This is to certify that the
thesis entitled

APPLICATION AND DEVELOPMENT OF NUMERICAL METHOD FOR MOVING BOUNDARY AND COMPLEX GEOMETRY PROBLEMS

presented by

Yanbing Li

has been accepted towards fulfillment
of the requirements for the

M. S. degree in Department of Mechanical Engineering

Major Professor's Signature

12/3/02

Date

THE APPLICATION AND DEVELOPMENT OF NUMERICAL METHOD FOR MOVING BOUNDARY AND COMPLEX GEOMETRY PROBLEMS

By

Yanbing Li

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

2002

ABSTRACT

THE APPLICATION AND DEVELOPMENT OF NUMERICAL METHOD FOR MOVING BOUNDARY AND COMPLEX GEOMETRY PROBLEMS

By

Yanbing Li

Various methods have recently been introduced to alleviate the difficulties associated with simulating moving boundary problems. Among them, the fictitious domain method is explored and methods to mitigate the errors associated with this technique are presented in this work. A summary of moving mesh strategies currently in use in commercial computational fluid dynamics software is first provided. Significant issues associated with moving grid techniques such as the interpolation errors, the human cost for mesh generation, and the quality of the resulting mesh are discussed by solving, with the help of a commercial code, a variety of fluid dynamics problems associated with in-cylinder flows of internal combustion engines. The Lagrange multiplier/fictitious domain method is then discussed by solving simple heat transfer problems. The fictitious domain method is based on the use of Lagrange multipliers that do not match with an underlying mesh. Such an approach introduces errors on the adjacent nodes and a parametric study of various factors affecting the quality of the results is performed. Two approaches for reducing the errors are proposed. A first method uses modified boundary conditions to reduce the errors on the adjacent nodes. The method is based on a predictor/corrector scheme and is called the "fictitious constraint" method. A second method consists of simply modifying the shape of the boundary by matching the complex boundary with the underlying mesh. Improvements in the solutions by using these techniques are illustrated with the help of one-dimensional and two-dimensional problems.

To my wife, my parents, and my grandmother in-law

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Roman Symbols

P:	pressure, *Psi*
u, v, w:	x, y, z component of velocity, m/s
g_x, g_y:	x, y component of body force, N
k :	turbulent kinetic energy, m^2/s^2
$C_{\varepsilon 1}, C_{\varepsilon 2}, C_{\varepsilon 3}$:	constants used in the ε balance equation
C_μ :	constant used in mixing length turbulence model
u_i :	velocity component , m/s
$\overline{u}_i$:	mean velocity component, m/s
u_i' :	fluctuation from the mean veiocity component $\overline{u}_i$, m/s
$\overline{p}$:	mean pressure, *Psi*
p' :	fluctuation from the mean pressure, *Psi*
P_{static}:	static pressure, *Psi*
T :	temperature, K
T_{static} :	static temperature, K
I :	turbulence intensity,

U: mean velocity magnitude, m/s

$\tilde{T}$: weak solution of the problem over the entire domain

$\widetilde{T_h}$: weak solution of the problem over finite element

T_h: finite dimensional spaces for Ω

E: fictitious domain element

E^{γ}: embedded boundary element

Greek Symbols

ρ : fluid density, kg/m^3

μ : fluid viscosity, $Pa \cdot s$

μ_t : turbulent fluid viscosity, $Pa \cdot s$

ε : turbulent dissipation rate, m^2/s^3

σ_k : turbulent Prandtl number for k

$\overline{\rho}$: mean fluid density, kg/m^3

ρ' : fluctuation from the mean fluid density, kg/m^3

Ω, ω : spatial domain

Γ, γ : boundaries of spatial domain

λ: weak solution of the problem on the boundary

λ_h: weak solution of the problem over the boundary elements

Λ_h: finite dimensional spaces for γ

$\Im_h$: discretization of the fictitious domain Ω

$\Im_h^\gamma$: discretization of the embedded boundary γ

Abbreviations

ATDC: after top dead center

BDC: bottom dead center

BFM: body-fitted mesh method

CA: crank angle

CFD: computational fluid dynamics

GCL: geometric conservation law

LM/FDM: Lagrange multiplier/fictitious domain method

TDC: top dead center

Chapter 1

Introduction

Computational fluid dynamics (CFD) still encounters difficulties when dealing with geometrically complex and dynamically complex problems [4]. Among them, moving boundary problems have been arising as a continuing challenging issue in a variety of important engineering application areas [2]. Typical examples are heat transfer problems and chemical reaction problems where there may be significant changes of material properties, physical-chemical properties or flow features. In these situations, the interfaces move under the interface of the flow field by internal or external forces and in turn affect the behavior of the flow, and may subject to instabilities under certain conditions [2,3]; the prediction of the mechanism in such moving boundary systems is very important but very difficult to analyze "numerically" due to existing of the complicating mechanisms around the interface region [2].

Numerous techniques exist for tracking moving boundaries [2,3]. Among them, Lagrangian methods, which are based on a moving grid technique, are widely used in commercial soft wares. The use of a body-fitted grid system that conforms to the actual geometry of the problem and is updated (redistribution and refinement) allows to track the interface (moving boundary) explicitly. The boundary condition can be applied at the exact location of the interface at each time step. Application of moving grid techniques to problems with complex geometry and extensive interfacial activities (e.g. large deformation and topological change) requires a strong remeshing; this corresponds to an increase of the human cost and thus limits its applications (such as design evaluations

with various geometry configurations). The formulation of a moving grid technique is reviewed and related issues such as geometric conservation law of the coordinate mapping and solution interpolation are discussed in chapter 2.

In the context of transient IC engine simulations, moving grid techniques are widely adopted in today's most commercial (FIRE, FLUENT, Star-CD) and no-commercial (Kiva) codes. Numerical simulations with such CFD platforms are performed to study the in-cylinder flow activities and help the design evaluation [11, 13, 18-26]. In chapter 3, a summary of the moving mesh strategies is provided; Simulations result using FIRE for an Generic IC engine intake and compression stroke process are then presented. Experimental data are also employed to verify the results. Finally the major issues that limit the application of moving gird technique are discussed based on the work of the in-cylinder flow simulations for Generic IC engine.

Fictitious domain methods, which can be term as one of the "combined Eulerian-Lagrangian" methods, has been proposed as a way to reduce the complexity of meshing for moving boundary problems. In this method, the original complex geometry associated to the problem at hand is embedded into an extended simpler fixed domain (the fictitious domain), the interface is tracked as an independent entities (either same order of dimension or low order of dimension of the extended domain), the computations are performed on the fixed fictitious domain mesh whose topology is independent of that of the interface and the original boundary conditions on the interface are enforced in the extended domain with the help of Lagrange multiplier [27-31] or by using a penalty method [32]. Therefore the solution is transformed to a one-domain approach and the

meshing task is simplified into two independent geometries, the discretizations then can be done separately and need not to be conform to each other. This obviates the need for repetitive remeshing.

The use of Lagrange multiplier to enforce the boundary conditions however may introduce errors [39]: the multipliers do not necessarily enforce the nodal boundary conditions at nodal locations and may fall within elements. This can introduce significant errors on the adjacent nodes when the extended solution involves a discontinuous derivative across the embedded boundary, e.g. enforcing a Dirichlet boundary condition within the extended domain. The errors are due to an inappropriate approximation of the local gradient at the location of a multiplier. A one-dimensional model problem is first used to study possible improvements. To reduce the errors, it is first possible to refine the mesh. It is observed that a uniform reduction in the element size changes the relative location of the Lagrange multiplier within an element and the errors are found to oscillate as the mesh is refined uniformly for a given extended domain. The errors are also functions of the relative locations of the Lagrange multipliers within the elements for a fixed element size. For fixed relative Lagrange multiplier position, the Lagrange multiplier/fictitious domain methods (LM/FDM) result in a lower-order convergence rate in L_2 norm compared to a solution based on a body conforming mesh. The magnitudes of the errors are related to the jump of the local solution gradient and the relative location of the multiplier within the corresponding element. The errors are minimized as the jump of the local gradients disappears. This is illustrated in chapter 4.

Finally in chapter 4, fictitious constraint method and shape reconstruction method are proposed to mitigate the errors: With fictitious constraint methods, instead of satisfying the original embedded Dirichlet boundary condition, a two-step predictor-corrector procedure is adopted to adjust the local boundary conditions to values that reduce significantly the errors. These local values are called the fictitious constraint and are obtained based on the computed solution obtained from the standard Lagrange multiplier/fictitious domain method. With the shape reconstruction methods, the boundary conditions are adjusted to be applied on the nearest adjacent nodes (edges) to the Lagrange multiplier, the LM/FDM computations are then performed with the reconstructed boundaries of the problem. Computation of the errors shows that the solutions obtained with these methods are less sensitive to the jump of the local gradients and the relative location of the Lagrange multipliers within the elements than the solutions of standard Lagrange multiplier/Fictitious domain method. A higher-order convergence ratio for L_2 norm can be achieved when the relative Lagrange multiplier position is fixed. This is tested by solving simple one- and two-dimensional problems.

Chapter 2

An Introduction to Moving Grid Technique

In moving grid methods, the motion of the moving boundary is explicitly known at each instant. a boundary fitted mesh (as shown in figure 2.1) is generated at each time step to explicitly track the interface's evolution. The corresponding grid motion can then be incorporated into the numerical scheme via the geometric conservation law [10]. When the motion of the interfaces begin to distort the mesh, regridding (grid redistribution/refinement) is needed and consequently the solution obtained on the old mesh needs to be interpolated onto the new mesh.

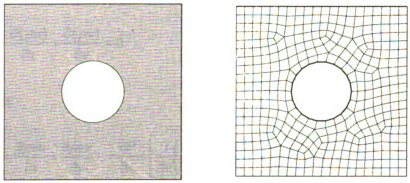


Figure 1 Discretization of the original problem (left) using a body fitted mesh (right)

There are three important issues related to moving grid technique. First the governing equations need to be transformed into a body-fitted coordinate system. Second, the geometric conservation law needs be satisfied to derive an equivalent differential relation that can take the mesh movement into account. Finally, solution interpolations are needed to transfer the data from the old grid to the new grid during a mesh regridding.

In this chapter, the fundamental fluid dynamics conservation laws are first presented for a Newtonian fluid. We then discuss these three issues: coordinate transformation using generalized curvilinear coordinates (Thompson et al. 1985), the geometric conservation law and interpolations of the solution.

To simplify the presentation, we will use 2D cased to highlight the issues involved, an extension to 3D geometrical geometry can follow the same concept without qualitative modifications.

2.1 The Governing Equations

The governing equations in Cartesian coordinates for two-dimensional, compressible flow can be written in dimensional form as:

Continuity:

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}=0 \tag{2.1}$$

x-momentum:

$$\frac{\partial \rho u}{\partial t}+\frac{\partial \rho u u}{\partial x}+\frac{\partial \rho u v}{\partial y}=-\frac{\partial P}{\partial x}+\left[\frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right)\right]+\rho g_x \tag{2.2}$$

y-momentum:

$$\frac{\partial \rho v}{\partial t}+\frac{\partial \rho u v}{\partial x}+\frac{\partial \rho v v}{\partial y}=-\frac{\partial P}{\partial y}+\left[\frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right)\right]+\rho g_y \tag{2.3}$$

For simplicity, we did not consider the source terms in the above equation.

2.2 Transformed Governing Equations in a Body-Fitted Coordinate System

Following W. Shyy's work in [2,3], the coordinate transformations are discussed below:

In order to transform the governing equation to a moving, body-fitted coordinate system using generalized curvilinear coordinates, we introduce a time-dependent invertible mapping transformation (for a two-dimensional space case):

$$\begin{aligned} x &= x(\xi,\eta,\tau) \\ y &= y(\xi,\eta,\tau) \\ t &= \tau \end{aligned} \tag{2.4}$$

The time varying irregular physical domain is then mapped to a fixed uniform computational space. The flow equations are then recast in a body-fitted curvilinear coordinate system (ξ,η) and are:

$$\frac{\partial}{\partial t}(J\rho)+\frac{\partial(\rho U)}{\partial \xi}+\frac{\partial(\rho V)}{\partial \eta}=0 \tag{2.5}$$

and

$$\begin{aligned} \frac{\partial}{\partial t}(J\rho u)+\frac{\partial(\rho U u)}{\partial \xi}+\frac{\partial(\rho V u)}{\partial \eta} = &-\left\{y_\eta \frac{\partial P}{\partial \xi}-y_\xi \frac{\partial P}{\partial \eta}\right\}+\frac{\partial}{\partial \xi}\left[\frac{\mu}{J}\left(q_1 u_\xi - q_2 u_\eta\right)\right]+ \\ &\frac{\partial}{\partial \eta}\left[\frac{\mu}{J}\left(q_3 u_\eta - q_2 u_\xi\right)\right]+\rho g_x \bullet J \end{aligned} \tag{2.6}$$

$$\begin{aligned} \frac{\partial}{\partial t}(J\rho v)+\frac{\partial(\rho U v)}{\partial \xi}+\frac{\partial(\rho V v)}{\partial \eta} = &-\left\{x_\xi \frac{\partial P}{\partial \eta}-x_\eta \frac{\partial P}{\partial \xi}\right\}+\frac{\partial}{\partial \xi}\left[\frac{\mu}{J}\left(q_1 v_\xi - q_2 v_\eta\right)\right]+ \\ &\frac{\partial}{\partial \eta}\left[\frac{\mu}{J}\left(q_3 v_\eta - q_2 v_\xi\right)\right]+\rho g_y \bullet J \end{aligned} \tag{2.7}$$

where the subscripts ξ and η denote $\partial/\partial\xi$ and $\partial/\partial\eta$, respectively; J is the Jacobian of the coordinate transformation $J = x_\xi y_\eta - x_\eta y_\xi$ and represents the volume element in the transformed coordinate.

and U and V are the contravariant velocity components

$$\begin{aligned} U &= (u-\dot{x})y_\eta - (v-\dot{y})x_\eta \\ V &= (v-\dot{y})x_\xi - (u-\dot{x})y_\xi \end{aligned} \tag{2.8}$$

where $x_\xi, x_\eta, y_\xi, y_\eta$ are the metrics of the coordinate transformation, and $\dot{x}, \dot{y}$ are the Cartesian components of the grid velocity vector defined as:

$$\dot{x} = \frac{x^n - x^{n-1}}{\Delta t}, \quad \dot{y} = \frac{y^n - y^{n-1}}{\Delta t} \tag{2.9}$$

where the superscripts (n) and $(n-1)$ denote the current time step and the previous time step, respectively. The metrics q_1, q_2, q_3 are defined as:

$$\begin{aligned} q_1 &= x_\eta^2 + y_\eta^2 \\ q_2 &= x_\xi x_\eta + y_\xi y_\eta \\ q_3 &= x_\xi^2 + y_\xi^2 \end{aligned} \tag{2.10}$$

The contravariant velocity component and Cartesian velocity components at the boundary are then computed to enforce mass conservation; and the kinematics condition can be enforced at the interface (moving boundaries) via the conformed curvilinear coordinate system.

After discretization, the physical domain Ω then is covered by a grid consisting of a fixed number of nodal points distributed over Ω and along its boundary.

2.3 The Geometric Conservation Law (GCL)

For moving boundary problems, part of the boundaries of the physical domain moves in time, the boundary-conforming character of the mapping transformation 2.4 implies a corresponding motion of the grid points in physical domain. Ignoring this grid motion can introduce errors in the computational flow fields with finite-difference method [5]. This issue can be addressed via an auxiliary equation derived via the concept of "geometric conservation law" (GCL) as discussed below:

Let A_e be an arbitrary, fixed element in the computational domain Ω_e enclosed by a smooth boundary ∂A_e, and let $A(t)=\left\{\vec{x}\middle|\vec{x}=\vec{x}(\vec{\xi},t)\forall\vec{\xi}\in A_e\right\}$ be the corresponding element in the physical domain Ω under the time-dependent coordinate transformation $\vec{x}=\vec{x}(\vec{\xi},t)$, then the change in volume of $A(t)$ equals the total flux through the surface $\partial A(t)$, which can be expressed as:

$$\frac{d}{dt}\int_{A(t)}d\vec{x}=\int_{\partial A(t)}\overrightarrow{x_t}\bullet d\partial A(t) \tag{2.11}$$

where $\overrightarrow{x_t}$ is the element velocity).This is the integral form of GCL [5,6],satisfying the differential GCL scheme that governs the special volume element under an arbitrary mapping will eliminate numerical oscillations and instabilities for solutions on moving grid [5].

In the differential scheme, grid motion and geometric conservation are handled in a natural way through the contravariant velocities and the Jacobian evaluations [2,3], A Jacobian transport equation is derived by considering a uniform density and velocity field,under a time dependent coordinate transformation. The following identity derived from the mass continuity equation results:

$$\frac{\partial J}{\partial t}+\frac{\partial}{\partial\xi}\left(-\dot{x}y_\eta+\dot{y}x_\eta\right)+\frac{\partial}{\partial\eta}\left(-\dot{y}x_\xi+\dot{x}y_\xi\right)=0 \tag{2.12}$$

Integrating the above equations using the same time integration scheme over the same control volume used for mass conservation leads to the following equation:

$$\frac{\left(J^n-J^{n-1}\right)}{\Delta t}+\left(-\dot{x}y_\eta+\dot{y}x_\eta\right)_e-\left(-\dot{x}y_\eta+\dot{y}x_\eta\right)_w+\left(-\dot{y}x_\xi+\dot{x}y_\xi\right)_n-\left(-\dot{y}x_\xi+\dot{x}y_\xi\right)_s=0 \tag{2.13}$$

This formulation ensures that the Jacobian is updated in every computational mesh according to the time dependent grid movements to guarantee geometric conservation in the discrete form of the conservation laws.

An alternative approach is using finite-volume method to interpret the metric and Jacobian, which establishes a direct physical relationship as following [9]:

If $\Delta\xi = \Delta\eta = 1$, then:

$$J = \begin{cases} area \quad of \quad element \quad in \ 2-D \\ volume \quad of \quad element \quad in \ 3-D \end{cases} \tag{2-14}$$

and

$$\begin{aligned} J\nabla\xi &= J\xi_x\vec{i} + J\xi_x\vec{j} + J\xi_x\vec{k} \\ &= area \ vector \ of \ the \ element \ face(3-D) \end{aligned} \tag{2-15}$$

This implies that with finite volume method, geometric conservation law is automatically satisfied.

2.4 Solution Interpolations

Moving grid techniques require grid rearrangement (regridding) when the boundary movement causes the grid to be skewed and unevenly distributed. At these situations, solution interpolations are necessary to transfer the data from the old grid to the new grid. The basis strategy for solution transfer is search and interpolate [10]: Each new grid is first located within an element of the old mesh; the solution value is then interpolated locally using element basis functions and element nodal values from the old grid.

For example, Assuming that the element e that containing node p with coordinates (x_p, y_p) has been identified, then the variable value u at p can be interpolated using the

nodal values and element basis functions in local reference element coordinates $(\xi,\eta)\in\widehat{\Omega}$:

$$u\left(x_p,y_p\right)=\sum_{j=1}^{N_e}u_j^e\widehat{\psi}_j\left(\xi_p,\eta_p\right) \tag{2.16}$$

where N_e is the number of nodes for element e, the local reference coordinates $\left(\xi_p,\eta_p\right)$ can be determined using the global value (x_p,y_p), the element basis function $\widehat{\psi}_j$ and nodal values (x_j^e,y_j^e) by the solving the following equation:

$$\begin{aligned} x_p-\sum_{j=1}^{N_e}x_j^e\widehat{\psi}_j\left(\xi_p,\eta_p\right)&=0\\ y_p-\sum_{j=1}^{N_e}y_j^e\widehat{\psi}_j\left(\xi_p,\eta_p\right)&=0 \end{aligned} \tag{2.17}$$

Chaper 3

The Application of Moving Grid Technique to the In-Cylinder Flow Simulation

3.1 In-Cylinder Flow Simulations of Internal Combustion Engine

Gas flow in a cylinder of an internal combustion engine has a profound influence on the performance of the engine [17,18,19]: the flow into the cylinder through the inlet valve or valves forms a impinging jet, and establishes organized motions in the cylinder (swirl - about the cylinder axis- and tumble-orthogonal to the cylinder axis). These in-cylinder flow characteristics are key issues to ensure stratification requirement (efficient convective transport of fuel to the plug; high turbulence level to initiate combustion; a stable and reproducible generated motion) and fundamental considerations for the exhaust emissions of an IC engine. To design an environmentally friendly (low emission) IC engine, which would also operate smoothly and produce high power with low fuel consumption, an improved understanding of these thermo and fluid dynamics of the in-cylinder process is thus very important.

