex.1050" "vertex.1051" "vertex.1052" "vertex.1053" \
   "vertex.1054" "vertex.1055" "vertex.1056" "vertex.1057" "vertex.1058" \
   "vertex.1059" "vertex.1060" "vertex.1061" "vertex.1062" "vertex.1063" \
   "vertex.1064" "vertex.1065" "vertex.1066" "vertex.1067" "vertex.1068" \
   "vertex.1069" "vertex.1070" "vertex.1071" "vertex.1072" "vertex.1073" \
   interpolate
edge create nurbs "vertex.1074" "vertex.1075" "vertex.1076" "vertex.1077" \
   "vertex.1078" "vertex.1079" "vertex.1080" "vertex.1081" "vertex.1082" \
```

```
    "vertex.1083" "vertex.1084" "vertex.1085" "vertex.1086" "vertex.1087" \
    "vertex.1088" "vertex.1089" "vertex.1090" "vertex.1091" "vertex.1092" \
    "vertex.1093" "vertex.1094" "vertex.1095" "vertex.1096" "vertex.1097" \
    "vertex.1098" "vertex.1099" "vertex.1100" "vertex.1101" "vertex.1102" \
    interpolate
edge create nurbs "vertex.1103" "vertex.1104" "vertex.1105" "vertex.1106" \
    "vertex.1107" "vertex.1108" "vertex.1109" "vertex.1110" "vertex.1111" \
    "vertex.1112" "vertex.1113" "vertex.1114" "vertex.1115" "vertex.1116" \
    "vertex.1117" "vertex.1118" "vertex.1119" "vertex.1120" "vertex.1121" \
    "vertex.1122" "vertex.1123" "vertex.1124" "vertex.1125" "vertex.1126" \
    "vertex.1127" "vertex.1128" "vertex.1129" "vertex.1130" "vertex.1131" \
    interpolate
edge create nurbs "vertex.1132" "vertex.1133" "vertex.1134" "vertex.1135" \
    "vertex.1136" "vertex.1137" "vertex.1138" "vertex.1139" "vertex.1140" \
    "vertex.1141" "vertex.1142" "vertex.1143" "vertex.1144" "vertex.1145" \
    "vertex.1146" "vertex.1147" "vertex.1148" "vertex.1149" "vertex.1150" \
    "vertex.1151" "vertex.1152" "vertex.1153" "vertex.1154" "vertex.1155" \
    "vertex.1156" "vertex.1157" "vertex.1158" "vertex.1159" "vertex.1160" \
    interpolate
edge create nurbs "vertex.1161" "vertex.1162" "vertex.1163" "vertex.1164" \
    "vertex.1165" "vertex.1166" "vertex.1167" "vertex.1168" "vertex.1169" \
    "vertex.1170" "vertex.1171" "vertex.1172" "vertex.1173" "vertex.1174" \
    "vertex.1175" "vertex.1176" "vertex.1177" "vertex.1178" "vertex.1179" \
    "vertex.1180" "vertex.1181" "vertex.1182" "vertex.1183" "vertex.1184" \
    "vertex.1185" "vertex.1186" "vertex.1187" "vertex.1188" "vertex.1189" \
    interpolate
edge create nurbs "vertex.1190" "vertex.1191" "vertex.1192" "vertex.1193" \
    "vertex.1194" "vertex.1195" "vertex.1196" "vertex.1197" "vertex.1198" \
    "vertex.1199" "vertex.1200" "vertex.1201" "vertex.1202" "vertex.1203" \
    "vertex.1204" "vertex.1205" "vertex.1206" "vertex.1207" "vertex.1208" \
    "vertex.1209" "vertex.1210" "vertex.1211" "vertex.1212" "vertex.1213" \
    "vertex.1214" "vertex.1215" "vertex.1216" "vertex.1217" "vertex.1218" \
    interpolate
edge create nurbs "vertex.1219" "vertex.1220" "vertex.1221" "vertex.1222" \
    "vertex.1223" "vertex.1224" "vertex.1225" "vertex.1226" "vertex.1227" \
    "vertex.1228" "vertex.1229" "vertex.1230" "vertex.1231" "vertex.1232" \
    "vertex.1233" "vertex.1234" "vertex.1235" "vertex.1236" "vertex.1237" \
    "vertex.1238" "vertex.1239" "vertex.1240" "vertex.1241" "vertex.1242" \
    "vertex.1243" "vertex.1244" "vertex.1245" "vertex.1246" "vertex.1247" \
```

```
   interpolate
edge create nurbs "vertex.1248" "vertex.1249" "vertex.1250" "vertex.1251" \
   "vertex.1252" "vertex.1253" "vertex.1254" "vertex.1255" "vertex.1256" \
   "vertex.1257" "vertex.1258" "vertex.1259" "vertex.1260" "vertex.1261" \
   "vertex.1262" "vertex.1263" "vertex.1264" "vertex.1265" "vertex.1266" \
   "vertex.1267" "vertex.1268" "vertex.1269" "vertex.1270" "vertex.1271" \
   "vertex.1272" "vertex.1273" "vertex.1274" "vertex.1275" "vertex.1276" \
   interpolate
edge create nurbs "vertex.1277" "vertex.1278" "vertex.1279" "vertex.1280" \
   "vertex.1281" "vertex.1282" "vertex.1283" "vertex.1284" "vertex.1285" \
   "vertex.1286" "vertex.1287" "vertex.1288" "vertex.1289" "vertex.1290" \
   "vertex.1291" "vertex.1292" "vertex.1293" "vertex.1294" "vertex.1295" \
   "vertex.1296" "vertex.1297" "vertex.1298" "vertex.1299" "vertex.1300" \
   "vertex.1301" "vertex.1302" "vertex.1303" "vertex.1304" "vertex.1305" \
   interpolate
edge create nurbs "vertex.1306" "vertex.1307" "vertex.1308" "vertex.1309" \
   "vertex.1310" "vertex.1311" "vertex.1312" "vertex.1313" "vertex.1314" \
   "vertex.1315" "vertex.1316" "vertex.1317" "vertex.1318" "vertex.1319" \
   "vertex.1320" "vertex.1321" "vertex.1322" "vertex.1323" "vertex.1324" \
   "vertex.1325" "vertex.1326" "vertex.1327" "vertex.1328" "vertex.1329" \
   "vertex.1330" "vertex.1331" "vertex.1332" "vertex.1333" "vertex.1334" \
   interpolate
edge create nurbs "vertex.1335" "vertex.1336" "vertex.1337" "vertex.1338" \
   "vertex.1339" "vertex.1340" "vertex.1341" "vertex.1342" "vertex.1343" \
   "vertex.1344" "vertex.1345" "vertex.1346" "vertex.1347" "vertex.1348" \
