NUMERICAL SIMULATION OF MULTI-DIMESIONAL FLOWS IN A GEAR PUMP

A Thesis

by

SURESH PATIL

Submitted in partial fulfillment of the requirement

for the Degree of

MASTER OF SCIENCE

IN

ENGINEERING PROGRAM

YOUNGSTOWN STATE UNIVERSITY
DECEMBER, 2006

NUMERICAL SIMULATION OF MULTI-DIMESIONAL FLOWS IN A GEAR PUMP

Suresh Patil

I hereby release this thesis to the public. I understand that this thesis will be made available from the Ohio Link ETD center and the Maag Library circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

12/07/2006

Suresh Patil — Date

Approvals:

12-7-06

Dr. Hyun W. Kim, Thesis Advisor — Date

12-7-06

Dr. Daniel H. Suchora, Committee Member — Date

12-07-2006

Dr. Hazel Marie, Committee Member — Date

12/12/06

Dr. Peter J. Kasvinsky, Dean of Graduate School — Date

ABSTRACT

FLUENT, commercial CFD (Computational Fluid Dynamics) software is used for the flow analysis of an external gear pump. This study is an extension of the previous work of FLUENT analyses of gear pumps, which has been conducted at Youngstown State University for the last two years.

The main objective of this study is to improve the computational fluid dynamics analysis of the gear pump by developing improved 2-D models and an initial 3-D model. A comprehensive analysis is performed for the 2-D flow with models having smaller gaps between gears. A case study of 3-D flow analysis is conducted on a 2.5-D model that is created from the 2-D model. The 3-D analysis is performed on supercomputer in a mode of remote parallel processing.

It is important to predict the fluid flow parameters such as pressure, velocity, turbulence in order to improve pump performance. These parameters are analyzed under various operating conditions such as different speeds and outlet pressures to investigate the effect of each parameter on the performance of the pump. The numerical outputs from the 2-D and 3-D analyses are compared with theoretical values. The results are consistent and reasonable.

The information contained in this thesis may help those who would like to conduct 3-D flow simulation of gear pump in the future.

Dedicated to my parents, wife and son

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor, Professor Hyun W. Kim for his advice, encouragement, and patience through out this work. My thanks go to Professor Daniel H. Suchora and Professor Hazel Marie for serving as the members of my thesis committee.

I would like to express my appreciation to John Murray, Shariq Khan and Nilesh Gandhi of Fluent Corporation for their valued suggestions in setting-up this simulation related to work. Also I am thankful to Ohio Supercomputer Center (OSC) for providing the supercomputer to use for this work and Jim Giuliani and Barbara Woodall for helping to overcome many hurdles in this whole process.

I would like to thank Dr. Alina Lazar and Dr. Michael Crescimanno for their valued suggestions in setting-up remote processing. Also I wish to thank Dennis Gajdos, Ben King, and Maureen Brajer for their time to help.

I am grateful to my wife Vrushali and son Ruturaj for their support, love and patience throughout two years. I would like to thank Mr. and Mrs. Killedar for their encouragement.

VITA

1972.....	Born in India.
1993.....	Bachelor of Engineering, Mechanical Engineering Shivaji University, Kolhapur, India.
1993-2005.....	Associate Project Manager Project Engineering Division Kirloskar Brothers Limited, Pune, India.
1997.....	Diploma in Business Administration Pune University, Pune, India.
1999.....	Master of Management Science, Marketing Management Pune University, Pune, India.
2005 - till present.....	Graduate Research Assistant, Department of Mechanical Engineering, Youngstown State University, Youngstown, USA.

FIELDS OF STUDY

Major Field: Mechanical Engineering

TABLE OF CONTENT

LIST OF TABLES

Table:

LIST OF FIGURES

Figure:

NOMENCLATURE

Variables and Constants:

$\vec{A}$	surface area vector
$\vec{A}_f$	area of face f
c	specific heat
$C_{1\varepsilon}$, $C_{2\varepsilon}$, $C_{3\varepsilon}$, C_μ	constants
G_k	turbulence kinetic energy due to the mean velocity gradients
G_b	turbulence kinetic energy due to buoyancy
g_x, g_y, g_z	gravitational acceleration in Cartesian coordinate system
$\hat{i}, \hat{j}, \hat{k}$	unit vectors in Cartesian coordinate system
k	kinetic energy
N_{faces}	number of faces enclosing cell
p	pressure
S_Φ, S_k, S_ε	source terms
t	time
T	temperature
u, v, w	velocity components in Cartesian coordinate system
V	cell volume
$\vec{V}$	velocity vector
x, y, z	Cartesian coordinates
Y_M	contribution of fluctuating dilatation
Γ_ϕ	diffusion coefficient of Φ
ε	kinetic energy heat dissipation rate
μ	viscosity
μ_t	turbulent viscosity
ρ	density
σ_k	turbulent Prandtl number for k

Chapter 1

INTRODUCTION

1.1. Introduction

The gear pump is one of the most widely used positive displacement pumps in hydraulic industry due to its unique advantages of simplicity and versatility in design and manufacture. These pumps are being used for generating low fluid flow but very high pressure, operating conditions which are frequently encountered in fluid power applications. Many hydraulic systems call for pumps generating high pressure. They are also light weight, small in size, highly efficient, with simplicity of servicing and reliability in operation under practical conditions. These demands are best met by gear pumps that have significant advantages over other types of pumps (1). Parker Hannifin, Viking Pump, Bosch are some of the leading gear pump manufacturers in USA. The Figure 1.1 shows one of the typical gear pump manufactured by Parker Hannifin Corporation.

Figure 1.1 - Gear Pump (Courtesy: Parker Hannifin Corporation)

Recently, Youngstown State University's Center for Hydraulics Research and Education has been developing a collaborative research and educational program in hydraulics to provide a cutting edge research on the gear pump. One of the objectives is to develop a computational laboratory capability for performing flow analysis to optimize the design and improve the efficiency, as well as reliability, of the gear pumps. The flow analysis is mainly performed with commercially available computational fluid dynamics (CFD) software packages. This study is part of the recent research efforts in which the CFD analysis of the gear pump has been performed in FLUENT (2).

1.2. History of Development

The lifting of water with the help of some mechanism is a very old concept of engineering. In ancient times water wheels were used with buckets for lifting water with minimum human efforts. In 200 BC, Ctesibius, a Greek inventor invented the reciprocating pump for pumping water. During the same era, the Greek mathematician, Archimedes invented a screw pump. No major change in pump design was done until the late fifteen century. In the year 1588, an Italian engineer Ramelli invented a water pump that was also used for pumping oil. Later, in 1636, a German engineer Pappenheim invented a gear pump for lubricating engines. In 1859, Jones modified Pappenheim's gear pump and created a double rotor gear pump with two teeth per gear (3).

Since the introduction of modern gear pumps in 1930, they have been widely used in polymer manufacturing plants and transportation industries. With continuous use of gear pumps for many years, the considerable technical advancement has been achieved. Recently Borghi performed major industrial research on developing and tailoring of CAD/CAE instruments for designing external gear pumps (4).

Noah D. Manring and Suresh B. Kasaragadda have performed research on the theoretical flow ripple of an external gear pump of similar size, using different numbers of teeth on driving and driven gears. According to them, the results show that the driving gear dictates the flow ripple characteristics of the pump while the driven gear dictates the pump size. Hence it is advantageous to design an external gear pump with a large number of teeth on the driving gear and fewer numbers of teeth on the driven gear. This design configuration will tend to reduce both the physical pump size and the amplitude of the flow pulsation, while increasing the natural harmonic frequencies of machine (5).

Jaakko Myllykylä has done research on the suction capability of an external gear pump. He collected data on a number of gear pumps to study the suction characteristics and formulated a theory for suction characteristics. He also showed that higher rotational speed can lead to cavitation. According to his research, the biggest uncertianity factor is the mathematical expression for the effect of housing shape characteristics (6).

FLUENT Inc. has done a simulation for the molten polymer flow analysis in the gear pump. The research result obtained showed that a high pressure region is formed where the gear teeth are closed in the outlet port (7). Gamez-Montero and Condia Macia studied the suction performance of the gerotor pumps and its influence on the fluid-born noise generation (8). Santosh Kini published a technical paper on numerical simulation of cover plate deflection in the gerotor pump. The study was performed by using computational fluid dynamics software CFD-ACE+ and helped to locate the place and amount of deflection in the side plate of gerotor pump. It also showed that computational results at a particular location within the fluid domain agree well with experimental results, which can be used to predict the cavitation and consequently minimize it (9).

1.3. Present Work in CFD Analysis at YSU

Yogendra Panta from the Department of Mechanical Engineering performed CFD analysis of gear pump using commercial software FLUENT. In his study, the simulation of 2-D flow in a gear pump was successfully developed for the very first time using FLUENT. The technique of moving dynamic mesh (MDM) was incorporated to generate flow against a high adverse pressure gradient. The research results provided reasonable outcomes and substantial information on modeling and executing FLUENT for solving the flow problem in gear pumps (10).

Jyotindra Killedar extended Panta's work of simulation gear pump flows in FLUENT. He conducted the analysis with improved 2-D models with smaller gap size between the gears and between gears and casing. The results of his study showed that the mass flow rate was fairly close to the theoretical value. Although no experimental verification has been presented, pressure contours and velocity vectors in fluid domain were regarded as reasonably accurate. He also performed cavitational analysis for the same models. The cavitation study shows that the pump does not cavitate below 1000 psi outlet pressure. However, according to his findings, the cavitation analysis can not be carried out for outlet pressures more than 1000 psi in FLUENT due to the software's limitation of numerical stability (11).

The present study is the continuation of research work on flow simulation mentioned above with the prime objective to obtain improved results by simulating flow in improved gear pump models. The study mainly focuses on simulation of 2-D and 2.5-D models by performing comprehensive flow analysis with models having smaller gaps. The simulation of 3-D flow analysis by using 2.5-D models was performed on supercomputer at Ohio Supercomputer Center (OSC).

Chapter 2

PHYSICAL MODEL AND PROBLEM DESCRIPTION

2.1 Gear Pump - PGP 640

Parker Hannifin Corporation is one of the leading manufacturers of hydraulic equipment including gear pumps. One of their models, PGP 640, has been selected for this study. PGP 640 is an external gear pump, having typical construction of two gears meshed with each other which are enclosed in an outer casing. Figure 2.1 and 2.2 shows the SolidWorks 3-D model of PGP 640. This gear pump consists of four basic components.

- Pump casing: Casing is the main component which accommodates both driving and driven gear and has inlet and outlet ports for suction and delivery.
- Driving gear: The driving gear is mounted on the driving shaft which is connected to prime mover.
- Driven gear: The driven gear is mounted on another shaft. The motion of this gear occurs due to meshing of this gear with driving gear.
- Cover Plate: The cover plate seals one side of the casing.

2.1.1 Specifications

Following are the specifications of PGP 640 pump.

Type : External gear pump

Pump Displacement : 50 cc/rev

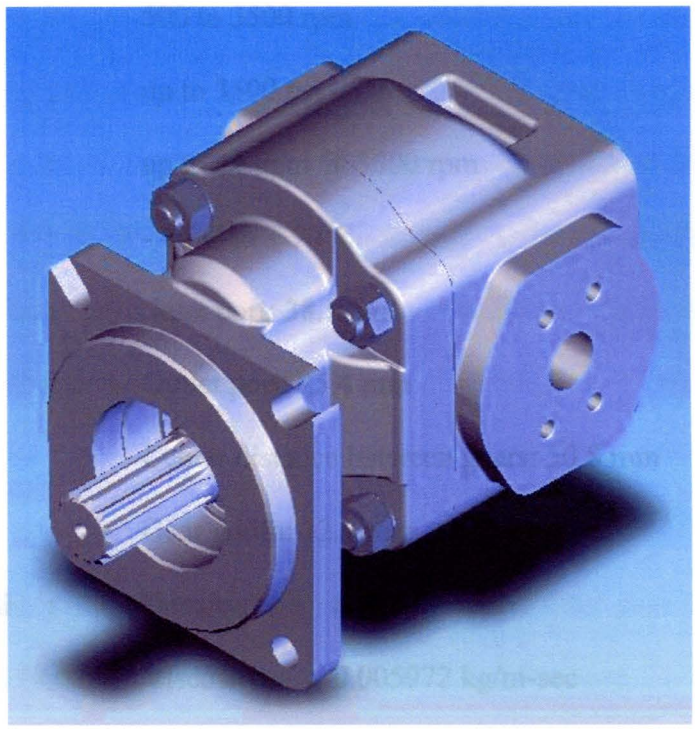

Figure 2.1 - PGP 640 - SolidWorks Model

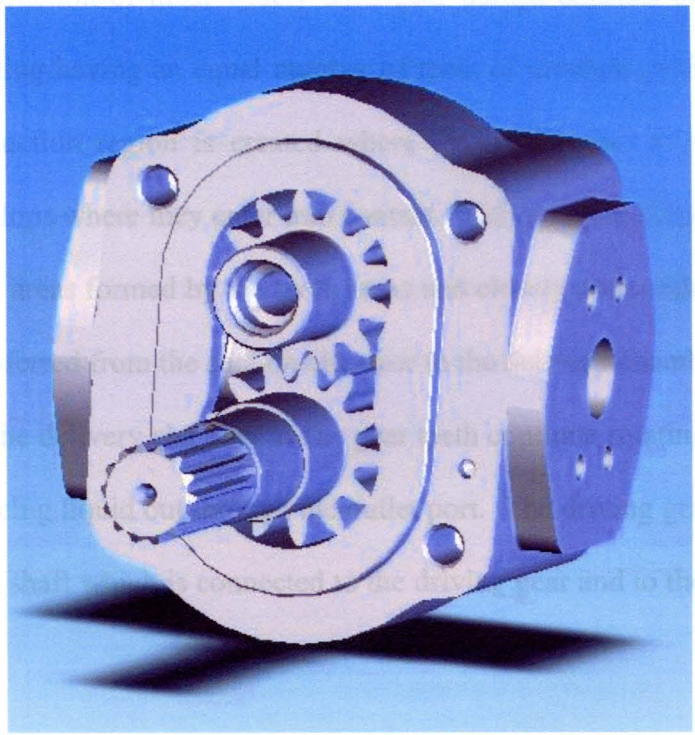

Figure 2.2 - PGP 640 - Internal Arrangement

Speed	:	500 to 3500 rpm
Pressures	:	up to 3500 psi
Flow rate	:	up to 41 gpm @ 3000 rpm
Ambient Temp	:	- 40°C to + 70°C
Dimensions	:	Inlet port: 38.1 mm
		Outlet port: 25.4 mm
		Center distance between gears: 50.8 mm
		Gear width: 32.9 mm
Properties of the oil	:	Density: 878.3 kg/sec.
		Viscosity: 0.005972 kg/m-sec

2.2 Working Principle

The standard construction of most widely used gear pumps consists of a pair of gears with external meshing having an equal number of teeth of involute profile. As the drive gear revolves, a suction region is created where the teeth cease to intermesh and a pressure zone develops where they enter into contact. With rotation of the gear rotors, the fluid trapped in the areas formed by the tooth areas and closely surrounded by the bore of the housing is transferred from the suction chamber to the delivery chamber. The pressure zone is created in the delivery chamber as the gear teeth continue rotating and come back in contact, thus forcing liquid out through the outlet port. The driving gear of the pump is driven by the drive shaft which is connected to the driving gear and to the drive by means of a spline joint.

Though the geometry of gear pump is simple, the fluid domain is very complex, which makes the simulation of fluid flow in gear pumps a difficult task. A major problem with gear pump modeling is the creation of very fine mesh in smaller gaps between the gears. These are the area, which are vulnerable to fail during a transient simulation. Hence, the gear pump simulation started with models having wider gaps, which was completed in previous studies by Panta and Killedar.

2.3 Problem Description

This research focuses on a refined fluid flow analysis with finer gaps in 2-D to achieve results very close to theoretical values. Simulation in 3-D by using 2.5-D models, which has been conducted on the OSC supercomputer, is also presented.

2.3.1 CFD Analysis of 2-D Flow

The CFD Analysis is performed by simulating 2-D flow in a gear pump under variety of operating conditions. The pressure contours, velocity vectors, pressure variation at particular points in the fluid domain, mass flow rate among other fluid properties have been monitored with respect to time. The effect of the parameters such as speed, outlet pressure, gaps between the gears on the performance of the gear pump has been studied.