For decades, the investigation of the in-cylinder flow patterns for IC engines have been achieved mainly by traditional methods based on extensive experiments: Engineers developed new combustion systems by making variations in previously successful configurations. Given the IC engine's high state of refinement and the physical complexity of the in-cylinder processes, this cut-and-try process is not sufficient to create the significant improvement now sought, plus the experiment set-up is time consuming and expensive.

The advent of computers and the possibility of performing "numerical" experiments based on computational fluid dynamics analysis provide a new way of the simulating and designing internal combustion engines. By easily examining the various tradeoffs that must be made to move current design toward the optimum, CFD enables the transient analysis in the operation condition, which can yields detailed information on the flow field in a timely and cost-effective manner, thus meets today's performance targets requirement.

However the three-dimensional flow numerical simulation about a helical/direct intake port-vertical/canted valve-cylinder system is remaining as a challenge to the CFD group The main reasons are as following [18, 19, 21]:

- It is extraordinarily difficult to generate computational grids of high quality due to complicated port shape, chamber shape, valve position and the valve lift and piston motion strategy.
- The flow field in an internal combustion engine is turbulent and comprises many time and length scales (from eddies that are essentially large enough to fill the available engine cylinder space down to eddies often substantially below a millimeters in size). It is impossible with present techniques and facilities to obtain a detailed numerical solution of the Navier-Stokes equations that can account for all the in-cylinder turbulence time and length scales, and proper turbulence models need to be introduced.

Fortunately, there are a number of commercial and in-house CFD codes have been developed today, each of employing one or more numerical methods and techniques to

circumvent these difficulties. Among them, FIRE, Star-CD, FLUENT and KIVA-3V are the most widely used codes in industries and academics.

3.2 Moving Grid Strategy in Current CFD Codes

Almost all these CFD software (FIRE, Star-CD, FLUENT,KIVA-3V) use the Lagrange based moving grid technique to track the valve and piston motion for the engine simulation [13,14, 15,16, 25], that is, the mesh attaches to the moving boundaries (valve surface and piston head) and deforms with the computational domain, and is subjected to the topology changes in the course of its evolution. There is three mesh motion strategies to accommodate the volume deformation associated with these in-cylinder motions: spring deforming, local regridding and layering.

3.2.1 Spring Deforming

In the spring-based deforming method, a block of mesh cells (single layer or multi-layers) deforms like springs due to the movement of the mesh boundary: the edges between any two mesh nodes are idealized as a network of interconnected a springs. A displacement at a given boundary node will generate a force proportional to the displacement along all the springs connected to the node, and so the displacement of the boundary node is propagated through the whole block [26]. The displacement of each node can be controlled by specifying different "spring coefficient" of each "springs" to get a better control over some specific region (especially the boundary layer regions). A simple example of the spring-based deforming is shown in Figure 2 for a cylindrical volume where one end of the cylinder is moving.

In the Spring Deforming Method, the connectivity of the mesh cells and nodes in the corresponding block remains the same, the topology and the resolution of the grid system do not change, only the nodes position may change due to the boundary motion. It is well suited for the situation where the motion of the interested region (piston head or valve surface) usually keeps the same (or reverse) direction and the displacement is not large enough to cause some ill-shaped (such as high aspect ratio) cells.

Spring deforming allows the use of a low grid resolution during the early time steps thus can speed the analysis through the initial transient behaviors. When the flow variables settle, layering (as discussed in 3.2.3) to a higher grid resolution can obtain the necessary accuracy.

Since this method always involves grid motions, it requires the Jacobin to be upgraded at each time step to satisfy the GCL.

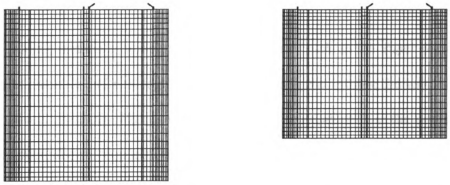


Figure 2 Spring deforming on a simple cylinder at the end of compression (Left: piston head at BDC, right: piston head at TDC)

3.2.2 Local Regridding

When the boundary displacement is large, the moving boundaries can distort the cells too much (especially for the gap region between valve surface and seat), which will create bad quality cells (e.g. high aspect ratios, twisted faces, negative volumes, negative volumes, etc.), which lead to a high degradation of a convergence rate and a poor accuracy as the solution is advanced to the next time step. Local regridding is thus necessary to adjust the mesh to the level with acceptable quality. This can be done either by changing the local mesh topology (figure) or using smoothing algorithms and others similar techniques to redistribute the grid nodes without changing the connectivity information. An interpolation is thus needed to maps the corresponding boundary conditions and flow variable data that is generated during the analysis from the old grid to the new grid between two adjacent time steps. This method may lead to the change of connectivity and grid resolution as topology adjustment is employed. Figure 3 gives such an example.

Local regidding method is widely used in the regions (such as the upper portion of the combustion chamber and port region around valve seat) that involve the valve movement and extensively meshing effort is needed to match the geometries at the position where the combustion deck is complex or a canted valve is involved in the motion. However, the application of hybrid mesh techniques (such as hexahedron elements couples with tetrahedral elements or pyramid element in FLUENT and Star-CD) can simplify this process.

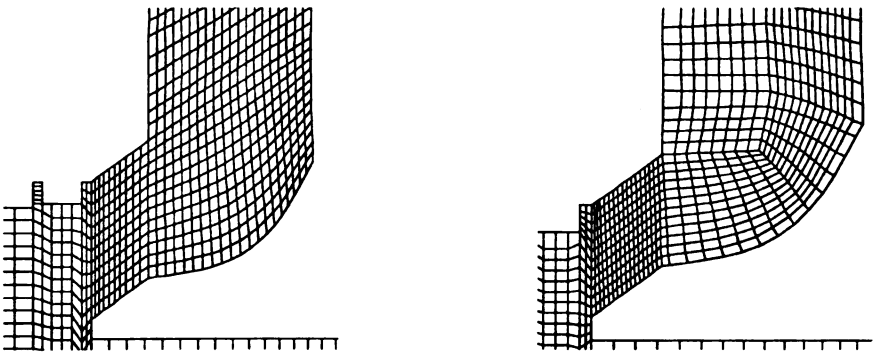


Figure 3 Local regridding applied to the valve region during intake stroke (Left: valve opening, right: valve closing)

3.2.3 Layering

When the boundary displacement is large, by simply adding or removing a cell layer in the boundary motion direction, the grid resolution can be adjusted to obtain a block of cells (single layer or multi-layers) with good aspect ratio and acceptable layer-thickness (which is very import for the boundary layer regions). The layering method is particularly suited in the cylinder region that is directly above the piston head, and can also be utilized in the regions above the valve, including the valve seat region.

Layering changes the resolution of the grid system and it can maintain the topology of grid system in the direction where there is no motion involved. An interpolation is also needed to map the data between different data set at different time steps. Consequently, the application of layering to a single layer block or multi-layers block determines the level of interpolation. A simple layering example is illustrated in Figure 4.

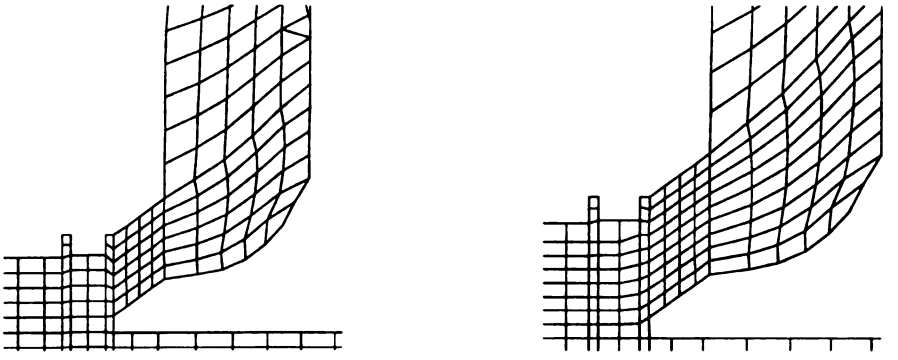


Figure 4 Layering applied to the valve region during intake stroke (Left: valve closing, right : valve opening)

The spring deforming technique and local regridding technique can be viewed as grid redistribution techniques while layering technique can be viewed as grid refinement method. The reason that we term them separately is that each technique is employed at specific situations for engine simulation.

The combination of these three techniques coupled with other meshing techniques (block structure mesh, unstructured mesh, hybrid mesh, automatic mesh generation) to deal with the meshing task in different portion of the computational volume enables the users to generate an acceptable computational mesh. The meshing time usually varies from 2-day to 2 weeks depending on the grid generator's automatic level and user's familiarity with the codes.

3.3 Turbulence modeling

For low speed (laminar) flows without heat transfer, the equations governing the conservation of mass and momentum can be used to describe the flow exactly for incompressible flows. Turbulence however, leads to rapid velocity fluctuations in both

space and time. So although these equations can properly describe the details of turbulent motions, it's too costly and often time consuming to obtain a solution with detailed information about both the time and space variations of flow variables. This leads to the concept of " turbulence modeling" [7, 8].

Most engineering models of turbulent flow are based on the use of the Reynold's averaging technique [7, 8], which assumes that the quantities (that appears in the N-S equations) at a given point in space and time are described as a superposition of some mean part, which may vary slowly with time and a random component, which varies rapidly. Mathematically, they can be expressed as:

$$\begin{aligned} u_i &= \overline{u}_i + u_i' \\ \rho &= \overline{\rho} + \rho' \\ p &= \overline{p} + p' \end{aligned} \tag{3.1}$$

with $\overline{u_i'} = \overline{\rho'} = \overline{p'} = 0$

Here the quantities with bar denote the mean values and those with primes are fluctuations.

Then the ensemble-averaged mass and momentum balance equations are given as (for incompressible fluids):

- The ensemble-averaged continuity equation:

$$\frac{\partial}{\partial t}\overline{\rho} + \frac{\partial}{\partial x}\left[\overline{\rho u} + \overline{\rho' u'}\right] + \frac{\partial}{\partial y}\left[\overline{\rho v} + \overline{\rho' v'}\right] = 0 \tag{3.2}$$

- The ensemble -averaged momentum equation:

$$\rho\frac{\bar{D}\bar{u}}{Dt}=-\frac{\partial\bar{p}}{\partial x}+\mu\nabla^2\bar{u}-\rho\left(\frac{\partial\overline{u'^2}}{\partial x}+\frac{\partial\overline{u'v'}}{\partial y}+\frac{\partial\overline{u'w'}}{\partial z}\right)$$

$$\rho\frac{\bar{D}\bar{v}}{Dt}=-\frac{\partial\bar{p}}{\partial y}+\mu\nabla^2\bar{v}-\rho\left(\frac{\partial\overline{u'v'}}{\partial x}+\frac{\partial\overline{v'^2}}{\partial y}+\frac{\partial\overline{v'w'}}{\partial z}\right) \tag{3.3}$$

$$\rho\frac{\bar{D}\bar{w}}{Dt}=-\frac{\partial\bar{p}}{\partial z}+\mu\nabla^2\bar{w}-\rho\left(\frac{\partial\overline{u'w'}}{\partial x}+\frac{\partial\overline{v'w'}}{\partial y}+\frac{\partial\overline{w'^2}}{\partial z}\right)$$

where $\frac{\bar{D}}{Dt}=\bar{u}\frac{\partial}{\partial x}+\bar{v}\frac{\partial}{\partial y}+\bar{w}\frac{\partial}{\partial z}+\frac{\partial}{\partial t}$

The additional unknown terms in the momentum equations are know as the Reynolds stress tensor expressed as:

$$\text{Reynolds stress tensor} = -\rho\begin{pmatrix}\overline{u'^2} & \overline{u'v'} & \overline{u'w'}\\ \overline{u'v'} & \overline{v'^2} & \overline{v'w'}\\ \overline{u'w'} & \overline{v'w'} & \overline{w'^2}\end{pmatrix} \tag{3.4}$$

The modeling of Reynolds stress tensor with other variables introduces different turbulence models (standard $k-\varepsilon$ model, RNG model, $k-\omega$ model...), among them standard $k-\varepsilon$ model is widely used in industry due to its numerical robustness and economy.

In this method, the Reynolds stress tensor terms can be expressed in terms of the mean rate of strain S_{ij} and turbulence viscosity μ_t:

$$-\rho\overline{u_i'u_j'}=2\mu_t S_{ij}-\frac{2}{3}\rho\delta_{ij}k \tag{3.5}$$

with :

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right); \quad \mu_t = C_\mu \rho \frac{k^2}{\varepsilon}; \text{ and } \delta_{ij} \text{ is the Kroneche delta function.}$$

The turbulence kinetic energy k and its dissipation ε can be solved from the following equation:

$$\begin{aligned} \rho \frac{Dk}{Dt} &= \rho(P_k - \varepsilon) + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \\ \rho \frac{D\varepsilon}{Dt} &= \rho\left(C_{\varepsilon 1} P_k + C_{\varepsilon 3} k \frac{\partial \overline{u_k}}{\partial x_k} - C_{\varepsilon 2}\varepsilon\right)\frac{\varepsilon}{k} + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial \varepsilon}{\partial x_j}\right] \end{aligned} \tag{3.6}$$

where the production of turbulence energy is:

$P_k = -\overline{u_i' u_j'}\frac{\partial \overline{u_i}}{\partial x_j}$ and $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, $C_{\varepsilon 3}$, σ_k are empirical constants.

It should also be noted that the application of $k-\varepsilon$ to engine simulations still has serious limitations [18]: the model assumes that the turbulent transport is in the same direction as the mean flow gradients and does not consider the effects of pressure-velocity correlations on the k equation and ε equations. And also the ε equation does not have a strong physical foundation.

3.4 Description of the Problem

A traditional direct port diesel engine (generic engine) was employed in the study; the geometric shape and engine parameters are listed in Figure 5 and Table 1:

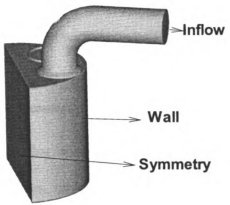


Figure 5 Geometry of the generic engine and the boundary definition

Table 1 Engine Specifications and Calculation Conditions

Bore x Stroke		80mm x 100mm
Squish		5.0 mm
Engine RPM		1500 Rpm
Compression Ratio		19.3
Connecting Rod		198 mm
Maximum Intake Valve Lift		8.68 mm
Intake Valve Opening		353.3 ATDC (after top dead center)
Intake Valve Closure		545.6 ATDC
Boundary Conditions	Inflow Boundary	$P_{static} = 1\ atm,$ $T_{static} = 300K$ $\mu_t = C_\mu \rho k^2 / \varepsilon = 100\mu$ $I = \sqrt{2k/3}/U = 0.1$ $U = f(Rpm)$
	Wall Boundary	adiabatic wall, no slip
	Symmetry Boundary	Symmetry
Initial Conditions		$u = v = w = 0$ $P_{static} = 1\ atm,$ $T_{static} = 300K$ $\mu_t = C_\mu \rho k^2 / \varepsilon = 100\mu$ $I = \sqrt{2k/3}/U = 0.1$ $U = f(Rpm)$

We assume a symmetry boundary across the engine symmetry plane thus there is no swirl generation during the simulation and we can use half engine simulation to reduce the computational time.

3.5 Numerical Setups of the Simulation

FIRE, a finite volume code that solves the ensemble averaged fully compressible conservation equations for mass, momentum, and energy, was used to perform the simulations. The Reynolds stresses are linked with the mean ensemble averaged properties with the standard $k-\varepsilon$ model. The standard " wall function" is used to bridge the viscosity-affected region between the wall and the fully turbulent region.

The $k-\varepsilon$ model constants were left to their default settings as listed in Table :

Table 2: Values of standard $k-\varepsilon$ model constants

C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	$C_{\varepsilon 3}$	σ_k
0.09	1.44	1.92	-0.373	1

The discretisation equations are solved by iteration following the SIMPLE velocity-pressure coupling algorithm.

The second order central total variation-diminishing scheme (CTVD) with minmod limiter was used for space differencing of all the variables. The first order fully implicit scheme is adopted for the temporal differencing of all the variables [14].

The numerical model was made based on the geometry CAD data (surface mesh) from another commercial code- Star-CD. The using of arbitrary multi-layer based spring deforming and layering technique were used to handle the valve and piston motion, and arbitrary local regriddings were employed to handle the valve seat region when the mesh

screwed during the valve motion. The resulting numerical models are shown in Figure 6

Table 3shows a summary of the model data.

Figure 6 Computational Mesh of the generic engine

Table 3 Numerical model data

Total number of cells (maximum)	404,184
Number of cells in port and valve (maximum)	99,504
Number of cells in cylinder (maximum)	304,680

3.6 Experiment Verifications

A validation based on the experiment data (provided by Dr. Shock) had been performed to check the code quality.

The average in-cylinder pressure shows good agreement with the experimental data; the difference is below 5%. (Figure 7)

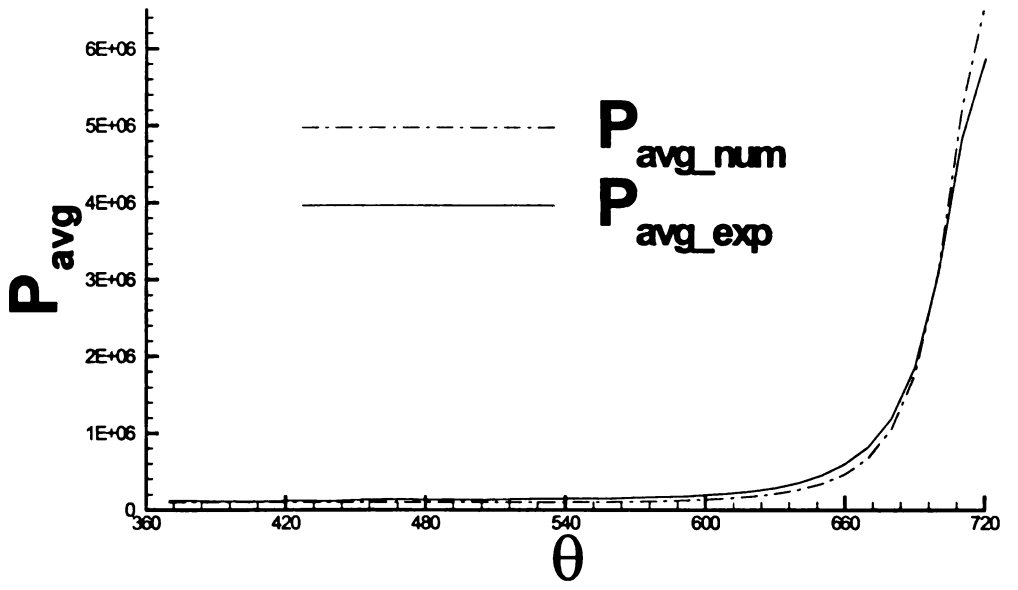


Figure 7 Average in-cylinder pressure comparision: dash line-numerical data, solid line-experimental data

The flow vector distribution at the symmetry plan are also compared (Figure 8-11, due to the limitation of the measurement, the experimental data only reflect a portion flow distribution in the whole flow field): the agreements of numerical results with experimental result at intake stroke (CA 450, 540 ATDC (after top dead center)) are satisfactory, but clear differences remain at compression stroke (CA 630, 660 ATDC): the flow pattern is almost completely wrong. This behavior may be caused by the inaccuracies or errors in either the numerical modeling or the experiments; currently there is no solid explanation for this behavior of the code.