   "vertex.1349" "vertex.1350" "vertex.1351" "vertex.1352" "vertex.1353" \
   "vertex.1354" "vertex.1355" "vertex.1356" "vertex.1357" "vertex.1358" \
   "vertex.1359" "vertex.1360" "vertex.1361" "vertex.1362" "vertex.1363" \
   interpolate
edge create nurbs "vertex.1364" "vertex.1365" "vertex.1366" "vertex.1367" \
   "vertex.1368" "vertex.1369" "vertex.1370" "vertex.1371" "vertex.1372" \
   "vertex.1373" "vertex.1374" "vertex.1375" "vertex.1376" "vertex.1377" \
   "vertex.1378" "vertex.1379" "vertex.1380" "vertex.1381" "vertex.1382" \
   "vertex.1383" "vertex.1384" "vertex.1385" "vertex.1386" "vertex.1387" \
   "vertex.1388" "vertex.1389" "vertex.1390" "vertex.1391" "vertex.1392" \
   interpolate
edge create nurbs "vertex.1393" "vertex.1394" "vertex.1395" "vertex.1396" \
   "vertex.1397" "vertex.1398" "vertex.1399" "vertex.1400" "vertex.1401" \
   "vertex.1402" "vertex.1403" "vertex.1404" "vertex.1405" "vertex.1406" \
```

```
    "vertex.1407" "vertex.1408" "vertex.1409" "vertex.1410" "vertex.1411" \
    "vertex.1412" "vertex.1413" "vertex.1414" "vertex.1415" "vertex.1416" \
    "vertex.1417" "vertex.1418" "vertex.1419" "vertex.1420" "vertex.1421" \
    interpolate
edge create nurbs "vertex.1422" "vertex.1423" "vertex.1424" "vertex.1425" \
    "vertex.1426" "vertex.1427" "vertex.1428" "vertex.1429" "vertex.1430" \
    "vertex.1431" "vertex.1432" "vertex.1433" "vertex.1434" "vertex.1435" \
    "vertex.1436" "vertex.1437" "vertex.1438" "vertex.1439" "vertex.1440" \
    "vertex.1441" "vertex.1442" "vertex.1443" "vertex.1444" "vertex.1445" \
    interpolate
edge create nurbs "vertex.1446" "vertex.1447" "vertex.1448" "vertex.1449" \
    "vertex.1450" "vertex.1451" "vertex.1452" "vertex.1453" "vertex.1454" \
    "vertex.1455" "vertex.1456" "vertex.1457" "vertex.1458" "vertex.1459" \
    "vertex.1460" "vertex.1461" "vertex.1462" "vertex.1463" "vertex.1464" \
    "vertex.1465" "vertex.1466" "vertex.1467" "vertex.1468" "vertex.1469" \
    "vertex.1470" interpolate
edge create nurbs "vertex.1471" "vertex.1472" "vertex.1473" "vertex.1474" \
    "vertex.1475" "vertex.1476" "vertex.1477" "vertex.1478" "vertex.1479" \
    "vertex.1480" "vertex.1481" "vertex.1482" "vertex.1483" "vertex.1484" \
    "vertex.1485" "vertex.1486" "vertex.1487" "vertex.1488" "vertex.1489" \
    "vertex.1490" "vertex.1491" "vertex.1492" "vertex.1493" "vertex.1494" \
    "vertex.1495" "vertex.1496" interpolate
edge create nurbs "vertex.1497" "vertex.1498" "vertex.1499" "vertex.1500" \
    "vertex.1501" "vertex.1502" "vertex.1503" "vertex.1504" "vertex.1505" \
    "vertex.1506" "vertex.1507" "vertex.1508" "vertex.1509" "vertex.1510" \
    "vertex.1511" "vertex.1512" "vertex.1513" "vertex.1514" "vertex.1515" \
    "vertex.1516" "vertex.1517" "vertex.1518" "vertex.1519" "vertex.1520" \
    "vertex.1521" "vertex.1522" interpolate
edge create nurbs "vertex.1523" "vertex.1524" "vertex.1525" "vertex.1526" \
    "vertex.1527" "vertex.1528" "vertex.1529" "vertex.1530" "vertex.1531" \
    "vertex.1532" "vertex.1533" "vertex.1534" "vertex.1535" "vertex.1536" \
    "vertex.1537" "vertex.1538" "vertex.1539" "vertex.1540" "vertex.1541" \
    "vertex.1542" "vertex.1543" "vertex.1544" "vertex.1545" "vertex.1546" \
    "vertex.1547" "vertex.1548" interpolate
edge create nurbs "vertex.1549" "vertex.1550" "vertex.1551" "vertex.1552" \
    "vertex.1553" "vertex.1554" "vertex.1555" "vertex.1556" "vertex.1557" \
    "vertex.1558" "vertex.1559" "vertex.1560" "vertex.1561" "vertex.1562" \
    "vertex.1563" "vertex.1564" "vertex.1565" "vertex.1566" "vertex.1567" \
    "vertex.1568" "vertex.1569" "vertex.1570" "vertex.1571" "vertex.1572" \
```

```
  "vertex.1573" "vertex.1574" interpolate
edge create nurbs "vertex.1575" "vertex.1576" "vertex.1577" "vertex.1578" \
  "vertex.1579" "vertex.1580" "vertex.1581" "vertex.1582" "vertex.1583" \
  "vertex.1584" "vertex.1585" "vertex.1586" "vertex.1587" "vertex.1588" \
  "vertex.1589" "vertex.1590" "vertex.1591" "vertex.1592" "vertex.1593" \
  "vertex.1594" "vertex.1595" "vertex.1596" "vertex.1597" "vertex.1598" \
  "vertex.1599" "vertex.1600" "vertex.1601" interpolate
edge create nurbs "vertex.1602" "vertex.1603" "vertex.1604" "vertex.1605" \
  "vertex.1606" "vertex.1607" "vertex.1608" "vertex.1609" "vertex.1610" \
  "vertex.1611" "vertex.1612" "vertex.1613" "vertex.1614" "vertex.1615" \
  "vertex.1616" "vertex.1617" "vertex.1618" "vertex.1619" "vertex.1620" \
  "vertex.1621" "vertex.1622" "vertex.1623" "vertex.1624" "vertex.1625" \
  "vertex.1626" "vertex.1627" "vertex.1628" interpolate
edge create nurbs "vertex.1629" "vertex.1630" "vertex.1631" "vertex.1632" \
  "vertex.1633" "vertex.1634" "vertex.1635" "vertex.1636" "vertex.1637" \
  "vertex.1638" "vertex.1639" "vertex.1640" "vertex.1641" "vertex.1642" \
  "vertex.1643" "vertex.1644" "vertex.1645" "vertex.1646" "vertex.1647" \