2.3.2 CFD Analysis of 3-D Flow

After the successful analysis of simulation of fluid flow with the 2-D models having finer gaps, the analysis is performed in 3-D by using FLUENT. The same 2-D model is extruded to construct a 2.5-D model. The full 3-D transient flow analysis is conducted on the above mentioned 2.5-D model. The parameters such as pressure contours, velocity vectors, pressure variation at a particular point and mass flow rate are also investigated in the 3-D flow analysis. The results of the 3-D fluid flow analysis are then compared with 2-D flow results.

Chapter 3

MATHEMATICAL MODELING OF GEAR PUMP

3.1 Mathematical modeling

The fluid motion in the fluid domain of gear pump is governed by the continuity, momentum, and energy equations. Due to the high speed of the gears, the flow in the fluid region is expected to be highly turbulent; hence additional equation for the turbulent kinetic and the rate of dissipation of kinetic energy are included.

Following are the assumptions which are incurred on the present analysis:

1. The fluid is Newtonian and is incompressible.
2. Body forces acting on the fluid are negligible.
3. The fluid was initially stationary.
4. There is no viscous dissipation.
5. There is no surface heat flux on casing.

3.2 Governing Equations

The Cartesian co-ordinate system is selected for mathematical modeling, having its origin at the center of the driving gear. By considering above said assumptions, the governing equations with boundary conditions can be expressed as follows.

3.2.1 Continuity Equation

The continuity equation for compressible fluid is

$$\frac{\partial \rho}{\partial t} + \nabla . \rho \vec{V} = 0 \qquad \ldots\ldots (3.1)$$

Where,

∇ is vector operator and

$$= \hat{i}\frac{\partial}{\partial x} + \hat{j}\frac{\partial}{\partial y} + \hat{k}\frac{\partial}{\partial z}$$

3.2.2 Momentum Equation

These equations of motion derived for a Newtonian fluid are the Navier-Stokes equation.

For incompressible fluid, these equations may be simplified as

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \qquad \ldots\ldots (3.2a)$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) = -\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \qquad \ldots\ldots (3.2b)$$

$$\rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = -\frac{\partial p}{\partial z} + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \qquad \ldots\ldots (3.2c)$$

Initial Condition,

At time $t \leq 0$: $\vec{V} = 0$

Boundary Conditions of velocity,

On the casing wall: $\vec{V} = 0$

On the gear surfaces: $\vec{V} = \vec{V}_s$

Boundary Conditions of pressure:

At inlet port: $P = P_i$

At outlet port: $P = P_o$

3.2.3 Energy Equations

The energy equation for compressible fluid is

$$\frac{\partial(c\rho T)}{\partial t} + \nabla.\left(\rho\vec{V}cT\right) = \nabla\left(k\nabla T\right) + S_\phi \qquad \text{.....(3.3)}$$

Boundary Conditions of temperatures:

At inlet port: $T = T_i$

At outlet port: $T = T_o$

On the gear surfaces: $\frac{\partial T}{\partial n} = 0$

3.2.4 Model Transport Equation

The standard k-ε model is a semi-empirical model based on model transport equations for the turbulence kinetic energy (k) and its dissipation rate (ε). The transport equations are based on assumptions that the flow is fully turbulent and the effect of the molecular viscosity is negligible.

Transport Equations for the standard k-ε model

$$\frac{\partial}{\partial t}(\rho\kappa) + \nabla\left(\rho\kappa\vec{V}\right) = \nabla\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right] + G_k + G_b - \rho\varepsilon - Y_M + S_k \qquad \text{.....(3.4)}$$

$$\frac{\partial}{\partial t}(\rho\varepsilon)+\nabla(\rho\varepsilon\vec{V})=\nabla\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\nabla\varepsilon\right]+C_{1\varepsilon}\frac{\varepsilon}{k}(G_k+C_{3\varepsilon}G_b)-C_{2\varepsilon}\rho\frac{\varepsilon^2}{k}+S_\varepsilon \quad .....(3.5)$$

Modeling the Turbulent Viscosity

$$\mu_t=\rho C_\mu\frac{k^2}{\varepsilon}$$

The model constants $C_{1\varepsilon}$, $C_{2\varepsilon}$, C_μ, σ_k and σ_ε have following values

$C_{1\varepsilon} = 1.44$, $C_{2\varepsilon} = 1.92$, $C_\mu = 0.09$, $\sigma_k = 0.3$ and $\sigma_\varepsilon = 1.3$

These values have been determined from experiments with air and water for fundamental turbulent shear flows including homogeneous shear flows and decaying isotropic grid turbulence.

Chapter 4

NUMERICAL MODELING OF GEAR PUMP

4.1 Computational Fluid Dynamics

The governing equations such as continuity, momentum, and energy can be used to express fluid motion in a gear pump. However, there are no known closed-form analytical solution methods available for many complex flows such as gear pump flow. Therefore, the analysis of such flows has often been resorted to numerical methods. The development of high speed computers and modern computing technology enabled complex numerical analysis and helped the computational fluid dynamics establish its place in modern fluid dynamics. Many commercial CFD software packages are available in the market including FLUENT, CFD-ACE and Star-CD. FLUENT has a unique feature of a Moving Dynamic Mesh (MDM) algorithm which helps in solving transient flow problems with changing fluid domain. Hence the FLUENT software package was selected for the gear pump flow simulation.

4.2 Methodology

Almost all commercially available CFD software utilizes the same solution procedure. The general steps are as follows:

1. The geometry of the fluid domain is defined.
2. The fluid domain is divided into discrete cells.
3. Boundary conditions, such as inlet pressure, outlet pressure, moving/stationery walls are defined.

4. Depending upon the characteristic of the problem, the governing equations are solved by an appropriate solver.

The process of solving the problem can be categorized into three major stages the preprocessor, solver, and postprocessor.

- **Preprocessor** – Geometry creation and meshing.
- **Solver** – Physical model defining, boundary conditions, equation solving.
- **Postprocessor** – Analysis and visualization of results.

For this study, SolidWorks and Gambit are used as the preprocessor whereas FLUENT is used as solver and postprocessor.

4.3 Preprocessor - Gambit

Gambit is a commercially available preprocessing software package designed to import, build, and mesh models for FLUENT and other scientific applications. Gambit performs fundamental steps of building, meshing, and creating zone types in a model. Gambit's combination with CAD interoperability, geometry cleanup, decomposition, and meshing tools makes it one of the easiest, fastest, and most straightforward preprocessors. As a state-of-the-art preprocessor for engineering analysis, Gambit has several geometry and meshing tools in a powerful, flexible, tightly integrated, and easy-to use interface. Most models can be built directly in Gambit or imported from major CAD systems. A comprehensive set of highly automated and size-function driven meshing tools ensures that the best mesh can be generated, whether structured, multi-block, unstructed or hybrid.

4.3.1 Geometry and Mesh Generation in Gambit

The construction of a 2.5-D gear pump model begins with building of a 3-D gear pump model in SolidWorks reasonably close to the actual gear pump (Figure 4.1). The 3-D SolidWorks model is then reduced to a simplified 2-D model consisting of meshed gears, casing and part of inlet and outlet as shown in Figure 4.2. The simplification of the geometry of the inlet and outlet regions reduces number of cells, data storage, and processing time. A justification is that the scope of study is limited to the fluid domain near gears rather than inlet and outlet region, which may have less impact on pump performance. The simplified 2-D model is then meshed with triangular cells keeping mesh skewness below 0.6. This extreme low skewness is required to prevent premature collapse of 3-D meshes in transient flows. The 2-D gear pump model is then extruded by using the cooper meshing technique. The 2.5-D model of gear pump is used for the simulation 3-D transient flows in a gear pump.

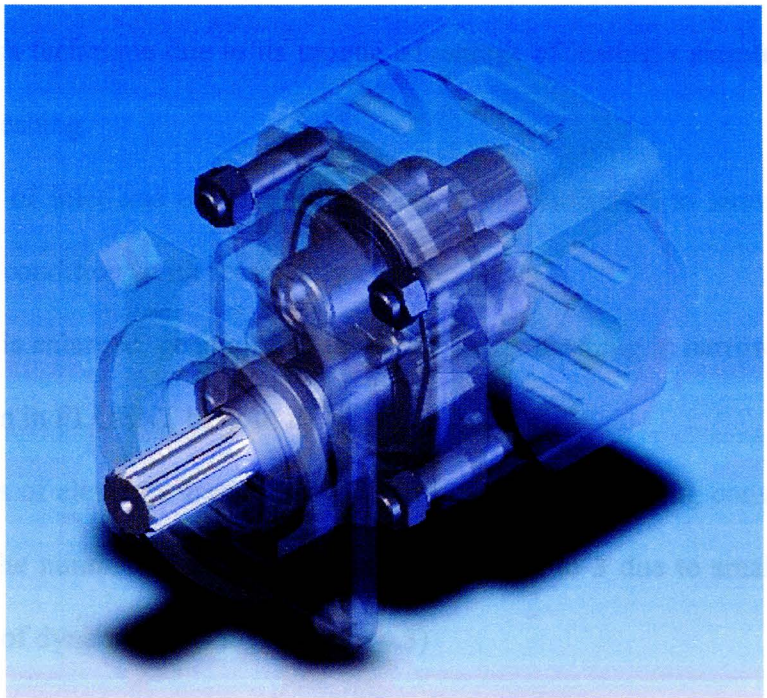

Figure 4.1 – 3-D Model of Gear Pump in SolidWorks

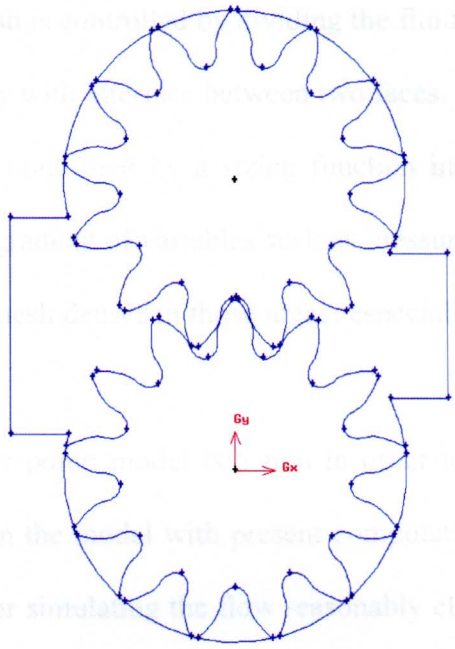

Figure 4.2 – Simplified Wireframe Model in 2-D

Due to complexity of the geometry, following points are considered while building the models for successful simulation.

1. The geometry of the fluid domain is made simple such that it is suitable for the cooper mesh technique due to its unique advantage of having a simple algorithm for dynamic meshing.
2. The length of inlet and outlet fluid passage is just sufficient to simulate the actual inlet/outlet conditions with reasonably accuracy.
3. The model is enlarged, generally 10 times, for better meshing in narrow gaps and then scaled down in FLUENT back to its original size.
4. More layers of elements are accommodated in fine gaps ensuring convergence of the solution. The number of layer of elements is limited to 3 due to small gap size and limitations of dynamic meshing. (Figure 4.3)

5. The skewness of the mesh is controlled by dividing the fluid domain in different faces and meshing it separately with interface between two faces.
6. The density of mesh is controlled by a sizing function in particular regions of the fluid domain where the gradient of variables such as pressure and velocity is high; the purpose is to make the mesh denser in those areas; especially within narrow gaps and near walls. (Figure 4.5)
7. The width of 2.5-D gear pump model is 5 mm in order to reduce number of cells, making it possible to run the model with present computational capability. Also it is kept sufficiently large for simulating the flow reasonably close to the actual flow in a gear pump. The width of gear pump model has been decided by trial and error.
8. The widthwise number of layers of cells in the meshed model was limited to five in order to limit the number of cells considering the data processing limitation as well as the accuracy of the results. (Figure 4.4)
9. The skewness of the 2-D as well as 2.5-D mesh is controlled below 0.6.

The 2-D models with three different gaps 15 μm, 22 μm, and 30 μm, are considered for study. Figure 4.5 shows the meshed 2-D model of the gear pump with the triangular mesh. The details of different meshed 2-D models are as follows:

Table 4.1 – 2-D Model Mesh Details

Gap between the gears	Number of cells	Skewness
15 μm	451,409	< 0.548
22 μm	464,961	< 0.543
30 μm	457,855	< 0.543

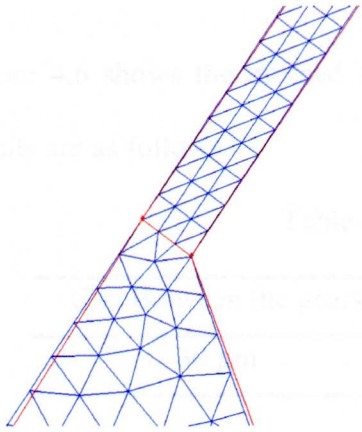

Figure 4.3 – Mesh in Gap between Gear and Casing in 2-D

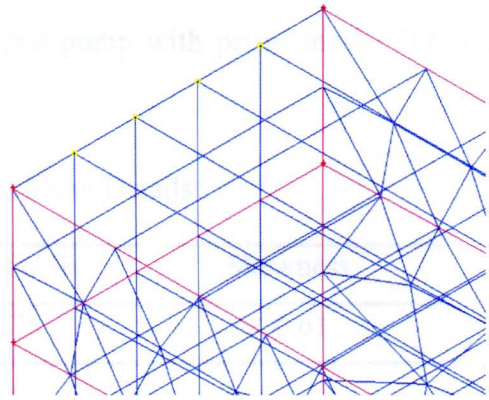

Figure 4.4 – Enlarged View of Inlet of 2.5-D Meshed Model showing 5 layers of prism elements.

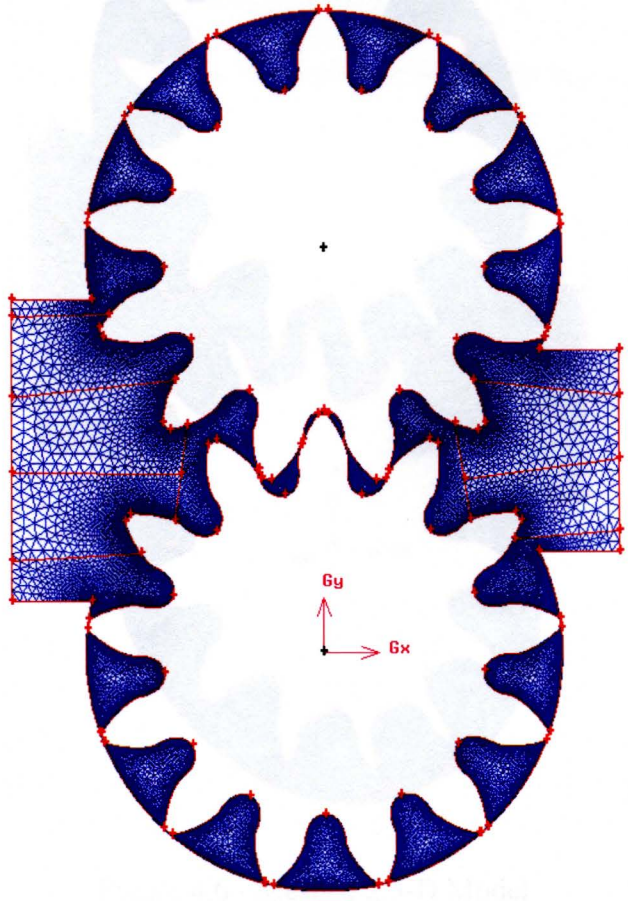

Figure 4.5 – Meshed 2-D Model

Figure 4.6 shows the meshed 2.5-D model of gear pump with prism mesh. The mesh details are as follows.