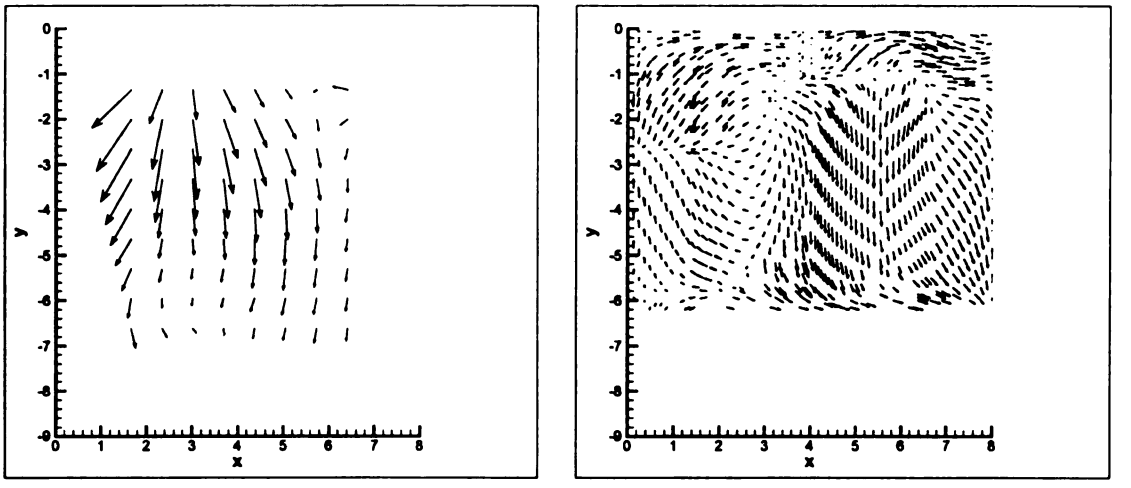


Figure 8 Flow pattern comparison at CA 450 ATDC (left-experimental data , right-numerical data)

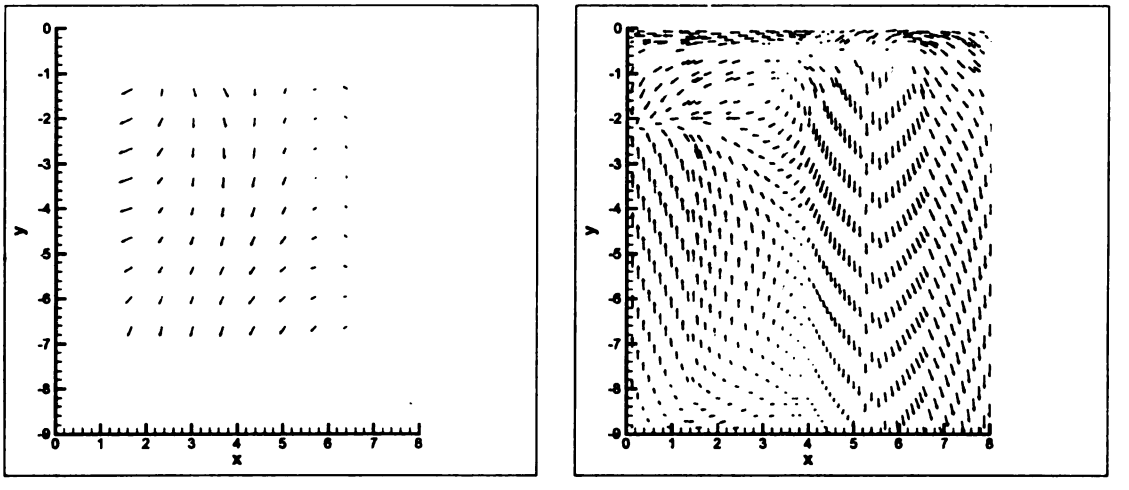


Figure 9 Flow pattern comparision at CA 540 ATDC (left-experimental data, right-numerical data)

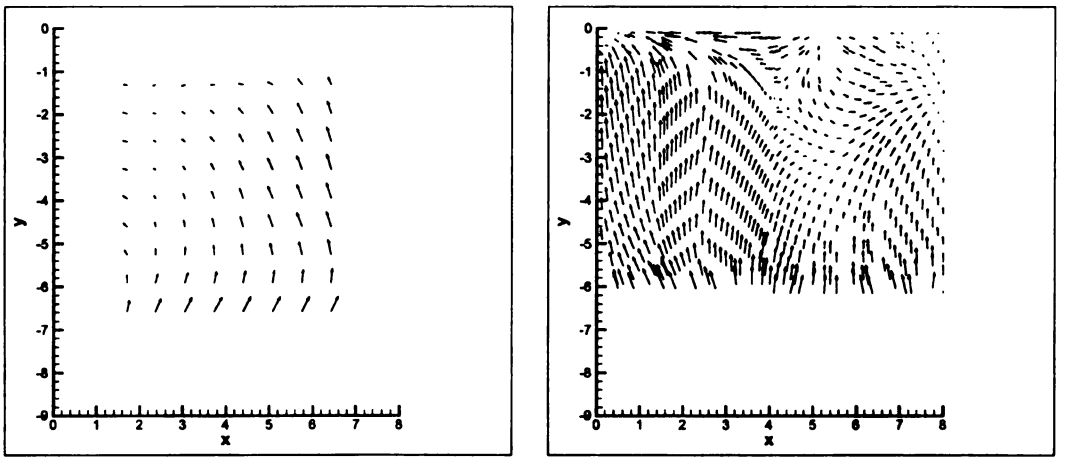


Figure 10 Flow pattern comparision at CA 630 ATDC (left-experimental data, right-numerical data)

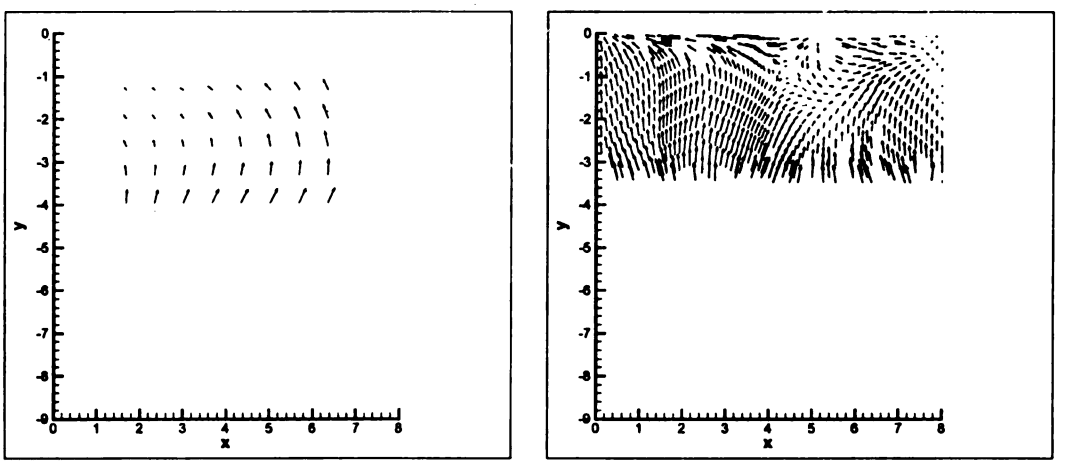


Figure 11 Flow pattern comparision at CA 660 ATDC (left-experimental data, right-numerical data)

3.7 Simulation Results Discussion

Simulation results are presented here for flow analysis. Three plans are chosen here (two vertical plans: x-plan and y-plan and one horizontal plan: z-plan) to reveal the flow pattern. These plans are defined in Figure 12: The x-plan cuts through the axis of the two

intake valves and normal to the horizontal plan. The y-plan cuts through the axis of the intake and outlet valves and normal to the horizontal plan. The z-plan is a horizontal plan and cuts through the middle of the cylinder.

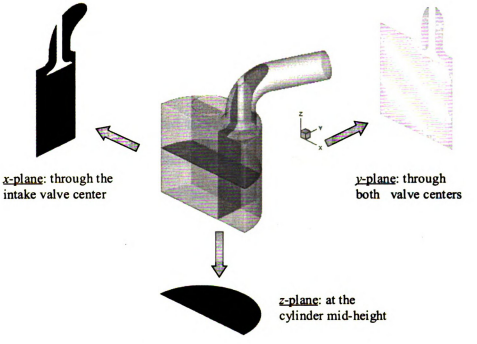


Figure 12 Selected plans for the generic engine

Figures 13-34 show the flow field in the form of velocity vector distribution at different degree ATDC along the vertical plans (Figures13-32) and horizontal plan (Figures 33, 34).

At the beginning of the intake stroke (Figure 13,14, CA 370,380 ATDC), with the valve opening and piston moving down, the flow is induced into the cylinder and forms an impinging jet around the valve seat region. As the valve lift and piston speed are low, the turbulence intensity level is not strong, the flow remains attached to the valve head and seat and then separates from the valve at the inner edge of the valve seat, part of the flow

will hit the cylinder wall, this results in the formation of obvious recircuilation motions under the cylinder head and valve (Figure 14, CA 380 ATDC): Two couples of opposite tumble motions are created-one couple that formed at the position between left side of valve seat and the top-left cylinder corner and one formed in the opposite direction. Later on we can see that the tumbles formed under the valve head have dominant influence on the flow pattern.

As the piston accelerates downside and the valve lift increases, (Figure 15-22, CA 390-460 ATDC), much of the directed energy in the jet is converted into turbulence, the turbulence level increases. The tumble in the right upside of cylinder corner can not find enough space to grow and finally breaks up. The other three tumbles motions are strengthened continuously as the distances from cylinder wall are far enough for them to grow (e.g., the tumble at the left corner can shift it position downside as the piston moves down). The structures of the two tumbles motions under the valve also changes as the turbulence intensity evolves. The left side one (counterclockwise) continues to grow and finally dominates the upper side in-cylinder flow field while the right side one does not grow and finally looses most of its energy (break up in to turbulence). Another tumble motion whose rotation direction is opposite the one under the valve is also created above the piston head at around CA 450 ATDC (Figure 21,22, CA 450, 460 ATDC) and dominates the lower part of the in-cylinder flow. At CA 460 ATDC, the turbulence level is strongest as the engine approaches its maximum valve lift and piston speed.

During the second half of the inlet stroke (Figure 23-25, CA 480-540 ATDC), as the piston slows down and the valve is going to close (which means the jet is coming to an end with fewer directed energy introduced), most of the tumble motions decays: the

intensity of the tumble that above the top of piston decreases markedly and finally attenuates into two small tumbles around the center position of the cylinder (Figure 25 CA 540 ATDC). Viscosity will also have some effects on these phenomena.

During the compression stroke (Figure 26-32, CA 550-700 ATDC), the tumble motions are amplified due to the increase of density and the changes in length scales (as engine compressed, the charged geometry is changed). The two tumbles located at the center of cylinder combine into one tumble motion that rotates in clockwise direction, and try to move upside to find more space to grow, the intensity of the upper tumble also increases and it shifts its location to the center position right below the valve head to get more tumble axis-cylinder distance.

As the piston approaching TDC (Figure 31,32, CA 680,700 ATDC), the vortexes created by the tumble cannot find sufficient room to maintain their form. They first combine into one vortex at CA 680 ATDC and then break up into turbulence at CA 700 ATDC. A crudely homogeneous condition can be observed at that time. The break up of the swirl motions (strictly speaking, these motions are recirculations in the cylinder radius direction since we assume a symmetry boundary condition across the cylinder symmetry plane) in the compression stroke can also be observed (Figure 33, from CA 550 ATDC to CA 660 ATDC). The breaking up of the tumble motions at CA 680 ATDC will also result in a changing over to small swirl motions (Figure 34 CA 680 ATDC to CA 700 ATDC).

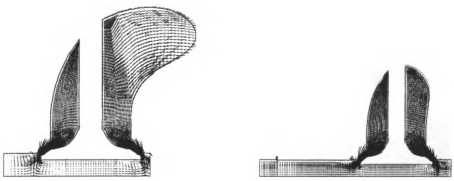


Figure 13 Velocity distributions at 370 ATDC (x-plane and y-plane)

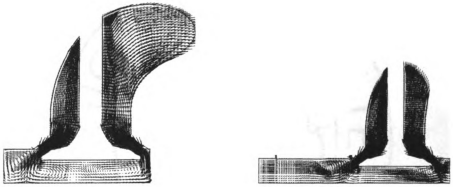


Figure 14 Velocity distributions at 380 ATDC (x-plane and y-plane)

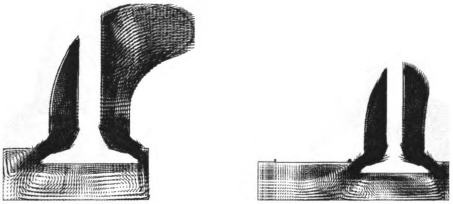


Figure 15 Velocity distributions at 390 ATDC (x-plane and y-plane)

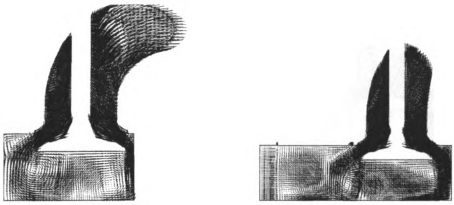


Figure 16 Velocity distributions at 400 ATDC (x-plane and y-plane)

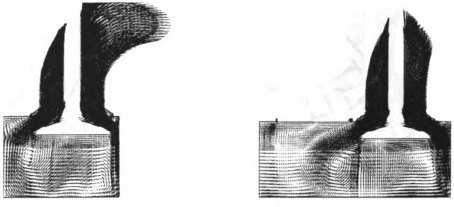


Figure 17 Velocity distributions at 410 ATDC (x-plane and y-plane)

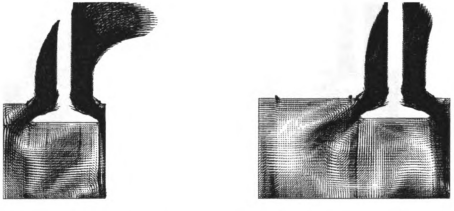


Figure 18 Velocity distributions at 420 ATDC (x-plane and y-plane)

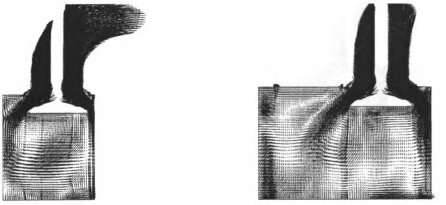


Figure 19 Velocity distributions at 430 ATDC (x-plane and y-plane)

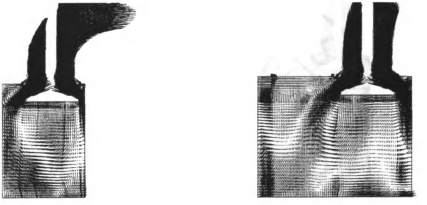


Figure 20 Velocity distributions at 440 ATDC (x-plane and y-plane)

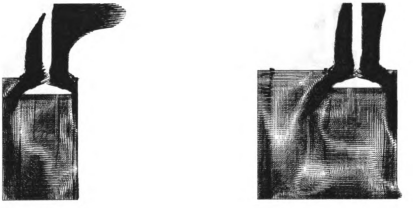


Figure 21 Velocity distributions at 450 ATDC (x-plane and y-plane)

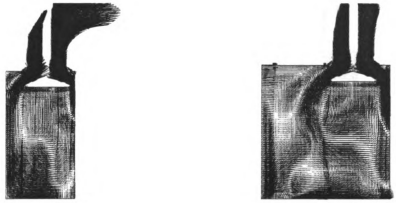


Figure 22 Velocity distributions at 460 ATDC (x-plane and y-plane)

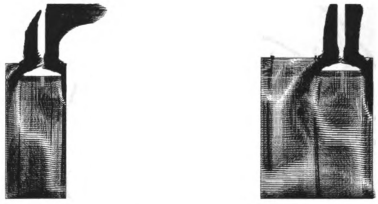


Figure 23 Velocity distributions at 480 ATDC (x-plane and y-plane)

Figure 24 Velocity distributions at 510 ATDC (x-plane and y-plane)

Figure 25 Velocity distributions at 540 ATDC (x-plane and y-plane)

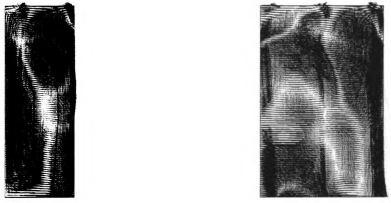


Figure 26 Velocity distributions at 550 ATDC (x-plane and y-plane)

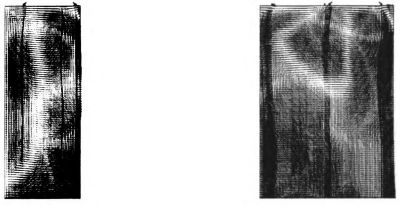


Figure 27 Velocity distributions at 580 ATDC (x-plane and y-plane)

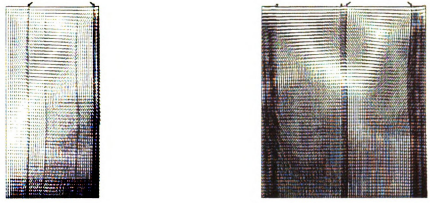


Figure 28 Velocity distributions at 610 ATDC (x-plane and y-plane)

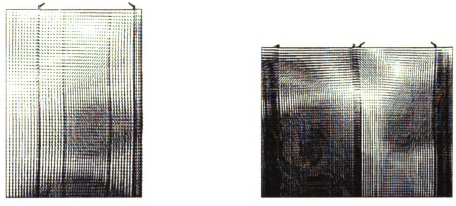


Figure 29 Velocity distributions at 640 ATDC (x-plane and y-plane)

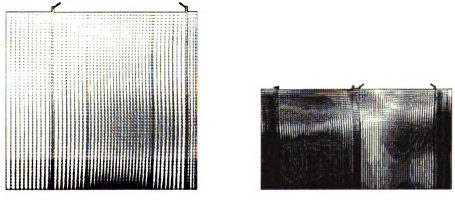


Figure 30 Velocity distributions at 660 ATDC (x-plane and y-plane)

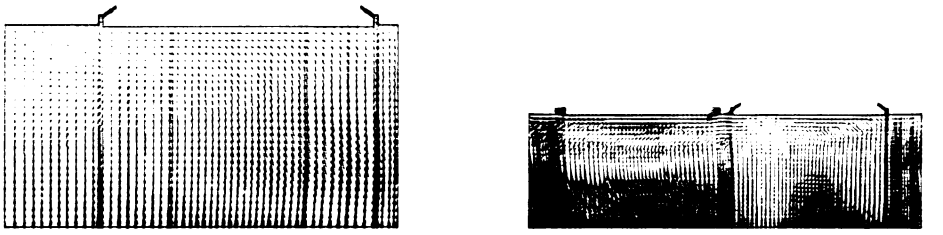


Figure 31 Velocity distributions at 680 ATDC (x-plane and y-plane)

Figure 32 Velocity distributions at 700 ATDC (x-plane and y-plane)

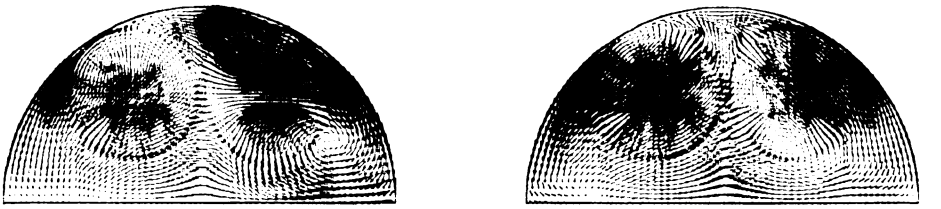


Figure 33 Velocity distributions at 550(left) and 660 (right) ATDC (z-plane)

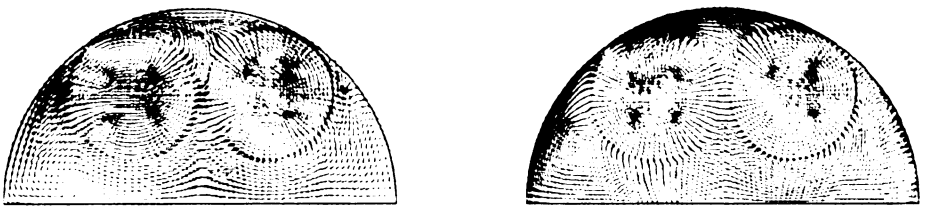


Figure 34 Velocity distributions at 680(left) and 700 (right) ATDC (z-plane)

3.8 Issues with Moving Grid Technique for In-cylinder Engine Simulation

There are three issues closely related to the moving grid technique: interpolation errors, human cost for mesh generation and the quality of grid system. Each of them will affect the numerical modeling process and the corresponding solution.