  "vertex.1648" "vertex.1649" "vertex.1650" "vertex.1651" "vertex.1652" \
  "vertex.1653" "vertex.1654" "vertex.1655" interpolate
edge create nurbs "vertex.1656" "vertex.1657" "vertex.1658" "vertex.1659" \
  "vertex.1660" "vertex.1661" "vertex.1662" "vertex.1663" "vertex.1664" \
  "vertex.1665" "vertex.1666" "vertex.1667" "vertex.1668" "vertex.1669" \
  "vertex.1670" "vertex.1671" "vertex.1672" "vertex.1673" "vertex.1674" \
  "vertex.1675" "vertex.1676" "vertex.1677" "vertex.1678" "vertex.1679" \
  "vertex.1680" "vertex.1681" "vertex.1682" interpolate
edge create nurbs "vertex.1683" "vertex.1684" "vertex.1685" "vertex.1686" \
  "vertex.1687" "vertex.1688" "vertex.1689" "vertex.1690" "vertex.1691" \
  "vertex.1692" "vertex.1693" "vertex.1694" "vertex.1695" "vertex.1696" \
  "vertex.1697" "vertex.1698" "vertex.1699" "vertex.1700" "vertex.1701" \
  "vertex.1702" "vertex.1703" "vertex.1704" "vertex.1705" "vertex.1706" \
  "vertex.1707" "vertex.1708" "vertex.1709" interpolate
edge create nurbs "vertex.1710" "vertex.1711" "vertex.1712" "vertex.1713" \
  "vertex.1714" "vertex.1715" "vertex.1716" "vertex.1717" "vertex.1718" \
  "vertex.1719" "vertex.1720" "vertex.1721" "vertex.1722" "vertex.1723" \
  "vertex.1724" "vertex.1725" "vertex.1726" "vertex.1727" "vertex.1728" \
  "vertex.1729" "vertex.1730" "vertex.1731" "vertex.1732" "vertex.1733" \
  "vertex.1734" "vertex.1735" "vertex.1736" interpolate
edge create nurbs "vertex.1737" "vertex.1738" "vertex.1739" "vertex.1740" \
  "vertex.1741" "vertex.1742" "vertex.1743" "vertex.1744" "vertex.1745" \
```

```
    "vertex.1746" "vertex.1747" "vertex.1748" "vertex.1749" "vertex.1750" \
    "vertex.1751" "vertex.1752" "vertex.1753" "vertex.1754" "vertex.1755" \
    "vertex.1756" "vertex.1757" "vertex.1758" "vertex.1759" "vertex.1760" \
    "vertex.1761" "vertex.1762" "vertex.1763" interpolate
edge create nurbs "vertex.1764" "vertex.1765" "vertex.1766" "vertex.1767" \
    "vertex.1768" "vertex.1769" "vertex.1770" "vertex.1771" "vertex.1772" \
    "vertex.1773" "vertex.1774" "vertex.1775" "vertex.1776" "vertex.1777" \
    "vertex.1778" "vertex.1779" "vertex.1780" "vertex.1781" "vertex.1782" \
    "vertex.1783" "vertex.1784" "vertex.1785" "vertex.1786" "vertex.1787" \
    "vertex.1788" "vertex.1789" "vertex.1790" interpolate
edge create nurbs "vertex.1791" "vertex.1792" "vertex.1793" "vertex.1794" \
    "vertex.1795" "vertex.1796" "vertex.1797" "vertex.1798" "vertex.1799" \
    "vertex.1800" "vertex.1801" "vertex.1802" "vertex.1803" "vertex.1804" \
    "vertex.1805" "vertex.1806" "vertex.1807" "vertex.1808" "vertex.1809" \
    "vertex.1810" "vertex.1811" "vertex.1812" "vertex.1813" "vertex.1814" \
    "vertex.1815" "vertex.1816" "vertex.1817" interpolate
edge create nurbs "vertex.1818" "vertex.1819" "vertex.1820" "vertex.1821" \
    "vertex.1822" "vertex.1823" "vertex.1824" "vertex.1825" "vertex.1826" \
    "vertex.1827" "vertex.1828" "vertex.1829" "vertex.1830" "vertex.1831" \
    "vertex.1832" "vertex.1833" "vertex.1834" "vertex.1835" "vertex.1836" \
    "vertex.1837" "vertex.1838" "vertex.1839" "vertex.1840" "vertex.1841" \
    "vertex.1842" "vertex.1843" "vertex.1844" interpolate
edge create nurbs "vertex.1845" "vertex.1846" "vertex.1847" "vertex.1848" \
    "vertex.1849" "vertex.1850" "vertex.1851" "vertex.1852" "vertex.1853" \
    "vertex.1854" "vertex.1855" "vertex.1856" "vertex.1857" "vertex.1858" \
    "vertex.1859" "vertex.1860" "vertex.1861" "vertex.1862" "vertex.1863" \
    "vertex.1864" "vertex.1865" "vertex.1866" "vertex.1867" "vertex.1868" \
    "vertex.1869" "vertex.1870" "vertex.1871" interpolate
edge create nurbs "vertex.1872" "vertex.1873" "vertex.1874" "vertex.1875" \
    "vertex.1876" "vertex.1877" "vertex.1878" "vertex.1879" "vertex.1880" \
    "vertex.1881" "vertex.1882" "vertex.1883" "vertex.1884" "vertex.1885" \
    "vertex.1886" "vertex.1887" "vertex.1888" "vertex.1889" "vertex.1890" \
    "vertex.1891" "vertex.1892" "vertex.1893" "vertex.1894" "vertex.1895" \
    "vertex.1896" "vertex.1897" "vertex.1898" interpolate
edge create nurbs "vertex.1899" "vertex.1900" "vertex.1901" "vertex.1902" \
    "vertex.1903" "vertex.1904" "vertex.1905" "vertex.1906" "vertex.1907" \
    "vertex.1908" "vertex.1909" "vertex.1910" "vertex.1911" "vertex.1912" \
    "vertex.1913" "vertex.1914" "vertex.1915" "vertex.1916" "vertex.1917" \
    "vertex.1918" "vertex.1919" "vertex.1920" "vertex.1921" "vertex.1922" \
```

```
   "vertex.1923" "vertex.1924" "vertex.1925" "vertex.1926" interpolate
edge create nurbs "vertex.1927" "vertex.1928" "vertex.1929" "vertex.1930" \
   "vertex.1931" "vertex.1932" "vertex.1933" "vertex.1934" "vertex.1935" \
   "vertex.1936" "vertex.1937" "vertex.1938" "vertex.1939" "vertex.1940" \
   "vertex.1941" "vertex.1942" "vertex.1943" "vertex.1944" "vertex.1945" \