Table 4.2 – 2.5-D Model Mesh Details

Gap Between the gears	Number of cells	Skewness
30 μm	2,289,275	< 0.61

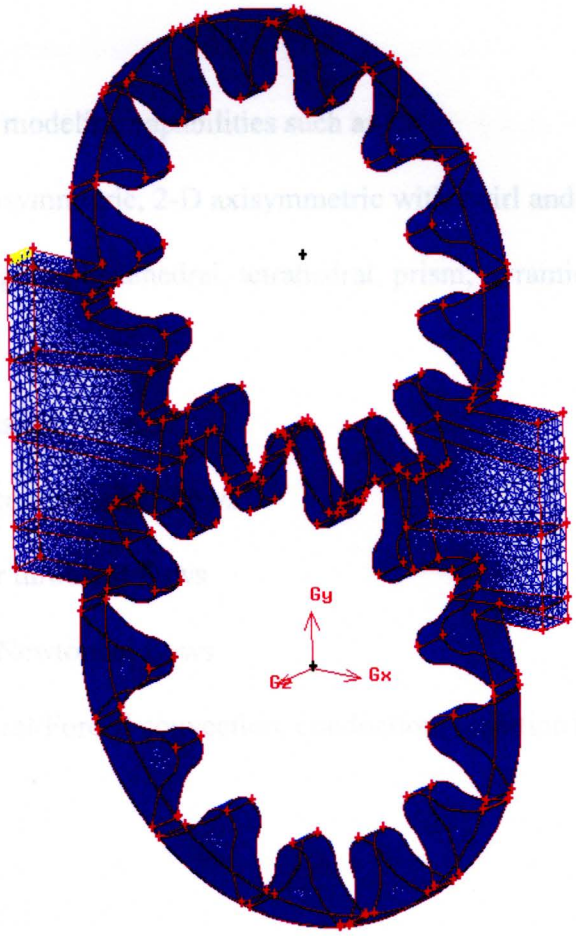

Figure 4.6 - Meshed 2.5-D Model

4.4 FLUENT – Solver

FLUENT is the state-of-the-art commercial Computational Fluid Dynamics software for modeling fluid and heat transfer problems in complex geometries. The numerical algorithm is based on the finite volume method. FLUENT is developed in C language. FLUENT has mesh flexibility including the ability to solve the flow problems using unstructured meshes in complex models. Hence, it is the most versatile software suitable for solving a problem like the gear pump. FLUENT is used both as a solver and postprocessor to review the results.

FLUENT has different modeling capabilities such as

- 2-D planer, 2-D axisymmetric, 2-D axisymmetric with swirl and 3-D flows.
- Quadrilateral, triangular, hexahedral, tetrahedral, prism, pyramid, and mixed element meshes.
- Steady-state or transient flows
- Incompressible or compressible flows
- Inviscid, laminar or turbulent flows
- Newtonian or non-Newtonian flows
- Heat transfer (Natural/Forced convection, conduction, radiation)
- Chemical reactions
- Cavitation model
- Fuel cell module and
- Dynamic mesh model for domains with moving and deforming mesh.

FLUENT has the unique capability of a Moving Dynamic Mesh (MDM), in which User Defined Function (UDF) or built-in profile can be used to simulate the moving boundary condition problems.

4.4.1 Segregated Solver

A segregated solver uses a solution algorithm of solving governing equations (continuity, momentum, and energy as well as turbulence) sequentially. The segregated solver is traditionally used for incompressible and mildly compressible flows. Hence, in this case, the segregated solver is used. The steps shown in figure are followed to arrive at the convergence

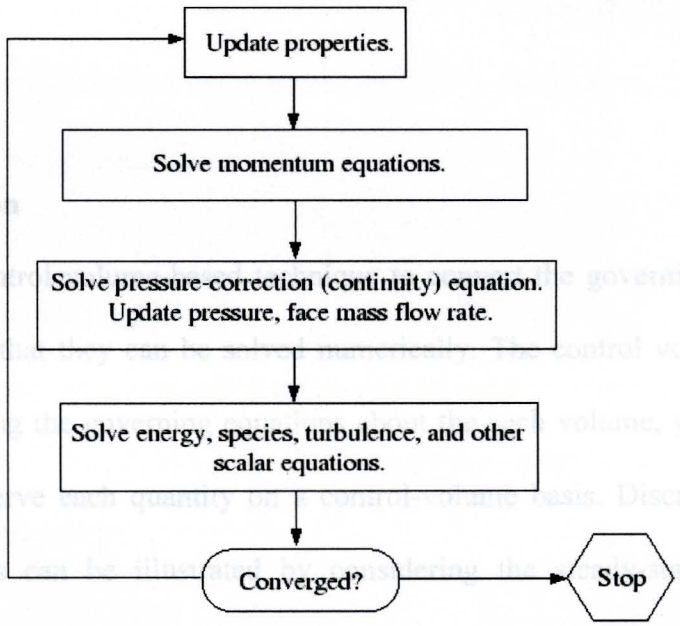

Figure 4.7 – Solver Algorithm

The segregated solver is similar to the coupled solver except for the approach of linearizing and solving the discretized equations.

4.4.2 Linearization

In segregated solution methods the discrete, non-linear governing equations are linearized to produce a system of equations for the dependent variables in every computational cell. Each discrete governing equation is linearized implicitly with respect to that equation's dependent variables. A point implicit (Guass-Seidel) linear equation is used in conjunction with an algebraic multi-grid (AMG) method to solve the resultant scalar system of equations for the dependent variable in each cell. This means that the segregated approach solves for a single variable filed by considering all cells at the same time and then solves for the next variable filed by again considering all cells at the same time, and so on.

4.4.3 Discretization

FLUENT uses a control volume-based technique to convert the governing equations to algebraic equations that they can be solved numerically. The control volume technique consists of integrating the governing equations about the each volume, yielding discrete equations that conserve each quantity on a control-volume basis. Discretization of the governing equations can be illustrated by considering the steady-state conservation equation for transport of a scalar quantity ϕ. Equation 4.1 is the equation integral form for an arbitrary control volume V.

$$\oint \rho \phi \vec{v}\, d\vec{A} = \oint \Gamma_\phi \nabla \phi . d\vec{A} + \int S_\phi dV \qquad \text{.....(4.1)}$$

The above equation can be discretized on a given cell as

$$\sum_{f}^{N_{faces}} \rho_f \vec{v}_f \phi_f . \vec{A}_f = \sum_{f}^{N_{faces}} \Gamma_\phi (\nabla \phi)_n . \vec{A}_f + S_\phi V \qquad \text{.....(4.2)}$$

By default FLUENT stores discrete values of the scalar ϕ at the centers of cell. However, face values ϕ_f are required for convection terms in equation 4.2 and must be interpolated from the cell center values. This is accomplished using an upwind scheme. For this model, a first-order upwind scheme is used, in which quantities at cell faces are determined by assuming that the cell-center values of any field variable represent a cell-average value and hold throughout the entire cell; face quantities are identical to the cell quantities. Thus in first-order upwind scheme, the face value ϕ_f is set equal to the cell-center value of ϕ in the upstream cell.

4.4.4 Pressure-velocity coupling

FLUENT has provided four methods for pressure-velocity coupling in the segregated solver: SIMPLE, SIMPLEC, PISO, and FSM. The present analysis of the gear pump is done with SIMPLE algorithm.

4.4.5 Convergence criteria

There is no universal metrics for judging convergence. Residual definitions that are useful for one class of problem are sometimes misleading for other classes of problem. Therefore it is a good idea to judge convergence not only by examining residual levels, but also by monitoring relevant integrated quantities such as drag. For most problems the default convergence criterion in FLUENT is sufficient. These criteria that the scaled residuals defined by equation decreased to 10^{-3} for all equations, except the energy equation for which the criteria are 10^{-6}. It is possible that if the initial guess is poor, the initial residuals are so large that a three-order drop in residual does not guarantee convergence. These are true for k and ε equations where initial guess are difficult.

4.4.6 Moving Dynamic Mesh (MDM) Modeling

The dynamic mesh model in FLUENT can be used to model flows where the shape of the domain continuously changes with time due to motion of the domain boundaries. The motion can either be a prescribed motion or an un-prescribed motion where the subsequent motion is determined based on the solution at the current time. The update of the volume mesh is handled automatically by FLUENT at each timestep based on the new position of the boundaries. To use the dynamic mesh model, it is needed to provide a starting volume mesh and the details of the motion of any moving zones in the model by using either boundary profiles, user-defined functions (UDFs), or the Six Degree of Freedom solver (6DOF). FLUENT facilitates to specify the description of motion on either face or cell zones. If the model contains moving and non-moving zones, it is needed to identify these zones by grouping them into respective face or cell zones. The

motion of the rigid body was specified by the linear and angular velocity of the center of gravity. Velocities can be specified by profiles or user-defined functions (UDF). UDF can be written for linear velocity, angular velocity or both as a function of time. In this case, angular velocity was specified to both gears by UDF. For the gear pump, it is essential to use UDF to define the motion of gears as the geometry of fluid domain changes with time. A UDF code was written in "C" language.

Chapter 5

PARALLEL PROCESSING FOR 3-D MODEL

5.1 Need of Parallel Processing

The complex fluid domain of gear pump, fine gaps, and the need to control the skewness of the mesh below 0.6 has increased the quantity of cells in the meshed 2.5-D model of gear pump to 2.5 million. During simulation of this model, it has been observed that the single workstation having dual processor of 2 GB RAM is not enough to process huge data generated by this large number of cells of the 2.5-D model. The limitations of processing the complex model with this huge number of cells on the present computing system called for use of a higher number of high RAM computers in parallel in which the data is shared among the nodes and processed in parallel. The supercomputer having multiple nodes connected in parallel with a suitable operating system provides the solution of solving problems with huge data. Ohio Supercomputer Center (OSC), located in Columbus, Ohio, is a non-profit organization and provides such reliable high performance computing and high performance communications infrastructure for a diverse statewide/regional community including education, academic research, industry and state government.

After several failed attempts in simulating 3-D models on workstation, it was decided to simulate the 3-D models on one of the supercomputer at Ohio Supercomputer Center - OSC Pentium 4 Cluster with a Linux operating system.

5.2 Parallel processing

Parallel processing refers to the concept of speeding-up the execution of a program by dividing the program into multiple fragments that can be executed simultaneously on its own processor. The evolution of parallel processing is a major advancement in modern computing. It enables researchers in many scientific fields to conduct large-scale simulations of complex physical phenomena.

The supercomputer is a cluster of many computers which are used for parallel processing. Parallel supercomputer architectures are mostly one of two classes, Distributed Memory or Shared Memory. Distributed Memory systems consist of a large number of computing nodes that are connected via an interconnection network. Each computing node has its own locally accessible memory bank. Memory banks of other nodes are accessible via messages passing through the interconnection network. The Shared Memory systems are made of a set of computing nodes with a large unified memory image or a global bank that is equally accessible by all nodes through shared memory bus. Distributed Memory systems can scale up to a very large number of computing nodes whereas shared-memory systems do not, due to limitation on bus bandwidth.

Typically parallel processing follows the domain decomposition approach where the computational domain is divided into smaller blocks of roughly the same size such that each node performs the same set of tasks on the portion of data that is assigned to it. Shared border data is updated between blocks through node-to-node communications.

Message Passing Interface (MPI) is the standard for conducting such communications. MPI is library of specifications designed to standardize how node-to-node communications are carried out.

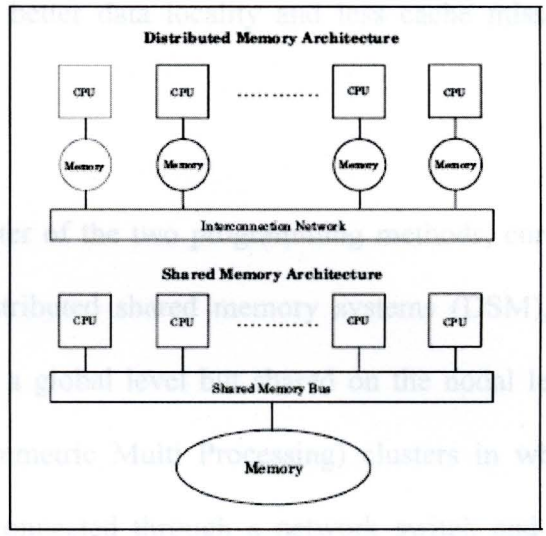

Figure 5.1 – Parallel Architecture

The domain decomposition approach also can be implemented on shared memory systems using MPI. However, programming of parallel shared systems generally adopts the task of decomposition of the model. Since the memory image, in shared memory systems, is unified, it is logical to decompose the task, rather than the data, into smaller tasks and execute them in parallel via computer directives such as OpenMP. OpenMP is a set of standard specifications for writing compiler directives and subroutines to achieve parallel programming on shared memory systems.

While OpenMP is limited to shared memory systems, MPI can be implemented on both shared and distributed platforms. On the other hand, OpenMP is much easier to code than MPI. Also OpenMP allows for incremental parallelism of serial codes whereas MPI doesn't. However, sometimes MPI is faster than OpenMP even on shared memory systems as a result of better data locality and less cache misses especially for large memory jobs.

In trying to get the better of the two programming methods, current parallel computers are built as hybrid distributed shared memory systems (DSM), i.e., systems that are distributed memory on a global level but shared on the nodal level. Examples of such systems are SMP (Symmetric Multi Processing) clusters in which a large cluster of computing nodes are connected through a network switch and each node consists of number of processors with a local shared memory bank. Programming on DSM platforms exploits the benefits of both MPI and OpenMP to achieve greater scalability. The coding strategy is to divide the computational domain into blocks and assign each block to a different node. Data is communicated between nodes via MPI; while each node, task decomposition programming is achieved using OpenMP (12).

5.3 OSC Linux Cluster

Ohio Supercomputer Center provided the OSC Pentium 4 Cluster for performing analysis. The OSC Pentium 4 Cluster is a distributed/shared memory system with a Linux operating system. The configuration of the Pentium 4 Cluster is two file server nodes, two front-end nodes, two hundred and fifty-six nodes and sixteen storage nodes. The file

server nodes each have two 2.4 GHz Xeon processors, 8 GB of memory and 500 GB of shared Fibre channel attached storage.

- 144 computer nodes for serial jobs configured with - Four gigabytes of RAM; Two, 2.4 GHz Intel P4 Xeon processors, each with 512 kB of secondary cache; One 80 gigabyte ATA100 hard drive; One 100Base-T Ethernet interface and one 1000Base-T Ethernet interface.
- 112 compute nodes for parallel jobs, configured with - Four gigabytes of RAM; Two, 2.4 GHz Intel P4 Xeon processors, each with 512 kB of secondary cache; One 80 gigabyte ATA100 hard drive; One Infiniband 10Gb interface; One 100Base-T Ethernet interface and one 1000 Base-T Ethernet interface.
- 1 front-end node configured with – Four gigabytes of RAM; Two, 2.4 GHz Intel P4 Xeon processors, each with 512 kB of secondary cache; One 80 gigabyte ATA100 hard drive; One 100Base-T Ethernet interface and one 1000Base-T Ethernet interface
- Useful link: www.osc.org/hpc

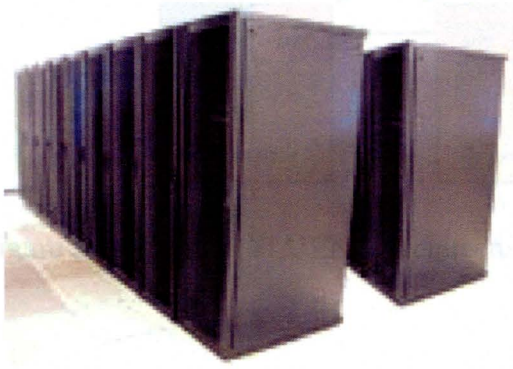

Figure 5.2 - OSC Pentium 4 Cluster (Courtesy: Ohio Supercomputer)

5.4 Parallel Processing in FLUENT

The FLUENT solver manages file input and output, data storage and flow field calculating using a single solver on a single computer. Also FLUENT has a parallel solver which allows a computing solution by using multiple processes that may be executing on the same computer or on different computers in a network. FLUENT parallel architecture is shown in Figure 5.3.