The interpolation error is introduced by the solution mapping process: once there is a need for local regridding or layering, the solution on old grid needs to be transformed into the new grid. As we have seen in section 2.4, the interpolated solution on the new grid is obtained from the local approximations using the nodal values of the corresponding element on the old grid, since the accuracy of the local approximation is highly depended on the chose of local basis function, it may not be a good representative of the actual solution and thus introduce errors. Figure 35 gives a simple one dimensional example of such interpolation: The numerical solution at the nodes on "old" grid agrees very well with the exact solution, while the interpolated one on the "new" grid lose accuracy.

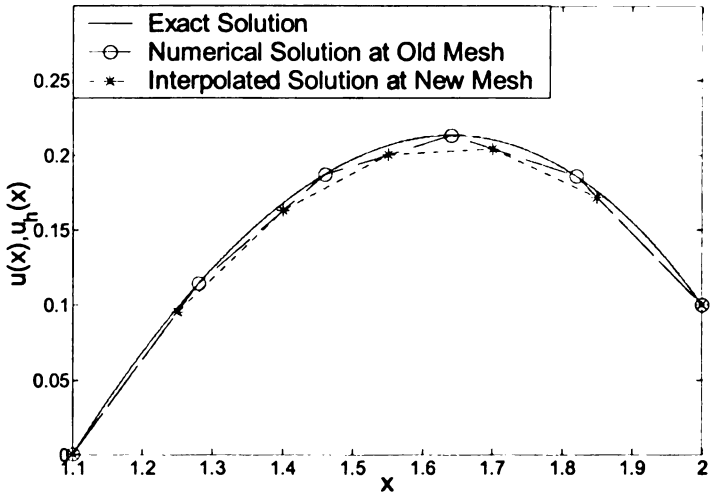


Figure 35 Solution interpolation from "old" grid to "new" grid introduces error

The second issue is related to the generation of computational grid to handle the valve/piston motion in the context of complicated engine geometry. As we have discussed earlier, moving grid technique requires the grid system to conform to the real engine geometry at the corresponding time step as well as preserve suitable topology to handle the valve/piston motion, it is usually done with a multiblock unstructured grid system and the quality of the grid is very difficult to control: the grid system may be skewed and unevenly distributed, especially in the region related to the valve motion. The bad quality grid will introduce error to the numerical solution and the error will transport and polluted others regions. Figure 36-38 give the example of a bad grid system and the corresponding errors are obtained based on the comparison of the numerical solution for a low-resolution grid (50k) with the one for a high-resolution grid (500k, we take it as the exact solution here)-Figure 36 shows the gird system for a 50K engine at CA 460 ATDC at selected x,y z plans and the corresponding velocity point-wise relative error and absolute error contours were draw in Figure 37 and 38 respectively.

Figure 36 Computational mesh at selected plans for a 50k generic engine numerical model

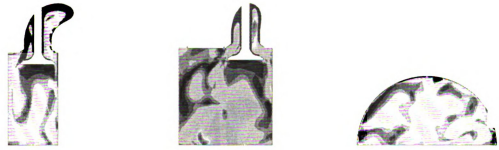


Figure 37 Velocity absolute error distributions at selected plans for a 50k generic engine numerical model

Figure 38 Velocity relative error distributions at selected plans for a 50k generic engine numerical model

And also because such a limitation (geometrical confirmation) exists, we find that the solution is highly sensitive to the grid topology and regridding strategy that were used in

the numerical model-to illustrate this, we employed three methods here to generate a 50K grid for the generic engine, each grid uses a different mesh strategy. In method I and II the grid topology in horizontal plan is exactly the same while the moving mesh part for the near-valve region are generated using different remeshing strategy (the time and topology for regridding are different); Method II and III use the same remeshing strategy for the near-valve region and differs from each other in the horizontal plan (different topology). Figure 39-41 shows the grids' difference of these three methods and Figure 42-47 show the different of the streamline contours for these methods (at CA460 and CA 630 ATDC).

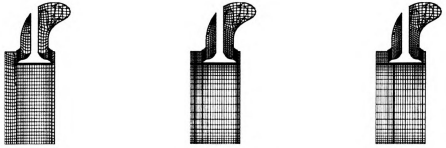


Figure 39 Computational mesh at selected plans for a 50k generic engine numerical model (Method I)

Figure 40 Computational mesh at selected plans for a 50k generic engine numerical model (Method II)

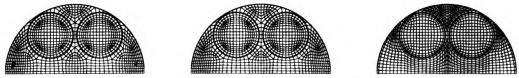


Figure 41 Computational mesh at selected plans for a 50k generic engine numerical model (Method III)

Figure 42 Streamline contours comparison (left: method I, middle: method II, right : method III) at selected plan (x-plan) for a 50k generic engine numerical model at CA 460 ATDC

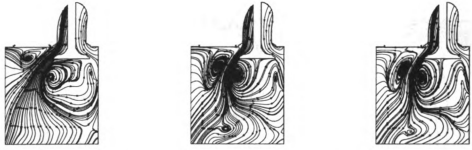


Figure 43 Streamline contours comparison (left: method I, middle: method II, right: method III) at selected plan (y-plan) for a 50k generic engine numerical model at CA 460 ATDC

Figure 44 Streamline contours comparison (left: method I, middle: method II, right: method III) at selected plan (z-plan) for a 50k generic engine numerical model at CA 460 ATDC

Figure 45 Streamline contours comparison (left: method I, middle: method II, right: method III) at selected plan (x-plan) for a 50k generic engine numerical model at CA 630 ATDC

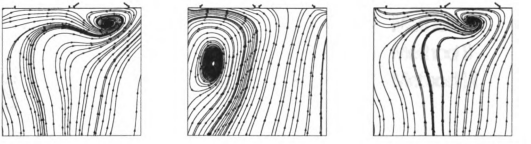


Figure 46 Streamline contours comparison (left: method I, middle: method II, right: method III) at selected plan (y-plan) for a 50k generic engine numerical model at CA 630 ATDC

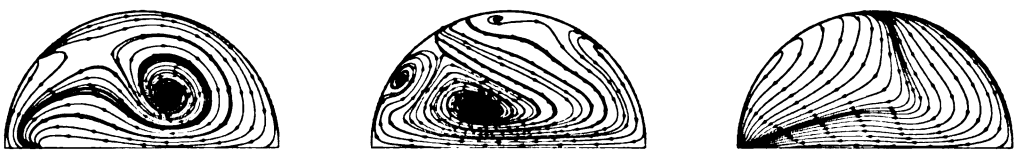


Figure 47 Streamline contours comparison (left: method I, middle: method II, right: method III) at selected plan (z-plan) for a 50k generic engine numerical model at CA 630 ATDC

We also compare the predicted turbulence kinetic energy for these three methods since turbulence kinetic energy is also one of the most important issue for engine simulation: an overproduction of turbulence kinetic energy changes spreading rate of the jet and caused different mixing in the cylinder. The averaged turbulence kinetics energy comparison is plotted in Figure 48, and the detailed TKE distributions at CA 460 and 630 are compared in Figure 49-52.

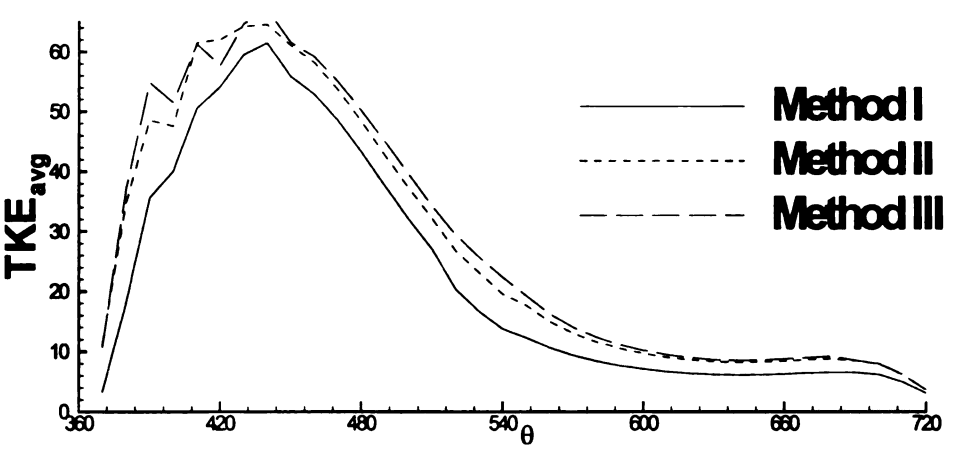


Figure 48 Average predicted in-cylinder turbulence kinetic energy comparison for three moving mesh strategies

Figure 49 Turbulence kinetic energy distribution comparison (left: method I, middle: method II, right: method III) at selected plan (y-plan) for a 50k generic engine numerical model at CA 460 ATDC

Figure 50 Turbulence kinetic energy distribution comparison (left: method I, middle: method II, right: method III) at selected plan (z-plan) for a 50k generic engine numerical model at CA 460 ATDC

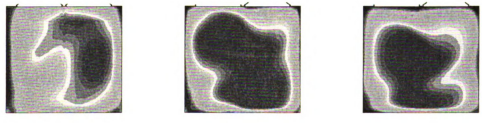


Figure 51 Turbulence kinetic energy distribution comparison (left: method I, middle: method II, right: method III) at selected plan (y-plan) for a 50k generic engine numerical model at CA 630 ATDC

Figure 52 Turbulence kinetic energy distribution comparison (left: method I, middle: method II, right: method III) at selected plan (z-plan) for a 50k generic engine numerical model at CA 630 ATDC

This tremendous difference in the prediction of turbulent kinetic energy will surely affect main engine parameters during a design process.

The third issue is well known: moving grid technique has a strong requirement for repetitive remeshing when intensive interface (moving boundaries) activity involved, which is time consuming and costly. This bad feature limits its application in engine model evaluation since every geometrical (port, valve, combustion chamber) configuration and possible lift strategies, we need to generate a separated set of grids to do the simulation, and such a human cost is huge.

Chapter 4

An Introduction to Lagrange Multiplier/Fictitious Domain Methods (LM/FDM), Methods for Reducing Errors in LM/FDM Applied to Heat Transfer Problems

4.1 Introduction

Fictitious domain methods offer the possibility of significantly reducing the difficulty of meshing complex domains. This is achieved extending a complex domain into a much simpler one that can be discretized easily (the extended domain is called the fictitious domain). The original problem is preserved in the extended domain by enforcing the original boundary conditions with the help of Lagrange multiplier [27-31] or by using a penalty method [32].

The fictitious domain method is especially useful when dealing with moving boundary problems. Two independent meshes can be then used: one fixed Cartesian mesh and an independent boundary mesh that "follows" the boundary and enforces the boundary condition. This obviates the need for repetitive remeshing. R.Glowinski et. al have developed recently various approaches based on the use of Lagrange multiplier distributed on the boundary of the moving object [27-29] and Lagrange multiplier distributed over the entire domain associated to a moving object [30, 31, 33-35].

Given its easy-implementation capability that which enables a fast turn around time in engineering modeling, Lagrange Multiplier/Fictitious domain methods can come to be a powerful technique for the design evaluation and engineering analysis if it accuracy can also be guaranteed. The use of multipliers to enforce an essential boundary condition however may result in a discontinuity in the first derivative of the solution (e.g.

temperature or velocity fields). This can introduce significant errors if the multipliers are not coincident to nodes of the underlying mesh [39]. This situation arises since most elements commonly used are C^0 continuous i.e. a discontinuity of the first derivative is allowed only at the boundaries of the elements. Errors are thus introduced on the adjacent nodes due to the impossibility of obtaining a C^0 solution within an element. The use of higher order polynomials as shape functions somewhat reduces the errors but such errors are nonetheless still considerable compared to the use of a mesh fitted to the boundary (see below in section 4.3 and 4.4)..

In this chapter, the Lagrange multiplier/fictitious domain method and its use in problems with Dirichlet boundary conditions is discussed. A parametric study for the introduced errors is conducted. Two possible approaches, Fictitious Constraint methods and Shape Reconstruction method, are proposed to mitigate the errors associated to the use of Lagrange multiplier with the corresponding element. For simplicity, we only consider one-dimensional and two-dimensional steady state conduction heat transfer problems and discuss the influence of the jump of local solution gradient and the presence of the multiplier within an element.

4.2 Lagrange Multiplier/Fictitious Domain Methods and the Dirichlet Problem

4.2.1 A Model Problem

The following Dirichlet boundary value problem is first considered:

$$
\left.\begin{array}{l}
\text{Given} \quad f \in H^{-1}(\Omega),\ T_0 \in L^2(\gamma), T_1 \in L^2(\gamma), \\
\text{find a function T such that} \\
-\Delta T = f \quad \text{in} \quad \Omega \setminus \omega, \\
T = T_0, \text{on} \quad \Gamma; \\
T = T_1, \text{on} \quad \gamma;
\end{array}\right\} \qquad (4.1)
$$

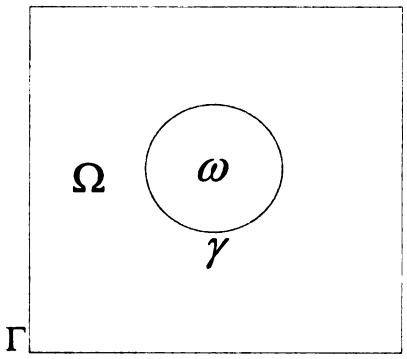


Figure 53 Illustration of the geometry associated to the Dirichlet problem presented in (4.1)

In this problem, Ω is a bounded domain in $\mathbb{R}^d$ ($d \geq 2$) including an inclusion ω, Γ and γ are the boundaries of Ω and ω.

4.2.2 Lagrange Multiplier/Fictitious Domain Formulation

The Lagrange multiplier/fictitious domain formulation consists of:

- Extending the domain $\Omega \setminus \omega$ to the larger square domain Ω, the original boundary condition on the γ becomes an embedded constraint for the extended problem
- Extending f in $H^1(\Omega \setminus \omega)$ to $\tilde{f}$ in $H^1(\Omega)$, and T in $H^1(\Omega \setminus \omega)$ to $\tilde{T}$ in $H^1(\Omega)$.

- Introducing Lagrange multipliers to enforce the embedded boundary condition on γ and making the augmented functional stationary.

The following equivalent Lagrange multiplier/fictitious domain formulation is thus obtained

$$\left.\begin{aligned}
&\text{Find } \tilde{T} \in H^1(\Omega),\ \lambda \in L^2(\gamma) \text{ such that} \\
&\alpha(\tilde{T},\ v) + \int_\gamma \lambda v d\gamma = \int_\Omega f v d\Omega \qquad \forall\ v \in H_0^1(\Omega) \\
&\int_\gamma \tilde{T}\mu d\gamma = \int_\gamma T_1 \mu d\gamma \qquad \forall\ \mu \in L^2(\gamma) \\
&\tilde{T} = T_0 \quad \text{on} \quad \Gamma \\
&\text{where} \quad \alpha(\tilde{T},\ v) \equiv \int_\Omega \nabla\tilde{T}\cdot\nabla v\ d\Omega
\end{aligned}\right\} \tag{4.2}$$

4.2.3 Finite Element Approximations

The approximation spaces for the discrete variables $\tilde{T}_h, \lambda_h$, are chosen as following:

$$\begin{aligned}
T_h &= \left\{ \widetilde{T_h} \middle| \widetilde{T_h} \in \left[C^0(\Omega)\right]^d, \widetilde{T_h}\big|_E \in [P_1]^d, \forall E \in \Im_h \right\} \\
\Lambda_h &= \left\{ \lambda_h \middle| \lambda_h\big|_{E^\gamma} = const., \forall E^\gamma \in \Im_h^\gamma \right\}
\end{aligned} \tag{4.3}$$

Where d=1,2, or 3. $\Im_h$ is a regular discretization of the fictitious domain Ω, $\Im_h^\gamma$ is the discretization of the embedded boundary γ, E and E^γ are the corresponding fictitious domain element/embedded boundary element. $\widetilde{T_h}$ is assumed globally C^0 continuous and locally (within each fictitious domain element) C^1 continuous.

Problem (4.2) can be posed as:

$$\left.\begin{aligned} &\text{Find } \widetilde{T_h} \in T_h,\ \lambda_h \in \Lambda_h \text{ such that:} \\ &\alpha\left(\widetilde{T_h},\ v\right) + \int_\gamma \lambda_h v d\gamma = \int_\Omega f v d\Omega \qquad \forall\ v \in T_h \\ &\int_\gamma \widetilde{T_h} \mu d\gamma = \int_\gamma T_1 \mu d\gamma \qquad \forall\ \mu \in \Lambda_h \\ &\widetilde{T_h} = T_0 \quad \text{on} \quad \Gamma \end{aligned}\right\} \tag{4.4}$$

R.Glowinski in 1994 [27] made the remark that the spaces T_h and Λ_h can be chosen to be independent and suggested to define Λ_h from the intrinsic geometrical properties of γ. This approach enables the uses of non-matching mesh sets. For problem (4.4), the discretization can be chosen as shown in Figure 54 i.e. two meshes are used, one underlying mesh and one to capture the boundary on which to enforce the boundary conditions.

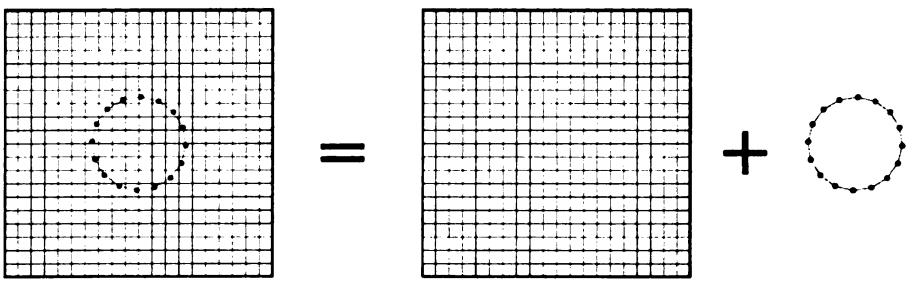


Figure 54. Discretization mesh set (left), discretization of fictitious domain Ω (middle) and embedded boundary γ (right)

This treatment provides substantial simplifications to the meshing task, and is particularly well suited to the situations where ω is subjected to a rigid body motion [27,28]. Both R. Glowinski et. al [27-31, 33-35] and Betrand et. al [32].employed this approach to construct the approximation spaces and simulate the flow around moving rigid bodies of various geometries.

4.3 Effects of the Presence of a Multiplier Within an Element

The use of non-matching meshes to solve a problem with an embedded boundary constraint often results in the Lagrange multipliers that are not located at the nodes of the underlying Cartesian mesh. When the extended solution of the problem is discontinuous (with respect to the first derivative of the variables) across the embedded boundary, the original governing equations (Eq. 4.1) defined in $\Omega \setminus \omega$ are not valid for the extended domain Ω even if the formulation of Eq. 4.2 still holds.

In a finite element approximation, the extended domain Ω is discretised into a collection of preselected finite elements, the solution over each element then can be approximated by a set of approximation functions derived from the interpolation theory [37]. These approximation functions are often algebraic polynomials which are smooth and process no discontinuities, which means when applied with LM/FDM, the solutions are still assumed to be smooth in the interfacial regions where the corresponding Lagrange multipliers is located, the approximation functions then can not track such a jump of the local gradient and thus introduces local errors in the numerical solution (as the adjacent nodes are adjusted to enforce the Dirichlet boundary conditions within the corresponding element).

Figure 55 illustrates a LM/FDM implementation for one-dimensional steady state heat conduction problem with non-matching mesh set. Obviously, the linear approximation of the unknowns cannot capture a discontinuity of the first derivative within an element.

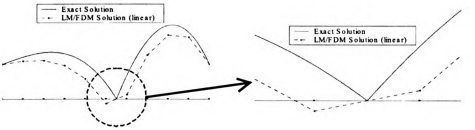


Figure 55 A linear approximation of the solution enforces the embedded boundary constraint but cannot capture a discontinuity in the first derivative at other location but the nodes. The right figure is an enlargement of the solution near the location of the multiplier.

In Figure 56, another example for a two dimensional case is provided. In this figure, the absolute and relative errors are shown and it can be seen that the errors are very high on the nodes surrounding the collocation points corresponding to the Lagrange multipliers.