   "vertex.1946" "vertex.1947" "vertex.1948" "vertex.1949" "vertex.1950" \
   "vertex.1951" "vertex.1952" "vertex.1953" "vertex.1954" interpolate
edge create nurbs "vertex.1955" "vertex.1956" "vertex.1957" "vertex.1958" \
   "vertex.1959" "vertex.1960" "vertex.1961" "vertex.1962" "vertex.1963" \
   "vertex.1964" "vertex.1965" "vertex.1966" "vertex.1967" "vertex.1968" \
   "vertex.1969" "vertex.1970" "vertex.1971" "vertex.1972" "vertex.1973" \
   "vertex.1974" "vertex.1975" "vertex.1976" "vertex.1977" "vertex.1978" \
   "vertex.1979" "vertex.1980" "vertex.1981" "vertex.1982" interpolate
edge create nurbs "vertex.1983" "vertex.1984" "vertex.1985" "vertex.1986" \
   "vertex.1987" "vertex.1988" "vertex.1989" "vertex.1990" "vertex.1991" \
   "vertex.1992" "vertex.1993" "vertex.1994" "vertex.1995" "vertex.1996" \
   "vertex.1997" "vertex.1998" "vertex.1999" "vertex.2000" "vertex.2001" \
   "vertex.2002" "vertex.2003" "vertex.2004" "vertex.2005" "vertex.2006" \
   "vertex.2007" "vertex.2008" "vertex.2009" "vertex.2010" interpolate
edge create nurbs "vertex.2011" "vertex.2012" "vertex.2013" "vertex.2014" \
   "vertex.2015" "vertex.2016" "vertex.2017" "vertex.2018" "vertex.2019" \
   "vertex.2020" "vertex.2021" "vertex.2022" "vertex.2023" "vertex.2024" \
   "vertex.2025" "vertex.2026" "vertex.2027" "vertex.2028" "vertex.2029" \
   "vertex.2030" "vertex.2031" "vertex.2032" "vertex.2033" "vertex.2034" \
   "vertex.2035" "vertex.2036" "vertex.2037" "vertex.2038" interpolate
edge create nurbs "vertex.2039" "vertex.2040" "vertex.2041" "vertex.2042" \
   "vertex.2043" "vertex.2044" "vertex.2045" "vertex.2046" "vertex.2047" \
   "vertex.2048" "vertex.2049" "vertex.2050" "vertex.2051" "vertex.2052" \
   "vertex.2053" "vertex.2054" "vertex.2055" "vertex.2056" "vertex.2057" \
   "vertex.2058" "vertex.2059" "vertex.2060" "vertex.2061" "vertex.2062" \
   "vertex.2063" "vertex.2064" "vertex.2065" "vertex.2066" interpolate
edge create nurbs "vertex.2067" "vertex.2068" "vertex.2069" "vertex.2070" \
   "vertex.2071" "vertex.2072" "vertex.2073" "vertex.2074" "vertex.2075" \
   "vertex.2076" "vertex.2077" "vertex.2078" "vertex.2079" "vertex.2080" \
   "vertex.2081" "vertex.2082" "vertex.2083" "vertex.2084" "vertex.2085" \
   "vertex.2086" "vertex.2087" "vertex.2088" "vertex.2089" "vertex.2090" \
   "vertex.2091" "vertex.2092" "vertex.2093" "vertex.2094" interpolate
edge create nurbs "vertex.2095" "vertex.2096" "vertex.2097" "vertex.2098" \
   "vertex.2099" "vertex.2100" "vertex.2101" "vertex.2102" "vertex.2103" \
```

```
   "vertex.2104" "vertex.2105" "vertex.2106" "vertex.2107" "vertex.2108" \
   "vertex.2109" "vertex.2110" "vertex.2111" "vertex.2112" "vertex.2113" \
   "vertex.2114" "vertex.2115" "vertex.2116" "vertex.2117" "vertex.2118" \
   "vertex.2119" "vertex.2120" "vertex.2121" "vertex.2122" interpolate
edge create nurbs "vertex.2123" "vertex.2124" "vertex.2125" "vertex.2126" \
   "vertex.2127" "vertex.2128" "vertex.2129" "vertex.2130" "vertex.2131" \
   "vertex.2132" "vertex.2133" "vertex.2134" "vertex.2135" "vertex.2136" \
   "vertex.2137" "vertex.2138" "vertex.2139" "vertex.2140" "vertex.2141" \
   "vertex.2142" "vertex.2143" "vertex.2144" "vertex.2145" "vertex.2146" \
   "vertex.2147" "vertex.2148" "vertex.2149" "vertex.2150" interpolate
edge create nurbs "vertex.2151" "vertex.2152" "vertex.2153" "vertex.2154" \
   "vertex.2155" "vertex.2156" "vertex.2157" "vertex.2158" "vertex.2159" \
   "vertex.2160" "vertex.2161" "vertex.2162" "vertex.2163" "vertex.2164" \
   "vertex.2165" "vertex.2166" "vertex.2167" "vertex.2168" "vertex.2169" \
   "vertex.2170" "vertex.2171" "vertex.2172" "vertex.2173" "vertex.2174" \
   "vertex.2175" "vertex.2176" "vertex.2177" "vertex.2178" interpolate
edge create nurbs "vertex.2179" "vertex.2180" "vertex.2181" "vertex.2182" \
   "vertex.2183" "vertex.2184" "vertex.2185" "vertex.2186" "vertex.2187" \
   "vertex.2188" "vertex.2189" "vertex.2190" "vertex.2191" "vertex.2192" \
   "vertex.2193" "vertex.2194" "vertex.2195" "vertex.2196" "vertex.2197" \
   "vertex.2198" "vertex.2199" "vertex.2200" "vertex.2201" "vertex.2202" \
   "vertex.2203" "vertex.2204" "vertex.2205" "vertex.2206" interpolate
edge create nurbs "vertex.2207" "vertex.2208" "vertex.2209" "vertex.2210" \
   "vertex.2211" "vertex.2212" "vertex.2213" "vertex.2214" "vertex.2215" \
   "vertex.2216" "vertex.2217" "vertex.2218" "vertex.2219" "vertex.2220" \
   "vertex.2221" "vertex.2222" "vertex.2223" "vertex.2224" "vertex.2225" \
   "vertex.2226" "vertex.2227" "vertex.2228" "vertex.2229" "vertex.2230" \
   "vertex.2231" "vertex.2232" "vertex.2233" "vertex.2234" interpolate
edge create nurbs "vertex.2235" "vertex.2236" "vertex.2237" "vertex.2238" \
   "vertex.2239" "vertex.2240" "vertex.2241" "vertex.2242" "vertex.2243" \
   "vertex.2244" "vertex.2245" "vertex.2246" "vertex.2247" "vertex.2248" \