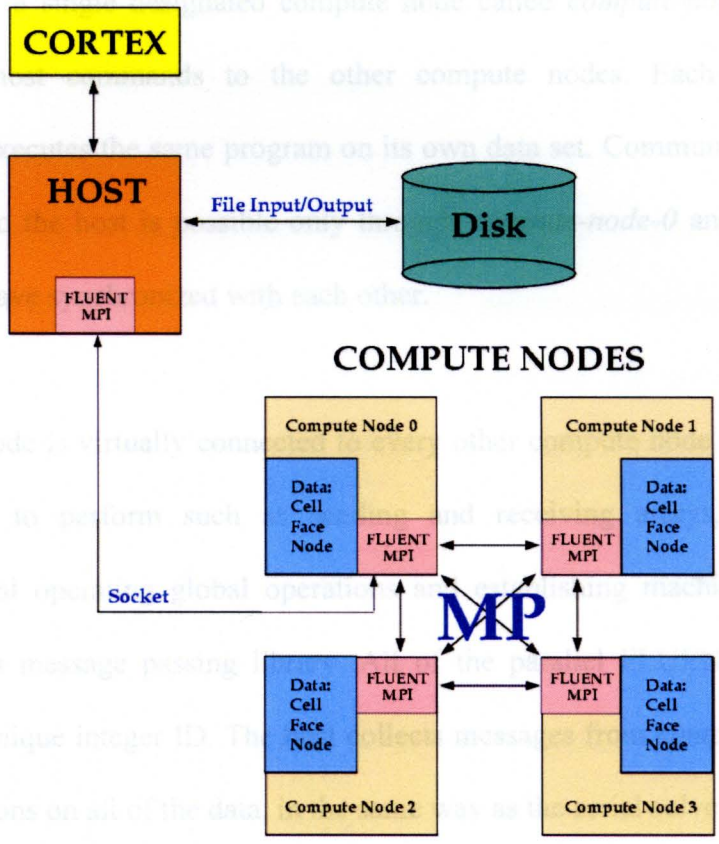

Figure 5.3 – Parallel FLUENT Architecture

Parallel processing in FLUENT involves an interaction between FLUENT, a host process and a set of compute-node processes. FLUENT interacts with the host process and the collection of computer nodes using a utility called *cortex* that manages FLUENT's user

interface and basic graphical functions. Parallel FLUENT splits up the grid and data into multiple partitions, and then assigns each grid partition to a different compute process (or node). The number of partitions is an integral multiple of the number of compute nodes available. FLUENT uses a host process that does not contain any grid data. Instead, the host process only interprets commands from FLUENT's graphics–related interface, *cortex.* The host distributes those commands to the other compute nodes via a socket communicator to a single designated compute node called *compute-node-0.* This node distributes the host commands to the other compute nodes. Each compute node simultaneously executes the same program on its own data set. Communication from the compute nodes to the host is possible only through *compute-node-0* and only when all compute nodes have synchronized with each other.

Each compute node is virtually connected to every other compute node and relies on its "communicator" to perform such as sending and receiving arrays, synchronizing, performing global operating global operations and establishing machine connectivity. FLUENT has its message passing library. All of the parallel FLUENT processes are identified by a unique integer ID. The host collects messages from *computer-node-0* and performs operations on all of the data, in the same way as the serial solver (13).

5.5 Grid Partitioning

FLUENT can partition the grid into groups of cells that can be solved on separate processors when it is read into the parallel solver. The partitioning of the grid may be done by using automatic partitioning algorithms when reading an unpartitioned grid into

a parallel solver or may it be performed manually in a serial solver. In the case of a large grid it is better to read them in to parallel solver (14).

5.5.1 Grid Partitioning Method

Partitioning the grid for parallel processing has three major goals

- Create partitions with equal numbers of cells
- Minimize the number of partition interfaces
- Minimize the number of partition neighbors

Balancing the partitions ensures that each processor has an equal load and that the partitions will be ready to communicate at about the same time. Since communication between partitions can be a relatively time-consuming process, minimizing the number of interfaces can reduce the time associated with data interchange. The partitioning schemes in FLUENT uses bisection algorithms, to create the partitions which have no limitation on the number of partitions.

5.5.2 Bisection Methods

The grid is partitioned using a bisection algorithm. The selection algorithm is applied to the parent domain and then recursively applied to the child sub-domains. The grid can be partitioned using one of the following algorithms listed below such as Cartesian Axes, Cartesian Strip, Cartesian X-, Y-, Z-Coordinate, Cartesian R axes, Cartesian RX-, RY-, RZ-coordinate, Cylindrical Axes, Cylindrical R-, Theta-, Z-Coordinate, Metis, Polar Axes, Polar R-Coordinate, Polar Theta-Coordinate, Principal Axes, Principal Axes, Spherical Axes, Spherical Rho-, Theta-, Phi-Coordinate. In this case, the X-Coordinate

5.5.3.1 Smooth

The smooth optimization scheme attempts to minimize the number of partition interfaces by swapping cells between partitions. The scheme traverses the partition boundary and gives cells to the neighboring partition if the interface boundary surface area is decreased. Figure 5.5 shows the mesh before and after smooth is applied.

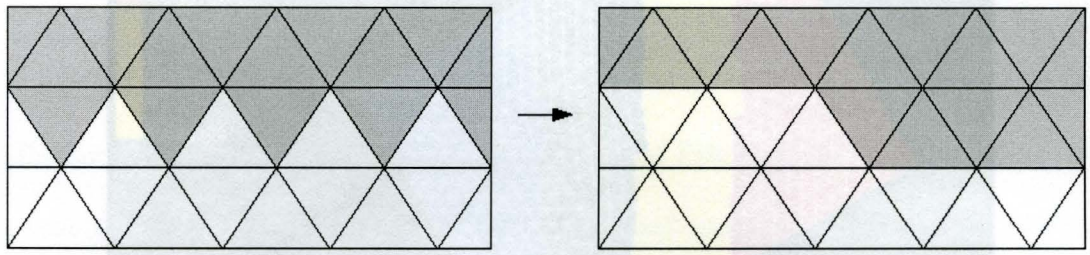

Figure 5.5 – Smooth

5.5.3.2 Merge

The merge optimization scheme attempts to eliminate orphan clusters from each partition. An orphan cluster is a group of cells with the common feature that each cell within the group has at least one face which coincides with an interface boundary. Orphan clusters can degrade multigrid performance and lead to large communication costs.

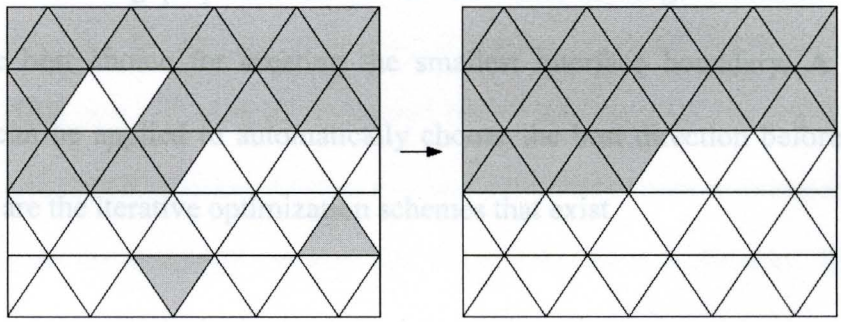

Figure 5.6 – Merge

Method is used for partitioning of the mesh. Figure 5.4 shows the how the partition is created by using the X-Coordinate Method.

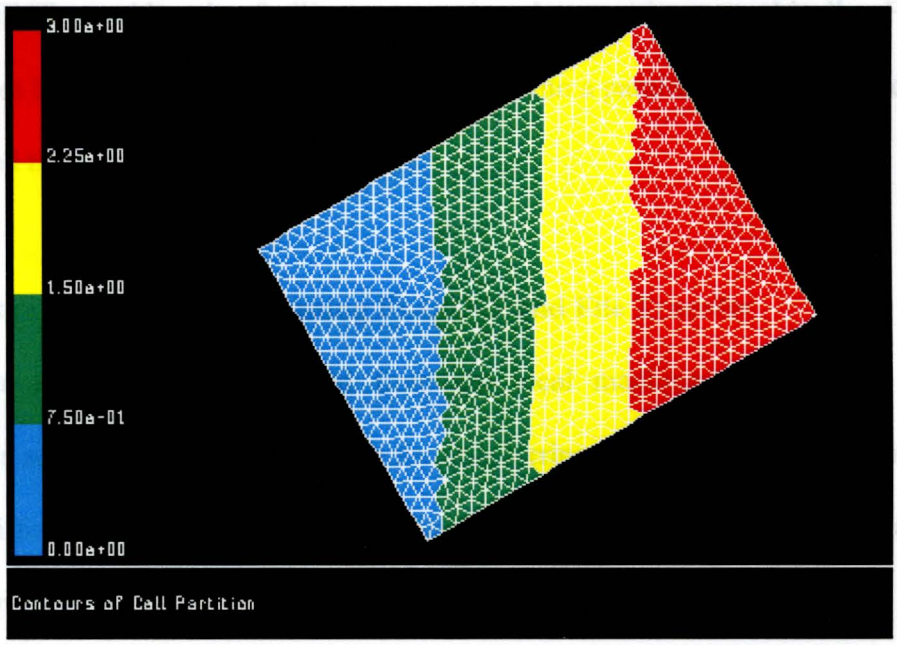

Figure 5.4 – Partition Created with Cartesian X-Coordinate Method

5.5.3 Optimizations

Additional optimization can be applied to improve the quality of the grid partitions. The heuristic of bisecting perpendicular to the direction of longest domain extent is not always the best choice for creating the smallest interface boundary. A "pre-testing" operation can be applied to automatically choose the best direction before partitioning. Following are the iterative optimization schemes that exist.

- Smooth
- Merge

5.5.4 Pre-testing and Checking

With the Principal Axes or Cartesian Axes Method, the bisection can be improved by testing different directions before performing the actual bisection. After partitioning a grid, FLUENT provides the facility to examine the partitions graphically. FLUENT distinguishes between two cell partition schemes within a parallel problem, the active cell partition and the stored cell partition.

5.6 Remote Login to Server for using FLUENT and File Transfer

The remote server can be accessed from the local machine for opening different applications such as FLUENT as well as for transferring files across the machine via the internet. Different software is available for login and using software application available on server from local machine.

5.6.1 Secure Shell (SSH)

Secure Shell is a program to log into another computer over network to execute commands in a remote machine and to move files from one machine to another. It provides a secure way to transmit data over TCP/IP network between computers over unsecured channels. This program is developed to other programs of remote login such as telnet, rsh, rlogin, and rcp, as these traditional BSD commands are vulnerable to different kinds of attacks. The X Window System also has a number of severe vulnerabilities. With SSH, it is secure for remote X sessions which are transparent to the user. There are two version of Secure Shell available: SSH1 and SSH2.

There are different free versions of SSH for different operating systems. For windows, the following free SSH ports are available.

- Robert O'Callahan's TTSSH, SSH1
- Gorben Chaffee's Command line port of SSH1 and SCP1
- Corinna Vinschen has ported OpenSSH to Cygwin
- PuTTY (SSH1 & SSH2) client for 32-bit windows
- FiSSH, Mass Confusion's 32-bit SSH1 client for Windows
- Cynus Win32 port of Secure Shell and Telnet client
- PenguiNet Secure Shell and Telnet client.
- Useful websites: www.ssh.com; www.ietf.org; www.onsight.com/faq/ssh

5.6.2 PuTTY

The PuTTY Secure shell is used for remote login to Linux cluster. PuTTY is a Telnet, Rlogin and SSH network client program for 32-bit Windows system. PuTTY is used to connect to remote UNIX/LINUX servers from a PC with a windows operating system. PuTTY provides Graphical User Interface for login in which the host name of the server, user login, and password can be entered. While logging in from a PC to a remote server, it needed to select one of the protocols Telnet, Rlogin, or SSH. The selection of a type of protocol changes a number in the "Port" box, due to provisions of different network ports by the server machine. The secured remote login protocol SSH was used for accessing the remote server. SSH has a feature of protecting data transfer from network attack known as spoofing. To prevent this attack, each server has a unique identifying code called a host key. PuTTY records the host key for each connected server in the Windows

Registry. Every time of login to the server, it checks the host key presented by the server to be sure it is the same key as it was the last time you logged in.

- Useful websites: www.putty.nl

5.6.3 WinSCP

WinSCP is an open source free SFTP client for Windows using SSH; Legacy SCP protocol is also supported. Its main function is safe copying of files between a local and a remote computer. It has two graphic user interfaces Norton –like and other is Explorer-like. The WinSCP can be used for navigation, uploading/downloading files, managing connections, editing/opening files, changing properties, synchronizing the local directory, and renaming, moving, duplicating and deleting files. WinSCP supports two secure transfer protocols, SFTP and SCP.

SCP (Secure Copy Protocol)

SCP is mostly used with SSH-1 which is universally supported on Unix-like platforms. The SCP protocol allows only file transfers in both directions. However WinSCP offers many other features such as cd, ls, pwd.

SFTP (SSH File Transfer Protocol)

SFTP is mostly operated as a subsystem of SSH-2 which supports even non-Unix-like platforms. The SFTP server does not need access to the shell. Hence it is more independent than other remote operating system.

- Useful websites: www.winscp.org; www.winscp.net

5.6.4 Portable Batch System (PBS)

Today's computing challenges often demand resources and skills not available within a single organization or at one location. Users need cost-effective services and tools that allow easy sharing of costly computing resources without knowing the specifications of each system environment. The Portable Batch system is an open source batch queuing system operating on networked, multi-platform UNIX-like environments. It is an extension of the NQS batch queuing system developed for NASA in the early to mid-1990s. There are two such tools:

- OpenPBS
- PBS Pro

In this case, OpenPBS is used for submitting the job to a Linux cluster. The PBS tools help the user in many ways such as:

- Determining resources requirements for a job
- Submitting jobs to PBS
- Monitoring job progress
- Copying job output to the specified directory
- Intimating users about the status of job submitted via email
- Finding when the job will be completed
- Useful website: www.openpbs.org

5.6.5 Exceed

Exceed is an X server program that provides the graphical features of an X server for use with remote Unix machines from Windows operating systems. Exceed is part of Hummingbird Communications package, which includes other communication tools.

Client/Server Software and X Windows

X Windows is client/server software where client software goes to a server to request service. In X Windows, X server software such as exceed runs on local machines and the client software runs on the remote server that uses the X server running on local machine to display the output.

- Useful website: www.uic.edu/depts/accc/software/exceed

Chapter 6

COMPUTATIONAL SETUP AND FLOW ANALYSIS

6.1 Flow Analysis in Solver – FLUENT

The flow is simulated in 2-D as well as in 3-D for the gear pump - PGP640. The 2-D analysis is performed on the workstation whereas 3-D analysis is performed on OSC supercomputer Pentium 4 Cluster. The analysis has been carried out for different operating conditions such as outlet pressure, speed and gap between the gears etc., in order to investigate the performance of the gear pump. For 2-D models of the gear pump, analysis is carried out for various cases with different combinations of the following parameters:

Table 6.1 – Operating Conditions Considered for 2-DAnalysis

Gap between Gears	Speeds	Outlet Pressures
• 15 μm	• 3500 rpm	• 3500 psi
• 22 μm	• 3000 rpm	• 3000 psi
• 30 μm	• 2500 rpm	• 2500 psi
	• 2000 rpm	

However, the analysis for only one case has been performed for a 3-D model due to extremely long execution time for just one tooth movement (of 27 deg.), huge data generation of several GB for one analysis and limitations of computing resources. The following case is selected for analysis:

Table 6.2 – Operating Conditions Considered for 3-D Analysis

Gap between Gears	Speeds	Outlet Pressures
• 30 μm	• 3000 rpm	• 3000 psi

6.2 Computational Setup

In FLUENT, the computational setup for the 3-D model of the gear pump is different from the setup for the 2-D model in some respects such as the boundary conditions and dynamic mesh zones.