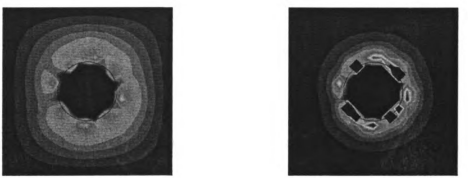


Figure 56 Absolute error (Left) /Relative Error (right) introduced by the LM/FDM for 2-D steady state heat conduction problem. E_{abs} and E_{rel} is defined as the point wise errors with $E_{abs} = |T - T_h|, E_{rel} = |T - T_h|/T$, the LM/FDM solution is obtained using gauss quadrature integration.

4.4 Parametric Study on the Sources of the Errors

Second order problems with single unknown and constant coefficients should have almost no error between the exact solution and the body fitted finite element solution at the nodes [37]. Clearly this is not the case for the Lagrange multiplier/fictitious domain

methods as observed from figure 55 and 56. The finite element solutions do not coincide with the exact solutions at the nodes.

To study the effect of a Lagrange multiplier located within an element, a simple one-dimensional problem is considered. The problem is shown in Figure 57.

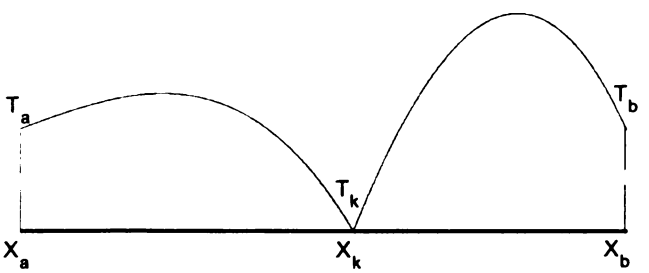


Figure 57 The extended problem possess discontinuity across the embedded boundary x_k: Original problem: $-\Delta T = x$ *in* (x_a, x_k) with boundary condition $T|_{x_a} = T_a$ *and* $T|_{x_k} = 0$; Extended problem: $-\Delta T = 0$ *in* (x_a, x_b) with embedded boundary constraint: $T|_{x_k} = 0$ and boundary condition $T|_{x_a} = T_a$, $T|_{x_b} = T_b$;

The exact solution to the above problem is given by:

$$T(x) = \begin{cases} a_1 x^3 + c_1 x + d_1 & x \in (x_a, x_k) \\ a_2 x^3 + c_2 x + d_2 & x \in (x_k, x_b) \end{cases} \tag{4.5}$$

where:

$$
\begin{aligned}
&a_1 = a_2 = -\frac{1}{6};\\
&c_1 = \frac{T_a + \frac{1}{6}\left(x_a^3 - x_k^3\right)}{x_a - x_k}; \quad c_2 = \frac{T_b + \frac{1}{6}\left(x_b^3 - x_k^3\right)}{x_b - x_k};\\
&d_1 = \frac{1}{6}x_k^3 - c_1 x_k; \quad d_2 = \frac{1}{6}x_k^3 - c_2 x_k;
\end{aligned}
\tag{4.6}
$$

This solution will be used below for studying the quality of the LM/FDM solution.

Two measures are used to estimate the quality of the solution: the energy norm and the L_2 norm. These two measures are defined as:

$$
\text{Energy norm: } \|e\|_1 = \|T - T_h\|_1 = \left\{ \int_0^1 \left(\frac{dT}{dx} - \frac{dT_h}{dx} \right)^2 dx \right\}^{\frac{1}{2}} \tag{4.7}
$$

$$
L_2 \text{ norm: } \|e\|_0 = \|T - T_h\|_0 = \left\{ \int_0^1 (T - T_h)^2 \, dx \right\}^{\frac{1}{2}} \tag{4.8}
$$

Clearly, other factors other than the element size that can affect the accuracy of the LM/FDM solution. It can be observed that by adjusting the relative position of the Lagrange multiplier inside the element or the value of the applied Dirichlet boundary condition at the auxiliary part, better LM/FDM solutions can be obtained: in Figure 58, when the Lagrange multiplier locates on the nodes' position of the corresponding element (left) or the exact solution for the extended problem does not possess discontinuity across the embedded boundary (right), there is no requirement of having a discontinuous slope inside the corresponding element and the LM/FDM solutions coincide with the exact solution at the nodes.

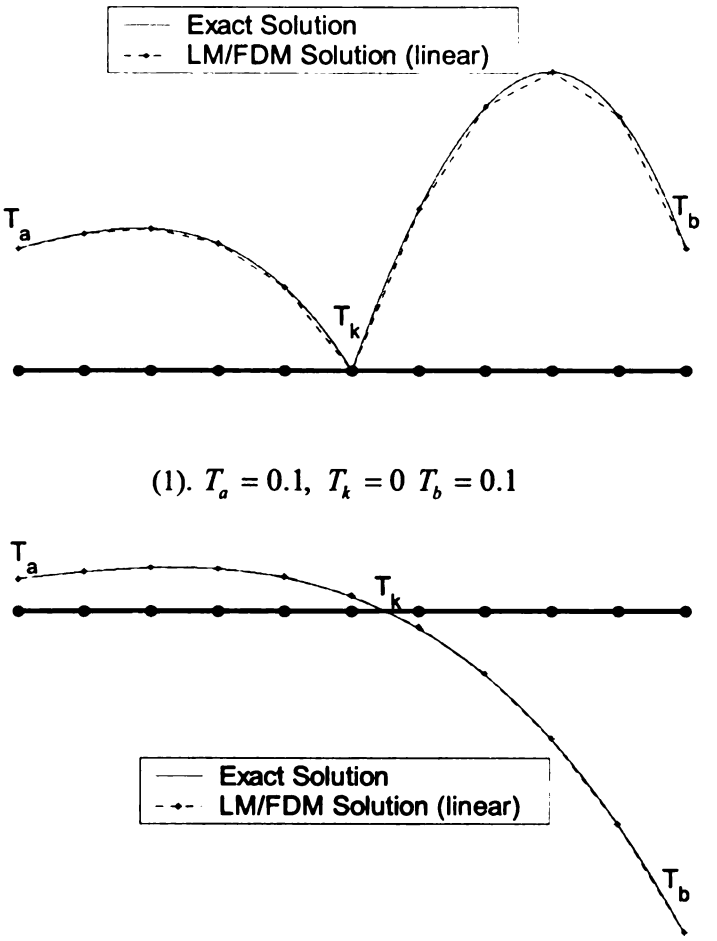


Figure 58 LM/FDM solutions coincide with the exact solution at nodes when no discontinuity locates inside the corresponding element: Lagrange multipliers locate on the adjacent element nodes (case-1); no discontinuity exists across the interface region (case-2)

Three influence factors are studied here: the element size (h), the relative location of the multipliers within the corresponding element (xk_{rel}, where $xk_{rel} = (xk - x_1)/h$ with x_k the location of the Lagrange multiplier and x_1 the adjacent node's location), and the value of the applied auxiliary Dirichlet boundary conditions (here a nondimensionless

parameter—the ratio of both the Dirichlet boundary conditions, $\alpha = T_b / T_a$ is first used to study the boundary condition's effect) which determines the jump of the local gradient of the solution.

4.4.1 Auxiliary Boundary Condition Influence

By adjusting the value of the auxiliary boundary condition (Figure 59), it can be observed that both the errors reach the minimum values at the specific condition $\alpha \approx -10$, which is compatible to the results of the body fitted finite element method (BFM methods, we employ this method in this article for the comparison, both methods use the linear elements here), and then gradually increase when α (correspond to the auxiliary boundary condition) changes in both directions.

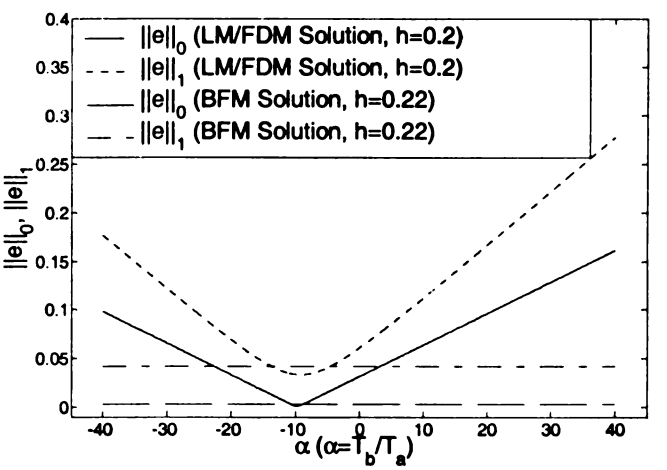


Figure 59. The influence of the auxiliary boundary condition ($xk_{rel} = 0.5$).

As already mentioned in section 4.3, through domain extension the exact solution may possess a jump of the local gradient across the embedded boundary γ, it usually cannot be avoided in the 2-D and 3-D case. By adjusting the auxiliary boundary condition, the magnitude of the jump varies. To make this clearer, another parameter θ (Figure 60) that

represents the magnitude of such a jump is introduced and its relationship with the errors is studied.

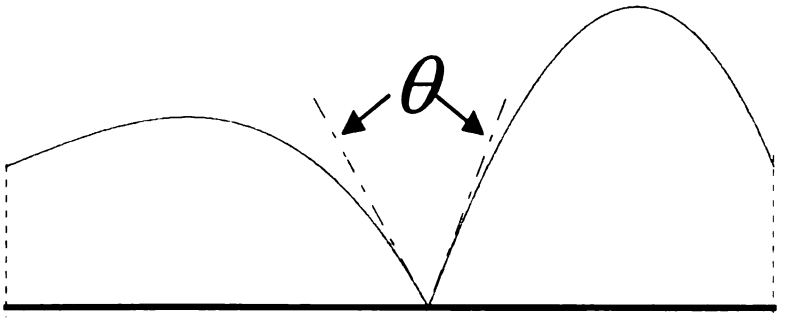


Figure 60. Definition of θ

Figure 61 shows the relationship between θ and the auxiliary boundary condition. At $\alpha = -10$, the exact solution smoothly across the embedded interface ($\theta = 180^o$). Changing auxiliary boundary condition in both direction($\alpha < -10$ and $\alpha > -10$), cross-boundary discontinuity exists and varies.

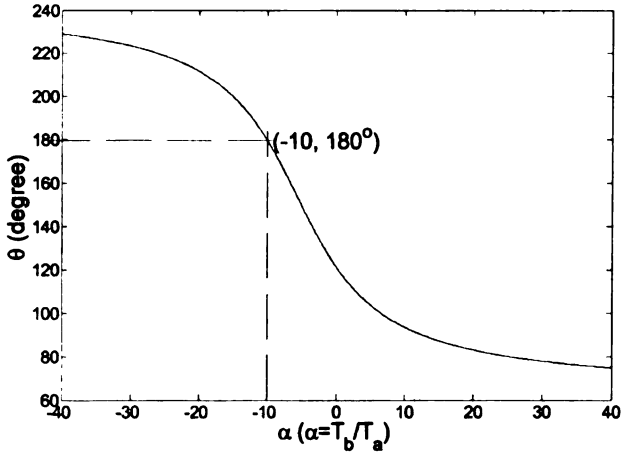


Figure 61. $\alpha - \theta$ relation (where $\alpha = -10$ corresponds to $\theta = 180^o$)

In LM/FDM methods, when the boundary mesh is not conformal to the underlying Cartesian mesh, this discontinuity will locate inside the corresponding fictitious domain

element (the interfacial element), which means the Lagrange multiplier is located within that element. For this local element region, the use of smooth weighting function approximations is inadmissible. It is very difficult to capture features of such a discontinuity thus introduces error. From figure 62 it can be seen that (in the case of the Lagrange multiplier locates inside element) when $\theta = 180^o$ (corresponds to $\alpha = -10$), which means the exact solution is smooth in the local interfacial region, minimum errors can be achieved. The errors increase gradually when the angle becomes sharp ($\theta < 180^o$) or blunt ($\theta > 180^o$).

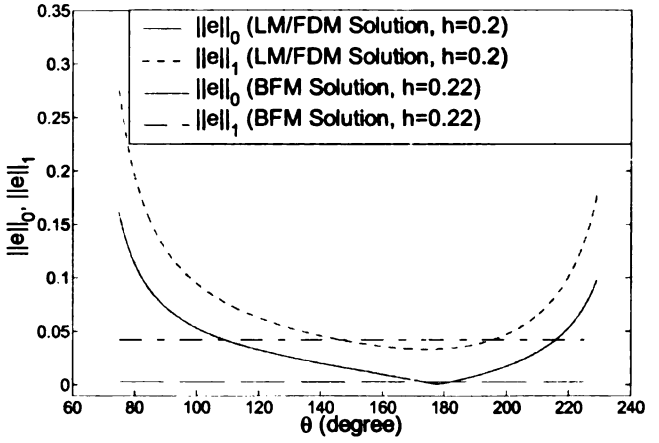


Figure 62. The influence of the local gradient jump ($xk_{rel} = 0.5$).

4.4.2 Mesh size and relative Lagrange multiplier inner element position influence

A decrease in the element size should significantly improve the precision. But from figure 63,64, it can be observed that the absolute error does not linearly change but exhibits some "oscillation" as the mesh is refined. A continuous reduction in the element size (here we assume uniform discretization) will change the relative location of the Lagrange multiplier within the corresponding interfacial element. This affects the accuracy of the local finite element approximations and the errors are greater than the

ones for the body-fitted finite element solutions except under some specific condition for which they are equal.

For a fixed fictitious domain, a change in the element size (here we assume uniform discretization) will cause the change of the relative location of the Lagrange multipliers within the corresponding interfacial element, it also oscillates with the element size (Figure 65). This oscillation is synchronous to the error-element size oscillation.

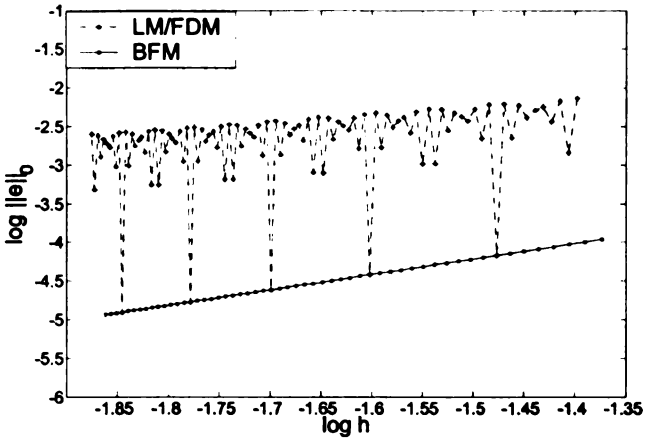


Figure 63 L_2 norm oscillates with the mesh refined (compared with BFM solution, $\alpha = 1$)

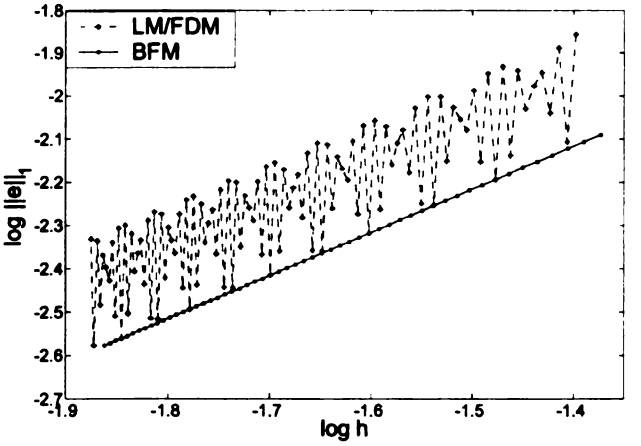


Figure 64 Energy norm oscillates with the mesh refined (compared with BFM solution, $\alpha = 1$)

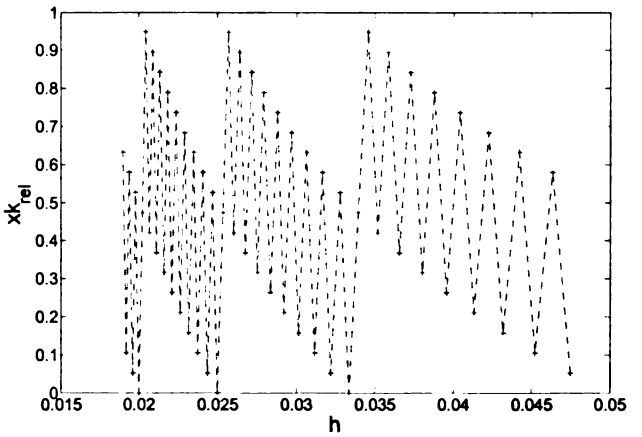


Figure 65 xk_{rel} oscillates with the mesh refined ($\alpha = 1$)

Thus the relative position of Lagrange multiplier also plays a key role in affecting the solution accuracy. It is then necessary to study the influence of element size and relative location of Lagrange multiplier within an element separately.

For fixed relative location of the Lagrange multiplier (figure 66,67), it can be seen that error linearly changed with element size. Plots of $\log \| e \|_0$ and $\log \| e \|_1$ versus $\log h$ show that:

1. When Lagrange multiplier located inside elements ($xk_{rel} \in (0,1)$), the errors can be expressed as:

$$\log \| e \|_0 = \log h + \log c_0 + \log f_0(xk_{rel}) \tag{4.9}$$

$$\log \| e \|_1 = \log h + \log c_1 + \log f_1(xk_{rel}) \tag{4.10}$$

which means convergence ratio of the LM/FDM solution is 1 in the L_2 norm and 1 in the energy norm. In this case, the errors are greater than the ones in the body-fitted finite element methods and the convergence ratio is smaller than in the body-fitted finite element methods (which equals 2).

2. When Lagrange multiplier located on the boundary of the correspond elements ($xk_{rel} = 0$), then:

$$\log \| e \|_0 = 2 \log h + \log c_0 \tag{4.11}$$

$$\log \| e \|_1 = \log h + \log c_1 \tag{4.12}$$

which is exactly the same errors as the body-fitted finite element methods: the convergence rate is 2 in the L_2 norm and 1 in the energy norm. Generally, such situation is very difficult to obtain in 2-D and 3-D when LM/FDM is employed with independent discretization for the fictitious domain and embedded boundary.

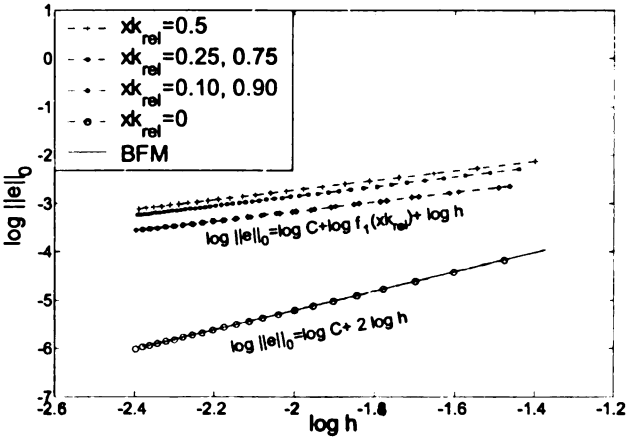


Figure 66. Plots of the L_2 norm of error versus element size. The log-log plots give the rates of the convergence in the L_2 norm. The rates of convergence are given by the slopes of the lines (the plots shown are for linear elements).

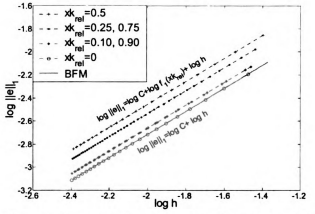


Figure 67. Plots of the energy norm of error versus element size. The log-log plots give the rates of the convergence in the energy norm. The rates of convergence are given by the slopes of the lines (the plots shown are for linear elements).

If the element size get fixed (figure 68), it can seen that the error is a parabolic-like function of the Lagrange multiplier relative location: It will reduce to the minimum value as it approaches the nodes of the corresponding element but will gradually increased and finally reach the peak value as it approaches the center of the element.

$$\| e \|_0 = f_0(h) + f(xk_{rel}) = f_0(h) + \sum_i a_i (xk_{rel} - 0.5)^i \tag{4.13}$$

$$\| e \|_1 = f_1(h) + f(xk_{rel}) = f_1(h) + \sum_j b_j (xk_{rel} - 0.5)^j \tag{4.14}$$

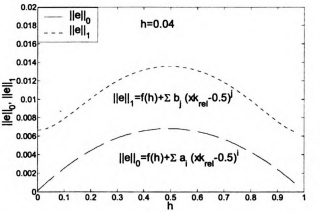


Figure 68. Plots of the L_2 /energy norm of error versus the relative location of Lagrange multiplier xk_{rel}.