   "vertex.2249" "vertex.2250" "vertex.2251" "vertex.2252" "vertex.2253" \
   "vertex.2254" "vertex.2255" "vertex.2256" "vertex.2257" "vertex.2258" \
   "vertex.2259" "vertex.2260" "vertex.2261" "vertex.2262" interpolate
edge create nurbs "vertex.2263" "vertex.2264" "vertex.2265" "vertex.2266" \
   "vertex.2267" "vertex.2268" "vertex.2269" "vertex.2270" "vertex.2271" \
   "vertex.2272" "vertex.2273" "vertex.2274" "vertex.2275" "vertex.2276" \
   "vertex.2277" "vertex.2278" "vertex.2279" "vertex.2280" "vertex.2281" \
   "vertex.2282" "vertex.2283" "vertex.2284" "vertex.2285" "vertex.2286" \
```

```
   "vertex.2287" "vertex.2288" "vertex.2289" "vertex.2290" interpolate
edge create nurbs "vertex.2291" "vertex.2292" "vertex.2293" "vertex.2294" \
   "vertex.2295" "vertex.2296" "vertex.2297" "vertex.2298" "vertex.2299" \
   "vertex.2300" "vertex.2301" "vertex.2302" "vertex.2303" "vertex.2304" \
   "vertex.2305" "vertex.2306" "vertex.2307" "vertex.2308" "vertex.2309" \
   "vertex.2310" "vertex.2311" "vertex.2312" "vertex.2313" "vertex.2314" \
   "vertex.2315" "vertex.2316" "vertex.2317" "vertex.2318" interpolate
edge create nurbs "vertex.2319" "vertex.2320" "vertex.2321" "vertex.2322" \
   "vertex.2323" "vertex.2324" "vertex.2325" "vertex.2326" "vertex.2327" \
   "vertex.2328" "vertex.2329" "vertex.2330" "vertex.2331" "vertex.2332" \
   "vertex.2333" "vertex.2334" "vertex.2335" "vertex.2336" "vertex.2337" \
   "vertex.2338" "vertex.2339" "vertex.2340" "vertex.2341" "vertex.2342" \
   "vertex.2343" "vertex.2344" "vertex.2345" "vertex.2346" "vertex.2347" \
   interpolate
edge create nurbs "vertex.2348" "vertex.2349" "vertex.2350" "vertex.2351" \
   "vertex.2352" "vertex.2353" "vertex.2354" "vertex.2355" "vertex.2356" \
   "vertex.2357" "vertex.2358" "vertex.2359" "vertex.2360" "vertex.2361" \
   "vertex.2362" "vertex.2363" "vertex.2364" "vertex.2365" "vertex.2366" \
   "vertex.2367" "vertex.2368" "vertex.2369" "vertex.2370" "vertex.2371" \
   "vertex.2372" "vertex.2373" "vertex.2374" "vertex.2375" "vertex.2376" \
   interpolate
edge create nurbs "vertex.2377" "vertex.2378" "vertex.2379" "vertex.2380" \
   "vertex.2381" "vertex.2382" "vertex.2383" "vertex.2384" "vertex.2385" \
   "vertex.2386" "vertex.2387" "vertex.2388" "vertex.2389" "vertex.2390" \
   "vertex.2391" "vertex.2392" "vertex.2393" "vertex.2394" "vertex.2395" \
   "vertex.2396" "vertex.2397" "vertex.2398" "vertex.2399" "vertex.2400" \
   "vertex.2401" "vertex.2402" "vertex.2403" "vertex.2404" "vertex.2405" \
   interpolate
edge create nurbs "vertex.2406" "vertex.2407" "vertex.2408" "vertex.2409" \
   "vertex.2410" "vertex.2411" "vertex.2412" "vertex.2413" "vertex.2414" \
   "vertex.2415" "vertex.2416" "vertex.2417" "vertex.2418" "vertex.2419" \
   "vertex.2420" "vertex.2421" "vertex.2422" "vertex.2423" "vertex.2424" \
   "vertex.2425" "vertex.2426" "vertex.2427" "vertex.2428" "vertex.2429" \
   "vertex.2430" "vertex.2431" "vertex.2432" "vertex.2433" "vertex.2434" \
   interpolate
edge create nurbs "vertex.2435" "vertex.2436" "vertex.2437" "vertex.2438" \
   "vertex.2439" "vertex.2440" "vertex.2441" "vertex.2442" "vertex.2443" \
   "vertex.2444" "vertex.2445" "vertex.2446" "vertex.2447" "vertex.2448" \
   "vertex.2449" "vertex.2450" "vertex.2451" "vertex.2452" "vertex.2453" \
```

```
   "vertex.2454" "vertex.2455" "vertex.2456" "vertex.2457" "vertex.2458" \
   "vertex.2459" "vertex.2460" "vertex.2461" "vertex.2462" "vertex.2463" \
   interpolate
edge create nurbs "vertex.2464" "vertex.2465" "vertex.2466" "vertex.2467" \
   "vertex.2468" "vertex.2469" "vertex.2470" "vertex.2471" "vertex.2472" \
   "vertex.2473" "vertex.2474" "vertex.2475" "vertex.2476" "vertex.2477" \
   "vertex.2478" "vertex.2479" "vertex.2480" "vertex.2481" "vertex.2482" \
   "vertex.2483" "vertex.2484" "vertex.2485" "vertex.2486" "vertex.2487" \
   "vertex.2488" "vertex.2489" "vertex.2490" "vertex.2491" "vertex.2492" \
   interpolate
edge create nurbs "vertex.2493" "vertex.2494" "vertex.2495" "vertex.2496" \
   "vertex.2497" "vertex.2498" "vertex.2499" "vertex.2500" "vertex.2501" \
   "vertex.2502" "vertex.2503" "vertex.2504" "vertex.2505" "vertex.2506" \
   "vertex.2507" "vertex.2508" "vertex.2509" "vertex.2510" "vertex.2511" \
   "vertex.2512" "vertex.2513" "vertex.2514" "vertex.2515" "vertex.2516" \
   "vertex.2517" "vertex.2518" "vertex.2519" "vertex.2520" "vertex.2521" \
   interpolate
edge create nurbs "vertex.2522" "vertex.2523" "vertex.2524" "vertex.2525" \
   "vertex.2526" "vertex.2527" "vertex.2528" "vertex.2529" "vertex.2530" \
   "vertex.2531" "vertex.2532" "vertex.2533" "vertex.2534" "vertex.2535" \
   "vertex.2536" "vertex.2537" "vertex.2538" "vertex.2539" "vertex.2540" \
   "vertex.2541" "vertex.2542" "vertex.2543" "vertex.2544" "vertex.2545" \