6.2.1 Setup for 2-D

The setup for 2-D model in FLUENT:

Computation Domain:

Inlet	stationary
Outlet	stationary
Casing	stationary
Gears	rotation at four different speeds

Boundary Conditions:

Inlet	pressure_inlet
Outlet	pressure_outlet
Casing	Wall
Gears	Wall

Solver:

Solver	Segregated
Formulation	Implicit
Space	2-D, double precision
Time	Unsteady

Viscous model:

Model	k-epsilon, standard
Model constant	$C_\mu = 0.09$, $C_{1\varepsilon} = 1.44$, $C_{2\varepsilon} = 1.44$, Prandtl Number = 1

Solution Control:

Pressure-Velocity Coupling	SIMPLE
Discretization	Pressure – Standard
	Momentum – First Order Upwind
	Turbulence Kinetic Energy – First Order Upwind
	Turbulence Dissipation Rate – First Order Upwind
Timestep	1e-06 sec.
Convergence criteria	Residual : 1e-06

Dynamic Mesh Modeling:

User Defined Function	Suitable for 2-D model – g3000.c (Please refer to Appendix I)
Dynamic Mesh zone	CW Gear :
	Type : Rigid body
	Motion : clock_wise
	CCW Gear :
	Type : Rigid body
	Motion : anticlock_wise

6.2.2 3-D Model Setup

In FLUENT, setup for 3-D model will be similar to 2-D model except following:

Computation Domain:

Inlet	Stationary
Outlet	Stationary
Casing	Stationary
Gears	Rotational speeds : 3000 rpm
Sidewalls	Stationary

Boundary Conditions:

Inlet	pressure_inlet
Outlet	pressure_outlet
Casing	Wall
Gears	Wall
Sidewall	Wall

Solver:

Solver	Segregated
Formulation	Implicit
Space	3-D, double precision
Time	Unsteady

Solution Control:

Pressure-Velocity Coupling	SIMPLE
Discretization	Pressure : Standard Momentum : First Order Upwind Turbulence Kinetic Energy : First Order Upwind Turbulence Dissipation Rate : First Order Upwind
Timestep	1e-07 sec.
Convergence criteria	Residual : 1e-03

Dynamic Mesh Modeling:

User Defined Function	Suitable for 3-D model to be ran on parallel processor: parallel_g3000_orig.c (Please refer to Appendix II)

Dynamic Mesh zone

CW Gear :

Type : Rigid body

Motion : clock_wise

CCW Gear :

Type : Rigid body

Motion : anticlock_wise

CCW Gear :

Type : Rigid body

Motion : anticlock_wise

Sidewall1 :

Type : Deforming

Geometry Definition :

Definition : Plane

Point on plane :

(any point on sidewall in fluid zone (X = -25.31771 mm, Y = 27.86212 mm, Z = 0.00053 mm)

Plane Normal :

(X = 0, Y = 0, Z = -1)

Meshing Option :

Methods : Smoothing

Remeshing

Min. Length Scale : 0.01 mm

Max. Length Scale : 0.70 mm

Max. Skewness : 0.7

Sidewall2 :

Type : Deforming

Geometry Definition :

Definition : Plane

Point on plane :

(any point on sidewall in fluid zone)

(X = -22.05935 mm, Y = 26.05279 mm,

Z = 5.00727 mm)

Plane Normal :

(X = 0, Y = 0, Z = 1)

Meshing Option :

Methods : Smoothing

Min. Length Scale : 0.01 mm

Max. Length Scale : 0.70 mm

Max. Skewness : 0.7

(Note: The second sidewall doesn't need remeshing option in Meshing Option Method)

6.3 Simulation on Supercomputer

The steps to be followed for simulation on Linux cluster -

- Open the networking software **WinSCP** which is the file transfer software on the remote machine through the internet.
- Login to the server by giving "*hostname*", your "*login ID*" and "*password*"
- Transfer the 2.5-D mesh file created in Gambit along with parallelized UDF in a folder created in the home directory.
- Open networking software **PuTTY** for opening a Linux window on workstation, which will be used for simulation.
- Login to the server by giving *"hostname"*, *"login ID"* and *"password"*.
- Submit request for the job with following **PBS** command

  >> qsub –V –I –L walltime=24:0:0 –l nodes=8:ppn=2

This **PBS** command requests the server to allot 8 nodes and 2 processors on each node for interactive job for 24 hrs.

- Keep the Linux window open.
- The server will prompt "*job 695829.nfs4.osc.edu:17001 ready*", when it will allot user specified number of nodes
- Start simulating the flow problem in FLUENT
- Export the variable to the server informing the address of your machine.

  >> export DISPLAY="ESB082.eng.ysu.edu":0.0

  Note: The name of your machine may be different.
- Create a file having information of nodes allotted to you named *"pnodes"*

  >>rm –f pnodes

  >>cat $PBS_NODEILE | sort > pnodes
- Load the FLUENT into your home directory.

  >> module load fluent6.2.16
- Open the FLUENT's GUI window

  >> fluent 3-ddp –t16 –cnf=pnodes -pnet
- Set up the problem by following same steps as 2-D simulation on a single workstation.
- Set the model for running.

Chapter 7

RESULTS OF 2-D FLOW ANALYSIS

7.1 Flow Analysis in 2-D

The results presented here are obtained from 2-D flow analysis performed for three different models having gaps of 30 μm, 22 μm and 15 μm between gears. Different cases are investigated by varying speed between 2000 rpm and 3500 rpm as well as outlet pressure from 2500 psi to 3500 psi. The simulation is initialized with inlet pressure as 0 psi. The convergence criteria for solution is used for simulation as 1e-6 for residual and the time step used as 1e-6 sec. Figure 7.1 shows convergence of solution at each time step.

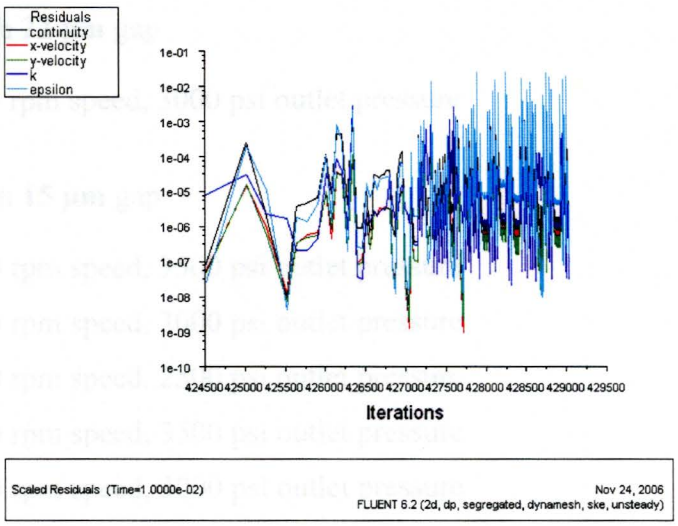

Figure 7.1 – Convergence of Solution

The simulations are performed for time steps up to 10,000. Following are cases with different models and operating conditions that model the performance of gear pump. The simulations provide reasonable output data regarding pressure, velocity, mass flow rate and more.

For 2-D Model with **30 μm** gap

Case 1: 3500 rpm speed, 3500 psi outlet pressure
Case 2: 3500 rpm speed, 3000 psi outlet pressure
Case 3: 3500 rpm speed, 2500 psi outlet pressure
Case 4: 3000 rpm speed, 3500 psi outlet pressure
Case 5: 3000 rpm speed, 3000 psi outlet pressure
Case 6: 3000 rpm speed, 2500 psi outlet pressure
Case 7: 2500 rpm speed, 3500 psi outlet pressure
Case 8: 2500 rpm speed, 3000 psi outlet pressure
Case 9: 2500 rpm speed, 2500 psi outlet pressure
Case 10: 2000 rpm speed, 3500 psi outlet pressure
Case 11: 2000 rpm speed, 3000 psi outlet pressure
Case 12: 2000 rpm speed, 2500 psi outlet pressure

For 2-D Model with **22 μm** gap

Case 13: 3000 rpm speed, 3000 psi outlet pressure

For 2-D Model with **15 μm** gap

Case 14: 3500 rpm speed, 3500 psi outlet pressure
Case 15: 3500 rpm speed, 3000 psi outlet pressure
Case 16: 3500 rpm speed, 2500 psi outlet pressure
Case 17: 3000 rpm speed, 3500 psi outlet pressure
Case 18: 3000 rpm speed, 3000 psi outlet pressure
Case 19: 3000 rpm speed, 2500 psi outlet pressure
Case 20: 2500 rpm speed, 3500 psi outlet pressure
Case 21: 2500 rpm speed, 3000 psi outlet pressure
Case 22: 2500 rpm speed, 2500 psi outlet pressure
Case 23: 2000 rpm speed, 3500 psi outlet pressure
Case 24: 2000 rpm speed, 3000 psi outlet pressure
Case 25: 2000 rpm speed, 2500 psi outlet pressure

7.2 30 µm Gap Model

The analyses of 30 µm model is conducted for four different speeds and three different outlet pressures in 2-D. The mass flow rate through suction, pressure and velocity at two points are monitored periodically. Also pressures contour and velocity vector plots are captured at every 25th time step. The simulation is performed for one of the cases with 3000 rpm and 3000 psi outlet pressure for 10000 time steps of 1e-6 sec.

7.2.1 Mass Flow Rate

The variation in mass flow rate is studied for 180 deg. rotation of gears. It is observed that the mass flow rate variation is sinusoidal, and steady. Figure 7.2 and 7.3 shows the mass flow rate with time. No change has been observed in the nature of variation of mass flow with time.

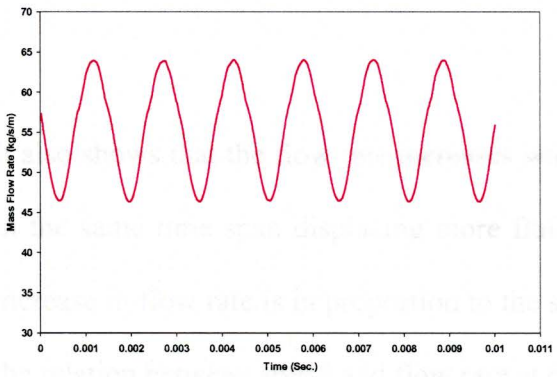

Figure 7.2 – Mass Flow Rate Results from Simulation (for 3000 rpm & 3000 psi)

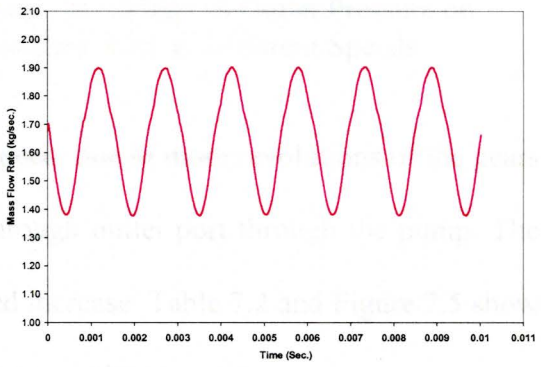

Figure 7.3 – Converted Mass Flow Rate for Actual Pump. (for 3000 rpm & 3000 psi)

The results also provide data on the effect of different speeds and outlet pressures on mass flow rate. Table 7.1 shows the mass flow rate at various speeds and outlet pressures. It shows that mass flow rate drops with increase in outlet pressure due to high resistance to flow. Figure 7.4 shows this effect on mass flow rate which is proportional to outlet pressure.

Table 7.1 – Mass Flow Rate for Different Speeds and Outlet Pressures.

	Speed (rpm)	Outlet Press-ure (psi)	Mass Flow Rate (kg/sec.)		
			Max.	Min.	Ave
Case 1	3500	3500	2.22	1.63	1.92
Case 2		3000	2.24	1.68	1.96
Case 3		2500	2.25	1.74	2.00
Case 4	3000	3500	1.88	1.34	1.61
Case 5		3000	1.90	1.38	1.64
Case 6		2500	1.92	1.44	1.68
Case 7	2500	3500	1.55	1.04	1.30
Case 8		3000	1.57	1.09	1.33
Case 9		2500	1.58	1.14	1.36
Case 10	2000	3500	1.21	0.75	0.98
Case 11		3000	1.24	0.80	1.02
Case 12		2500	1.25	0.85	1.05

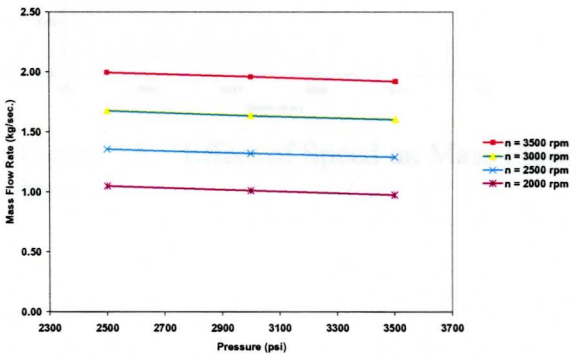

Figure 7.4 – Effect of Outlet Pressure on Mass Flow Rate at Different Speeds

It also shows that the flow rate increases with speed, due to more revolutions of the gears in the same time span displacing more fluid through outlet port through the pump. The increase in flow rate is in proportion to the speed increase. Table 7.2 and Figure 7.5 show the relation between speed and flow rate at different outlet pressures.

The mass flow rate is less than the theoretical flow rate for all the cases, which is due to the slip through the gaps between casing and gear as well as between gears from the high pressure region to the low pressure.

Table 7.2 - Theoretical and Numerical Mass Flow Rates

	Theoretical Mass Flow Rate (kg/sec.)	Numerical Mass Flow Rate (kg/sec.)		
Outlet Pressure (PSI)		3500	3000	2500
Speed (rpm)				
3500	2.56	1.92	1.96	2.00
3000	2.20	1.61	1.64	1.68
2500	1.83	1.30	1.33	1.36
2000	1.46	0.98	1.02	1.05

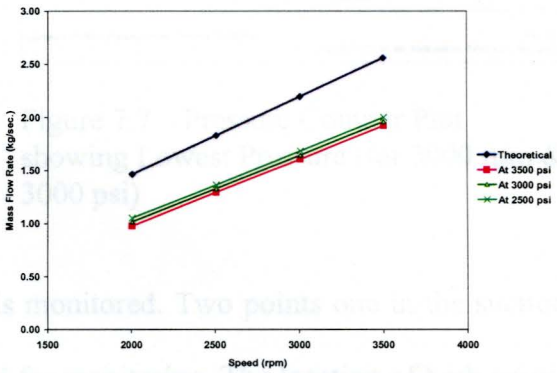

Figure 7.5 – Effect of Speed on Mass Flow Rate

7.2.2 Pressure

Along with mass flow rate, the pressure contours are also obtained from the simulation providing details of pressure variation in fluid domain of the gear pump. The pressure contours are captured at every 25th timestep to observe variation in highest and lowest pressure in the fluid zone with respect to time. In this case with 3000 rpm and 3000 psi outlet pressure, the maximum pressure reached to 4220 psi and falls to a minimum of -8440 psi. Figure 7.6 and 7.7 shows the highest and lowest pressure respectively. The negative pressure is observed in the gap between the gears teeth where the pressure region changes from high to low generating high velocity flow through narrow gaps.

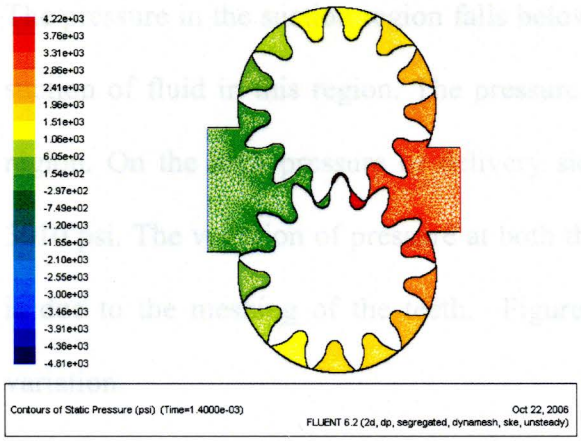

Figure 7.6 – Pressure Contour Plot showing Highest Pressure (for 3000 rpm & 3000 psi)

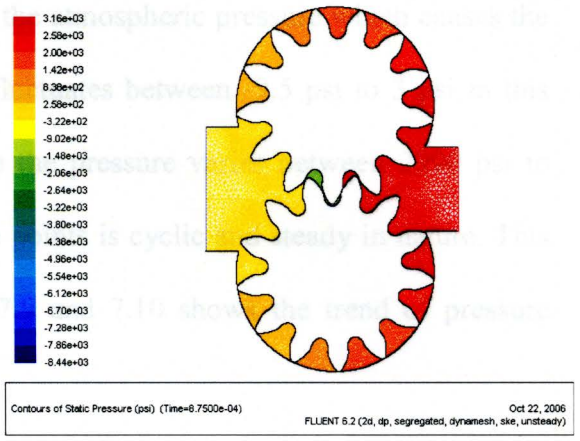

Figure 7.7 – Pressure Contour Plot showing Lowest Pressure (for 3000 rpm & 3000 psi)

Also, pressure at two points in fluid region is monitored. Two points one in the suction region and other in delivery region are located for monitoring. The location of both points is shown in Figure 7.8.