The jump of the local gradient of the solution across the immersed boundary in the extended domain and the presence of the Lagrange multiplier within an element influence the accuracy of the LM/FDM solution. As a result, the numerical solution is very sensitive to relative location of the Lagrange multiplier within an element, the LM/FDM solution then cannot achieve the same level accuracy as the body fitted method's (BFM's) solution if the same type of element is used (figure 69).

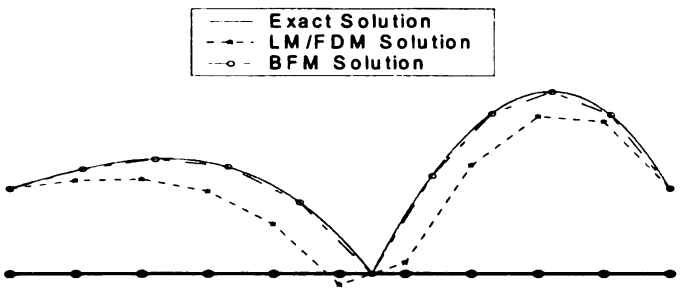


Figure 69. Solution comparison of the LM/FDM methods and BFM methods (both use linear element) : BFM solution possess no error at the node while LM/FDM introduces errors

4.5 Reducing the error

There are several ways to reduce the errors, obviously include mesh refinement (figure 70), and adjusting the Dirichlet boundary condition for the auxiliary domain in the extended problem (figure 72). But as shown already, the LM/FDM solution is very sensitive to the location of the Lagrange multiplier inside element, mesh refinement for the fixed fictitious domain may not produce an improved solution (figure 71). And also, as we have discussed in section 4.4, even the relative location of Lagrange multiplier within an element is fixed, LM/FDM methods converge slower than the body fitted methods. Regarding the auxiliary boundary condition adjustment, it only works for the

external domain-extending problem and the correct direction of the adjustment is not known in most cases.

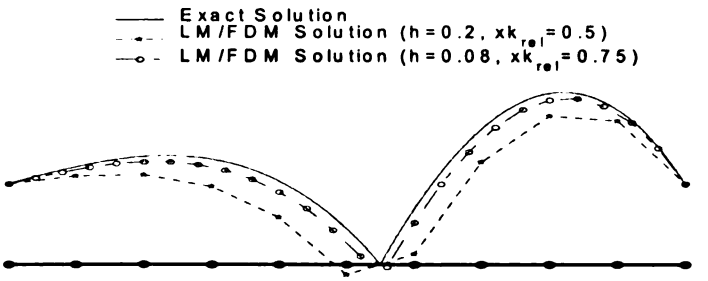


Figure 70. Mesh refinement may get a better solution

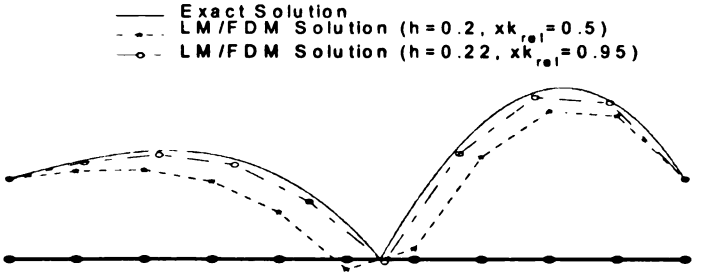


Figure 71. Mesh refinement may produce a worse solution: with a coarser mesh (h=0.22), we can even get a better solution (compared with h=0.2)

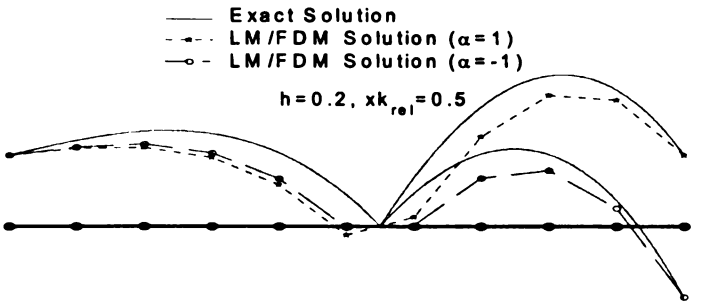
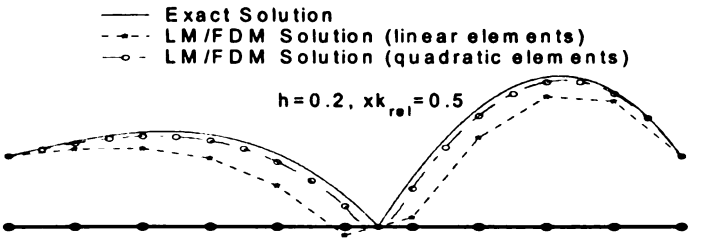


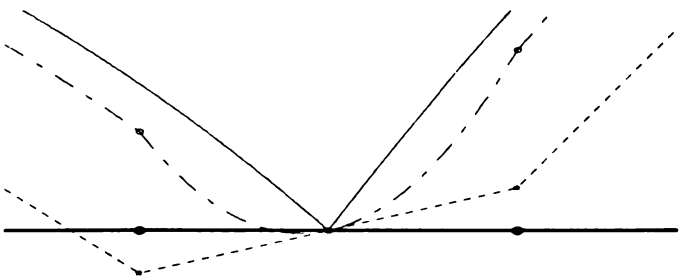
Figure 72. By adjusting auxiliary boundary condition adjustment from $T_b = 0.1$ (corresponds to $\alpha = 1$) to $T_b = -0.1$ (corresponds to $\alpha = -1$), a better solution can be obtained

High-order element can also provide a better solution because it use high-order polynomials which can approximate the discontinuity much more accurate than linear

element (figure 73). But the error analysis shows it also is sensitive to the relative inner element location of the Lagrange multiplier and the L_2 norm does not show much improvement compared to the BFM solution and LM/FDM linear element solution. Thus in LM/FDM employing high-order elements is also not a featherable way for reducing error(figure74 and 75), plus, both mesh refinement and high order element is expensive.



(1) solution for high-order element



(2) local zoom in

Figure 73 High order element can get a better solution (1) because it can approximate the discontinuity much more accurate (2).

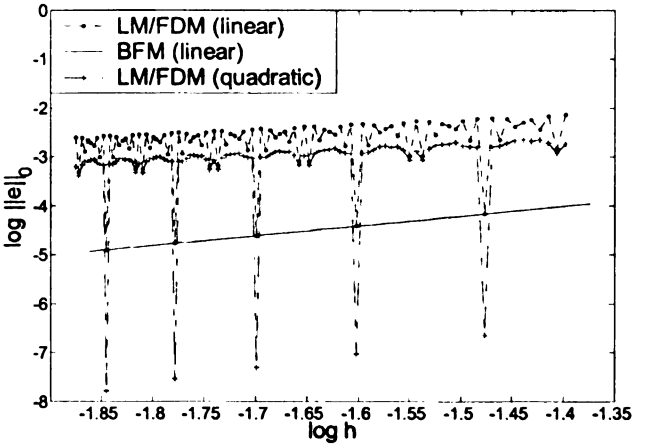


Figure 74 L_2 norm oscillates with the mesh refined for quadratic elements

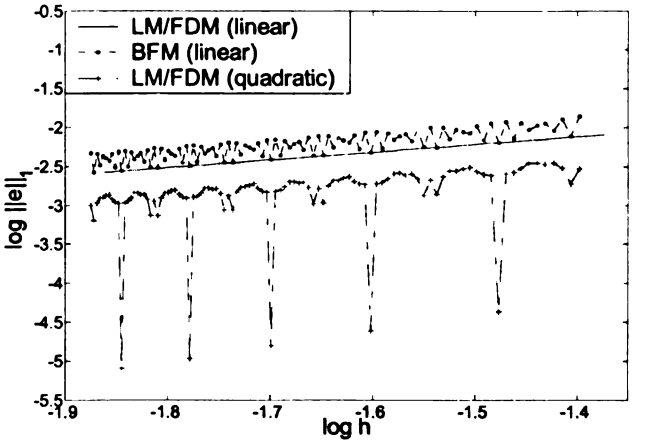


Figure 75 Energy norm oscillates with the mesh refined for quadratic elements

4.6 Fictitious Constraint Methods

The multipliers enforce specified constraints within the elements but induce errors on the adjacent nodes. A possible approach to mitigate the errors on the adjacent nodes might consist of adjusting the embedded boundary condition to a value that reduces the error locally. This approach thus introduces artificial boundary conditions that are referred to here as the fictitious constraints. In this method, the embedded boundary condition is adjusted to an arbitrary value that can be predicted from the standard LM/FDM solution. This approach can be illustrated by considering a linear approximation T^*, which satisfies a specific embedded boundary condition $T_{x_k} = T_k^*$ (T_k^* is the "Fictitious Constraint "), that coincides with the exact solution at the nodes (Figure 76), then obviously $T_k^* = T_1 * (1 - xk_{rel}) + T_2 * xk_{rel}$ (T_1, T_2 are the nodal values for the interfacial element $[X_1, X_2]$) and may not be the actual embedded boundary condition $T_{x_k} = T_k$ if a discontinuity in the first derivative of the solution exists.

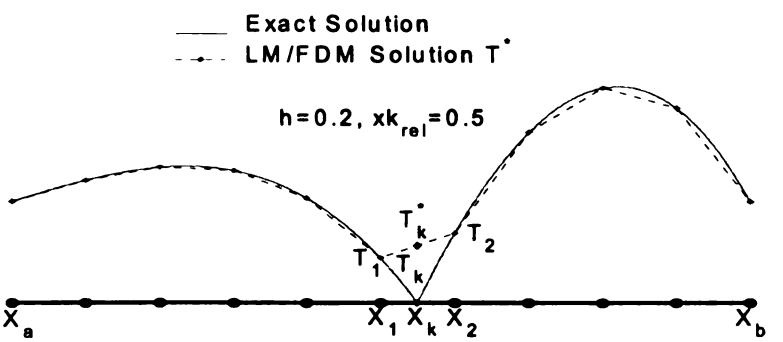


Figure 76. A LM/FDM Solution which satisfying a specific embedded boundary condition ("Fictitious Constraint ") coincides with the exact solution at nodes

Such a bridge up with a proper value for "Fictitious Constraint " T_k^* modifies the extended problem as following:

$$\left.\begin{aligned} &\text{Find } \tilde{T} \in H^1(\Omega),\ \lambda \in L^2(\gamma) \text{ such that} \\ &\alpha(\tilde{T},\ v) + \int_\gamma \lambda v d\gamma = \int_\Omega f v d\Omega \qquad \forall\ v \in H_0^1(\Omega) \\ &\int_\gamma \tilde{T}\mu d\gamma = \int_\gamma T_k^* \mu d\gamma \qquad \forall\ \mu \in L^2(\gamma) \\ &\tilde{T} = T_0 \quad \text{on} \quad \Gamma \\ &\text{where} \quad \alpha(\tilde{T},\ v) \equiv \int_\Omega \nabla\tilde{T}\cdot\nabla v\ d\Omega \end{aligned}\right\} \tag{4.15}$$

The solution for this modified equation can coincide with the solution for equation 4.2 in most regions except the interfacial region.

Knowledge of the exact solutions for the two different non-interfacial regions $[X_a, X_1]$ and $[X_2, X_b]$ of Figure 76 makes it simple to construct a smooth solution define on $[X_1, X_2]$ that satisfies the governing equation $\alpha(\tilde{T},\ v) + \int_\gamma \lambda v d\gamma = \int_\Omega f v d\Omega$ with boundary condition $[T_1, T_2]$ and the corresponding fluxes at position X_1, X_2. The combined solution of these three regions $[X_a, X_1]$,$[X_1, X_2]$ and $[X_2, X_b]$ also satisfied the governing equation 4.2 and the corresponding boundary condition, only the embedded constraint is different from the original problem. Its value can be adjusted depending on the selection of the element type.

In this case the "Fictitious Constraint " can be adjust to the theoretical condition $T_{x_k} = T_k^*$ because the exact solution of $[T_1, T_2]$ is already known, but in practice, a two-step predictor-corrector procedure based on the solution obtained for standard LM/FDM allows to find the appropriate value of the fictitious constraint.

The procedure consists of first assuming that the exact solution in the neighborhood (must cover at least 2 elements) of the interfacial region (in both the original domain side and auxiliary domain side) is smooth, almost linear and does not possess any other discontinuity (Figure 77). It can then be assumed that the solution near the multiplier is linear (such as in zone 1 defined in $[X_0, X_k]$ and zone 2 defined in $[X_k, X_3]$).

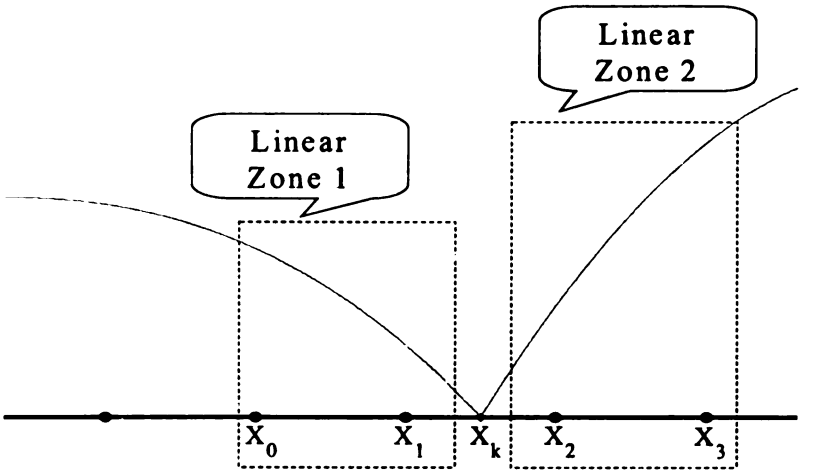


Figure 77. Linearity assumption of the boundary-neighboring zones: Variable distributions in each separated boundary-neighboring zones (such as zone 1 and zone 2 which belong to different domain-original domain or auxiliary domain) have similar slope.

In second it is assumed there are sufficient number of elements in each zone so that the variable distribution in the interfacial region such as $[X_1, X_k]$ ($[X_k, X_2]$ can be predicted from the neighboring element region $[X_0, X_1]$ ($[X_2, X_3]$ in the same linear zone.

Third, the derivative of the LM/FDM solution is assumed to approximate the derivative of the exact solution in regions except near the interfacial element region (Figure 78).

The slope computed near the multiplier is extrapolated from each sides to the slope within the element associated to a multiplier (such as regions $[X_0, X_1]$ and $[X_2, X_3]$).

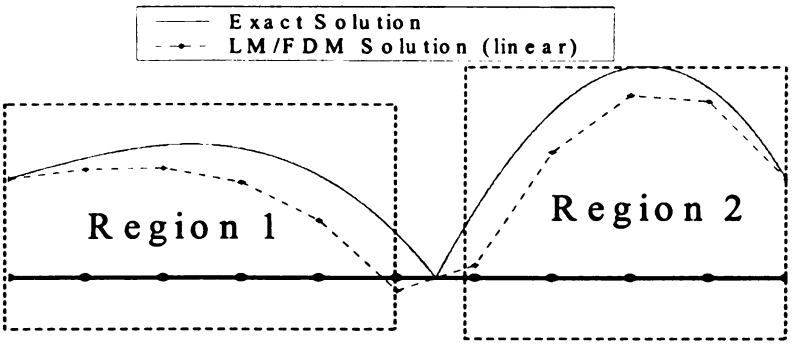


Figure 78. Derivative similarity assumption: Variable derivative of the LM/FDM solution in each region (region 1 or region 2) is similar to the exact solution.

The predicted solution from standard LM/FDM then can be used to obtain more accurate nodal values for the interfacial elements and the "Fictitious Constraint" can be obtained by the corrections based on these nodal values. This is described in detail below with Figure 79 for the problem described in Figure 57 as following:

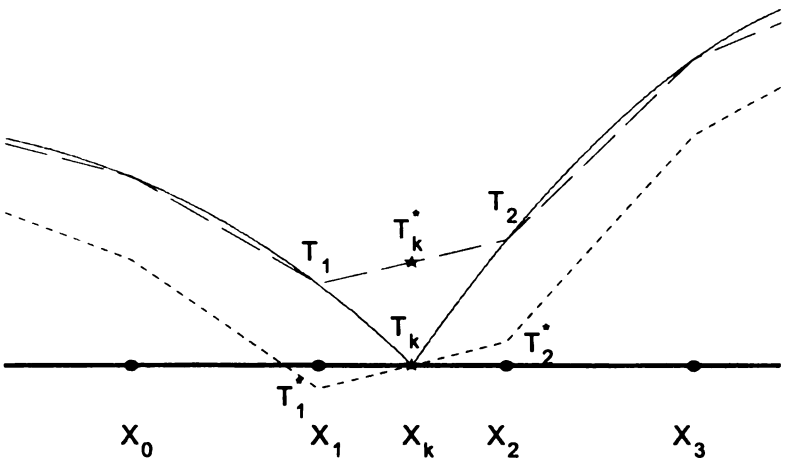


Figure 79. Implementation of fictitious constraint method in 1-D(The solid line is the exact solution; the dotted line is the LM/FDM solution and dashed line the fictitious constraint solution).

1. First computing, using LM/FDM method (enforcing the original boundary condition) a solution T^* (which takes value T_1^*, T_2^* at nodal X_1, X_2 respectively). For this solution the derivative in the non-interfacial region is similar to the one of the exact solution (assumption 3). Hence the derivatives (in the LM/FDM solution) for regions $[X_0, X_1]$ (corresponds to the derivative k_1) and $[X_2, X_3]$ (corresponds to the derivative k_2) can be estimated.

2. Once k_1 and k_2 are known, from assumptions 1 and 2, the derivative of the solution in region $[X_1, X_k]$ ($[X_k, X_2]$) is similar to that in region $[X_0, X_1]([X_2, X_3])$, with assumption 3, it can be treated as the same value obtained from the LM/FDM solution in region $[X_0, X_1]([X_2, X_3])$, which implies the slopes in the interfacial regions $[X_1, X_k]$ ($[X_k, X_2]$) is $k_1(k_2)$. The nodal values T_1, T_2 for this interfacial element then can be approximated by a linear extrapolation from the boundary condition to the adjacent nodes as:

$$\begin{aligned} T_1 &= k_1 x_{k_{rel}} + T_k \\ T_1 &= k_2 (1 - x_{k_{rel}}) + T_k \end{aligned} \qquad (4.16)$$

3. The fictitious constraint thus can be computed based on the adjusted nodal value and the relative location of the Lagrange multiplier inside $[X_1, X_2]$:

$$T_k^* = T_k + T_1 (1 - x_{k_{rel}}) + T_2 x_{k_{rel}} \qquad (4.17)$$

4. The computed fictitious constraint is finally used as boundary condition and the problem is solved again.

A much more accurate can then be obtained (Figure 80).

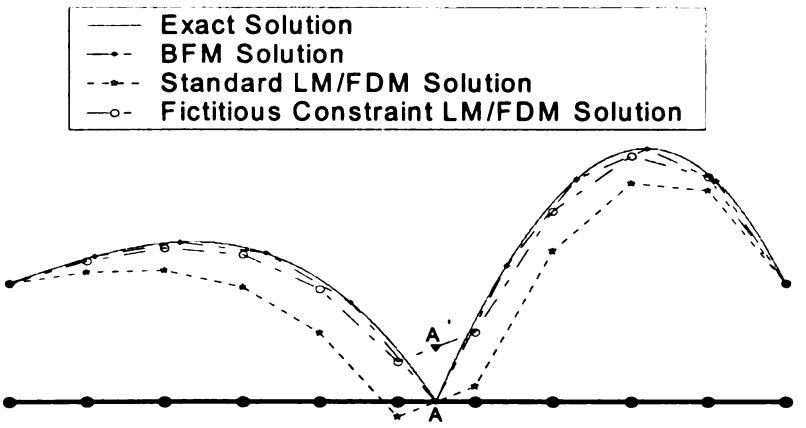


Figure 80. "Predict-Correct Fictitious Constraint" methods obtaines better solution. Note the boundary constraint is adjust from original position A to A'

The error analysis also reflects such improvements: The L_2 norm and energy norm is smaller than that obtained from the standard LM/FDM methods (figure 81, 82) but still shows some oscillations. They do approach nonetheless the results for standard body fitted finite element methods. When compare to the standard LM/FDM method, the errors are not very sensitive to relative location of the Lagrange multiplier inside elements (figure 83 84); the convergence ratio for L_2 norm is also improved (figure 85), and the errors are also not sensitive to the adjustment of the boundary condition (figure 86).