   "vertex.2546" "vertex.2547" "vertex.2548" "vertex.2549" "vertex.2550" \
   interpolate
edge create nurbs "vertex.2551" "vertex.2552" "vertex.2553" "vertex.2554" \
   "vertex.2555" "vertex.2556" "vertex.2557" "vertex.2558" "vertex.2559" \
   "vertex.2560" "vertex.2561" "vertex.2562" "vertex.2563" "vertex.2564" \
   "vertex.2565" "vertex.2566" "vertex.2567" "vertex.2568" "vertex.2569" \
   "vertex.2570" "vertex.2571" "vertex.2572" "vertex.2573" "vertex.2574" \
   "vertex.2575" "vertex.2576" "vertex.2577" "vertex.2578" "vertex.2579" \
   interpolate
edge create nurbs "vertex.2580" "vertex.2581" "vertex.2582" "vertex.2583" \
   "vertex.2584" "vertex.2585" "vertex.2586" "vertex.2587" "vertex.2588" \
   "vertex.2589" "vertex.2590" "vertex.2591" "vertex.2592" "vertex.2593" \
   "vertex.2594" "vertex.2595" "vertex.2596" "vertex.2597" "vertex.2598" \
   "vertex.2599" "vertex.2600" "vertex.2601" "vertex.2602" "vertex.2603" \
   "vertex.2604" "vertex.2605" "vertex.2606" "vertex.2607" "vertex.2608" \
   interpolate
edge create nurbs "vertex.2609" "vertex.2610" "vertex.2611" "vertex.2612" \
```

```
    "vertex.2613" "vertex.2614" "vertex.2615" "vertex.2616" "vertex.2617" \
    "vertex.2618" "vertex.2619" "vertex.2620" "vertex.2621" "vertex.2622" \
    "vertex.2623" "vertex.2624" "vertex.2625" "vertex.2626" "vertex.2627" \
    "vertex.2628" "vertex.2629" "vertex.2630" "vertex.2631" "vertex.2632" \
    "vertex.2633" "vertex.2634" "vertex.2635" "vertex.2636" "vertex.2637" \
    interpolate
edge create nurbs "vertex.2638" "vertex.2639" "vertex.2640" "vertex.2641" \
    "vertex.2642" "vertex.2643" "vertex.2644" "vertex.2645" "vertex.2646" \
    "vertex.2647" "vertex.2648" "vertex.2649" "vertex.2650" "vertex.2651" \
    "vertex.2652" "vertex.2653" "vertex.2654" "vertex.2655" "vertex.2656" \
    "vertex.2657" "vertex.2658" "vertex.2659" "vertex.2660" "vertex.2661" \
    "vertex.2662" "vertex.2663" "vertex.2664" "vertex.2665" "vertex.2666" \
    interpolate
edge create nurbs "vertex.2667" "vertex.2668" "vertex.2669" "vertex.2670" \
    "vertex.2671" "vertex.2672" "vertex.2673" "vertex.2674" "vertex.2675" \
    "vertex.2676" "vertex.2677" "vertex.2678" "vertex.2679" "vertex.2680" \
    "vertex.2681" "vertex.2682" "vertex.2683" "vertex.2684" "vertex.2685" \
    "vertex.2686" "vertex.2687" "vertex.2688" "vertex.2689" "vertex.2690" \
    "vertex.2691" "vertex.2692" "vertex.2693" "vertex.2694" "vertex.2695" \
    interpolate
edge create nurbs "vertex.2696" "vertex.2697" "vertex.2698" "vertex.2699" \
    "vertex.2700" "vertex.2701" "vertex.2702" "vertex.2703" "vertex.2704" \
    "vertex.2705" "vertex.2706" "vertex.2707" "vertex.2708" "vertex.2709" \
    "vertex.2710" "vertex.2711" "vertex.2712" "vertex.2713" "vertex.2714" \
    "vertex.2715" "vertex.2716" "vertex.2717" "vertex.2718" "vertex.2719" \
    "vertex.2720" "vertex.2721" "vertex.2722" "vertex.2723" "vertex.2724" \
    interpolate
edge create nurbs "vertex.2725" "vertex.2726" "vertex.2727" "vertex.2728" \
    "vertex.2729" "vertex.2730" "vertex.2731" "vertex.2732" "vertex.2733" \
    "vertex.2734" "vertex.2735" "vertex.2736" "vertex.2737" "vertex.2738" \
    "vertex.2739" "vertex.2740" "vertex.2741" "vertex.2742" "vertex.2743" \
    "vertex.2744" "vertex.2745" "vertex.2746" "vertex.2747" "vertex.2748" \
    "vertex.2749" "vertex.2750" "vertex.2751" "vertex.2752" "vertex.2753" \
    interpolate
edge create nurbs "vertex.2754" "vertex.2755" "vertex.2756" "vertex.2757" \
    "vertex.2758" "vertex.2759" "vertex.2760" "vertex.2761" "vertex.2762" \
    "vertex.2763" "vertex.2764" "vertex.2765" "vertex.2766" "vertex.2767" \
    "vertex.2768" "vertex.2769" "vertex.2770" "vertex.2771" "vertex.2772" \
    "vertex.2773" "vertex.2774" "vertex.2775" "vertex.2776" "vertex.2777" \
```

```
   "vertex.2778" "vertex.2779" "vertex.2780" "vertex.2781" "vertex.2782" \
   interpolate
face create skin "edge.1" "edge.2" "edge.3" "edge.4" "edge.5" "edge.6" \
   "edge.7" directions 0 0 0 0 0 0 0
face create skin "edge.7" "edge.8" "edge.9" "edge.10" "edge.11" "edge.12" \
   "edge.13" directions 0 0 0 0 0 0 0
face create skin "edge.13" "edge.14" "edge.15" "edge.16" "edge.17" "edge.18" \
   "edge.19" directions 0 0 0 0 0 0 0
face create skin "edge.19" "edge.20" "edge.21" "edge.22" "edge.23" "edge.24" \
   "edge.25" directions 0 0 0 0 0 0 0
face create skin "edge.25" "edge.26" "edge.27" "edge.28" "edge.29" "edge.30" \
   "edge.31" directions 0 0 0 0 0 0 0
face create skin "edge.31" "edge.32" "edge.33" "edge.34" "edge.35" "edge.36" \
   "edge.37" directions 0 0 0 0 0 0 0
face create skin "edge.37" "edge.38" "edge.39" "edge.40" "edge.41" "edge.42" \
   "edge.43" directions 0 0 0 0 0 0 0
face create skin "edge.43" "edge.44" "edge.45" "edge.46" "edge.47" "edge.48" \
   "edge.49" directions 0 0 0 0 0 0 0
face create skin "edge.50" "edge.51" "edge.52" "edge.53" "edge.54" "edge.55" \