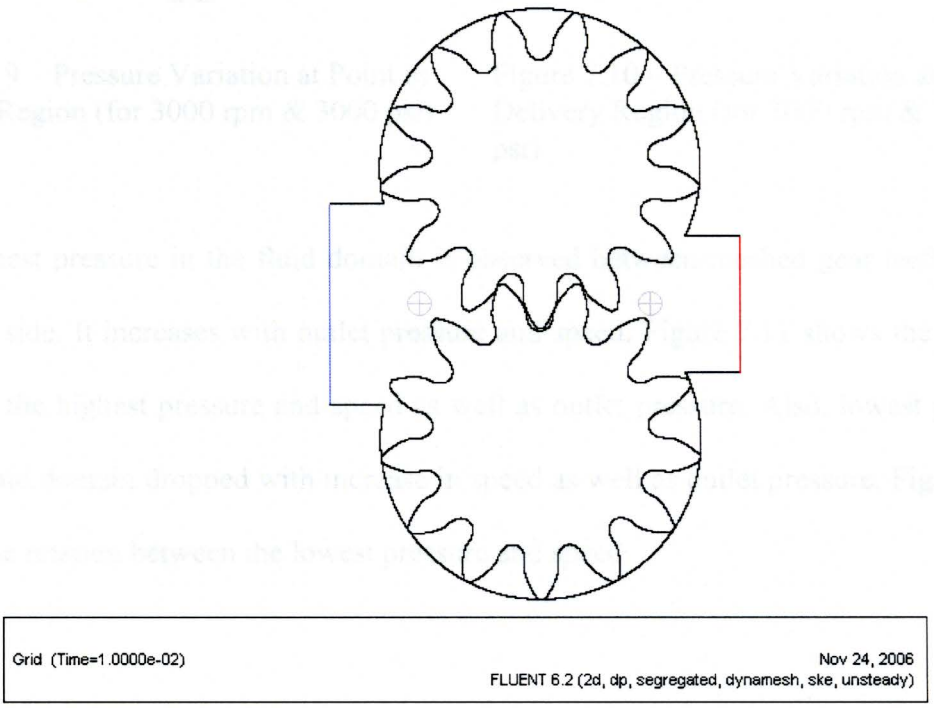

Figure 7.8 – The Point Locations for Pressure Monitoring

The pressure in the suction region falls below the atmospheric pressure which causes the suction of fluid in this region. The pressure fluctuates between -9.5 psi to 3 psi in this region. On the high pressure or delivery side the pressure varies between 2990 psi to 3010 psi. The variation of pressure at both the points is cyclic and steady in nature. This is due to the meshing of the teeth. Figure 7.9 and 7.10 shows the trend of pressure variation.

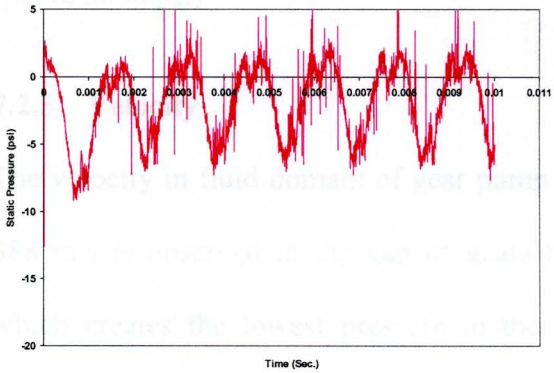

Figure 7.9 – Pressure Variation at Point in Suction Region (for 3000 rpm & 3000 psi)

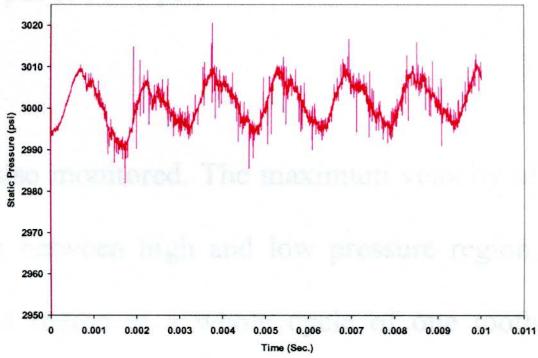

Figure 7.10 – Pressure Variation at Point in Delivery Region (for 3000 rpm & 3000 psi)

The highest pressure in the fluid domain is observed between meshed gear teeth on the delivery side. It increases with outlet pressure and speed. Figure 7.11 shows the relation between the highest pressure and speed as well as outlet pressure. Also, lowest pressure in the fluid domain dropped with increase in speed as well as outlet pressure. Figure 7.12 shows the relation between the lowest pressure and speed.

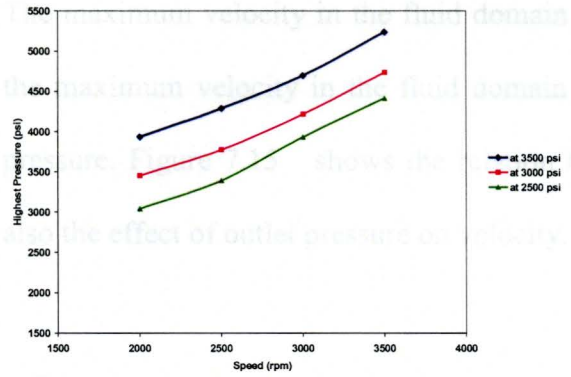

Figure 7.11 – Comparison Between Highest Pressure in Fluid Zone and Speed (for 3000 rpm & 3000 psi)

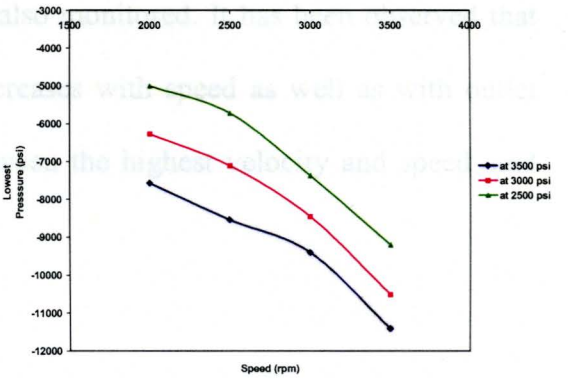

Figure 7.12 – Comparison Between Lowest Pressure in Fluid Zone and Speed (for 3000 rpm & 3000 psi)

7.2.3 Velocity

The velocity in fluid domain of gear pump is also monitored. The maximum velocity of 388 m/s is observed in the gap of gears teeth between high and low pressure region, which creates the lowest pressure in the fluid region in a whole cycle of one tooth movement. Figure 7.13 and 7.14 shows the high velocity of flow between the smallest gaps.

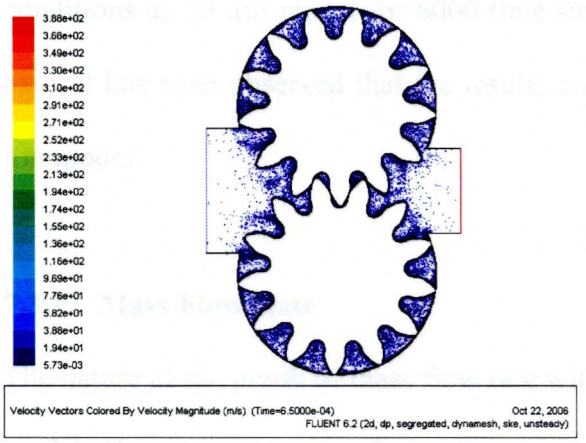

Figure 7.13 – Velocity Vector Plot (for 3000 rpm & 3000 psi)

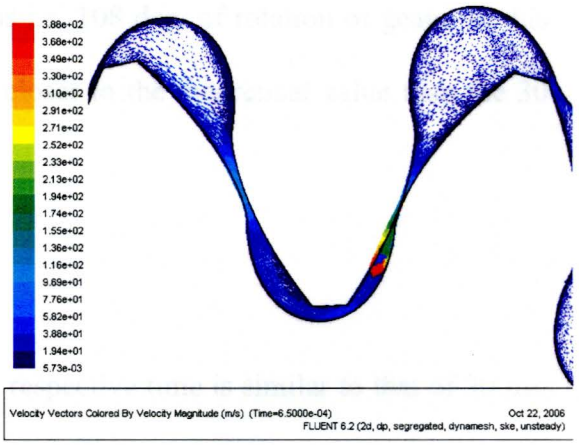

Figure 7.14 – Enlarged View of Velocity Vector Plot (for 3000 rpm & 3000 psi)

The maximum velocity in the fluid domain is also monitored. It has been observed that the maximum velocity in the fluid domain increases with speed as well as with outlet pressure. Figure 7.15 shows the relation between the highest velocity and speed, and also the effect of outlet pressure on velocity.

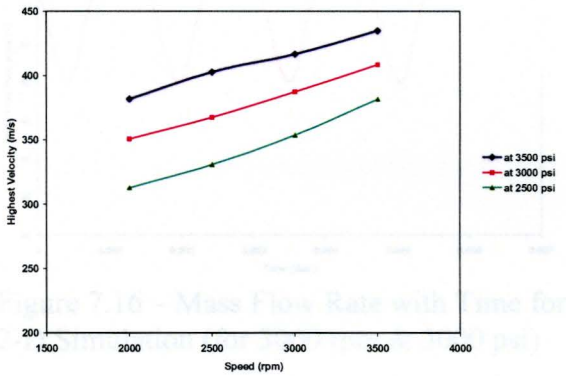

Figure 7.15 – Comparison between Highest Velocity in Fluid Zone and Speed (for 3000 rpm & 3000 psi)

7.3 15 µm Gap Model

The 2-D analyses for 15 µm gap model has been performed with same external conditions as 30 µm model for 6000 time steps i.e. 108 deg. of rotation of gears. In this case, it has been observed that the results are closer to the theoretical value than the 30 µm model.

7.3.1 Mass Flow Rate

The nature of the graph of mass flow rate with respective time is similar to that of 30 µm model. However, the mass flow rate in this case is higher and closer to the theoretical mass flow rate. The nature of fluid flow variation is shown in Figure 7.16 and 7.17 for the

case with 3000 rpm and 3000 psi. Other trends such as change in mass flow rate with respective to speed and pressure are similar.

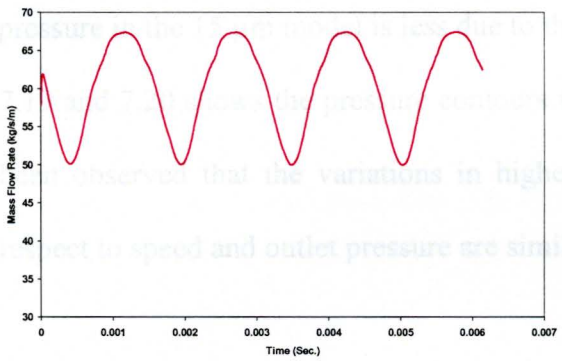

Figure 7.16 – Mass Flow Rate with Time for 2-D Simulation (for 3000 rpm & 3000 psi)

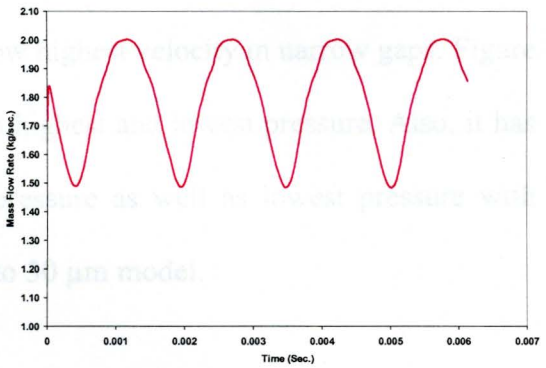

Figure 7.17 – Converted Mass Flow Rate with Time (for 3000 rpm & 3000 psi)

Table 7.3 – Mass Flow Rate at Different Speeds and Outlet Pressures.

	Speed (rpm)	Outlet Press-ure (psi)	Mass Flow Rate (kg/sec.) Max.	Min.	Ave
Case 1	3500	3500	2.34	1.75	2.04
Case 2		3000	2.34	1.79	2.07
Case 3		2500	2.35	1.83	2.09
Case 4	3000	3500	2.00	1.45	1.72
Case 5		3000	2.01	1.49	1.75
Case 6		2500	2.01	1.54	1.77
Case 7	2500	3500	1.66	1.16	1.41
Case 8		3000	1.67	1.20	1.43
Case 9		2500	1.68	1.24	1.46
Case 10	2000	3500	1.30	0.81	1.06
Case 11		3000	1.33	0.91	1.12
Case 12		2500	1.34	0.95	1.14

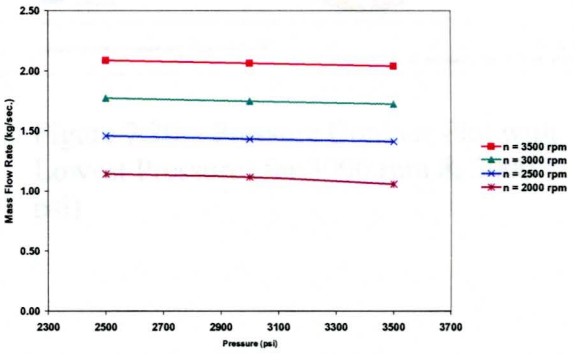

Figure 7.18 – Effect of Outlet Pressure on Mass Flow Rate at Different Speeds

7.3.2 Pressure

In the case of pressure contours in this model, it has been observed that its behavior is similar to the 30 μm model pressure contour. The highest pressure in this case is higher than the 30 μm model due to smaller gaps between the teeth. Also the drop in lowest pressure in the 15 μm model is less due to the low highest velocity in narrow gaps. Figure 7.19 and 7.20 shows the pressure contours with highest and lowest pressure. Also, it has been observed that the variations in highest pressure as well as lowest pressure with respect to speed and outlet pressure are similar to 30 μm model.

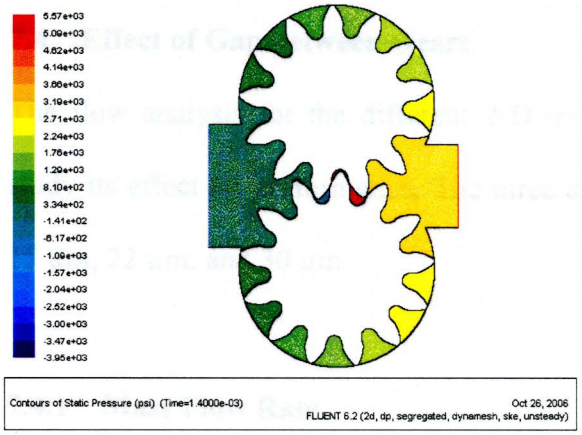

Figure 7.19 – Pressure Contour Plot with Highest Pressure (for 3000 rpm & 3000 psi)

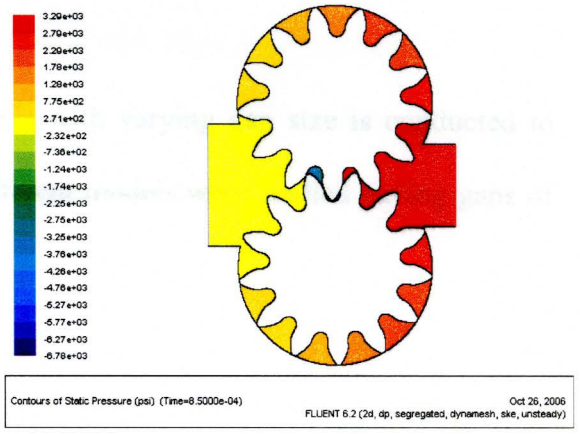

Figure 7.20 – Pressure Contour Plot with Lowest Pressure (for 3000 rpm & 3000 psi)

7.3.3 Velocity

The nature of variation in velocity with respect to speed and outlet pressure is also similar to the 30 μm model, except the highest velocity is lower. The velocity vector diagram is shown in Figure 7.21 and 7.22

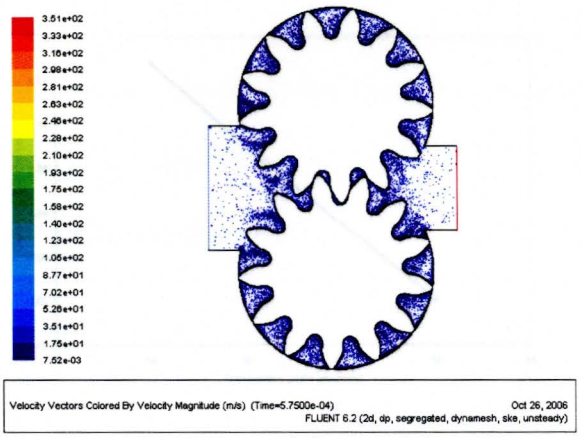

Figure 7.21 – Velocity Vector Plot (for 3000 rpm & 3000 psi)

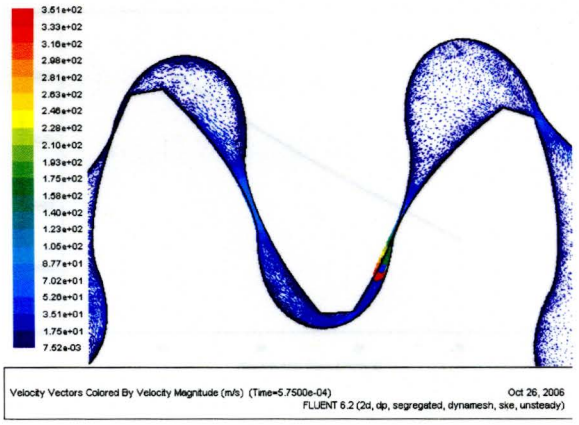

Figure 7.22 – Enlarged view of Velocity Vector Plot (for 3000 rpm & 3000 psi)

7.4 Effect of Gap between Gears

The flow analysis for the different 2-D models with varying gap size is conducted to study its effect on performance. The three different models were studied having gaps of 15 μm, 22 μm, and 30 μm.