We also need to point out that when the discontinuity angle satisfy some specific condition ($\theta \in [160^0, 195^0]$ for this problem), the standard LM/FDM solution is more accurate than the fictitious constraint method. The solution is then almost smooth across the boundary so LM/FDM can approximate well the exact solution.

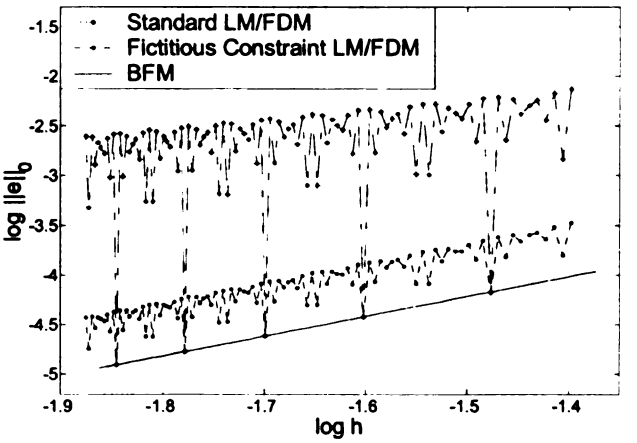


Figure 81. L_2 norm comparison for three methods : Standard LM/FDM, Fictitious Constraint LM/FDM, Body Fitted methods

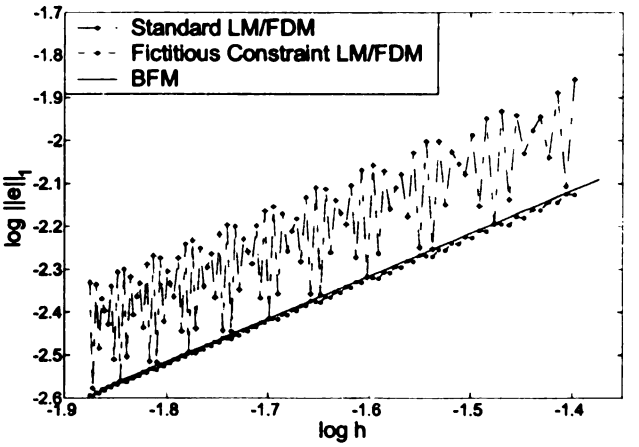


Figure 82. Energy norm comparison for three methods: Standard LM/FDM, Fictitious Constraint LM/FDM, Body Fitted methods

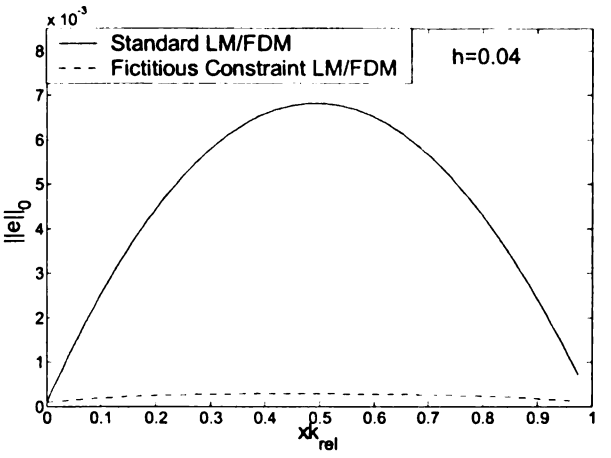


Figure 83. L_2 norm is not very sensitive to relative location of the Lagrange multiplier inside elements for Fictitious Constraint LM/FDM methods

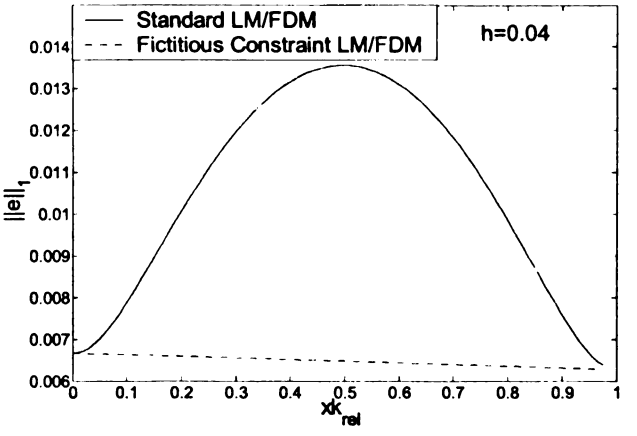


Figure 84. Energy norm is not very sensitive to relative location of the Lagrange multiplier inside elements for Fictitious Constraint LM/FDM methods

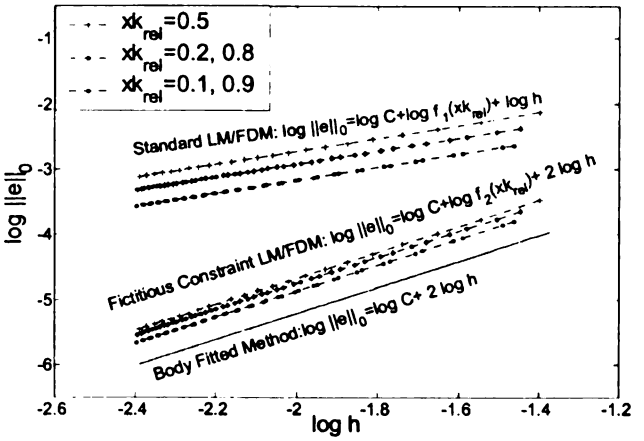


Figure 85. Plots of the L_2 norm of error versus element size. The log-log plots give the rates of the convergence in the L_2 norm. The rates of convergence are given by the slopes of the lines (the plots shown are for linear elements).

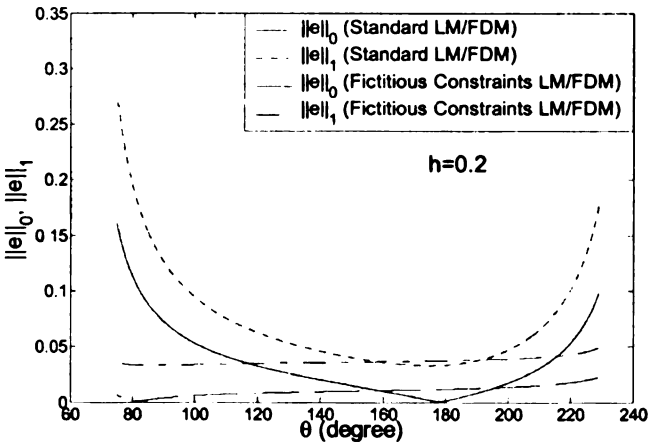
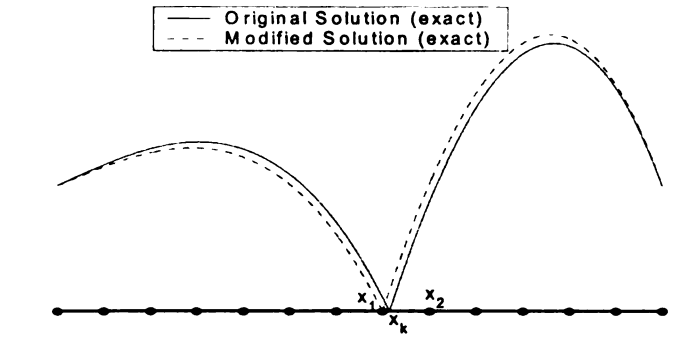


Figure 86. L_2 norm and energy norm is not very sensitive to the local gradient jump for Fictitious Constraint LM/FDM methods

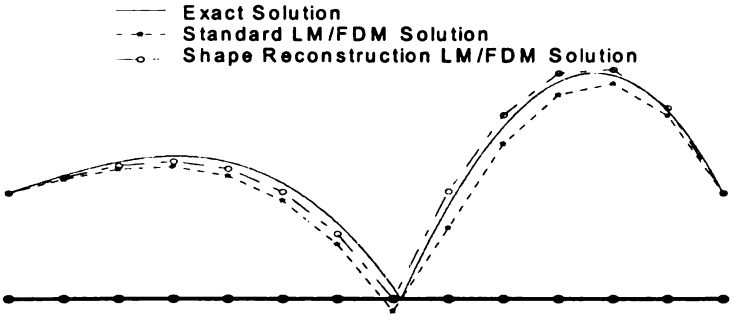
4.7 Shape Reconstruction Method

Since the embedded boundary condition on γ imposes a very strong constraint on the solution, properly relaxation of this constraint may also reduce error. One of the easiest way to realize this idea is shifting the location of Lagrange multiplier to the nearest nodal points in the corresponding interfacial element. Figure 87 illustrates this idea. Since we

assume that the mesh is fine enough and no great variations near the interface region, the modified solution should approach the original solution.



(a)

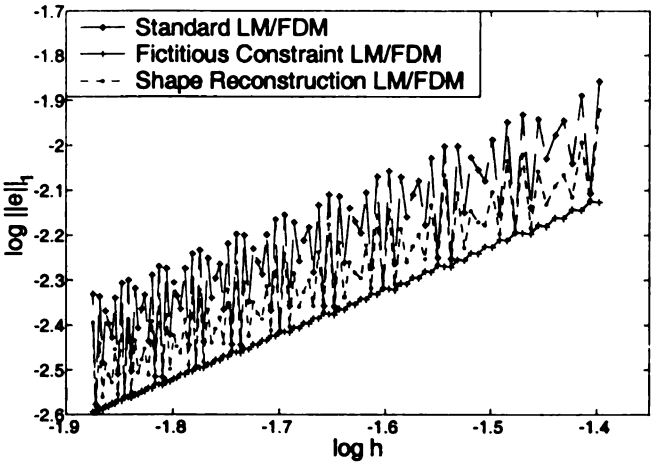


(b)

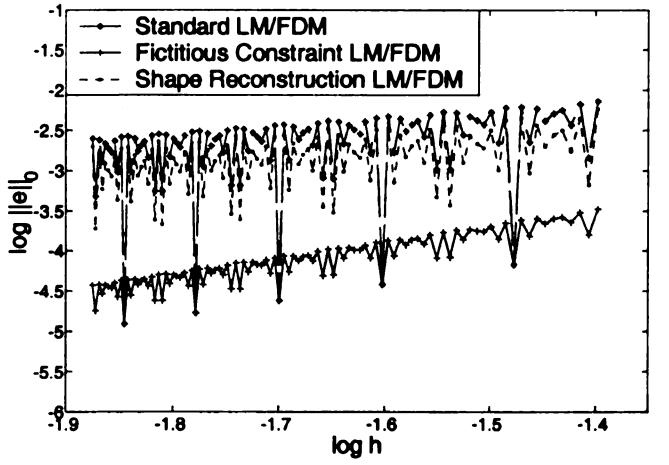
Figure 87. Shifting the location of Lagrange multiplier from x_k to the nearest node x_1 in the corresponding interfacial element (a) will obtain a better solution (b).

The errors analysis also shows such improvement under some specific conditions: The L_2 norm and energy norm still show the oscillation but they are smaller than that obtained from the standard LM/FDM methods and greater than the fictitious constraint methods (figure 88); Compare to the standard LM/FDM method, the errors are not very sensitive to relative location of the Lagrange multiplier within elements (figure 89) and the discontinuity angle (figure 90), but much more sensitive if compared with the one from Fictitious Constraint methods. When the discontinuity angle satisfy some specific

condition ($\theta \in [130^0, 210^0]$ for this problem), the solution of standard LM/FDM is more accurate than that of Shape Reconstruction method.

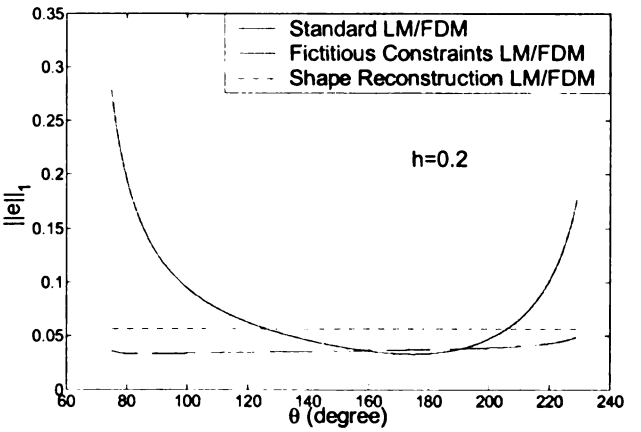


(a)

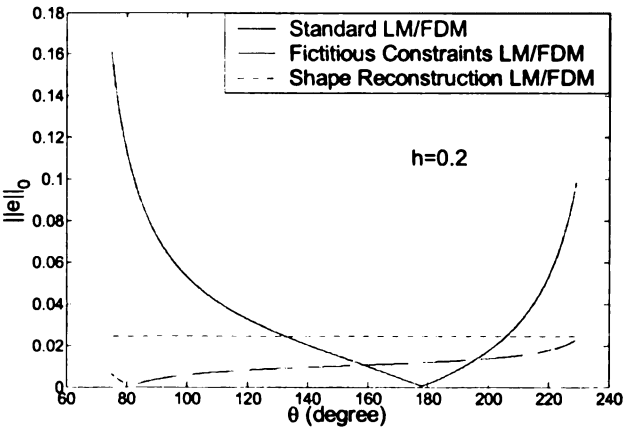


(b)

Figure 88. Energy norm (a) and L_2 norm (b) comparisons for three methods: Standard LM/FDM, Fictitious Constraint LM/FDM, Shape Reconstruction LM/FDM

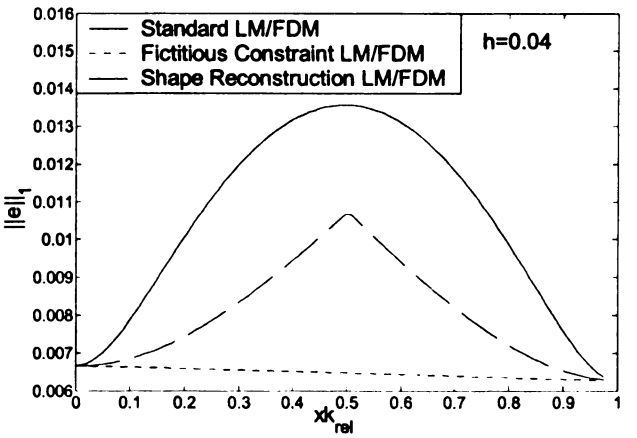


(a)

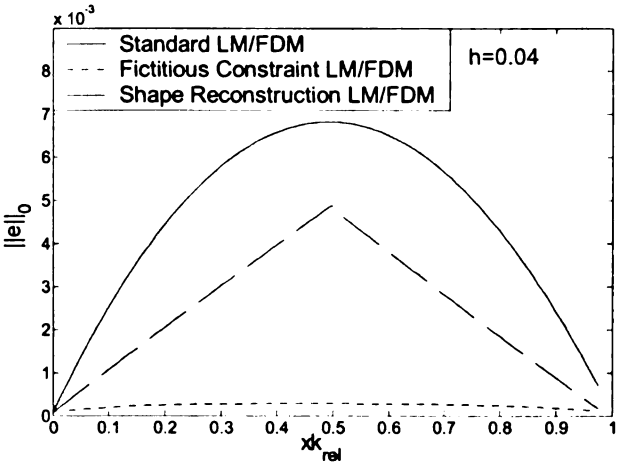


(b)

Figure 89. Energy norm (a) and L_2 norm (b) are not very sensitive to local gradient jump (compare to the standard LM/FDM methods) but much more sensitive (compare to the fictitious constraint LM/FDM) for Shape Reconstruction methods



(a)



(b)

Figure 90. Energy norm (a) and L_2 norm (b) are not very sensitive to relative location of the Lagrange multiplier inside elements (compare to the standard LM/FDM methods) but much more sensitive (compare to the fictitious constraint LM/FDM) for Shape Reconstruction LM/FDM methods

4.8 LM/FDM in Two Dimensions

The essential concepts about stand LM/FDM, Fictitious Constraint LM/FDM and Shape Reconstruction LM/FDM that are outlined in the above sections do not change in two or three dimensions. The only difficulties that emerge are related to the computation of the boundary integrations associated with the embedded boundary γ ($\int_\gamma \lambda v d\gamma$, $\int_\gamma \tilde{T}\mu d\gamma$ and $\int_\gamma T_1 \mu d\gamma$ in equation 4.2 for example) and the implementation of the Fictitious Constraint and shape reconstruction ideas in a higher dimension space.

4.8.1 Boundary integrations

Typically these embedded boundary integration can be evaluated through Gauss-Legendre Quadrature [27] which is summarized as following (2-D case):

Suppose the function for boundary γ take the following forms: $\begin{cases} x = x(s) \\ y = y(s) \end{cases}$, then the integration $\int_\gamma f(x,y)d\gamma$ can be evaluated through the formula:

$\int_\gamma f(x,y)d\gamma = \int_{s1}^{s2} f(x(s),y(s))\sqrt{x^2(s)+y^2(s)}ds$, which changes to the one dimensional problem $\int_a^b F(x)dx$ which then can be approximated using Gauss-Legendre quadrature pending on the degree of function $F(x)$.

Another alternative approach developed by Betrand et al [32] in 1997 is the collocation methods which employs the Dirac delta function to enforce the embedded boundary constraint $T = T_0\big|_\gamma$ pointwisely, in this method, the space for Λ_h is defined as a collection of control points $\{\overline{X_i}\}_{i=1}^N$ which discretized the embedded boundary γ, then Λ_h can be expressed as:

$$\Lambda_h = \left\{ \lambda_h \middle| \lambda_h = \sum_{i=1}^{N} \lambda_i \delta(\overline{X} - \overline{X_i}), \lambda_1, \lambda_2, ... \lambda_N \in R^2 \right\}, \tag{4.18}$$

where $\delta(\bullet)$ is the Dirac delta function. The boundary integration then can be easily obtained as:

$$\int_\gamma \lambda F(\overline{X})d\gamma = \int_\gamma \sum_{i=1}^{N} \lambda_i \delta(\overline{X} - \overline{X_k})F(\overline{X})d\gamma = \sum_{i=1}^{N} \lambda_i F(\overline{X_i}) \tag{4.19}$$

This treatment simplify the computation process, but introduces errors since it only enforce the boundary condition at the collocation points. Figure 88 ,89 show the different between these two approach.

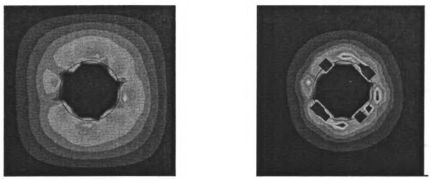


Figure 91 Absolute errors (left) and relative error (right) for the stand LM/FDM using Gauss Legendre quadrature

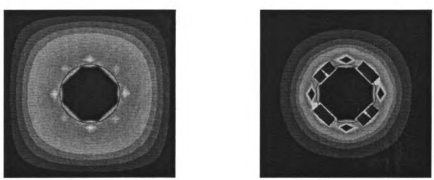


Figure 92 Absolute errors (left) and relative error (right) for the stand LM/FDM using Collocation method

4.8.2 Implementation of Fictitious Constraint LM/FDM in 2-D

The application of fictitious constraint methods in 2-D is straightforward. A heat transfer example is provided to illustrate the implementation in the context of linear element combined with the using of collocation method (the integration using Gauss-Legendre quadrature method is similar):

Suppose we have an interfacial element E_0 with a collocation point x_k locates inside (Figure 93). A numerical solution T_{std} is already obtained from standard LM/FDM.