   "edge.56" directions 0 0 0 0 0 0 0
face create skin "edge.56" "edge.57" "edge.58" "edge.59" "edge.60" "edge.61" \
   "edge.62" directions 0 0 0 0 0 0 0
face create skin "edge.62" "edge.63" "edge.64" "edge.65" "edge.66" "edge.67" \
   "edge.68" directions 0 0 0 0 0 0 0
face create skin "edge.68" "edge.69" "edge.70" "edge.71" "edge.72" "edge.73" \
   "edge.74" directions 0 0 0 0 0 0 0
face create skin "edge.74" "edge.75" "edge.76" "edge.77" "edge.78" "edge.79" \
   "edge.80" directions 0 0 0 0 0 0 0
face create skin "edge.80" "edge.81" "edge.82" "edge.83" "edge.84" "edge.85" \
   "edge.86" directions 0 0 0 0 0 0 0
face create skin "edge.86" "edge.87" "edge.88" "edge.89" "edge.90" "edge.91" \
   "edge.92" directions 0 0 0 0 0 0 0
face create skin "edge.92" "edge.93" "edge.94" "edge.95" "edge.96" "edge.97" \
   "edge.98" directions 0 0 0 0 0 0 0
edge create straight "vertex.1" "vertex.1422"
edge create straight "vertex.29" "vertex.1445"
edge create straight "vertex.1393" "vertex.2754"
edge create straight "vertex.1421" "vertex.2782"
face create wireframe "edge.1" "edge.50" "edge.147" "edge.148" real
```

```
face create wireframe "edge.49" "edge.98" "edge.149" "edge.150" real
face create wireframe "edge.147" "edge.100" "edge.103" "edge.106" "edge.109" \
   "edge.112" "edge.116" "edge.119" "edge.122" "edge.149" "edge.146" \
   "edge.143" "edge.140" "edge.136" "edge.133" "edge.130" "edge.127" \
   "edge.124" real
face create wireframe "edge.148" "edge.101" "edge.104" "edge.107" "edge.110" \
   "edge.113" "edge.115" "edge.118" "edge.121" "edge.150" "edge.145" \
   "edge.142" "edge.139" "edge.137" "edge.134" "edge.131" "edge.128" \
   "edge.125" real
volume create stitch "face.1" "face.2" "face.3" "face.4" "face.5" "face.6" \
   "face.7" "face.8" "face.9" "face.10" "face.11" "face.12" "face.13" \
   "face.14" "face.15" "face.16" "face.17" "face.18" "face.19" "face.20" real
volume creflect "volume.1" multiple 1 vector 0 1 0 origin 0 0 0
volume creflect "volume.1" "volume.2" multiple 1 vector 1 0 0 origin 0 0 0
volume unite volumes "volume.1" "volume.2" "volume.3" "volume.4"
volume create height 7 radius1 0.165 radius3 0.165 offset 0 0 -1 zaxis frustum
volume subtract "volume.2" volumes "volume.1"
volume create height 0.07 radius1 0.136 radius3 0.165 offset 0 0 0.035 zaxis frustum
volume create height 0.07 radius1 0.2 radius3 0.2 offset 0 0 0.035 zaxis frustum
volume subtract "volume.4" volumes "volume.3"
volume move "volume.4" offset 0 0 0.56
volume subtract "volume.2" volumes "volume.4"
face create width 10 yzplane rectangle
volume split "volume.2" faces "face.92" connected
volume delete "volume.2" lowertopology
face create width 1 xyplane rectangle
face cmove "face.99" multiple 1 offset 0 0 0.7
face cmove "face.99" multiple 1 offset 0 0 -0.3
volume split "volume.3" faces "face.100" "face.99" "face.101" connected
solver select "FLUENT 5/6"
volume mesh "volume.4" cooper source "face.83" "face.100" size 0.02
volume mesh "volume.7" tetrahedral size 0.003
volume mesh "volume.5" tetrahedral size 0.01
volume mesh "volume.6" cooper source "face.17" "face.99" "face.101" \
   "face.107" size 0.01
volume mesh "volume.3" cooper source "face.93" "face.101" size 0.02
physics create "water" ctype "FLUID" volume "volume.4" "volume.3" "volume.5" \
   "volume.6" "volume.7"
physics create "inlet" btype "MASS_FLOW_INLET" face "face.83"
```

```
physics create "outlet" btype "OUTFLOW" face "face.93"
physics create "sym" btype "SYMMETRY" face "face.92" "face.110" "face.102" \
   "face.104" "face.109" "face.108"
physics create "pipe1" btype "WALL" face "face.82" "face.103" "face.85" \
   "face.18"
physics create "pipe2" btype "WALL" face "face.98" "face.105" "face.106" \
   "face.94"
physics create "DBVext" btype "WALL" face "face.7" "face.8" "face.6" "face.5" \
   "face.4" "face.3" "face.2" "face.1" "face.25" "face.26" "face.24" "face.23" \
   "face.22" "face.21" "face.38" "face.40" "face.17"
physics create "DBVint" btype "WALL" face "face.15" "face.16" "face.14" \
   "face.13" "face.12" "face.10" "face.11" "face.9" "face.34" "face.35" \
   "face.33" "face.31" "face.32" "face.29" "face.30" "face.28"
save name "DBV3DFLOW-36kPa.dbs"
```

APPENDIX E

A Labview Script for Water Tunnel Data Acquisition System

The output signals from the transducer were amplified by a Validyne CD101 conditioner and transferred to a NI PCI-6024E A/D card, which was plugged in an HP desktop computer. A script was developed using the commercial code, Labview, to analyze the digital signals from the transducer. Fig. E-1 shows a block diagram of the data acquisition system, where the pressure signals are processed and the root mean square (RMS), maximum and minimum magnitudes of 500 signal samples per time are monitored by a front panel and recorded by an output file.