7.4.1 Mass Flow Rate

It has been observed that the mass flow rate has been increased with finer gaps in the gears. Figure 7.23 shows how the gap affects the mass flow rate of gear pump. The theoretical mass flow rate for 3000 rpm is 2.20 kg/sec. With finer gap size the mass flow rate results approaches the theoretical value.

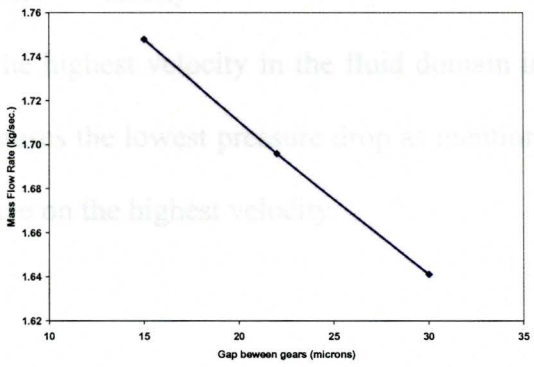

Figure 7.23 – Effect of Gap on Mass Flow Rate (for 3000 rpm & 3000 psi) (Theoretical value of mass flow rate 2.20 kg/sec.)

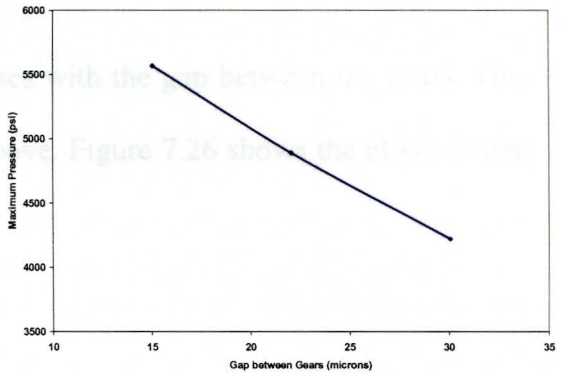

Figure 7.24 – Effect of Gap on Highest Pressure (for 3000 rpm & 3000 psi)

7.4.2 Pressure

The highest pressure in the fluid domain increases with finer gaps between the gears. This is basically due to low slip between the gaps. However the lowest pressure drops even further with increase in gap size. Figure 7.24 and 7.25 show the relation.

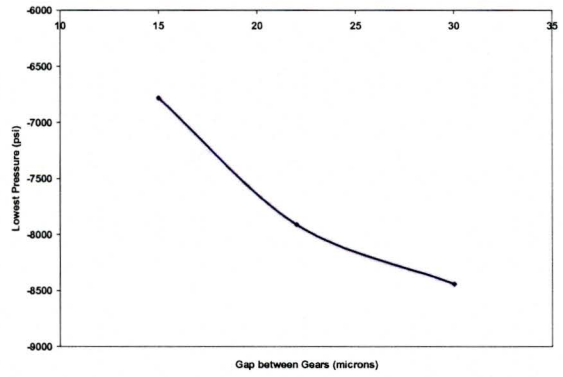

Figure 7.25 – Effect of Gap on Lowest Pressure (for 3000 rpm & 3000 psi)

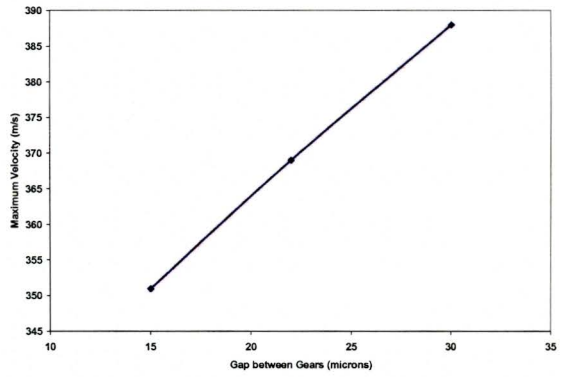

Figure 7.26 – Effect of Gap on Highest Velocity (for 3000 rpm & 3000 psi)

7.4.3 Velocity

The highest velocity in the fluid domain increases with the gap between the gears. This causes the lowest pressure drop as mentioned above. Figure 7.26 shows the effect of gap size on the highest velocity.

Chapter 8

RESULTS OF 3-D FLOW ANALYSIS

8.1 Flow Analysis in 3-D

The gear pump flow is simulated in 3-D by using a 2.5-D model of the gear pump. The supercomputer OSC Pentium 4 cluster is used for performing the analysis in FLUENT. Due to limitations of the computing resources and model run time, the analysis is performed on 2.5D model with 30 μm for only one case with 3000 rpm and 3000 psi outlet pressure. The simulation is performed with convergence criteria for solution of 1e-3 for residual and the time step of 1e-7 sec. The solution is initialized with inlet pressure of 0 psi. It is run for total 8000 time steps corresponding to 14.4 deg. of gears rotation, which is half tooth movement (27.7 deg. is one tooth movement). Figure 8.1 shows the convergence of solution at each time steps.

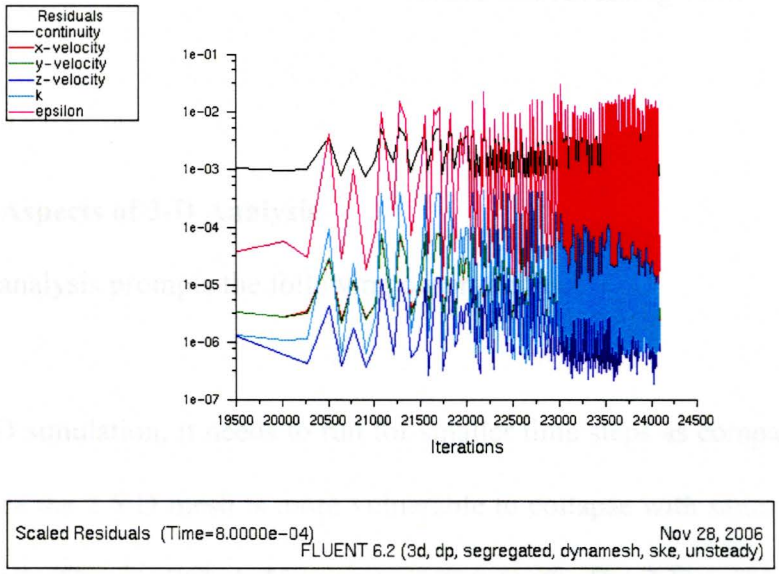

Figure 8.1 – Convergence of solution

A maximum of 32 nodes are used for the simulation due to the limited availability of nodes. The effect of the number of nodes on analysis time is investigated in this study. Figure 8.2 shows the effect of the number of computer nodes on time required for processing. The time required each iteration decreases with more number of nodes. However, no significant time reduction in processing is achieved after a certain number of nodes.

Table 8.1 – Number of Computer Nodes and Processing Time.

Number of Computer Nodes	Time per Iteration (Sec.)
24	12.29
16	15.56
12	19.78
10	24.67
8	30.63

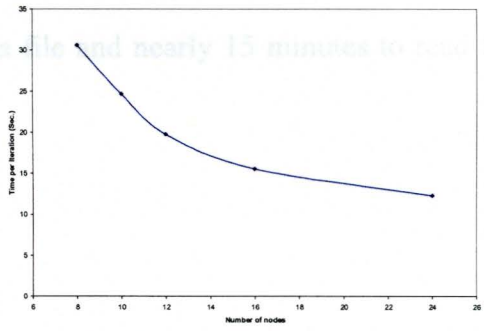

Figure 8.2 – Effect of Number of Computer Nodes on Processing Time.

8.2 Crucial Aspects of 3-D Analysis

The 3-D flow analysis prompts the following observations.

1. For the 3-D simulation, it needs to run for smaller time steps as compared to the 2-D model, since the 2.5-D mesh is more vulnerable to collapse with same time step than the 2-D mesh. This is attributed to using a prism mesh. The 2-D simulation runs with a time step of 1e-6 sec., whereas 3-D analysis requires time step of 1e-7 sec to run

without failure. This makes for an increase in the number of time steps for 3-D analysis by 10 times, requiring 200,000 time steps for one complete rotation of the gears in contrast to 20,000 for 2-D model. This makes the process of running the simulation prohibitively time consuming.

2. The 3-D simulation of the gear pump flow generates pressure contours and velocity vectors files of size nearly 1 GB each, which creates huge amount of data.
3. To capture pressure contour and velocity vector data for animation without any error, one needs to capture these plots very frequently, every 25th time step, which is also time consuming and generates unacceptably huge data sets.
4. It takes 25 minutes to save one case and data file and nearly 15 minutes to read the case and data file into the cluster.

8.3 3-D Analysis Results

The various parameters such as mass flow rate, pressure contours, velocity vectors in the fluid domain, and pressure at particular point are monitored while running the simulation. The results from the 3-D analysis are compared with 2-D results in Table 8.2.

Table 8.2 – Comparison of 2-D and 3-D Analysis Results

	3-D Analysis	2-D Analysis
Lowest Mass Flow Rate (kg/sec.)	1.434	1.38
Highest Pressure (psi)	3210	3180
Lowest Pressure (psi)	- 8480	- 7550
Highest Velocity (m/s)	388	365

8.3.1 Mass Flow Rate

The lowest mass flow rate in case of 3-D analysis is 1.434 kg/sec, whereas it is 1.38 kg/sec in case of 2-D analysis. The average mass flow rate for 2-D model is 1.68 kg/sec. Hence, in proportion to the lowest flow rate of 1.434 kg/sec in 3-D, the average mass flow rate shall be 1.74 kg/sec. As compared to the theoretical flow rate of 2.20 kg/sec for 3000 rpm, the mass flow rate from 3-D analysis is closer than the mass flow rate in 2-D. This shows the improvement in results in comparison to 2-D results. Figure 8.3 and 8.4 show the mass flow rate with respect to time of the 3-D analysis before and after conversion to a full scale model.

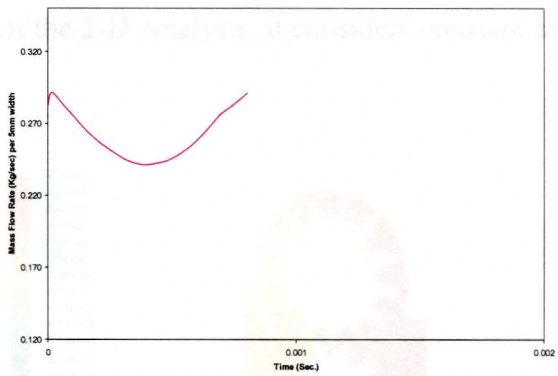

Figure 8.3 – Mass Flow Rate with Time from for 3-D simulation (for 3000 rpm & 3000 psi)

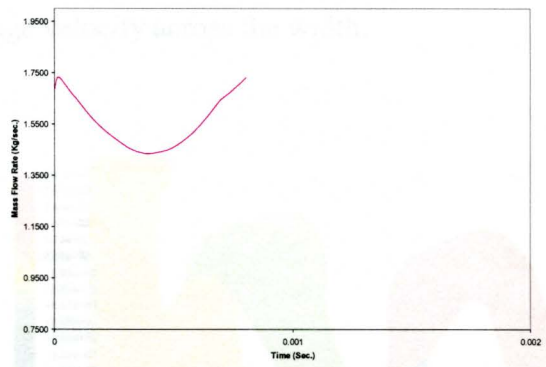

Figure 8.4 – Converted Mass Flow Rate with Time (for 3000 rpm & 3000 psi)

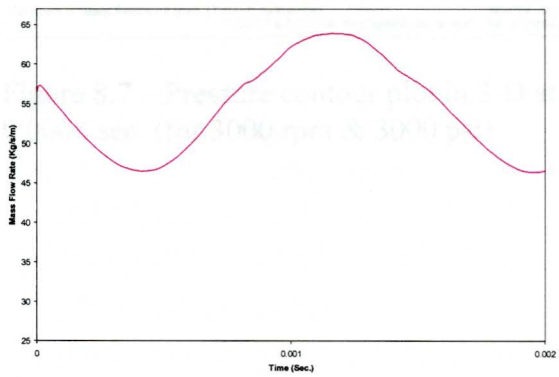

Figure 8.5 – Mass Flow Rate with Time from 2-D (for 3000 rpm & 3000 psi)

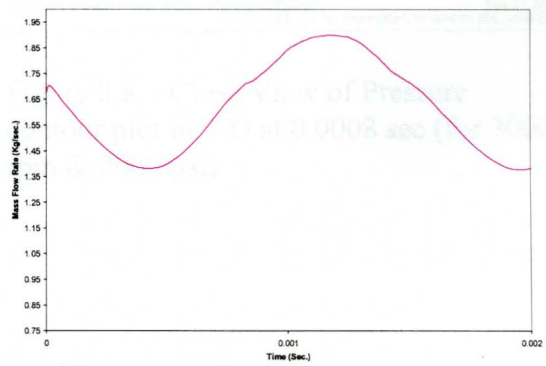

Figure 8.6 – Converted Mass Flow Rate with Time (for 3000 rpm & 3000 psi)

8.3.2 Pressure

In the case of the 3-D Analysis, the highest pressure at 0.0008 sec. (i.e. 8000th time step of 1e-7 sec.) is 3210 psi, whereas the lowest pressure is -8480 psi. Comparatively the same time of 0.0008 sec. (800th time step of 1e-6 sec. for 2-D Analysis), the highest pressure in the case of the 2-D Analysis is 3180 psi and lowest pressure is -7550 psi. Figure 8.7, 8.8, 8.8, and 8.10 show the pressure contours for both the cases. It may be noted that the highest pressure in the 3-D model is larger, which shows that it will be closer to actual value. However, the lowest pressure at this timestep decreases. It may be due to an increase in velocity at the center of fluid flow due to the effects of the sidewall in the case of the 3-D model which may be observed in Figure 8.11. Normally in the case of the 2-D Analysis, it considers constant average velocity across the width.