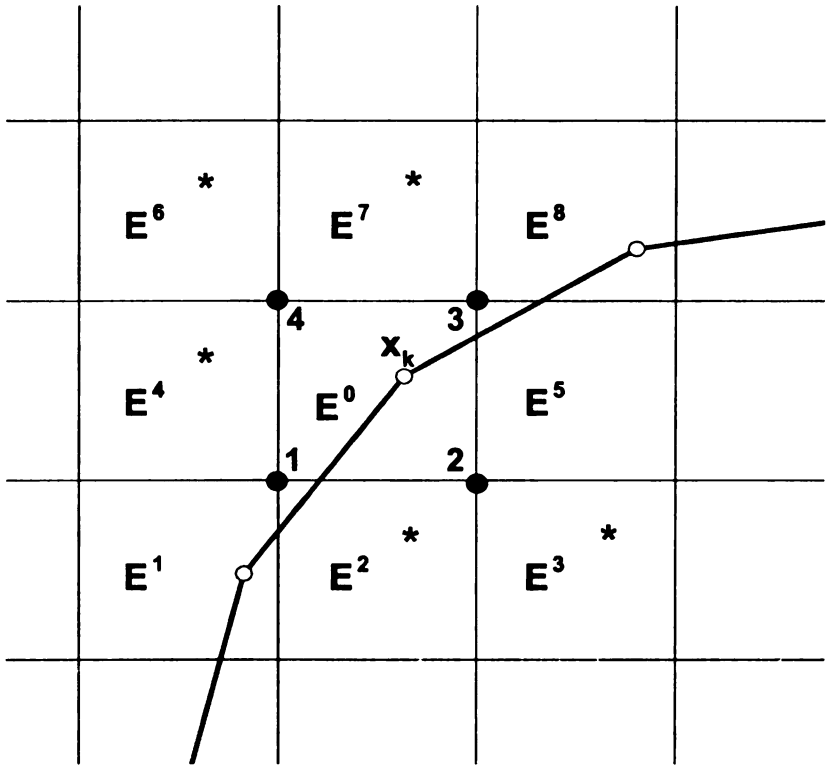


Figure 93 Illustration of the implementation fictitious constraint LM/FDM

Then the fictitious constraint T^* at location x_k can be obtained from the following formula:

$$T^*\Big|_{x_k} = T_0 + \sum_{i=1}^{4} T_{i-pre}^{E0}\phi_i\Big|_{\overline{X}=\overline{Xk_{rel}}}, \tag{4.20}$$

$$\text{where } T_{i-pre}^{E0} = \frac{1}{n}\sum_{k}\left(T_i^{Ek} - \sum_{j=1}^{4} T_j^{Ek}\phi_j\Big|_{\overline{X}=\overline{Xk_{rel}}}\right)$$

Where the following are defined:

$T_{i-std}^{E_k}$: Solution obtained from the standard LM/FDM for the i th (i=1,2,34) node of element E_k;

$T_{i-pre}^{E_0}$: Predicted solution for the i th (i=1,2,34) node of interfacial element E_0;

$\overline{Xk_{rel}}$: the collocation points x_k's relative position in element's local coordinate;

n: the total number of similarity neighboring elements;

$k = N_1, N_2, ... N_n$ is the corresponding similarity neighboring element.

For example, node 1 in interfacial element E_0 has four neighbors (E_0, E_1, E_2, E_4), among these neighbors only the one located in the non-interfacial region can be used for computations, these elements are referred as the similarity neighbors (E_2, E_4). A more accurate nodal value (for node 1 in E_0) then can be predicted from following:

$$T_{1-pre}^{E_0} = \frac{1}{2} \sum_{k=2,4} \left(T_1^{E_k} - \sum_{j=1}^{4} T_j^{E_k} \phi_j \Big|_{\overline{X}=\overline{Xk_{rel}}} \right) \tag{4.21}$$

Table 4 lists all the relative information for the correction of the fictitious constraint at location x_k in E_0.

Table 4 Computing the fictitious constraint

Node # in E_0	Neighbors	Similarity Neighbors	Nodal value Correction
1	E_0, E_1, E_2, E_4	E_2, E_4	$T_{1-pre}^{E_0} = \frac{1}{2} \sum_{k=2,4} \left(T_1^{E_k} - \sum_{j=1}^{4} T_j^{E_k} \phi_j \Big\vert_{\overline{X} = \overline{Xk_{rel}}} \right)$
2	E_0, E_2, E_3, E_5	E_2, E_3, E_5	$T_{2-pre}^{E_0} = \frac{1}{3} \sum_{k=2,3,5} \left(T_2^{E_k} - \sum_{j=1}^{4} T_j^{E_k} \phi_j \Big\vert_{\overline{X} = \overline{Xk_{rel}}} \right.$
3	E_0, E_5, E_7, E_8	E_5, E_7	$T_{3-pre}^{E_0} = \frac{1}{2} \sum_{k=5,7} \left(T_3^{E_k} - \sum_{j=1}^{4} T_j^{E_k} \phi_j \Big\vert_{\overline{X} = \overline{Xk_{rel}}} \right)$
4	E_0, E_4, E_6, E_7	E_4, E_6, E_7	$T_{4-pre}^{E_0} = \frac{1}{3} \sum_{k=4,6,7} \left(T_4^{E_k} - \sum_{j=1}^{4} T_j^{E_k} \phi_j \Big\vert_{\overline{X} = \overline{Xk_{rel}}} \right.$
Fictitious Constraint	$T^* \Big\vert_{x_k} = T_0 + \sum_{i=1}^{4} T_{i-pre}^{E_0} \phi_i \Big\vert_{\overline{X} = \overline{Xk_{rel}}}$,		

Figure 94 and 95 show the errors obtained for the Fictitious Constraint method with the collocation approach and Gauss-Legendre quadrature approach, respectively. We can see the fictitious constraint method for these two approaches provides significantly more accurate results than previous LM/FDM.

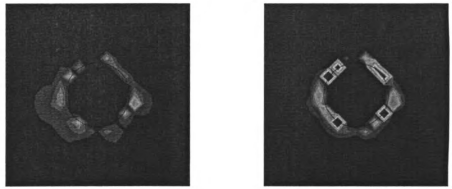


Figure 94 Absolute errors (left) and relative error (right) for the Fictitious constraint LM/FDM using Gauss Legendre quadrature

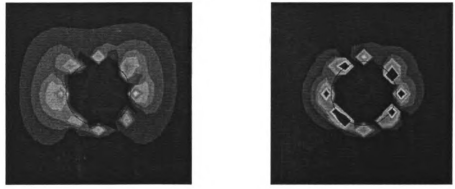


Figure 95 Absolute errors (left) and relative error (right) for the Fictitious constraint LM/FDM using collocation method

4.8.3 Implementation of Shape Reconstruction LM/FDM in 2-D

The implementation of shape reconstruction LM/FDM in 2-D is much more easier. This can be shown in Figure 96. For collocation approach, the location of the Lagrange multiplier at x_k can be simply moved to the nearest nodal position x_k^*. For the Gauss-Legendre quadrature approach, the edge that connects the two neighboring Lagrange multipliers needs to be reconstructed, one simple way to do that is just use the edges that

connect the nearest node in the corresponding interfacial element. Other approaches may also be feasible.

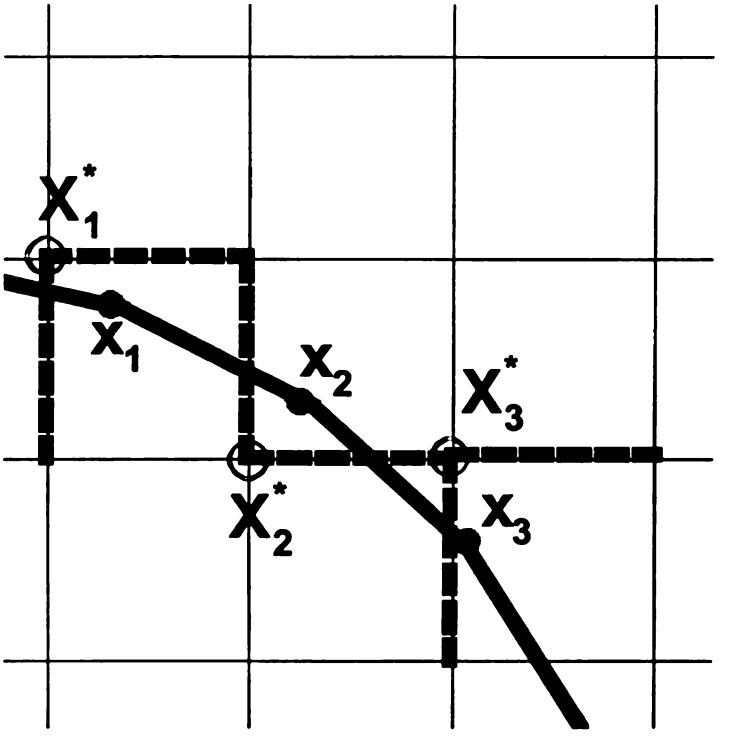


Figure 96 Illustration of the Shape Reconstruction in 2-D, where x_k represents the location of the Lagrange multipliers and x_k^* represents the reconstructed location (the nearest nodal position), the thick-solid line represents the original boundary, and the thick-dashed line represents the reconstructed boundary.

Figure 97 and 98 show the errors obtained for the Shape reconstruction method with the collocation approach and Gauss-Legendre quadrature approach, respectively. The results with gauss Legendre quadrature approach provide good results than the one from standard LM/FDM. With the collocation approach, the results are not good.

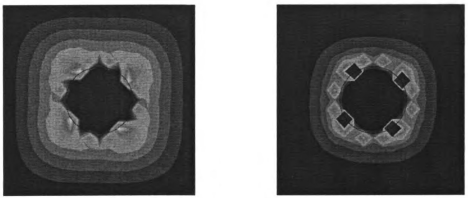


Figure 97 Absolute errors (left) and relative error (right) for the shape reconstruction LM/FDM using Gauss Legendre quadrature

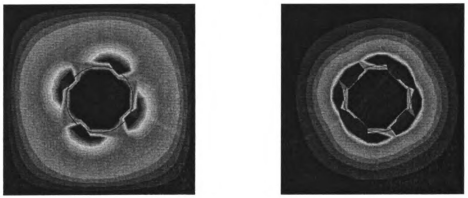


Figure 98 Absolute errors (left) and relative error (right) for the shape reconstruction LM/FDM using collocation method

Chapter 5

Conclusions

This project explored the applicability and use of moving grid technique in simulating in-cylinder flows for internal combustion engine. Current CFD codes use three strategies: spring deforming, local regridding and layering to handle the moving mesh for engine simulation. Simulations for a generic IC engine intake and compression stroke process have been performed using FIRE- an engine simulation code based on the above moving mesh strategies. The in-cylinder flow process are studied and analyzed. Three major issues that limit the application of moving gird technique: interpolation errors, human cost for mesh generation and the quality of grid system, are discussed based on the work of the in-cylinder flow simulations for Generic IC engine.

Fictitious domain methods reduce the complexity of meshing by using a simpler auxiliary domain and augmenting a functional to implement the original boundary conditions in the extended auxiliary domain. A possible fictitious domain approach, attractive for its simplicity in 2-D and 3-D, is based on using Lagrange multipliers combined either with a "collocation-like" method or the Gauss Legendre quadrature based boundary integration. The presence of a Lagrange multiplier within an element in LM/FDM introduces significant errors on the adjacent nodes when the extended problems process discontinuity across the embedded boundary, the magnitude of the errors is related to the jump of the local gradient across the embedded boundary and the relative location of the multiplier with respect to the adjacent nodes. Compared to the body fitted methods, this reduces the convergence ratio in L_2 norm with fixed relative Lagrange multiplier position

for the LM/FDM methods. Either adjusting the boundary condition with the factitious constraint or changing the location of the Lagrange multiplier to the adjacent nodes can provide improved numerical solutions for 1-D and 2-D steady state heat conduction problem. Further investigations and improvement methods are desired.

APPENDIX

Turbulence Boundary Layer

1. The Concept of y^+

The dimensionless symbol y^+ is related to the characteristics of near-wall turbulence flows. In flows along solid boundaries, there is a region of intertia-dominated flow far away form the wall and a thin layer within which viscous effects are important. Close to the wall, viscous effects dominate the flow and the mean flow velocity only depends on the distance from the wall:

$$u^+ = \frac{u}{u_\tau} = \left(\frac{\rho u_\tau y}{\mu} \right) = f(y^+)$$

The above equation is termed law of the wall and contains definitions for the two dimensionless group y^+ and u^+. u_τ is the friction velocity and is defined as :

$$u_\tau = \frac{\tau_w}{\rho}$$

y^+ provides a useful measure of the influence of the viscous layer (near the wall) given a flow velocity. The near-wall region can be largely subdivided into three layers. In the innermost layer (viscous sublayer), the flow is almost laminar and viscosity plays an important role in the momentum transfer. In the outer layer (fully turbulence layer), turbulence plays a major role. This region is also called the log-law region as u^+ is a straight-line function of y^+. There is an intermediate region between these two layers where the effect of viscosity and turbulence are equally important.

2. Wall Function

Two approaches available to model the near-wall region. In one approach, the viscous-affected inner region (viscous sublayer and buffer layer) is not resolved. Instead, semi-empirical formulas called "wall functions" are used to bridge these regions between the wall and the fully turbulent region. With wall functions, the need to modify the turbulence models to account for the presence of the wall is obviated. Another approach modifies the turbulence models to enable the viscosity-affected region to be resolved with a mesh all the way to the wall, including the viscous sublayer.

The wall function approach saves considerable computational resources because the viscosity-affected near wall region, where the solution variables change most rapidly, does not need to be resolved. This provides a practical option for turbulence flow simulations.

However, the wall function approach is inadequate in situations where low Reynolds number effects prevail in the flow domain and the assumptions underlying the wall functions cease to be valid. Such situations require near-wall models that are valid in the viscosity-affected region and hence can be integrated all the way to the wall.

3. Near Wall Mesh Generation for Wall Functions

The region near the wall is meshed finer than the rest of the cross section, as it contain the maximum amount of gradients, The distance form the wall at the wall-adjacent cells must be determined by considering the range over which the log-law is valid. The first grid point away form the wall is usually placed in the log-law region with $y^+ \in [30,300]$.

At least five points must be placed in the boundary region to resolve the gradients sufficiently for most flow situations.

BIBLIOGRAPHY

[1] Merle C. Potter and John F. Foss, Fluid Mechanics, Great Lakes Press, INC., MI, 1982

[2] Shyy W, Udaykumar HS, Rao MM, and Smith RW, Computational Fluid Dynamics with Moving Boundaries, Taylor and Francis, Washington DC. 1996

[3] W Shyy and M Francois, HS Udaykumar, N N'dri and R Tran-Son-Tay, Moving boundaries in micro-scale biofluid dynamics, App Mech Rev vol 54, no 5, September 2001

[4] W.Shyy, S.S. Thakur, H.Ouyang, J.Liu, and E. Blosch, Computational Techniques for complex transport phenomena, New York, Cambridge, 1997

[5] P.D. Thomas and C.K. Lombard, Geometric conservation law and its application to flow computations on moving grid, AIAA Vol. 17 No. 10 , pp.1030-1037, 1979

[6] Weiming Cao, Weizhang Huang, and Robert D. Russell, A moving mesh method based on the geometric conservation law, SIAM J. SCI. Comput. Vol. 24 No. 1, pp. 118-142, 2002

[7] Ching-Jen Chen, Shenq-Yuh Jaw, Fundamentals of turbulence modeling, Washington, DC, Taylor & Francis, 1998.

[8] Stanisic, M.M . The mathematical theory of turbulence, New York, Springer-Verlag, 1988.

[9] Dale A. Anderson, John C. Tannehill and Richard H. Pletcher, Computational fluid mechanics and heat transfer, Washington, DC, Taylor & Francis, 1997

[10] Graham F. Carey, Computational Grid, Generation, adaptation, and solution strategies, Washington, D.C, London, Taylor & Francis, 1997

[11] Hozef Arif, Application of computational fluid dynamics to the modeling of flow in horizontal wells, Master thesis, Standford University, June 1999

[12] A.D. Young, Boundary layers, Washington, DC, AIAA, BSP Professional Books, 1989

[13] FLUENT Inc. FLUENT 5 user guide, Vol 1-4, Fluent Inc., Lebanon, New Hampshere, USA, 1998

[14] AVL Inc., FIRE 7.2b user guide, AVL LIST GmbH, Graz, Austria, 2000

[15] Amsden A.A, Kiva-3V a block structured KIVA program for Engines with vertical or canted valves, Las Alamos National Laboratory, LA-13313-MS, 1997

[16] Adapco Inc. Star-CD user guide Version 3.10 B, adapco Inc. Melville, New York, 2000

[17] John L. Lumley, Engines, an introduction, New York, Cambridge, 1999

[18] J.I. Ramos. Internal combustion engine modeling, New York, Hemisphere Pub. Corp., 1989.

[19] N.C. Markatos, Computer simulation for fluid flow, heat and mass transfer and combustion in reciprocation engines, New York, Hemisphere Publishing Corporation, 1989

[20] B.Basara, Computations of automotive flows using the second-moment closure, ECCOMAS September 2000

[21] Jiang Yong, Zhu Ning , Qiu Rong and Fan Weicheng, Transient flow analysis of the helical intake port and cylinder of a direct injection diesel engine, Japan, Fluid and Heat Engineering study, Vol 36 No. 2, 2002

[22] Yang-Liang Jeng, Rong-Che Chen and Chao-He Chang, Study of tumbling motion generated during intake in a bowl-in-piston engine, Taiwan, Journal of Marine Science and technology, Vol. 7 No.1, pp. 52-64 1999

[23] Henk, K., Numerical simulation and experimental verification of DI diesel intake port designs, http://www.cyclone.nl/ports/ports.htm

[24] Bassem Ramadan, A study of swirl generation in DI engines using Kiva-3V, 11th international multidimensional engine modeling user's group meeting, MI, 2001.

[25] P.J. Colucci, D.Lee, C.K.Lim, G.Goldin, In-Cylinder engine modeling developments at Fluent, 12th international multidimensional engine modeling user's group meeting, MI, 2002.

[26] Gisoo Hyun and Mitsuharu Oguma, 3-D CFD analysis of the mixture formation process in an LPG DI SI engine for heavy duty vehicles, 12th international multidimensional engine modeling user's group meeting, MI, 2002.

[27] R.Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for Dirichlet problem and applications, Comput. Meth. Appl. Mech. Eng. 111, pp. 283-303, 1994

[28] R.Glowinski, T.-W. Pan, J. Periaux, A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations, Comput. Meth. Appl. Mech. Eng. 112, pp. 133-148, 1994

[29] R.Glowinski, T.-W. Pan, J. Periaux, A Lagrange multiplier/fictitious domain method for the Dirichlet problem. Generalization to some flow problems, Japan J.Industrial Appl. Math. 12, pp. 87-108, 1995

[30] R.Glowinski, T.-W. Pan, J. Periaux, Fictitious domain methods for imcompressible viscous flow around moving rigid bodies, in : J.R. Whiteman (Ed.), The Mathematics of Finite Elements and Applications, Highlight 1996, Wiley, Chichester, England, pp. 155-174, 1997

[31] R.Glowinski, T.-W. Pan, J. Periaux, Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies, , Comput. Meth. Appl. Mech. Eng. 151, pp. 181-194, 1997

[32] F. Bertrand, P.A. Tanguy, F. Thibault, A three-dimensional fictitious domain method for incompressible fluid flow problems, Internat. J. numer. Meth. Fluids 25, pp. 719-736, 1997

[33] R.Glowinski, T.-W. Pan, J. Periaux, Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies, , Comput. Meth. Appl. Mech. Eng. 151, pp. 181-194, 1998

[34] R.Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Internat. J. Multiphase Flow 25, pp. 755-794, 1998

[35] R.Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, , J. Periaux, A distributed Lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flow, Comput. Meth. Appl. Mech. Eng. 184, pp. 241-267, 2000

[36] V.Girault and R.Glowinski, Error Analysis of a fictitious domain method applied to a Dirichilet problem. Japan J. of Ind. and App.Math. 12, (1995) pp. 487-514,

[37] J.N. Reddy, An introduction to the Finite element method, 2d ed. McGraw-Hill, New York/Hemisphere, Washington, DC, 1993.

[38] Antony Jameson, The present status, challenges and future developments in computational fluid dynamics, Proceedings of the 7th AGARD Fluid Dynamics Panel Symposium, Seville, October 1995.

[39] Yanbing Li and Andre Benard, A method for reducing the errors in fictitious domain methods applied to heat transfer problems, accepted by The 6th ASME-JSME Thermal Engineering Joint Conference, March 16-20, 2003