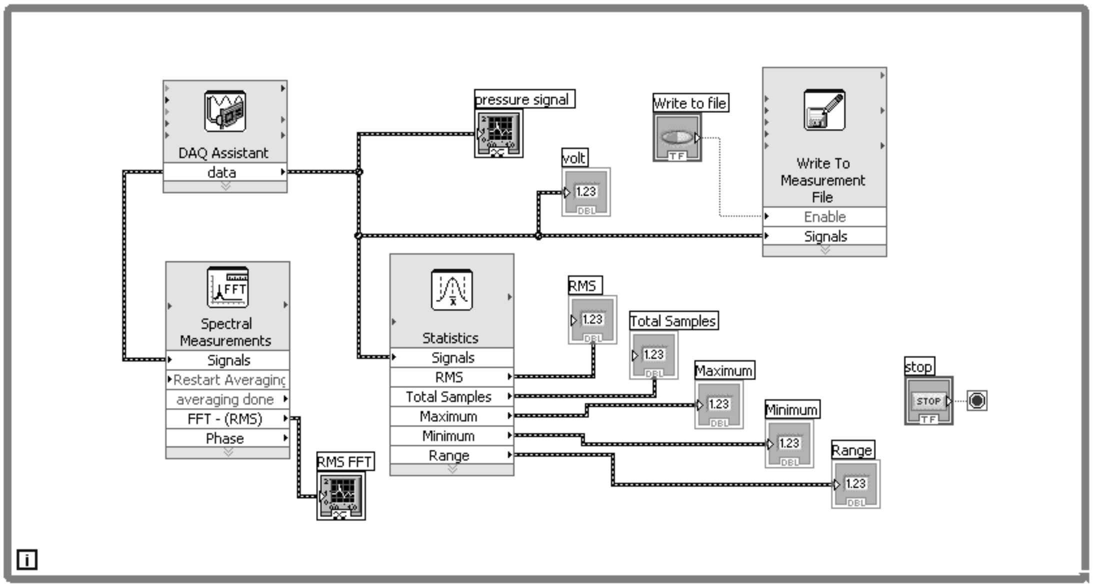
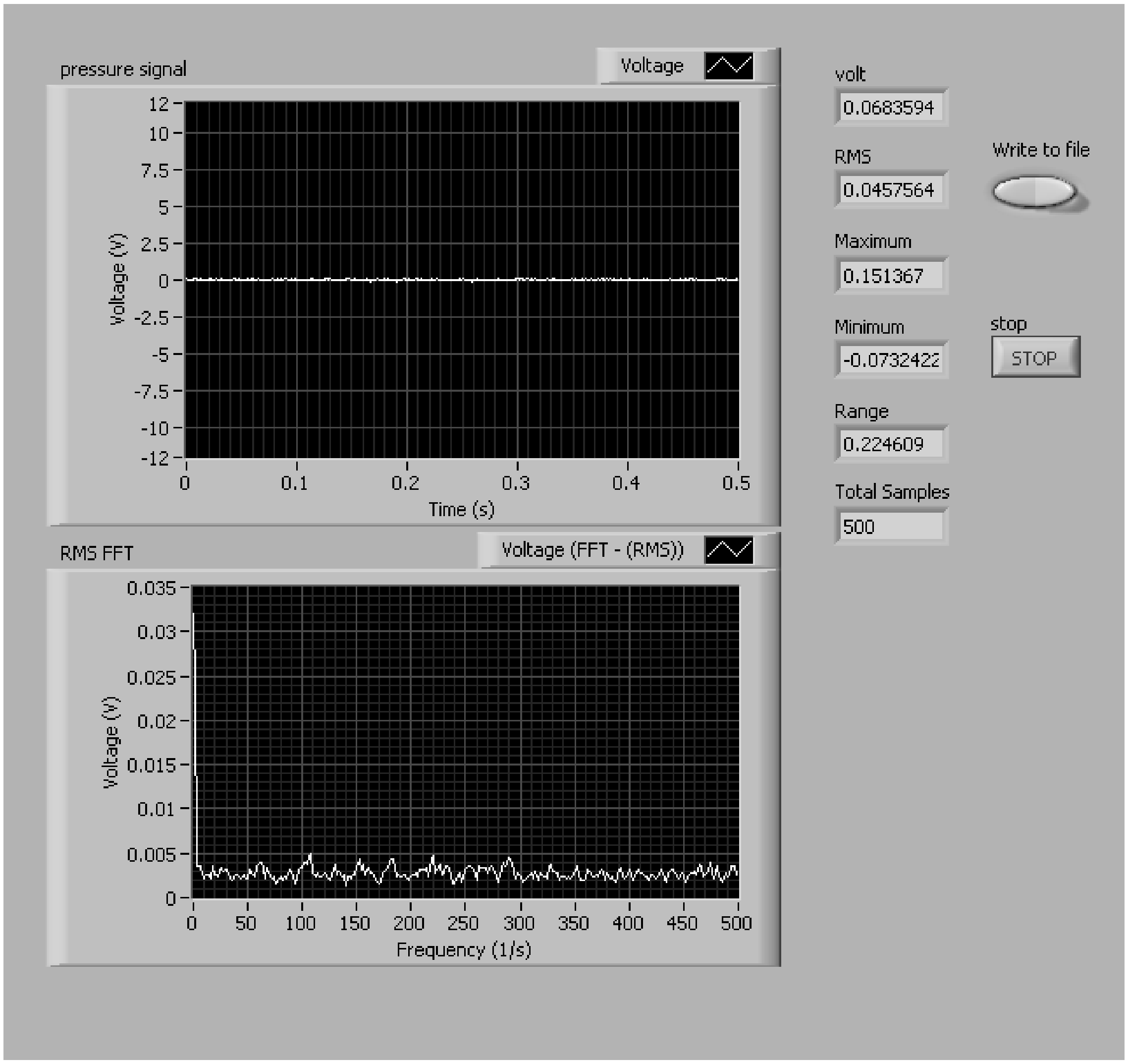


Fig. E-1 Block diagram of pressure signal acquisition system

Figure E-2 shows the front panel of the data acquisition system. When the data acquisition system starts to run, test operator can watch the transient date curves from the voltage-time and FFT monitors on the left of the panel, read digital numbers on the right, and save 500 data samples to a file per clicking the button of "write to file". When a test is finished, the data measurement can be terminated by clicking the red stop button on the panel.



volt: 0.0683594

RMS: 0.0457564

Maximum: 0.151367

Minimum: -0.0732422

Range: 0.224609

Total Samples: 500

Fig. E-2 Front panel of pressure signal acquisition system