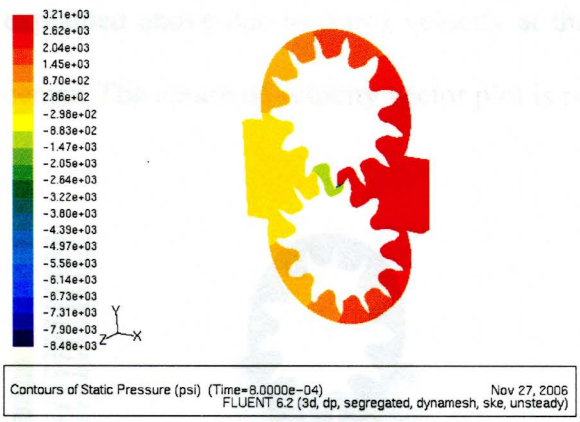

Figure 8.7 – Pressure contour plot in 3-D at 0.0008 sec. (for 3000 rpm & 3000 psi)

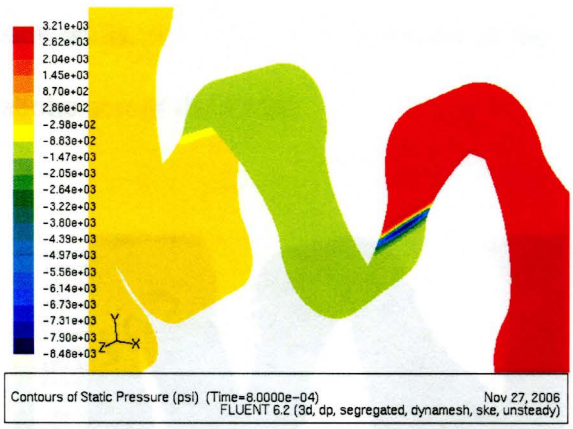

Figure 8.8 – Close View of Pressure contour plot in 3-D at 0.0008 sec (for 3000 rpm & 3000 psi)

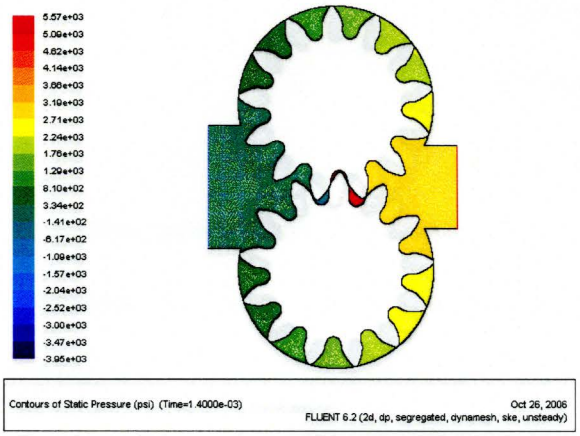

Figure 8.9 – Pressure Contour Plot in 2-D at 0.0008 sec. (for 3000 rpm & 3000 psi)

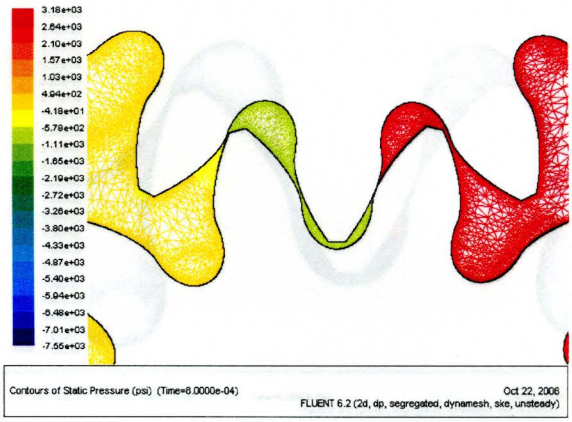

Figure 8.10 – Close View of Pressure Contour Plot in 2-D at 0.0008 sec (for 3000 rpm & 3000 psi)

8.3.3 Velocity

The velocity vector plots are also compared for both models. The highest velocity in case of 3-D analysis is 388 m/s at the same time 0.0008 sec., whereas for 2-D it is 365 m/s. As explained above due to 0 m/s velocity at the side walls, the velocity is increased at the center. The nature of velocity vector plot is parabolic across the width.

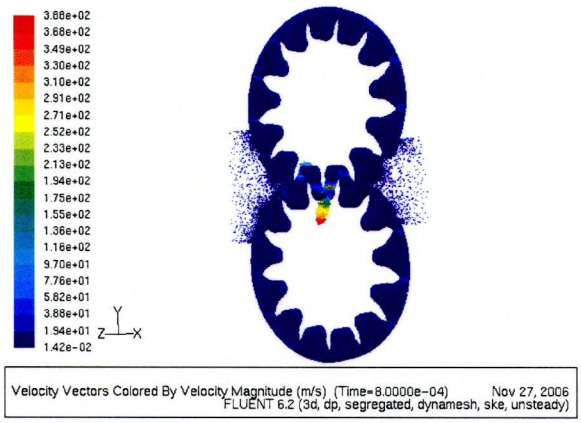

Figure 8.11 – Velocity Vector plot in 3-D at 0.0008 sec. (for 3000 rpm & 3000 psi)

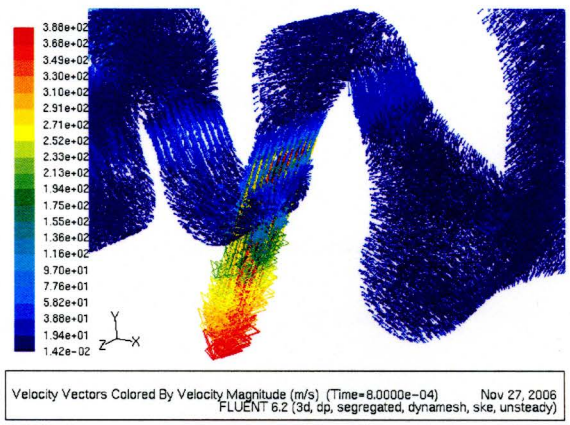

Figure 8.12 – Close View of Velocity Vector Pressure Contour Plot in 3-D at 0.0008 sec (for 3000 rpm & 3000 psi)

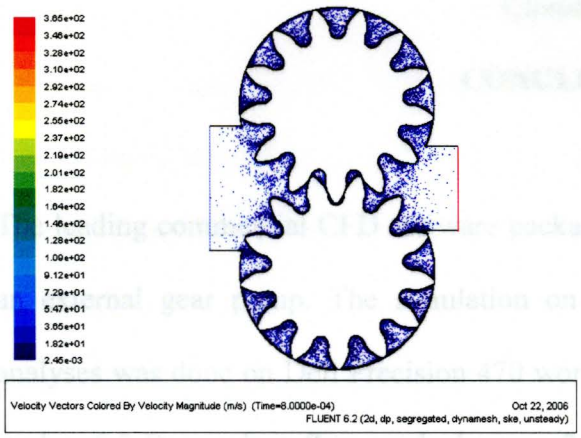

Figure 8.13 – Velocity Vector Plot in 2-D at 0.0008 sec. (for 3000 rpm & 3000 psi)

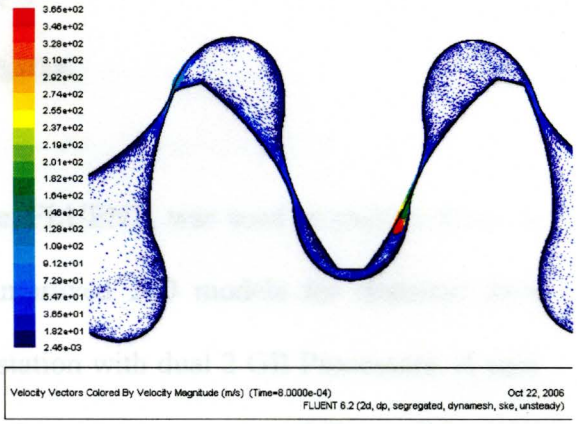

Figure 8.14 – Enlarged View of Velocity Vector plot in 2-D at 0.0008 sec (for 3000 rpm & 3000 psi)

Chapter 9

CONCLUSION

The leading commercial CFD software package, FLUENT, was used to analyze flows in an external gear pump. The simulation on improved 2-D models for transient flow analyses was done on Dell Precision 470 workstation with dual 2 GB Processors. A case study of 3-D transient flow analysis on a 2.5-D model was successfully conducted in remote parallel processing on a cluster of Pentium 4 computers at the Ohio Supercomputer Center in Columbus, Ohio.

The 2-D simulation included 25 case studies for three different models with gap sizes of 15 μm, 22 μm, and 30 μm. Three different outlet pressures of 2500 psi, 3000 psi and 3500 psi were applied for each model at four different speeds of 2000 rpm, 2500 rpm, 3000 rpm, and 3500 rpm. The results showed that the size of the gap is the most pertinent parameter affecting the accuracy of the pump capacity. However, the reduction of gap increases the instability and complexity of computing so much that a careful examination of all these factors is warranted in making acceptable computational models.

The 3-D simulation was performed on 32 nodes of OSC Pentium computer for a single case study of 3000 rpm and 3000 psi outlet pressure due to limitation of computing time, storage space and other computational resources. The results of the 3-D simulation show that mass flow rates are closer to the theoretical values than those of 2-D analysis. They also show that the highest pressure in fluid zone is higher than 2-D results.

Although it seems clear that 3-D simulation gives superior results over 2-D simulation, it is inadvisable to make a general conclusion on the accuracy of the 3-D simulation at this time due to insufficient data. It may be noted that many computing problems, such as the generation of extremely large data sets, extremely long execution time, and the complexity of modeling and interfacing, present users with serious questions regarding the economic viability of the 3-D simulation in remote parallel processing. However, as more experience in modeling is accumulated, it may be reasonable to assume that these problems can be solved or substantially improved in the near future to make 3-D simulation an essential tool for engineering design and analysis. This thesis, through a process of trails and errors, found a way to successfully conduct remote parallel processing and documented detailed information regarding the procedure. The information contained herein may open the door for those who would like to take on the challenge of 3-D flow simulation of gear pumps.

BIBLIOGRAPHY

1. E. M. Yudin - ***Gear Pumps Principal Parameters and Their Calculation***, National Lending Library for Science and Technology, England, 1967.
2. H. W. Shawn Kim – ***Computational Fluid Dynamics Analysis of Gear Pumps,*** Department of Mechanical Engineering, Youngstown State University, Youngstown.
3. Julia Novak and Kimberly Gates – ***The World of Pumps***, Department of Civil Engineering, Virginia Tech University, Blacksburg, VA, 2002.
4. Massimo Borghi – ***Fluid Power Research Group,*** DIMEC, Department of Mechanical and Civil Engineering, University of Modena and Reggio Emilia, Modena, Italy.
5. Noah D Manring and Suresh B Kassaragadda – ***The Theoretical Flow Ripple of a External Gear Pump,*** Department of Mechanical and Aerospace Engineering, University of Missouri-Columbia, 2003.
6. Jaakko Myllykylä - ***Semi-Impirircal Model for The Suction Capability of an External Gear Pump IHA,*** Tampere University of Technology, Tampere, Finland, 1999.
7. FLUENT Solution – ***Gear Pump Solution,*** Example X219, FLUENT Inc., USA.
8. Gamez-Montero and Condina Macià – ***Fluid Dynamic Behavior of an Internal Rotary Pump Generated by Trochoidal Profiles,*** Department of Fluid Mechanics, Technical University of Catalonia, Barcelona, Spain.

9. Santosh Kini, Nick Mapara, Richard Thoms, Peter Chang, and Michal Nemec – ***Numerical Simulation of Cover Plate Deflection in the Gerotor Pump,*** ESI-CFD Inc., Huntsville, USA & STT Technologies, Inc., Concord, USA, 2004

10. Yogentdra Panta – ***Numerical Flow Analysis of Gear Pump,*** Department of Mechanical Engineering, Youngstown State University, Youngstown, USA, 2004

11. Jyotindra Killedar – ***CFD Analysis of Gear Pump,*** Department of Mechanical Engineering, Youngstown State University, Youngstown, USA, 2005

12. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar - ***Parallel Computing,*** Second Edition, Addison-Wesley, New York, 2003

13. FLUENT Inc. – ***Introduction to Parallel Processing***, chapter 32, User's Guide

14. FLUENT Inc. – ***Partitioning the Grids***, chapter 32, User's Guide

15. Robert Fox and Alan McDonald - ***Introduction of Fluid Mechanics***, Third Edition, Jon Willey & Sons, New York,1985.

16. Robert Granger – ***Fluid Mechanics,*** Holt Rinehart and Winston, New York, 1985.

17. James Sullivan – ***Fluid Power,*** Reston Publishing Company, Virginia

18. ***International Journal of Fluid Power,*** published by FPNI and TuTech, Vol. 3, No. 1, April 2002.

19. Luc Machiels – ***Simulation and Theory of Randomly Forced Turbulence,*** Thése No. 1724, Présetée Au Département De Génie Mécanique, École Polytechnique Fédérale De Lausanne, Switzerland, 1997.

20. Spyros Gavrilakis – ***Numerical Simulation of Low-Reynolds-Number Flow through a Straight Square Duct, Turbulence,*** IMHEF-DME, École Polytechnique Fédérale De Lausanne, Switzerland, 1997.

21. John C Tannehill, Dale A Anderson, and Richard H Pletcher - ***Computational Fluid Mechanics and Heat Transfer***, Second Edition, Taylor & Francis, Washington, DC, 1997.

22. Charles Bookman – ***Linux Clustering***, New Riders, New York, 2003.

APPENDIX I

User Defined Function (UDF) for PC

"g3000.c"

```
/*UDF starts*/
#include "udf.h"
#include "dynamesh_tools.h"
DEFINE_CG_MOTION(clock_wise, dt, vel, omega, time, dtime)
{
        NV_S (vel, =, 0.0);
        NV_S (omega, =, 0.0);
        omega[0] = 0;
        omega[1] = 0;
        omega[2] = -314;
Message("\nThis is Clockwise Gear\n");
Message("\nCG_Omega for clock_wise: %g, %g, %g\n", omega[0], omega[1],
          omega[2]);
Message("\nCG Position for clock_wise: %g, %g, %g\n", NV_LIST(DT_CG(dt)));
Message("\nCG Orientation for clock_wise: %g, %g, %g\n",
          NV_LIST(DT_THETA(dt)));
}
DEFINE_CG_MOTION(anticlock_wise, dt, vel, omega, time, dtime)
{
        NV_S (vel, =, 0.0);
        NV_S (omega, =, 0.0);
        omega[0] = 0;
        omega[1] = 0;
        omega[2] = 314;
 Message("\nThis is Anti Clockwise Gear\n");
 Message("\nCG_Omega for anticlock_wise: %g, %g, %g\n",  omega[0], omega[1],
          omega[2]);
 Message("\nCG Position for anticlock_wise: %g, %g, %g\n", NV_LIST(DT_CG(dt)));
 Message("\nCG Orientation for anticlock_wise: %g, %g, %g\n",
          NV_LIST(DT_THETA(dt)));
}
/* UDF ends*/
```

APPENDIX II

User Defined Function (UDF) for Parallel Processing

"parallel_g3000_orig.c"

```
/*UDF starts*/
#include "udf.h"
#include "dynamesh_tools.h"
DEFINE_CG_MOTION(clock_wise, dt, vel, omega, time, dtime)
{
        NV_S (vel, =, 0.0);
        NV_S (omega, =, 0.0);
        omega[0] = 0;
        omega[1] = 0;
        omega[2] = -315;
 #if !RP_NODE
 Message("\nThis is Clockwise Gear\n");
 Message("\nCG_Omega for clock_wise: %g, %g, %g\n", omega[0],omega[1],
            omega[2]);
 Message("\nCG Position for clock_wise: %g, %g, %g\n",
            NV_LIST(DT_CG(dt)));
 Message("\nCG Orientation for clock_wise: %g, %g, %g\n",
        NV_LIST(DT_THETA(dt)));
 #endif
}

DEFINE_CG_MOTION(anticlock_wise, dt, vel, omega, time, dtime)
{
         NV_S (vel, =, 0.0);
        NV_S (omega, =, 0.0);
        omega[0] = 0;
        omega[1] = 0;
        omega[2] = 315;
 #if !RP_NODE
 Message("\nThis is Anti Clockwise Gear\n");
 Message("\nCG_Omega for anticlock_wise: %g, %g, %g\n",  omega[0],
        omega[1], omega[2]);
 Message("\nCG Position for anticlock_wise: %g, %g, %g\n",
        NV_LIST(DT_CG(dt)));
 Message("\nCG Orientation for anticlock_wise: %g, %g, %g\n",
        NV_LIST(DT_THETA(dt)));
 #endif
}
/* UDF ends*/
```